OUTPUT FOR USER V S X

OF (ND-PATCH-SIN-00:VSX)VSX-PART-1:LIST;1

GENERATED 12.39.11 JANUARY 18, 1985

PRINTED 16.55.38 JANUARY 22, 1985

```
    1  %FILE-NAME: Sintran III VSX
    2  %% :: Part One
$NONUMB

 SINTRAN III - VSX PRELIMINARY J

          OBELIX

 *       ND 100/CX.1053.       *

 --- SINTRAN III BATCH PROCESSOR ---

 USER SINTRAN ENTERED AT 15.10.17     23 SEPTEMBER    1984

MAXIMUM TIME IS     999 MINUTES
```

```
@CC VALID FOR SINTRAN III VSE VERSION J

@CC MODIFIED 84.09.12 (KGT)

@SET-MEM-CON 151000,0,-1

@DATCL
 15.10.20      23 SEPTEMBER   1984
@CC

@CC     ********************************************

@CC

@CC     CXCOMP-BATCH

@CC     BATCH TO COMPILE AND ASSEMBLE SIII CORE RES.,

@CC     OP.SEG., OP2.SEG., AND SYS.SEG.

@CC     COMPILE SIII-COM AND CDR

@CC

@CC     ******************************************

@NPL

NORD PL APRIL 1977

@FL048
@STLIB CXCPU
@STLIB NRTD
@DEV (S-S-J)SIN-DATA,CX-OBJ,CX-OBJ
@DEV (S-S-J)FILSYS-DEF
@DEV (S-S-J)XMSG-VALUES
SYMBOL SLAKK=0                  % TEMPORARY SYMBOL FOR FREE SPACE
SYMBOL BCSTA=110000             % START OF COMMAND SEGMENT
SYMBOL OP2BG=110000             % START OF S-S-P AND MAIL SEGMENT
SYMBOL ERRFL=110000             % START OF ERROR PROGRAM SEGMENT
SYMBOL 5BFPAG=76                % LOGICAL PAGE NUMBER FOR BUFFER PAGE
SYMBOL 5UBFPAGE=77              % LOGICAL PAGE NUMBER FOR USER WINDOW IN VSX
SYMBOL L12LGP=74                % LOGICAL PAGE WINDOW FOR TERMINAL INPUT DRIVER
SYMBOL L10LGP=77                % LOGICAL PAGE WINDOW FOR TERMINAL OUTPUT DRIVER
SYMBOL L4LGP=75                 % LOGICAL PAGE WINDOW FOR TERMINAL HANDLING ON LEVEL 4
SYMBOL 3BFPAG=5BFPAG+300
@DEV (S-S-J)SIN-VARIABLES
@DEV (S-S-J)SINA
@DEV (S-S-J)SINI
@DEV (S-S-J)SINB-1
@DEV (S-S-J)FILSYS-SYS
@DEV (S-S-J)SINB-2
@DEV (S-S-J)SINC-1
@DEV (S-S-J)SINC-2
@DEV (S-S-J)SIND
- END OF COMPILATION
0 ERRORS DETECTED
@CONT
```

```
@DEV (S-S-J)SINE-2,CX-OB1,CX-OB1
@DEV (S-S-J)SINE-3
- END OF COMPILATION
0 ERRORS DETECTED
@CONT

@DEV (S-S-J)SINF,CX-OB4,CX-OB4
- END OF COMPILATION
0 ERRORS DETECTED
@CONT

@DEV (S-S-J)SSCOM,CX-OB2,CX-OB2
@DEV (S-S-J)COS-TAD-RES-CODE
@DEV (S-S-J)SINB-X
@DEV (S-S-J)CDR1
@DEV (S-S-J)CDR2
@DEV (S-S-J)XMSG-SYSTABS
@DEV (S-S-J)CDR3
- END OF COMPILATION
0 ERRORS DETECTED
@CONT

@DEV (S-S-J)EX-MRES-SINA,CX-OB32,CX-OB32
@DEV (S-S-J)MRES-SINI
@DEV (S-S-J)MRES-SEGADM
@DEV (S-S-J)ND500-DRIVER-1
@DEV (S-S-J)ND500-DRIVER-2
@DEV (S-S-J)MRES-CDR1
@DEV (S-S-J)MRES-CDR2
@DEV (S-S-J)MRES-CDR3
@DEV (S-S-J)MRES-SSCOM
@DEV (S-S-J)COS-TAD-POF-CODE
- END OF COMPILATION
0 ERRORS DETECTED
@CONT

@DEV (S-S-J)PIT3-SEGMENT,CX-PIT3-OBJ,CX-PIT3-OBJ
- END OF COMPILATION
0 ERRORS DETECTED
@CONT

@DEV (S-S-J)PIT3-PIT0-CODE,CX-PIT3-PIT0-OBJ,CX-PIT3-PIT0-OBJ
- END OF COMPILATION
0 ERRORS DETECTED
@CONT

@DEV (S-S-J)PIT3-POF-CODE,CX-PIT3-POF-OBJ,CX-PIT3-POF-OBJ
- END OF COMPILATION
0 ERRORS DETECTED
@CONT

@DEV (S-S-J)EX-START-SINTRAN,CX-OBJ-START,CX-OBJ-START
```

```
- END OF COMPILATION
0 ERRORS DETECTED
@NPL

NORD PL APRIL 1977

@DEV (S-S-J)SIN-DATA
@DEV (S-S-J)NPL-STCDRIV,STC-DRIV,STC-DRIV
- END OF COMPILATION
0 ERRORS DETECTED
@CONT


@DEV (S-S-J)WINCHESTER:NPL,WINCH-8,WINCH-8
- END OF COMPILATION
0 ERRORS DETECTED
@CONT


@DEV (S-S-J)BFDIS-DRIVER:NPL,BFDIS-DRIVER,BFDIS-DRIVER
- END OF COMPILATION
0 ERRORS DETECTED
@FMAC

- MACF -
IMAGE-FILE ;
XXX=10000;YYY=200;ZZZ=7000; )9TABL XXX YYY XXX
0/126306  0
0/000000  CORES
CXCPU
)SYSDF
)9ASSM FILELIST
 **** 000000 DIAGNOSTICS ****


)9SLPL
)9ASSM CX-OBJ,CX-SINLIS
 **** 000000 DIAGNOSTICS ****

)9RLPL

)9ASSM 1,0,CX-BINSIN:SYMB

CSSLO:002034
0<CSSLO
)BPUN ZRO

RESI2:006355
CSSL:021757
RESI2<CSSL
)BPUN ZRO

BGSYS:022000
CONTX:033774
BGSYS<CONTX
)BPUN ZRO

BCSTA:110000
OPEND:172205
BCSTA<OPEND
```

```
)BPUN ZRO

)9ASSM 1,,CX-SULIST:SYMB

)ULIST

)9ASSM 1,,CX-SLIST:SYMB

)LIST

)9EXIT
@CONT

)CLEAR
)9ASSM CX-SLIST,0,0
 **** 000000 DIAGNOSTICS ****


)9SLPL
)9ASSM CX-OB1,CX-SIN2LIS,0
 **** 000000 DIAGNOSTICS ****

)9RLPL

)9ASSM 1,0,CX-BIN2SIN:BPUN

% UNDEFINED SYMBOLS IN SERVICE-PROGRAM AND MAIL.
% ALL SYMBOLS FROM RESIDENT,SYSTEM-SEGMENT AND OPCOM SEGMENT
% SHOULD BE DEFINED
?
TSLST ASBPR AEBPR MBSPR EN000 EN200
EN300 EN400 EN500 EN600 EN700 E1000
E1100 E1200 E1300 E1400 E1500 E1600
E1700 E2000 E2100 E2200 E2300 E2400
BACKT F1204 BPTMP GBTIN BPTMP 5NASE
MAX10 MAX11 MAX12 MAX13 ITB10 ITB11
ITB12 ITB13 ITE10 ITE11 ITE12 ITE13
9SMRE SEMSE TMRTA ETMRT BACKT XGTDF
XGTDF BACKT DIRTA NAMTA IOXTA CERNE
5DBFL DIRTA ENDDT SPTAB ENDSP  GET0
 GET1  GET2  GET3  GET4  GET5 CCTAB
HEAPR TRAPR SPTAB ENDSP SSPTA XSTDF
XGTDF CCSTA CCTAB CCNOX XGTDF IOBUT
TIOBU PRVTT XGTDF SGAND  SGOR CUMTA
GSYSI COPYB COPYB GNEXM GUSEN GMUSI
GDATE COPYB XSTDF GDATE XGTDF GDATE
BACKT TMRTE XSTDF XGTDF TUSSY GUSEN
TUSSY COPYB GUSEN LDATE LAMAR FXCTA
NSWPA NINIT LAMAR NINIT LAMAR LAMAR
GPIBH USIZE DSIZE GPIBT R3BUF G3BUF
DBTRA DBTRA R3BUF BACKT XGTDF GUSEN
BACKT PRVTT XGTDF XGTDF XGTDF GUSEN
 BADM TSLBR TSLPR TSLNE TSLLP TSLSY
TSLTU TSLLO TSLHT TSLHA TSLES TSLBR
TSLPR TSLTI TSLNE UPLTS TSLSY TSLTU
TSLLO TSLHT TSLHA TSLES TSLPR TSLTI
TSLBR TSLNE BACKT GBTIN APRVT MBPRV
ASBPR AEBPR MBSPR XGTDF GTSLP TSLST
XSVTS FPTSL TSLST XSVTS PRVTT XGTDF
DFPTS TSLST XGTDF DFPTS DFDIL XDILF
```

```
XP131 DILBU XDILF DILBU XP131 DILBU
DILBU XDILF XP131 XDILF XP131 DILBU
DILBU XDILF XP131 MCLGF MCLGO MCLGB
TNMCA CMCLG MCLGF MCLGO MCLGF MCLGO
CMCLG MCLGB TNMCA CSWLG TFPFS CFPFS
TPFRS CPFRS TPFL4 CPFL4 TPFL1 CPFL1
SNWPS SWLGO CCSWL TFPFS TPFRS TPFL4
TPFL1 SNWPS CFPFS CPFRS CPFL4 CPFL1
CSWLG SWLGO CCSWL CCSWL TFPFS CFPFS
TPFRS CPFRS TPFL4 CPFL4 TPFL1 CPFL1
SNWPS USIOX NSWPA NINIT CUMTA BPTMP
PRVTT SBPRT EBPRT BPTMP APRVT MBPRV
 SG41 ONTTM TTMWA BPTMP BPTMP ONTTM
TTMWA ASBPR AEBPR MBSPR BACKT BBCHT
EBCHT XGTDF GBTIN APRVT MBPRV

OP2BG:110000
OP2EN:173774
OP2BG<OP2EN
)BPUN ZRO

)9ASSM 1,,CX-ULIST22:SYMB
)ULIST

)9ASSM 1,,CX-LIST22:SYMB
)LIST

)9TSS
@CONT

)CLEAR
0<177777
)ZERO
173777/000000  OP2BG;OP2EN            % TO DEFINE LIMITS FOR OP2.SEGMENT
)9ASSM CX-SLIST,0,0
 **** 000000 DIAGNOSTICS ****

)SYSDF
)9ASSM CX-LIST22
 **** 000000 DIAGNOSTICS ****


)9SLPL
)9ASSM CX-UB4,CX-ESINLIST
 **** 000000 DIAGNOSTICS ****

)9ASSM 1,0,CX-BINESIN:BPUN
ERRFL:110000  ;EESYS:123033
ERRFL<EESYS
)BPUN ZRO

)9ASSM 1,,CX-ESULIST:SYMB
)ULIST
)9ASSM 1,,CX-ESLIST:SYMB
)LIST
1,0,0$

%=================================================================
5BFPA:000076                          % BUFFER WINDOW PAGE
00000:000000  CSSLO:002034            % SINTRAN RESIDENT LOWER ADDRESS PAG  (<5BFPA)
```

```
RESI2:006355  CSSL:021757                  % SINTRAN RESIDENT CONFIG. INDEPENDANT
BGSYS:022000  SYEND:025770                 % SINTRAN (BG PROCESS.) SYSTEM SEGMENT PART
BGSYS:022000  CONTX:033774                 % ENTIRE SYSTEM SEGMENT
BCSTA:110000  OPEND:172205                 % COMMAND SEGMENT
OP2BG:110000  OP2EN:173774                 % SIN-SERV-PROG AND MAIL SEGMENT
ERRFL:110000  EESYS:123033                 % ERROR PROGRAM SEGMENT
%==================================================================
)9EXIT

 BATCH USER LOGGED OFF AT 15.56.37      23 SEPTEMBER    1984
```

TIME USED IS 39 MINS 10 SECS OUT OF 46 MINS 19 SECS

% :: Part One numeric Symbol List

```
=========================================================================
    SINTRAN III/VSX VERSION I                  16.14.36     23 SEPT  1984
=========================================================================

XXXXX     XXXX    XXXXX   XXXXXX         XXXX    XX  XX  XXXXX
XXXXXX   XXXXXX   XXXXXX  XXXXXX        XXXXXX   XX  XX  XXXXX
XX  XX   XX  XX   XX  XX     XX         XX  XX   XXX XX  XX
XXXXXX   XX  XX   XXXXX      XX    XXXX  XX  XX   XXXXXX  XXXX
XXXXX    XXXXXX   XXXXX      XX    XXXX  XX  XX   XXXXXX  XXXX.
XX       XXXXXX   XX  XX     XX         XX  XX   XX XXX   XX
XX       XX  XX   XX  XX     XX         XXXXXX   XX  XX   XXXXX
XX       XX  XX   XX  XX     XX          XXXX    XX  XX   XXXXX
=========================================================================
    SINTRAN III/VSX VERSION I                  16.14.37     23 SEPT  1984
=========================================================================
```

SYMBOL LIST WITH VALUES IN ASCENDING ORDER.

```
=========================================================================

P0    =000000   S0    =000000   S90   =000000   DS0   =000000   DDR   =000000
GSI   =000000   DY0   =000000   50K   =000000   DOU0  =000000   BENA  =000000
FRES  =000000   LMPP  =000000   LMCN  =000000   BITR  =000000   READ  =000000
RING1=000000    XMSGX=000000    TLINK=000000    DPREG=000000    XTDPR=000000
BXTRG=000000    XTADR=000000    RESLI=000000    BMEM0=000000    5ECH0=000000
ZROUS=000000    TDRAD=000000    5TLRE=000000    5BPAS=000000    5TDUM=000000
BECH0=000000    CNAME=000000    RPREG=000000    DIERC=000000    SECW0=000000
MCMAG=000000    7CUEL=000000    BBPRO=000000    BPRTM=000000    TXBBP=000000
SVBPR=000000    TXSVB=000000    DIL1C=000000    7DATA=000000    EMPTY=000000
FSEND=000000    X21L0=000000    XCHAI=000000    X21C1=000000    X21SA=000000
X21NL=000000    XRRDR=000000    X2FOK=000000    NPTIG=000000    XPKIC=000000
DTRIG=000000    MBREA=000000    LAMPP=000000    LAMCN=000000    VDPGL=000000
VDCPF=000000    DKNA1=000000    URFIL=000000    DIRFL=000000    DVNAM=000000
BDATA=000000    BDIRI=000000    ULOCK=000000    MESSF=000000    OLOCK=000000
0TBIT=000000    CXNOC=000000    XFDUM=000000    XRSOK=000000    SLAKK=000000
7NCML=000000    SEGLI=000000    PAGLI=000000    DPAGL=000000    5UEM0=000000
1ILLC=000000    P1    =000001   S1    =000001   DDW   =000001   ANI   =000001
,DA   =000001   POK   =000001   BIML  =000001   DPIN  =000001   ITOM  =000001
BBID  =000001   TRIG  =000001   BITW  =000001   ALEVL=000001    RING2=000001
STATU=000001    DXREG=000001    TDXRE=000001    BXARG=000001    RTRES=000001
2INBT=000001    5BREA=000001    ZCTTY=000001    XDFOP=000001    CXTST=000001
5BCOM=000001    5ESC0=000001    5TLOW=000001    BOTTE=000001    CPARA=000001
RXREG=000001    SECTR=000001    MC5SP=000001    CBPTE=000001    BPCFI=000001
SVTSL=000001    DIL1U=000001    7DBRE=000001    7BDAT=000001    FRECV=000001
X21C2=000001    X21SC=000001    XWPCR=000001    X2FSM=000001    NPFUN=000001
NPOPC=000001    MCONT=000001    FFREL=000001    MSGN5=000001    LAMNP=000001
LAMLP=000001    VDPGR=000001    DKNA2=000001    FRACC=000001    UWFIL=000001
,WRIT=000001    DUNIT=000001    BPART=000001    UDIRI=000001    BROAF=000001
0DIRI=000001    OPBIT=000001    SPROG=000001    CXSPM=000001    XFDCT=000001
XDINF=000001    XMTN0=000001    XRISN=000001    XXEIE=000001    7ZMEM=000001
BPAGL=000001    5DEMA=000001    ALOGN=000001    5UETE=000001    P2    =000002
S2    =000002   DS1   =000002   DST   =000002   GRI   =000002   ,SA   =000002
```

```
DY1  =000002   ALEV =000002   DOU2 =000002   7RFI =000002   XPTN =000002
BNDC =000002   BITA =000002   5FIX =000002   5CIMS=000002   FIXED=000002
ALMSZ=000002   DTIM1=000002   DTIME=000002   DTREG=000002   BXBRG=000002
BWLIN=000002   5BWLI=000002   5CLDV=000002   2OUTB=000002   BSAVE=000002
5SPEC=000002   ZCESC=000002   5BUSE=000002   5ESC2=000002   5THIG=000002
BITER=000002   CMAND=000002   RTREG=000002   SECSY=000002   BPRFL=000002
BPWAR=000002   PRVUE=000002   DILLI=000002   7DECO=000002   FULLN=000002
MXMEL=000002   FMXRE=000002   BBYTC=000002   X21C3=000002   X21SB=000002
XRRSR=000002   X2FIL=000002   MMEMO=000002   PPROM=000002   FKICK=000002
N5STA=000002   LAMPR=000002   USBLS=000002   LUNIT=000002   BLOCN=000002
UPART=000002   OPART=000002   OSBIT=000002   SPERI=000002   CXFNA=000002
CSPFN=000002   XFGET=000002   XDSBP=000002   XMROU=000002   XRUNN=000002
XXIOW=000002   DSKTY=000002   7ZSCR=000002   LOGAD=000002   PAGPR=000002
PAGPH=000002   5UEIE=000002   P3   =000003   S3   =000003   RA   =000003
S91  =000003   IRI  =000003   BSEG =000003   OSEG =000003   BITC =000003
ER71 =000003   MADR =000003   MLEVL=000003   5OPSE=000003   LDTSZ=000003
DTIM2=000003   DAREG=000003   BXLRG=000003   TYPRI=000003   5NORE=000003
2ECHO=000003   DCONT=000003   5HDUP=000003   5LOGO=000003   5BESC=000003
5TREA=000003   BXOTT=000003   CPROT=000003   RAREG=000003   MAXCY=000003
IMCBP=000003   BPSOK=000003   PRVMA=000003   DILRA=000003   7RQIN=000003
FULLS=000003   7ECKM=000003   BMBYT=000003   X21C4=000003   XWSAR=000003
X2FIC=000003   PNOPC=000003   MOPEN=000003   FWKIC=000003   UMAGT=000003
COMPA=000003   DLOCK=000003   BBUFF=000003   UINDP=000003   OINDP=000003
OIBIT=000003   SQUEU=000003   XFREL=000003   XDPUS=000003   XMTHI=000003
XRDDF=000003   XXBIN=000003   LOADI=000003   7ZFPA=000003   5INHB=000003
EPAGP=000003   5UETL=000003   P4   =000004   S4   =000004   DS2  =000004
DY2  =000004   5BAD =000004   DOU4 =000004   DACT =000004   GOOD =000004
PCNT =000004   BITD =000004   BITX =000004   FLAG =000004   BLEVL=000004
5RTSG=000004   MMESI=000004   DTIN1=000004   DTINT=000004   DDREG=000004
XUSEG=000004   ISTAT=000004   2BRKM=000004   5FIMO=000004   ZTSPE=000004
5ABJO=000004   K5ESC=000004   5ERRO=000004   5CONT=000004   RDREG=000004
DIEWC=000004   POLSY=000004   IMBPR=000004   PRVFL=000004   DILWA=000004
7IRQI=000004   7BMMX=000004   BUDIS=000004   BCHAI=000004   XWCHL=000004
X2FIS=000004   MIOIN=000004   FLOAD=000004   SRTCS=000004   VDPFU=000004
URBYP=000004   DNUMB=000004   DRESE=000004   RWFIE=000004   OCBIT=000004
SQSEM=000004   XFRHD=000004   XDPUR=000004   XMTRE=000004   XRNSP=000004
XXMCE=000004   BACKG=000004   7NPAS=000004   5SVSE=000004   5UERU=000004
P5   =000005   S5   =000005   ISI  =000005   BAD  =000005   5RQI =000005
BITN =000005   5ERRS=000005   DTIN2=000005   DLREG=000005   XURSE=000005
XTRSE=000005   MLINK=000005   5TERM=000005   5CTRL=000005   ZCNTR=000005
5LBLO=000005   5REMO=000005   5WCON=000005   REFOR=000005   BTBIN=000005
PRVFB=000005   DILSM=000005   7CONF=000005   BHEAD=000005   X2MFU=000005
XWTDR=000005   X2FNM=000005   REGPO=000005   MPIOC=000005   FUNLO=000005
VDPME=000005   ERRSE=000005   USBLP=000005   CPAG1=000005   CPAGE=000005
PART1=000005   DNAME=000005   OFACC=000005   UINDX=000005   OINDX=000005
OABIT=000005   SQIOS=000005   XFWHD=000005   XDGES=000005   XMKIK=000005
XRIPT=000005   XXIEN=000005   DEMAN=000005   5SPRO=000005   5SEGS=000005
5UEPR=000005   S6   =000006   DS3  =000006   DY3  =000006   DOU6 =000006
FULL =000006   X2M1 =000006   XNTP =000006   FSEG =000006   MTOR =000006
5FILS=000006   STADR=000006   DSREG=000006   XUBIM=000006   DUBIM=000006
BXBIT=000006   MFUNC=000006   5IBDV=000006   5WECH=000006   5WRQI=000006
5ALEC=000006   5LCHA=000006   5RERU=000006   RSREG=000006   RESCY=000006
5SPRF=000006   BPLOG=000006   YSVBP=000006   DAC1C=000006   7CORQ=000006
X2MST=000006   XRTSR=000006   X2FNR=000006   FSTAR=000006   UOPEN=000006
CPAG2=000006   LWBIT=000006   OFFTP=000006   OMBIT=000006   SSTOP=000006
XFREA=000006   XDGER=000006   XMTPS=000006   XRMMP=000006   XXIFL=000006
5SREE=000006   5UECM=000006   S7   =000007   ,ST  =000007   SEGM =000007
5REG =000007   5XON =000007   CHKO =000007   PWCR =000007   5MACD=000007
DBREG=000007   XUBI1=000007   TRLRE=000007   5INVR=000007   5HDMA=000007
ZPREG=000007   2RPAG=000007   IOTRA=000007   5USES=000007   5CFIL=000007
```

```
ALTFO=000007   5BRKF=000007   TTMCO=000007   YSVTS=000007   DAC1U=000007
BOXNO=000007   BUFUL=000007   7CORS=000007   X2MFA=000007   X2M1B=000007
X21BL=000007   DCE19=000007   XWTCR=000007   X2FNI=000007   PIOCI=000007
FSTOP=000007   PMFUN=000007   VDPDI=000007   UCLOS=000007   LNUMB=000007
5PHOE=000007   OFFLG=000007   OLBIT=000007   SABOR=000007   XFWRI=000007
XFSYS=000007   XRUNM=000007   XXIRT=000007   RTREF=000007   5UEST=000007
S10  =000010   DS4  =000010   DY4  =000010   MLEV =000010   CCLR =000010
X2M2 =000010   PRSR =000010   LMLP =000010   LMNP =000010   ALEVB=000010
5RTFI=000010   USDVS=000010   WLINK=000010   BITMA=000010   5WLIN=000010
5BITM=000010   XTDXR=000010   XUBI2=000010   DUBM2=000010   HSTAT=000010
5FLOP=000010   ZXREG=000010   2WPAG=000010   RTCLD=000010   STDEV=000010
5XOFF=000010   TDOND=000010   5ESCL=000010   5ND5E=000010   DIERO=000010
DISPN=000010   5ESCF=000010   9IVAL=000010   DIOUV=000010   TXCBP=000010
5PRVT=000010   TXSVT=000010   DILST=000010   MAINF=000010   IGNAC=000010
7ESCA=000010   X2MCP=000010   XRRTS=000010   X2F01=000010   SLMAX=000010
VDCFU=000010   UDROB=000010   ADVAN=000010   DBUFB=000010   DBPAG=000010
PAVA1=000010   PAVAI=000010   5RCFI=000010   OFBUF=000010   OFNBR=000010
UOFLG=000010   OTMBI=000010   BBASP=000010   SPINX=000010   XFSCM=000010
XRMTL=000010   XXNBF=000010   CURPR=000010   DALOG=000010   5UEES=000010
5MT  =000011   S11  =000011   GSO  =000011   X2M3 =000011   5LOGR=000011
ACTSE=000011   BITM1=000011   XUBI3=000011   MTRAN=000011   5CRDL=000011
ZTREG=000011   ZTADR=000011   2TIME=000011   KSETD=000011   SETDV=000011
5XDEV=000011   5WESC=000011   5LOGI=000011   3FLOP=000011   5NOSL=000011
9OUVA=000011   DILDE=000011   ANCHA=000011   7DCON=000011   X2MSL=000011
X2MCL=000011   X21PM=000011   XWRTC=000011   X2FNC=000011   KPROS=000011
PIOCO=000011   VDPNU=000011   UMROB=000011   REVER=000011   BUFFE=000011
PAVA2=000011   BDBIT=000011   5REMP=000011   OFLIB=000011   OFLOC=000011
OFPDI=000011   BFSSP=000011   HEADE=000011   XFMST=000011   XFSEC=000011
XRSMF=000011   XXRIN=000011   MQUEU=000011   5UETM=000011   S12  =000012
DS5  =000012   DY5  =000012   X2M4 =000012   WEOF =000012   OFCB =000012
5IFS2=000012   CUMSI=000012   ACTPR=000012   BITM2=000012   XUBI4=000012
5TMOU=000012   M144B=000012   ZAREG=000012   2SETO=000012   DFOPP=000012
5OXON=000012   5WLOC=000012   3STRE=000012   7CUTY=000012   1DTU1=000012
1TUSE=000012   DILBP=000012   1DILB=000012   CDFIL=000012   7SVSI=000012
X2MSS=000012   XRTTS=000012   X2FFF=000012   PINIT=000012   XPIPR=000012
VDLIN=000012   UDWOB=000012   DBLOC=000012   SECTO=000012   ECCBT=000012
OFSCR=000012   BNFIQ=000012   TRAIL=000012   SPLEN=000012   XFOPN=000012
XFROU=000012   XRPRV=000012   XXNMM=000012   BTIMQ=000012   5UELO=000012
S13  =000013   5ABS =000013   5COM =000013   IINI =000013   7LUN =000013
X2M5 =000013   OFNB =000013   5PGU =000013   5IRTS=000013   BRESL=000013
BITM3=000013   5BRES=000013   XUBI5=000013   ZDREG=000013   2CIBU=000013
DERRO=000013   5CAPI=000013   5LOC2=000013   STDEL=000013   3FLTI=000013
1DTU2=000013   DILBA=000013   DDIBA=000013   2DILB=000013   RSICH=000013
SINIT=000013   MINFR=000013   OFSET=000013   X2MS2=000013   XWTTC=000013
X2FNP=000013   PIOCA=000013   PIPOW=000013   VDCST=000013   UMUID=000013
REWIN=000013   DBLO1=000013   DBLOA=000013   DTUSE=000013   NFLAG=000013
MTBIT=000013   INDB2=000013   OCOMM=000013   ,PRIN=000013   XFCLS=000013
XFFWD=000013   XRISY=000013   XXNIM=000013   BEXQU=000013   5UEAL=000013
PGC  =000014   S14  =000014   DY6  =000014   5INT =000014   PRIO =000014
5WIP =000014   ECODS=000014   5N100=000014   RSEGM=000014   BITM4=000014
XUBI6=000014   5ISET=000014   ZLREG=000014   2COBU=000014   BUFST=000014
6XOFF=000014   5FFGP=000014   MEMA1=000014   MEMAD=000014   DIEWO=000014
3FRES=000014   5DTU1=000014   5TUSE=000014   2DIBA=000014   DILCO=000014
LIBEG=000014   NPFBU=000014   BSINI=000014   MSSTA=000014   BUFFI=000014
7TMOD=000014   X2FAB=000014   MASTA=000014   VDDST=000014   UMRUS=000014
ERASE=000014   DBLO2=000014   DDAUF=000014   10BIT=000014   PTRNS=000014
INDB1=000014   OFBLZ=000014   OBACK=000014   FILEN=000014   XFSND=000014
XFOPS=000014   XFRES=000014   XFBNC=000014   XRNRO=000014   XXCLS=000014
BSEGL=000014   S15  =000015   MAX  =000015   ECCR =000015   5ISMS=000015
5SEXI=000015   WINDO=000015   BITM5=000015   XUBI7=000015   5RWAI=000015
```

```
5SWWA=000015   5CONC=000015   BISYM=000015   ZSREG=000015   5FRER=000015
5IESC=000015   USIDX=000015   5MLGI=000015   MEMA2=000015   3DOUA=000015
7SATY=000015   5DTU2=000015   DIL1D=000015   DILDA=000015   DILBO=000015
SWICH=000015   PLMSG=000015   UANTM=000015   TDTAF=000015   TDTAD=000015
7TTYP=000015   X2FCP=000015   PNBOX=000015   DROPS=000015   VDMBO=000015
VDMBX=000015   UGUSN=000015   BACKS=000015   DKFUN=000015   DTAPE=000015
OBFIL=000015   1USER=000015   NLOCK=000015   OFWRT=000015   OFDIR=000015
ORESE=000015   LASTP=000015   XFRCV=000015   XFPON=000015   XFWOK=000015
XFUSG=000015   XFHIP=000015   XFRRO=000015   XRIIV=000015   XXCHE=000015
SGMAX=000015   S16  =000016   DY7  =000016   5REP =000016   IDLE =000016
HPEK =000016   INDX =000016   5INNE=000016   RTDLG=000016   BITM6=000016
5RTOF=000016   5RFIL=000016   BIHDL=000016   ZBREG=000016   DOLDP=000016
BHOLD=000016   5PAER=000016   5LBRK=000016   5SPSI=000016   CMAD1=000016
CMADR=000016   3DOUB=000016   BPRTS=000016   DIL2D=000016   NPHBU=000016
LMDAT=000016   PFMSG=000016   TDTAL=000016   7CESC=000016   X2FEC=000016
PKICK=000016   XNDPR=000016   VDMB1=000016   UGUIO=000016   FORSP=000016
DPNTO=000016   DMAIN=000016   DRBIT=000016   NTLEN=000016   INFLG=000016
OFOBJ=000016   OWRTE=000016   PAGEN=000016   XFPST=000016   XFWAK=000016
XRNEI=000016   XXNOR=000016   USEGM=000016   S17  =000017   ISO  =000017
PTN  =000017   OFIP =000017   5WPM =000017   5IPT3=000017   BITM7=000017
5RTSI=000017   5WAIT=000017   5BACK=000017   5IOBT=000017   OLDPA=000017
HENTE=000017   5BLOC=000017   5BFUL=000017   5RDEV=000017   MXTIM=000017
5LSTA=000017   CMAD2=000017   3ILLF=000017   DILNS=000017   CUDBU=000017
CURBU=000017   TDBTP=000017   7DESC=000017   X2FAC=000017   BCFLA=000017
VDCNR=000017   VDUNI=000017   UCRAL=000017   REWUN=000017   DPNT1=000017
DENTE=000017   USFIL=000017   CTBIT=000017   PERMF=000017   OFIP1=000017
UUSED=000017   OUSED=000017   SUBIN=000017   REMAI=000017   XFGST=000017
XFWTF=000017   XRNXM=000017   XXICM=000017   ND500=000017   D0   =000020
D90  =000020   S20  =000020   NTP  =000020   DY8  =000020   ER0  =000020
BLEV =000020   XRSA =000020   LMPR =000020   55SDS=000020   ARSEG=000020
XTDTR=000020   5XRTD=000020   CFREE=000020   BCHNU=000020   CLEDE=000020
M2UNT=000020   DIERT=000020   IOLOG=000020   TXBPR=000020   TXUEF=000020
DIL1L=000020   DILLA=000020   LASBU=000020   SOHSC=000020   SLIPO=000020
CHNST=000020   7USID=000020   X2FBR=000020   SIB50=000020   SICCO=000020
VDABS=000020   VDCME=000020   UMDLF=000020   DPNT2=000020   MBLEN=000020
OFIP2=000020   BYTEN=000020   XFSIN=000020   XRILN=000020   XX100=000020
DPGPR=000020   D1   =000021   S21  =000021   XON  =000021   ER1  =000021
LRSA =000021   OFOP =000021   55NSG=000021   AACSE=000021   FYLLE=000021
HTABL=000021   PVEFU=000021   BADTA=000021   WFLAG=000021   DIL2L=000021
N5RDF=000021   INIAD=000021   IDBST=000021   7PASS=000021   PIPRO=000021
KICKP=000021   SIAD1=000021   SIDRT=000021   ULIOP=000021   DPNT3=000021
BIFIL=000021   OFOP1=000021   BUFHA=000021   XFSRL=000021   XRNXL=000021
XXMON=000021   D2   =000022   RTIN =000022   ER20 =000022   5FIUS=000022
MINBH=000022   BSTAT=000022   CLOGD=000022   RHSTA=000022   ICORA=000022
DILGF=000022   DFPNT=000022   POOLP=000022   NDPRO=000022   XNMAI=000022
PIMBH=000022   SIAD2=000022   USBYP=000022   PNEXT=000022   OFOP2=000022
PCBNK=000022   PCORA=000022   XFABR=000022   XRNXD=000022   XXMMC=000022
D3   =000023   ER2  =000023   XOFF =000023   SEND =000023   5OP2S=000023
MAXBH=000023   TSTAT=000023   6TTYN=000023   ROFIL=000023   DFDEV=000023
IBLOA=000023   DILFL=000023   DDILF=000023   REMBY=000023   7SYCN=000023
PNMAI=000023   PIMBL=000023   O2SEG=000023   UINIT=000023   PPREV=000023
PLEFT=000023   OFIND=000023   OFIOD=000023   OFPAG=000023   BBLEN=000023
PCADR=000023   XFABW=000023   XRNTR=000023   XXFBI=000023   D4   =000024
.SP  =000024   ER3  =000024   RTBH =000024   FRSG1=000024   NINSZ=000024
NNSWS=000024   RDLOO=000024   IMAXB=000024   DBPRO=000024   BCHOS=000024
DIEWT=000024   IMAXW=000024   DILFU=000024   BSEND=000024   7USCN=000024
NPMAI=000024   URSOF=000024   SECTP=000024   OFOUD=000024   SNPAG=000024
XFMLK=000024   XRTRA=000024   XXPER=000024   RTSTA=000024   D5   =000025
CL7  =000025   ER4  =000025   RTBL =000025   SSEG =000025   OFFP =000025
SPSEG=000025   FRSG2=000025   ON5MS=000025   DBADR=000025   IFUNC=000025
```

```
DLLOG=000025   IDBAD=000025   RECEI=000025   MOTRG=000025   7FBSI=000025
PREQU=000025   URLFE=000025   DBLEN=000025   NMUSM=000025   SURUS=000025
UENTE=000025   OFLAG=000025   SNLIN=000025   XFMUL=000025   XRTRP=000025
XXILN=000025   SEGST=000025   ER5  =000026   TERM =000026   DY12 =000026
5FIU2=000026   2LAST=000026   RDHII=000026   CBUAD=000026   RIFIL=000026
DIFTC=000026   RERRC=000026   IRETW=000026   DLDRI=000026   VENTX=000026
MOLRG=000026   DRFUN=000026   7RESE=000026   X21SY=000026   PISTT=000026
XPMAI=000026   USPER=000026   NMUSD=000026   SURTM=000026   UNAME=000026
ONAME=000026   SCOND=000026   XFM2P=000026   XRTFE=000026   XXROU=000026
CORMS=000026   ER6  =000027   5NNET=000027   2RTDS=000027   NOCHA=000027
CHARI=000027   BCHIS=000027   DEDFA=000027   MTFLG=000027   DLALO=000027
RMLNR=000027   LDOTR=000027   BXTAD=000027   7RECO=000027   PIMPG=000027
USPEF=000027   DTLEN=000027   INDX1=000027   NLPAG=000027   XFP2M=000027
XRTRT=000027   XXHER=000027   CORAD=000027   UEMAX=000027   ER7  =000030
DER0 =000030   PSIZ =000030   BLST =000030   MLEVB=000030   XTDAR=000030
2GETR=000030   CNOCH=000030   BREGB=000030   5BREG=000030   DIERR=000030
MRSTA=000030   5MRST=000030   TXICB=000030   DLAUN=000030   LIPOI=000030
FRETR=000030   LDOXR=000030   ACTOU=000030   7DUMM=000030   PIOCN=000030
MLITE=000030   UMGFI=000030   SPMOD=000030   XFRIN=000030   XRTIS=000030
XXRO2=000030   SBLST=000030   DDBLS=000030   DBLST=000030   DPAGP=000030
ER8  =000031   PINO =000031   2EXIO=000031   XNOCH=000031   SSREF=000031
DIL1F=000031   DILFA=000031   MISTA=000031   LDOLR=000031   RSPNU=000031
7STRQ=000031   MBOXH=000031   MLICP=000031   UDSCN=000031   INDA1=000031
SPAGE=000031   XFCRD=000031   XRBLK=000031   XXTAS=000031   ER9  =000032
2MSG =000032   DER2 =000032   5ACCS=000032   ZOPRG=000032   WHSTA=000032
STRSE=000032   DIL2F=000032   SBYTS=000032   OSSRE=000032   7STRS=000032
X2F20=000032   CPUNR=000032   MLICA=000032   USTEM=000032   WSNLI=000032
NOSIN=000032   XFSTD=000032   XRSYD=000032   XXMER=000032   DRT  =000033
DDD0 =000033   ER10 =000033   5XSG1=000033   2ALTN=000033   DI1LA=000033
DILGL=000033   CUIBU=000033   OIFUN=000033   7KEYI=000033   X2F30=000033
RTBOX=000033   MLIMX=000033   UGDIE=000033   WSNPA=000033   NUMIN=000033
XFDIB=000033   XRNLS=000033   XXNER=000033   ECL7 =000034   DDD1 =000034
ER11 =000034   5XSG2=000034   2ALTF=000034   DIEWR=000034   MCLRG=000034
DI2LA=000034   CUUBU=000034   OICOR=000034   7BADT=000034   MLIWO=000034
PBSIZ=000034   UFDFD=000034   LFCOU=000034   XFRIB=000034   XRTRE=000034
RGSI =000035   DDD2 =000035   ER12 =000035   2IOUT=000035   ZOARG=000035
1XNDA=000035   DXNDA=000035   OIMAX=000035   X2F50=000035   MCURB=000035
MDCUR=000035   MLIAD=000035   UCOPA=000035   SPFNA=000035   XFWIB=000035
XRRNA=000035   DDD3 =000036   ER13 =000036   5BADM=000036   2NOWA=000036
WERRC=000036   2XNDA=000036   SQERR=000036   OOLDP=000036   LCURB=000036
ACCRL=000036   MLIMS=000036   UDELP=000036   UPASS=000036   OTYPE=000036
XFPRV=000036   XRBUS=000036   XXNGA=000036   PXT  =000037   DDD4 =000037
ER14 =000037   5RT2S=000037   1XNWD=000037   DXNWD=000037   RNACO=000037
7OPSV=000037   UFOBJ=000037   UDATE=000037   XFRTN=000037   XRNSE=000037
XXRI2=000037   UEBSZ=000037   ER15 =000040   BLEVB=000040   5RRUS=000040
XTDDR=000040   ZOSRG=000040   DBPRE=000040   DIECO=000040   TXIBP=000040
TXFLL=000040   2XNWD=000040   SNACO=000040   7ESRS=000040   HINIF=000040
MLIPA=000040   VDCDI=000040   USTER=000040   5FORM=000040   ONEXT=000040
XFRRH=000040   XRRPN=000040   XXINP=000040   BGFPA=000040   ER16 =000041
5PT3S=000041   DBACT=000041   DALFU=000041   CTRCH=000041   7CERS=000041
MASTB=000041   UEXPF=000041   UDENT=000041   OPREV=000041   XFDUB=000041
XRUKS=000041   XXBER=000041   BGLPA=000041   FCST =000042   ER17 =000042
5ISPS=000042   FLAGB=000042   DALCM=000042   BADAN=000042   7ISRQ=000042
XBBNK=000042   URENF=000042   RFORM=000042   TSPTR=000042   OACCE=000042
XFWDF=000042   XRMFL=000042   XXRUT=000042   RTFPA=000042   SMI  =000043
RTUT =000043   ER18 =000043   SPSGM=000043   2CLOS=000043   2CLSE=000043
SBHOL=000043   TDOPD=000043   EUSAD=000043   USDAD=000043   7ISRS=000043
XBSTR=000043   URTLI=000043   UPAVA=000043   OFTYP=000043   XFDBK=000043
XRROV=000043   XXNXM=000043   RTLPA=000043   SMO  =000044   ER19 =000044
TOIPD=000044   LUSAD=000044   DIESP=000044   DALCD=000044   MITRG=000044
```

```
OBCOU=000044   7NOWT=000044   XBEND=000044   ODEVN=000044   XFSMC=000044
XRRFU=000044   XXNLD=000044   CCFPA=000044   ER21 =000045   TDIND=000045
ACQFP=000045   MILRG=000045   TSTAD=000045   7TNOW=000045   XBSAV=000045
MLIST=000045   UPUS1=000045   UPUSE=000045   OUSER=000045   XFDMM=000045
XRNRB=000045   XXUSR=000045   CCLPA=000045   ER22 =000046   ACQHP=000046
LDITR=000046   REMSI=000046   7NWRE=000046   XXUBF=000046   UPUS2=000046
ONDEX=000046   XFALM=000046   XRURT=000046   XXN10=000046   SYSNO=000046
ER23 =000047   ACQBH=000047   LDIXR=000047   CURME=000047   7RLOC=000047
XXSBK=000047   UNDEX=000047   OCOUN=000047   XFFRM=000047   XRSNR=000047
XXNSG=000047   HWINF=000047   7IAM =000050   ER24 =000050   XTDLR=000050
2NOPE=000050   DIUEX=000050   IBYTS=000050   LDILR=000050   NOBDI=000050
XXSBF=000050   MAILF=000050   OOPEN=000050   X5FUN=000050   XRRND=000050
XXNI2=000050   ER25 =000051   CURID=000051   ISSRE=000051   POOLL=000051
7EDRS=000051   XXHOM=000051   INDX2=000051   DFIAC=000051   ODATC=000051
XRNNA=000051   XXMIN=000051   SPRS =000052   ER26 =000052   2MTER=000052
IIFUN=000052   7TREP=000052   FIXID=000052   X21ST=000052   USPRV=000052
XRISE=000052   XXTAI=000052   SINVE=000052   RPRS =000053   ER27 =000053
IICOR=000053   SWBUF=000053   X21PL=000053   INDA2=000053   USNXT=000053
ODATR=000053   STORX=000053   XRIRQ=000053   XXWS3=000053   ACQU =000054
ER28 =000054   TDOSI=000054   DIUET=000054   IIMAX=000054   LWPHY=000054
X21KO=000054   UFREE=000054   BUFFA=000054   XRNGA=000054   XXN33=000054
REVLE=000054   ER29 =000055   IOLDP=000055   XWAIT=000055   UFRIE=000055
ODATW=000055   FPAR1=000055   XRRNL=000055   XXVSX=000055   GENDA=000055
ER30 =000056   IMTFL=000056   MESSI=000056   FPAR2=000056   XRNCO=000056
XXNCX=000056   ER31 =000057   XUNIT=000057   RSRET=000057   X21SL=000057
OPAGE=000057   FPAR3=000057   XXWSY=000057   ER32 =000060   XTDSR=000060
5WORK=000060   BTYPR=000060   TXXSV=000060   ILSAV=000060   RSCUR=000060
PRSCU=000060   VDCNU=000060   DREAD=000060   NBYTS=000060   ER33 =000061
OLSAV=000061   IQUEU=000061   DWRTE=000061   OBYTE=000061   LNKSP=000061
CXSPJ=000061   CSPJ1=000061   LINR =000062   ER34 =000062   2RMAX=000062
BPRCL=000062   OQUEU=000062   SAVFN=000062   CSPJ2=000062   XXILL=000062
IDNTS=000062   ER35 =000063   XNOWU=000063   DNACO=000063   WAKEF=000063
DCOMP=000063   OPOIN=000063   STORT=000063   CSPJ3=000063   2ERM =000064
HXCC =000064   ER36 =000064   2ERMS=000064   5WCBU=000064   UNRST=000064
BUSTA=000064   BUALL=000064   DBHCA=000064   LIBSI=000064   ER37 =000065
2QERM=000065   NEWUN=000065   X2DBR=000065   SOUR1=000065   SOURX=000065
ER38 =000066   2ISIZ=000066   2XIBU=000066   SELUN=000066   X2DLS=000066
SOUR2=000066   EXTDS=000066   ER39 =000067   2OSIZ=000067   COMCO=000067
BUSIS=000067   X2DSR=000067   RFIEL=000067   ER40 =000070   XTDBR=000070
2CMND=000070   BUFAD=000070   X2DSL=000070   PBBYT=000070   VDCHS=000070
SBUFR=000070   NOCOP=000070   ER41 =000071   2DESC=000071   X2DUI=000071
FSPME=000071   SPUME=000071   ER42 =000072   2EESC=000072   X2DUO=000072
FNAME=000072   NAME1=000072   TABLE=000072   ER43 =000073   2SMAX=000073
DWONO=000073   X2DPI=000073   ER44 =000074   2SBYT=000074   INHBT=000074
X2DCN=000074   L12LG=000074   LOST =000075   ER45 =000075   2RBYT=000075
X2DST=000075   L4LGP=000075   ER46 =000076   2SBLZ=000076   COTAB=000076
X2DDF=000076   5BFPA=000076   USTAR=000076   ER47 =000077   2STBC=000077
COFLA=000077   X2DPC=000077   5UBFP=000077   L10LG=000077   UREST=000077
2RT  =000100   ER48 =000100   RETRN=000100   X2DER=000100   IDEVN=000100
FZERO=000100   XSNUL=000100   URPRO=000100   2SET =000101   ER49 =000101
AKMCH=000101   X2DPS=000101   XSLET=000101   DASAX=000101   DDASA=000101
ER50 =000102   2ABSE=000102   MXCHN=000102   X2DLA=000102   HMAXB=000102
DMAXB=000102   XSNAM=000102   ER51 =000103   2INTV=000103   IDADR=000103
X2DLI=000103   NOBYT=000103   MAXBY=000103   XSCNM=000103   HENTF=000103
ER52 =000104   POTES=000104   2HOLD=000104   X2DCI=000104   NOCHR=000104
XSGNM=000104   UNAFL=000104   ER53 =000105   2ABOR=000105   X2DC1=000105
XSGNI=000105   FLPT3=000105   ISTR =000106   ER54 =000106   2CONC=000106
DFDCW=000106   X2DC2=000106   FIXCL=000106   ER55 =000107   2DSCN=000107
X2DC3=000107   XSGMG=000107   CNTXL=000107   ER56 =000110   2PRIO=000110
X2DC4=000110   HBRST=000110   CAMID=000110   ER57 =000111   2UPDA=000111
```

```
X2DC5=000111   5BRST=000111   XSDRN=000111   9TIM0=000111   ER58 =000112
2CLAD=000112   TDISI=000112   X2DC6=000112   HECHS=000112   XSDSY=000112
XSDMC=000112   8CLCN=000112   ER59 =000113   2CLOC=000113   X2DMP=000113
X2DMM=000113   5ECHS=000113   XSGSY=000113   XSGMC=000113   ER60 =000114
2TUSE=000114   X2DRL=000114   5FYLL=000114   XSLKI=000114   2FIX =000115
ER61 =000115   X2DSP=000115   MLFLA=000115   XSTIN=000115   9TIM1=000115
ER62 =000116   2UNFI=000116   X2DOC=000116   XSTCL=000116   ER63 =000117
2RFIL=000117   X2DCC=000117   XSTDC=000117   9TIM2=000117   ER64 =000120
LV10B=000120   2WFIL=000120   X2DBC=000120   FSP00=000120   XSCRS=000120
9TIM3=000120   ER65 =000121   2WAIT=000121   X2D00=000121   X2FXX=000121
XSNSP=000121   9TIM4=000121   ER66 =000122   ABS6 =000122   2RESR=000122
X2D02=000122   XSGIN=000122   9TIM5=000122   ER67 =000123   2RELE=000123
X2D05=000123   XSDLO=000123   9TIM6=000123   ER68 =000124   2PRSR=000124
X2D06=000124   2PRES=000124   XSLEK=000124   9CL00=000124   ER69 =000125
2PRLS=000125   X2D10=000125   2PREL=000125   XSNET=000125   9CL01=000125
ER70 =000126   2DSET=000126   DFDCR=000126   X2D11=000126   XSSCI=000126
XSMAX=000126   9CL02=000126   ER72 =000127   2DABS=000127   X2D12=000127
9CL03=000127   ER73 =000130   LV11B=000130   2DINT=000130   X2D13=000130
9CL04=000130   ER74 =000131   2ABST=000131   X2DL3=000131   9CL05=000131
ER75 =000132   2MCAL=000132   9CL06=000132   ER76 =000133   ACL7 =000133
2MEXI=000133   ATIM1=000133   ATIME=000133   ER77 =000134   2RTEX=000134
2RTXT=000134   ATIM2=000134   ER78 =000135   2RTWT=000135   MTIM1=000135
MTIME=000135   ER79 =000136   2RTON=000136   MTIM2=000136   ER80 =000137
2RTOF=000137   IMASK=000137   ER81 =000140   LV12B=000140   2WHER=000140
HBUFA=000140   ABUFA=000140   NOPGS=000140   ER82 =000141   2IOSE=000141
LBUFA=000141   ENDCO=000141   TODF =000142   ER83 =000142   2ERRM=000142
5HENT=000142   USEGA=000142   ER84 =000143   2RSIO=000143   SPL12=000143
SPFLA=000143   ADRMA=000143   ER85 =000144   2MAGT=000144   UZERO=000144
2ACM =000145   ER86 =000145   5MESS=000145   BUFAS=000145   ER87 =000146
XALTS=000146   ER88 =000147   2CAMA=000147   XSGFN=000147   2GL  =000150
ER89 =000150   LV13B=000150   5CBUF=000150   USRTB=000150   ER90 =000151
2GRTD=000151   SPJNA=000151   SPJN1=000151   CACHL=000151   ER91 =000152
2GRTN=000152   SPJN2=000152   ECORM=000152   ER92 =000153   2IOXN=000153
SPJN3=000153   XTMRT=000153   ER93 =000154   2ASSI=000154   ESGTA=000154
ER94 =000155   2PLOT=000155   MRTLA=000155   ER95 =000156   2TRAC=000156
XERDE=000156   ER96 =000157   2ENTS=000157   ENDPA=000157   ER97 =000160
2FIXC=000160   PPRTD=000160   ER98 =000161   2INST=000161   SPMES=000161
XMSGU=000161   ER99 =000162   2OUTS=000162   2WRQI=000163   ER100=000163
2WSEG=000164   2WSBC=000164   2WSGB=000164   ER101=000164   2DIW =000165
ER102=000165   2DOLW=000166   ER103=000166   USLOG=000166   2REEN=000167
ER104=000167   FREEC=000167   ER105=000170   ER106=000171   PRINC=000171
ER107=000172   FORCE=000172   ER108=000173   LCACH=000173   ER109=000174
UCACH=000174   ER110=000175   MBSYM=000175   ER111=000176   CPSTA=000176
X21EN=000177   ER112=000177   CORMB=000177   DCORM=000177   2XMSG=000200
ER113=000200   CBSIZ=000200   ER114=000201   BUFMA=000201   ER115=000202
LRESP=000202   BUSIZ=000203   ER116=000203   CSTAR=000203   ER117=000204
ALTMA=000204   ER118=000205   MONNO=000205   ER119=000206   9ERRP=000206
ER120=000207   ER121=000210   SQELS=000210   ER122=000211   ER123=000212
ER124=000213   ABPRO=000213   ER125=000214   RTTER=000214   ER126=000215
FLRTT=000215   ER127=000216   ACCFL=000216   ER128=000217   RTACC=000217
2DOPE=000220   ER129=000220   NBRTP=000220   ER130=000221   NBSRT=000221
ER131=000222   ICCRT=000222   ER132=000223   PACTA=000223   ER133=000224
PIOAC=000224   ER134=000225   PGNFL=000225   ER135=000226   CSGST=000226
ER136=000227   LGCOL=000227   ER137=000230   PN500=000230   ER138=000231
PDDEB=000231   ER139=000232   BUFBA=000232   ER140=000233   ECBKF=000233
ER141=000234   VDDFL=000234   ER142=000235   SPACC=000235   ER143=000236
BYPIN=000236   2SFAC=000237   ER144=000237   FPOFP=000237   ER145=000240
LPOFP=000240   ER146=000241   7EPOF=000241   ER147=000242   CURMA=000242
ER148=000243   TDFPA=000243   ER149=000244   TDFBA=000244   ER150=000245
LAMBA=000245   DLAMD=000245   ER151=000246   LAMDT=000246   ER152=000247
```

```
LAMAC=000247   ER153=000250   GNLAM=000250   ER154=000251   GNLPR=000251
SIZF =000252   ER155=000252   X21PP=000253   ER156=000253   ER157=000254
PIMON=000255   ER158=000255   ER159=000256   7RTDL=000256   ER160=000257
BUFER=000257   ER161=000260   SREBB=000260   DSREB=000260   2SYCN=000261
ER162=000261   SREBA=000261   ER163=000262   XZRTT=000262   ER164=000263
EXSEC=000263   ER165=000264   ARTFP=000264   ER166=000265   ARTLP=000265
ER167=000266   RWPOF=000266   ER168=000267   RRPOO=000267   ER169=000270
SSPOO=000270   ER170=000271   REMUS=000271   ER171=000272   ER172=000273
ER173=000274   LUSER=000274   ER174=000275   ER175=000276   ER176=000277
OPFTA=000277   ER177=000300   ER178=000301   ER179=000302   PSYSN=000302
ER180=000303   ER181=000304   2SIBA=000305   ER182=000305   PRJPA=000305
ER183=000306   ER184=000307   ER185=000310   MAXOP=000310   ER186=000311
INIFL=000311   ER187=000312   CRTRF=000312   ER188=000313   CNVRT=000313
TER00=000314   ER189=000314   2LAMU=000315   TER01=000315   ER190=000315
2SLRM=000316   TER02=000316   ER191=000316   ER192=000317   ER193=000320
ER194=000321   ER195=000322   ER196=000323   ER197=000324   ER198=000325
ER199=000326   ER200=000327   ER201=000330   ER202=000331   ER203=000332
ER204=000333   ER205=000334   ER206=000335   ER207=000336   ER208=000337
ER209=000340   SSPLE=000340   FIXPA=000340   9MCMX=000340   ER210=000341
SWPFL=000341   ER211=000342   CPULO=000342   ER212=000343   NMATP=000343
ERM  =000344   SEGMA=000344   FSABC=000344   SEGMB=000345   SEGMC=000346
NSEGA=000347   FNABC=000347   NSEGB=000350   NSEGC=000351   TSEGA=000352
TSEGB=000353   TSEGC=000354   TSEGS=000355   SEGPR=000355   PNUMB=000356
CPNUM=000357   NUMBE=000360   WIND1=000361   SEGRE=000362   SAVEF=000363
SRTRE=000364   SGCNU=000365   MAXP =000366   FIXMA=000367   MASSN=000370
7CPCO=000372   VDCRE=000372   7ERRS=000373   7WHO =000374   VDCSE=000374
MASSU=000374   7POLL=000375   7REJE=000376   3BFPA=000376   7EOP =000377
MSTOR=000400   NUSER=000400   NOBJE=000400   ABLPA=000400   TSBAN=000404
TSSIZ=000405   ERRFA=000406   PMONE=000412   MONEN=000412   STUPR=000446
PRWAI=000455   RWAIT=000456   RW   =000461   IRWAI=000466   5CLOA=000501
5RTLS=000503   PSTUP=000524   PRW  =000526   QBSEM=000527   CXLOC=000547
CXULO=000552   5BADS=000600   5BDSE=000601   9TMCT=000660   K5NOS=001000
RECBO=001000   9MCTA=001040   5DMSE=001201   5LOGS=001202   5HSEM=001203
5MLSE=001205   SWAPP=001400   CALLP=001437   BBCAL=001516   MFBBC=001523
BBCLB=001530   MCAL =001543   MONDE=001557   MRET =001644   WT   =001657
DNALT=001723   DALTO=001731   BALTO=001736   RALTO=001742   ALTON=001746
ALTOF=001761   FIACC=001777   LV10 =002000   FREQU=002000   K1024=002000
GAPIT=002005   CSSLO=002034   LUDV =002100   HUDV =002117   RDVDF=002400
RDVLO=002500   RDVHI=002600   LV11 =004000   K5LOC=004000   CORES=006355
RESI2=006355   XRTCH=006355   RTCHE=006360   RETRT=006411   RETXI=006413
ABRET=006415   RETST=006417   RETRW=006421   M61RE=006423   RET  =006427
RT   =006463   PRIOR=006472   ABORT=006521   PRSRV=006647   RESRV=006654
PRLS =006715   RELES=006723   WHERE=006745   RTWT =006776   PBRTE=007003
BRTEX=007003   PSBRT=007004   RTEXT=007022   PRTEX=007061   IFTRM=007063
MALTN=007124   MALTF=007167   RDSC =007226   LOGPH=007377   RTENT=007464
RESER=007632   RELEA=007643   XLOCK=007662   XUNLO=007670   BRESE=007745
BRELE=007772   LV12 =010000   TOEXQ=010112   TOWQU=010126   FREXQ=010166
FRWQU=010173   DMSTR=010241   ICLK =010245   TTIMQ=010310   FTIMQ=010354
STRT =010377   MND  =010403   KALNX=010406   KALDR=010413   MONTH=010465
MNTH1=010466   INTV =010512   DINTV=010521   SET  =010536   DSET =010545
ABSET=010575   ABS1 =010646   DABST=010657   HOLD =010713   TMOUT=010761
TIME =011015   CLOCK=011024   CLADJ=011106   UPDAT=011221   CLCON=011304
CCNN =011315   TUSED=011352   STERM=011363   SLV10=011453   SLV11=011453
SLV12=011453   SLV13=011453   ID10 =011477   WT10 =011501   ID11 =011540
WT11 =011542   ID12 =011574   WT12 =011576   ID13 =011635   WT13 =011637
ERR22=011732   TWT12=012007   TWT11=012011   TWT10=012013   TWT05=012015
TWT04=012017   XFIPV=012062   PVLAD=012103   P0PVL=012121   P1PVL=012124
P2PVL=012127   P3PVL=012132   P4PVL=012135   P5PVL=012140   P6PVL=012143
P7PVL=012146   P8PVL=012151   P9PVL=012154   P10PV=012157   P11PV=012162
P12PV=012165   P13PV=012170   ENT14=012204   IOB14=012207   RET14=012212
```

```
MRET1=012212   BEG14=012217   FINST=012243   ERR14=012250   PENTO=012271
ENTO =012273   WDATA=012300   RDATA=012306   RTACT=012317   XRTAC=012333
XMXSE=012346   XSETU=012350   MXSET=012351   TCNTI=012372   B14  =012374
TIMER=012402   MRFI =012610   MRFO =012612   MOFIL=012623   MOFIA=012647
MOFIB=012651   MOFI2=012653   500MT=012712   500RF=012715   500WF=012721
MAGTP=012725   XRFIL=013006   XWFIL=013010   XMRW =013036   XRPAG=013212
XWPAG=013214   YFGET=013216   YFPUT=013220   COMMO=013422   WAITF=013720
ABSTR=013771   MTRNS=014056   RETRA=014101   FDTMS=014117   MTMRS=014121
OUTBT=014135   INBT =014256   SSPAL=014373   CHBME=014407   M8RET=014615
WDX  =014640   TERWD=014706   IORES=014741   FIORE=014743   CXXRT=014760
CXRTA=014762   IOSET=015003   CIBUF=015100   COBUF=015105   ISIZE=015173
T1P01=015224   OSIZE=015230   T1P02=015260   GZTRE=015305   FDATA=015334
BRKM =015450   ECHOM=015541   T1P03=015607   MSTTY=015626   MGTTY=015630
T1P04=015707   NOWAI=015737   TNOWA=015742   T1P05=016015   EXIOX=016033
LSTC =016110   SETPA=016156   IPTCH=016177   PUTW =016214   IGTCH=016231
GETW =016236   ISTDV=016243   OSTDV=016255   INIOS=016307   BRKTB=016327
ECHTB=016337   TTIMR=016347   TETTO=016461   TTPUT=016514   TRTPU=016522
HXCOD=016531   DIOUT=016537   TTOMR=016541   DMOUT=016543   PDMOU=016641
IONIO=016653   MLTTO=016656   MCLR =016713   CLBUF=016717   CEXIT=016724
TEXIT=016725   TEXIA=016726   CTRTI=016730   CTRTO=016746   TRGET=017016
RSTDE=017023   BSTDE=017027   DTAPT=017040   CONCT=017065   DSCNT=017117
DDRIV=017200   DMONI=017226   PICKF=017234   PICKL=017241   PICKX=017250
PICKY=017255   PSTMR=017262   PSTDR=017266   MTERM=017276   T1P06=017427
GTMOD=017460   MCDES=017562   MCEES=017564   T1P07=017601   MSDAE=017607
MGDAE=017611   T1P08=017622   USCNT=017635   SYCNT=017637   GDEVT=017707
TREPP=017717   GETXM=017725   EXABS=017733   9EXRE=017737   9ETRE=017740
9ESSA=017740   9EARE=017741   9EDRE=017742   9ERRA=017744   9ERR =017752
LV13 =020000   STESC=020015   ACTES=020046   ESCAP=020054   RESCA=020070
LOCAC=020235   USESC=020244   MESCA=020253   T1P12=020264   BBRTW=020452
BRTWT=020456   YBRTW=020463   USRTW=020470   XBRTW=020473   MOCOM=020634
BGERR=020672   ESCON=020735   T1P13=020766   ESCOF=021012   GGLOC=021056
PPLOC=021074   STPIO=021141   STDPI=021145   GETPT=021151   LDPIO=021151
LDDPI=021155   2BDUM=021255   2BDBR=021262   2BSRE=021267   2BREC=021271
XRETB=021473   RETBA=021476   TORTL=021543   OMEXI=021605   MMEXI=021614
STUSE=021620   EGURS=021656   SVTXT=021674   1IREA=021702   1DREA=021705
1RREA=021710   1IWRI=021713   1DWRI=021716   1RWRI=021721   BABOR=021725
ACCSE=021740   SPLMO=021743   SALMO=021747   CSSL =021757   CSSL1=021757
V0   =022000   FV0  =022000   BGSYS=022000   V3   =022003   FV3  =022003
V6   =022006   FV6  =022006   FV11 =022011   V12  =022014   SAVTA=022015
INDEX=022020   TTNO =022021   TTIFI=022022   CPNT =022023   OPNT =022024
CSTRI=022025   OSTRI=022026   PASST=022027   SPAST=022030   ACTPA=022031
CPLIS=022037   TEXTA=022045   GPSEG=022046   BCSEG=022047   STPNT=022050
BSTAR=022051   BREST=022052   LDADR=022053   HDADR=022054   LDDAD=022055
HDDAD=022056   2BXAD=022057   W789P=022060   BCHFL=022063   BINDV=022064
FOBUF=022065   FILNO=022066   NULL =022067   P3RET=022070   S3RET=022070
OPSEG=022071   RTSEG=022072   FILSE=022073   OP2SE=022074   COMTA=022075
RERNU=022076   RERPR=022077   RN1  =022100   RN2  =022101   RRTPR=022102
CURUS=022103   BATAB=022104   SARDF=022105   TIMON=022107   1TIMO=022107
2TIMO=022110   TIOOF=022111   1TIOO=022111   2TIOO=022112   PRJN =022113
REMLI=022114   CHNR =022115   BATIN=022116   KTRVF=022117   MSTPN=022120
MOSTK=022121   FLMAI=022122   SCRSI=022123   ENTCT=022124   SSPAS=022124
MAXCT=022125   NAROU=022126   FILER=022127   ENTER=022130   LEAVE=022131
OUTTE=022132   REESS=022133   XXSAV=022134   5COBS=022134   XCSTR=022134
EMOD =022135   NAMSA=022135   EDSVL=022136   EDSVB=022137   EDOSG=022140
NCOMP=022141   M4LRG=022142   CI4DF=022143   EDIRS=022144   BMECH=022145
COBST=022146   CMDFF=022147   PANAM=022150   ESCBL=022151   BGFIE=022161
MONBL=022170   BFIEL=022200   YBFIE=022200   DFS1 =022220   DFS2 =022264
DCOM =022330   STBEG=022351   STEND=022527   COMST=022527   FIBUF=022614
WORKA=022701   DWORK=022701   TYPS =022766   TYPD =022771   SC100=022774
XSC10=023006   BAUSE=023007   MODST=023013   MSTEN=023037   USPAR=023040
```

```
FLBGT=023045   FLQER=023046   FLLIP=023047   6PASS=023050   MINUS=023051
LACTP=023052   MOFTY=023053   INLOG=023054   UEFLG=023055   UEDAT=023056
UEXRE=023106   UECMR=023107   BACTP=023112   TXAPP=023113   2BFIL=023114
2B64K=023115   2BBLC=023116   2BMCA=023117   2BLRE=023120   2BXRE=023121
2BOLD=023122   2BSEG=023123   2BSST=023124   2BNWR=023125   2BRWP=023126
2BMEM=023130   22BRW=023131   2BALT=023133   1XUST=023134   3BDUR=023135
XBMRE=023151   COMEN=023155   MENTR=023421   TQCOP=023553   XQCOP=023574
RTDCO=023622   ESCQE=023736   XEROP=023751   ESCOP=023753   ENTOP=024017
XJABO=024054   BDBRK=024110   BGBRK=024125   BSBRK=024153   MSG  =024206
SREEC=024260   REEC =024262   RCOM =024315   TOOPS=024337   RECOX=024343
DUMPX=024345   LODX =024410   XMACD=024463   XGBRK=024550   SBRKD=024621
ABADM=024726   GET1L=025005   PUT1L=025010   CHMEM=025212   USET =025214
3FILE=025233   XALTO=025255   BAPRO=025303   TOUS =025320   IER  =025557
OBAER=025576   JABOR=025616   BXBAP=025627   BATBL=025672   BERNR=025702
BERTY=025703   BLREG=025704   CM144=025712   NDNTO=025736   NDNCO=025740
SMSGC=025742   PRNAM=025754   CT500=025764   CPUT5=025766   SYEND=025770
TDVN =026000   CUSER=026001   USDI =026002   USNO =026003   SCUSE=026004
SUSDN=026005   CRTRE=026006   OFLCK=026007   RUSNA=026010   LOCUS=026021
RSYSN=026032   RPRPA=026042   FACFL=026052   INSFG=026053   WPECT=026054
DRFSG=026055   UWLOG=026056   FILUL=026057   FENTL=026060   FWFLA=026061
SCRWR=026062   FXSPF=026063   XSPFM=026064   XXSPM=026065   STACK=026145
ESTCK=027045   AASTC=027054   ASTCK=027054   CSTCK=027056   XSTCX=027067
SPUSH=027060   SPOP =027103   SBINB=027115   FINBT=027121   SBOUT=027171
FOUTB=027175   SFILC=027337   FILCA=027342   OPCAL=027423   EOPCA=027475
GURSE=027520   OPSYS=027544   OP2SY=027547   ERSYS=027552   SFILS=027556
FILSY=027561   SVCAL=027644   FSVCA=027655   RTSVC=027663   RSCAL=027673
FRSCA=027702   RTRSC=027711   MMREE=027723   EFILS=030005   EOPSY=030011
SAVUS=030045   RTSVU=030070   USAVU=030104   RTUSU=030117   ERMSG=030143
QERMS=030145   SPOOL=030233   RPOOL=030234   NPOOL=030235   SDFLA=030236
DV100=030237   OPTAB=030240   OPSPO=030300   CONTX=033774   XKXXX=041000
XRXXX=041100   PRKEY=052163   MASKE=076033   KBACT=105210   KABAC=105410
BCSTA=110000   OP2BG=110000   ERRFL=110000   SVERA=110003   PTSIN=110003
SVLET=110014   SVER0=110016   SVER1=110020   SVER2=110022   SVER4=110026
SVER5=110030   USTX5=110034   LUSTX=110035   USTX4=110036   UDD4 =110040
TSLAN=110051   USTXT=110060   UDD  =110061   STAGP=110062   STOGP=110074
XSBPR=110111   RESIP=110151   RESOP=110154   EXUNA=110157   LSBUF=110160
DBPCO=110161   STOPS=110162   RESES=110176   SRTON=110177   SMEND=110205
XSTOP=110206   SMSTR=110232   SMENT=110250   SM3LE=110273   XBLOG=110301
SMXLE=110306   SM2LE=110321   SMLEA=110332   NSTAR=110357   SMCHT=110373
SMYES=110433   XSRTO=110441   SMABL=110451   XRRTO=110452   CCBRE=110462
CCBRS=110521   SM1AB=110544   STSLU=110562   RINPA=110613   NWBPA=110646
SMGCO=110656   NW2PA=111017   SMEDI=111020   MLGRS=111152   OISYS=111302
EDIRT=111521   TSGNN=111552   TSGNA=111560   VDUST=111575   VDUSB=111607
GETSO=111617   SRCSH=111631   LIREE=111632   SMSRC=111657   SMTMT=111720
LRESE=111725   REEPN=112000   REECS=112001   REECO=112002   SMTIM=112007
SAGPA=112110   SMSGP=112115   SMAGP=112122   SMGPA=112130   REECE=112456
REECT=112457   SMKGP=113061   SMCRE=113120   SMORE=113125   SMCWR=113133
SMOWR=113140   SMBAC=113167   SMSCA=113174   SMCCL=113216   SMOUT=113230
SMDDE=113317   SM2DE=113401   SMDOC=113442   SMOCT=113473   REETE=113503
FREES=113503   SM3OC=113514   55DLR=113526   SMDTD=113534   SMDEC=113540
SMPER=113600   DLRET=113601   SMSPA=113663   SMCRL=113675   SM2TC=113704
5DFRE=113713   5DLRE=113715   SMTCO=113715   SMTCI=113723   SMWIN=113735
SMDYN=114012   SMTAC=114037   55DFR=114045   SMFIB=114051   DFPRE=114054
DPREE=114060   REEDU=114064   PTERM=114136   6PINO=114150   6BFTY=114162
6DFTY=114165   6PRTN=114170   6TILB=114176   6PARE=114211   6PSEG=114235
6PLOG=114256   6PADD=114267   6PSPN=114275   6FFHE=114307   6FFTR=114322
6TILL=114334   6P3RE=114345   RETUA=114345   RETSG=114346   6PRTF=114347
6RTLR=114352   6CTAB=114360   6TINC=114361   6BLCK=114362   6BLOC=114363
6CFIL=114364   6FINA=114365   NWORD=114366   6RPAR=114367   6WPAR=114374
6IDMA=114401   6IDTA=114405   6EIDT=114411   6SINT=114415   6MACM=114424
```

```
6TERR=114434   SMCIL=114434   6ILLA=114446   6STRN=114456   6ADRM=114463
6XALT=114464   6CFUP=114465   SMFAM=114474   SMTXF=114506   SMCAM=114517
SCOMT=114532   CRSRT=114621   CRLRT=114633   CRESE=114646   DLREE=114653
BPOPE=115031   PROPE=115033   TALRE=115053   RESRT=115063   RTLRM=115066
ILLA =115103   ILLP =115113   STSTA=115124   STRES=115135   STRHI=115147
STRLO=115160   STROF=115170   STRTE=115200   AMBSU=115212   FTPPR=115225
GGSGM=115230   DMACN=115251   STSUP=115337   5OPCO=116227   OPCOR=116307
OPCOM=116331   ERRS =116446   EROPC=116447   TFILE=116456   OPCFI=116461
UECHE=116525   REMES=116576   REMRU=116600   SETOL=116612   COMSB=116617
UECOM=116625   UELOG=116653   SINSE=117036   SSMON=117057   SSSLE=117153
SSLEA=117154   IOUT =117221   6ERRP=117221   6ERRO=117223   SMFIL=117235
6OOPE=117240   RSIO =117252   GBGSZ=117275   MBECH=117316   SINCO=117446
DRTDE=117562   DSTEN=120006   RIMAG=120142   RMEMO=120162   RSAVE=120202
SELAR=120237   ASKAR=120365   LEGSC=120411   CHRTD=120434   IMTAB=120456
IITAB=120504   ISTAB=120533   FFMTA=120565   FFITA=120573   FFSTA=120601
ITIMT=120624   RTIMT=120630   CERR1=120737   6SOPE=120772   6MOPE=120776
6CLOS=121015   INORO=121055   R1IMA=121102   R1MEM=121110   W1IMA=121114
W1MEM=121122   R1SAV=121126   W1SAV=121161   WXSAV=121201   WIMBA=121214
IIDTA=121251   RIDTA=121253   IEIDT=121255   REIDT=121257   PIKEY=121314
ADUMP=121626   ODUMP=121630   IBACK=122200   RBACK=122202   DUSSU=122536
DURSU=122603   DURPR=122655   DVSTR=123011   DFPRM=123016   9FINF=123314
9CABL=123314   9CLOG=123315   9CUNI=123316   1CFST=123317   9CFST=123317
2CFST=123320   1CFSI=123321   9CFSI=123321   2CFSI=123322   9CCDF=123323
9CFNO=123324   SMGFI=123327   DLSGF=124143   DEFSG=124145   CRBIN=125316
IIIOX=125604   RFIOX=125606   LISCO=125714   ABENT=126200   OUTUS=126231
XOUTU=126255   PASSW=126261   PROJN=126267   XRTPR=126276   AFILN=126304
WPROP=126313   CHVAR=126417   SWDEN=126611   LOGIN=126720   ILUTA=126737
RLUTA=126741   DUMCA=127113   SSDNU=127331   DBSPR=127445   LIADD=127575
ISPHE=127635   RSPHE=127637   SCHPR=130061   PROJ1=130263   PROJ3=130325
CDAFI=131361   DISPR=132076   CDRTC=132326   ISYSE=132537   CLACI=132604
SCPRO=132620   UUELO=132654   SCSFS=132770   UEPRE=133043   UEADM=133137
CHIOB=133404   XINST=133771   T1P15=134026   XMREM=134032   DUMF =134062
WAIF =134065   XMCHK=134100   REMCM=134162   CCCOM=134274   SENCT=134403
CENCT=134405   FLOGO=134430   BLOGO=134433   BINPE=134523   BOUPE=134535
JOBAB=134547   BILCM=134575   BILPA=134577   BOBOR=134625   SGWPR=134664
SGWPE=134670   JAB2 =134677   LOGOU=134731   ALOGO=135002   DFHDL=135072
LIHDL=135076   MBZME=135130   RSCOL=135161   CMRFI=135343   CMWFI=135345
STMSY=135444   FNO  =135445   MESMA=135446   ORBIT=135451   ST500=135564
PLPRO=135637   RECFI=135643   RECOV=135647   LOUTT=136033   XMLGR=136042
FRUSE=136051   DATIM=136052   BITTA=136054   CONTI=136216   GOTOU=136225
EILCO=136237   CHSML=136252   DUMP =136322   MESBU=136454   LOAD =136504
PLACE=136506   MEMOR=136647   ZMEMO=136701   SETME=136710   XRTLO=137046
MLCTA=137051   ZRTLO=137053   WRTLO=137060   PYRTL=137065   PZRTL=137072
YRTLO=137077   CSTAT=137157   SMMAI=137165   PDATC=137310   DATCL=137334
COPYF=137433   LISTT=137522   LISTX=137526   LISTS=137717   YOUHA=137735
MLHEL=137771   TIMOU=140040   MLINI=140042   ADDD =140120   MLSTO=140122
MLRUN=140125   MOVB =140131   MOVBF=140132   LIDEV=140157   MLSEN=140176
MLBRO=140200   ISETP=140300   ICLEP=140301   ICLNR=140302   ICHRE=140303
LRTPR=140404   EXPAN=140404   MLDSE=140455   STOPT=140605   MLDBR=140634
CHTER=140641   ENTRT=140664   WHOIS=140704   GETRN=141042   CCDUM=141056
UNEXT=141057   MLLME=141120   MLLBR=141122   SETUN=141156   SETAV=141210
SSETE=141252   SETER=141255   MLDLM=141346   MLDLB=141350   GETER=141407
INIBU=141443   READM=141576   INCOM=141663   NEXIN=141664   WMESS=141672
LIICO=141720   RBUFF=141763   WBUFF=141770   BTSTO=142011   BTCLE=142032
HELP =142041   XBTLO=142054   BTLOA=142074   5BUFR=142134   SSBUF=142234
TERMO=142254   LAMFU=142334   PAR1 =142335   PAR2 =142336   PAR3 =142337
T1P09=142337   LAPLI=142340   PARA1=142341   PARA2=142342   PARA3=142343
NSLAR=142344   LWIAR=142355   NFLAD=142366   YESNO=142372   TPPAG=142405
TNPAG=142421   OPERA=142426   TLID =142433   TRFP =142441   PTLAM=142532
WFOPE=142534   REUSE=142640   MOVNN=143100   MOVNA=143101   MOVNP=143102
```

```
MOVAN=143103   MOVAA=143104   MOVAP=143105   MOVPN=143106   MOVPA=143107
MOVPP=143110   PFLAM=143203   LDATX=143300   LDXTX=143301   LDDTX=143302
LOGSY=143302   LDBTX=143303   STATX=143304   STZTX=143305   STDTX=143306
CRLAM=143452   DELAM=143541   LAPRO=143612   STAPL=143673   LAREA=144010
STOPL=144033   LAINF=144156   PERCE=144325   LLAMC=144461   SLAMC=144560
DEFSH=144570   DEFHI=144574   SGPIB=144614   STAHI=144760   STOHI=144764
PRHIS=145043   XPRHI=145064   KGPIB=145261   SGPBS=145304   EXECI=145320
CSTTY=145360   GVAL =145405   T1P14=145430   CGTTY=145433   GEPNE=145456
GPIIM=145477   SMOOP=145522   CDSPF=145550   FUNCT=145551   DEFES=145652
T1P10=145730   DEFLO=145771   CDESC=146060   CEESC=146065   T1P11=146144
CSBSI=146400   TECBU=146632   PETEC=147033   TAPEF=147067   NTERM=147155
DEFTE=147160   LITER=147264   ENTRM=147400   DITRM=147402   ACHSU=147742
LIPBU=147746   STWOR=147747   CCCIN=147756   CINBT=147765   PELIP=150033
DEFLI=150074   RCALA=150114   LISLI=150143   ENLIP=150217   DILIP=150221
LDVFC=150240   PIOF =150405   SEX  =150406   REX  =150407   PONN =150410
PION =150412   IOXT =150415   EXAM =150416   DEPO =150417   PASET=150426
ERINI=150473   ERPRI=150505   MEMLI=150603   CAON =150642   CAOFF=150647
MBDYN=150657   CHTAC=150702   CCOLD=150714   LOOKA=150736   UECMD=150747
ABLOO=151007   ABL1 =151100   GCOM =151210   MONED=151337   EDIT =151341
MBTER=151530   TERMS=151675   VDUTT=152126   VDUBS=152140   GETOL=152150
SRCHO=152162   SGPAR=152231   AGPAR=152236   CCTCO=152237   GPAR =152244
LISTR=152602   STRFI=152767   STRNU=152776   OILLP=153003   DEC2 =153015
CLFOR=153032   RTOUT=153057   PRIRT=153112   CREAD=153154   OREAD=153161
CWRIT=153167   OWRIT=153174   BACK =153227   SCAB =153234   3OUTT=153250
TEXTN=153311   SMFFL=153330   OCTU =153337   DTDEC=153360   DECU =153364
CRLF =153421   TCO2 =153430   TCO  =153441   UELGN=153454   TCI  =153457
WINB =153510   STFLG=153562   ETCI =153564   SMTRS=153672   SRCHI=153675
SRHIN=153675   3ENTE=153727   L3EAV=153752   LEAVX=153765   LEAV2=154000
3LEAV=154011   ROFIP=154044   RBGUF=154171   EXECC=154223   TIMUS=154312
CTIMU=154317   TMTIM=154330   ACER =154377   STABA=154424   STOBA=154426
XBADM=154430   SRMMO=154500   SLCMO=154502   TOOPC=154511   ACCOU=154542
INITA=155001   CACCO=155223   SACCO=155225   OPCHE=155537   DTSLI=155643
NIMPL=155645   RSTAC=155736   RSTOC=155740   DEFEL=156171   LIRTL=156174
GETVA=156304   SAVE =156332   CHKRN=156347   CHCLA=156373   DDIRN=156376
DOLDD=156410   DNEWD=156424   DDEVN=156440   DLOGU=156450   SDLOG=156460
DFRFL=156472   DUSEN=156511   DSPAC=156520   DNUSE=156532   DUSEI=156543
DPASS=156553   DOPAS=156562   DNPAS=156573   DFRIN=156604   DACCE=156614
DFNAM=156642   DFSNA=156651   DPAGE=156660   DNFIL=156671   DDVNU=156707
CMLTS=156713   DPACC=156720   DFACC=156752   DOACC=157004   DOBJI=157034
DACCM=157045   DFNUM=157071   DBLSZ=157101   DBYTP=157111   DBLOP=157122
DOUTF=157133   DSOUD=157143   CPTSL=157150   DDESD=157156   DSOUF=157173
DDESF=157203   DSDEV=157216   SCDEV=157232   DPFNA=157243   DNUCO=157260
BFAD =157273   MNCHK=157306   VONA =157317   STYPE=157327   DSGEN=157337
DTYPE=157351   FINRD=157364   LTSPR=157404   FINRS=157412   PRSFN=157436
PSTSP=157460   PNLPA=157514   PNPAG=157533   PANLI=157545   PIBSF=157557
PAASF=157573   PINAP=157606   PPIAC=157621   PFOID=157655   DUFQU=157676
DRSND=157701   DRUND=157715   DPPAW=157730   DNENT=157743   GLPAR=157774
CGTTS=160117   ITSLI=160155   XRTSL=160157   9MOCO=161056   9IHEN=161057
9OHEN=161115   9HNTC=161153   X9HNT=161163   IPFI =161175   OPFI =161205
MODE =161262   EXHEN=161417   RBPAR=161530   BATCH=161625   BANUM=161776
REMCO=162007   TYREM=162021   LOGDI=162027   CHFTR=162075   APREB=162244
APPBA=162265   XAPPB=162334   SIM2O=162531   CHIOP=162561   LIBAT=162753
LIREQ=163062   LIBQU=163105   DELBE=163360   DELRE=163403   ABJOB=163746
ABBA =164025   CHBAT=164066   SCEDU=164172   KGPAR=164340   CLEBA=164367
LIVER=164476   SNAMS=164533   NAMSR=164542   RTFEL=164647   INAMS=164647
STCH =164711   SEGFN=164733   YBCSE=164734   XBLPA=164735   X5DSK=164736
DRUMF=164737   MXTSO=164740   MXTS1=164741   MXTS2=164742   MXTS3=164743
ENDVT=164744   CMMCL=164746   IEXI =164771   OLDST=165233   SWPLO=166062
MOVFL=167522   MOVSY=167527   FCPUL=167545   CMCPU=167577   ESCMA=170076
FLOGD=170345   FFLOG=170370   CMCHT=170520   RTRAN=170626   WTRAN=170650
```

```
WXTRA=170674   UPDSI=170763   INBRP=171010   CDVCO=171170   COPCO=171275
LIVOL=171410   CREVO=171410   CPUFI=171410   OSCPR=171457   MAIL =171535
TRTER=171557   SERVS=171603   RELOA=171626   ALOAF=171726   ADR2B=171730
LDRAD=171733   XSWTP=171734   TLBUF=172000   KLRC1=172117   LDEND=172205
OPEND=172205   BAUTI=172337   SSCLD=173652   RSCLD=173730   OP2EN=173774
XTTNO=177621   CFLUN=177622   XTTIF=177622   FDIST=177623   SFCOM=177623
FRTRY=177624   DOORL=177624   DINCR=177625   FLMQU=177626   FLMFA=177627
FRTPR=177630   STEPR=177630   WDCNT=177631   FCNT1=177632   FCNT2=177633
OPFLG=177634   PRECP=177634   DATAF=177635   SECT =177636   SCTIB=177637
OLDTR=177640   DTRCK=177640   NEWTR=177643   CCBWO=177644   FCOMF=177644
BFDEV=177645   MSCIB=177646   FMEMH=177646   FMEMD=177646   FMEML=177647
OPWCH=177650   SCTTR=177651   WCOUN=177651   FSTA1=177652   FSTA2=177653
CALIB=177654   LASMH=177654   LAMAD=177654   PYERA=177654   LASML=177655
MREMW=177656   REMWO=177656   FDRIV=177657   LREMW=177657   FRETU=177660
FDIMO=177661   FDIFO=177662   XBCHF=177663   LFADD=177666   XECRA=177701
SVLWK=177706   WDSCT=177706   OSVWK=177707   NWLBB=177710   NWLBA=177711
OCMD1=177712   OCMD2=177713   OSVBA=177714   OSVBB=177715   OSVCO=177716
OSVWC=177717   SPACO=177720   ADMIA=177720   MTLRG=177720   SPAFL=177721
ADMIS=177721   CMTRE=177721   CORCU=177722   ALTPO=177722   MRECC=177722
SLONG=177723   BADDR=177723   NMTRE=177723   ECCFL=177724   BADIP=177724
CMWCN=177724   CPAT1=177725   BADTR=177725   CTACN=177725   CDISP=177726
WANKN=177726   ADNST=177726   NFDIA=177726   CPAT2=177727   NOSEC=177727
XEIXT=177727   TYPEC=177730   RTZFL=177730   XEREJ=177730   SVLBB=177731
XENUS=177731   SVLBA=177732   SEEKF=177732   SHSTA=177732   XENTO=177732
TINFO=177733   ERRC1=177733   XENRU=177733   PECH7=177734   ERRC2=177734
SHEAD=177734   XEPCL=177734   SSEC =177735   SRTRY=177735   XERND=177735
SWTRY=177736   SSTEP=177736   CERRC=177736   XENSE=177736   TRTZ =177737
SMARG=177737   MAXUN=177737   XENOS=177737   SVLCO=177740   MACOU=177740
ACCBU=177740   XEILR=177740   SCADR=177741   LCYLI=177741   MRETU=177741
DMRET=177741   XEIDP=177741   MARGC=177742   DERRC=177742   MWRIN=177742
XEITL=177742   BUSFL=177743   MWSTA=177743   XENDP=177743   PBRK7=177744
SVLCA=177744   MLOAD=177744   NOWFL=177744   PNOTF=177744   XEDRI=177744
PNI  =177745   CLRG =177745   SVLWC=177745   CFLRG=177745   XEXBF=177745
TRG  =177746   PE1  =177746   TADRG=177746   XEROV=177746   ARG  =177747
PNOME=177747   XERNA=177747   DRG  =177750   NOWH =177750   PFULL=177750
XEPVR=177750   XRG  =177751   NOWL =177751   PNOTV=177751   XEPRV=177751
CTRG =177752   CTADR=177752   PILSL=177752   XEIPN=177752   CARG =177753
CADRG=177753   X21T2=177753   PSLBS=177753   XEILM=177753   CDRG =177754
PILF =177754   IN5MS=177754   ANACS=177754   XEMFL=177754   CXRG =177756
RSIST=177755   AFRET=177755   ESCBU=177755   XEIMA=177755   BRECH=177756
ERCNT=177756   ATTNI=177756   FCRCE=177756   TMPBU=177756   VDBFR=177756
XEILF=177756   ROUSP=177757   SERRB=177757   REMTI=177757   BADTY=177757
XENVI=177757   NCBRK=177760   WERRB=177760   TRGIN=177760   USTAT=177760
RSELR=177760   VDBUF=177760   XEWNA=177760   CTTYP=177761   AERRB=177761
INDAT=177761   GPXTR=177761   TBUSA=177761   XEBNC=177761   CESCP=177762
TACNS=177762   UCLIN=177762   GPDZI=177762   RLTSA=177762   VDMTT=177762
XEAIN=177762   BRKMA=177763   TACOU=177763   ATINT=177763   GPUZI=177763
RLLSA=177763   VDRTP=177763   XEBFC=177763   TSPEE=177764   COMFL=177764
MWCNT=177764   SOFTA=177764   GPRUN=177764   ANTOR=177764   NOBUF=177764
VDFUN=177764   XEMCH=177764   BLSZ =177765   XSAC =177765   CNTRE=177765
GPBFL=177765   TMLRE=177765   FBSIZ=177765   VDMAD=177765   XENDM=177765
SIAD =177766   DFLAG=177766   TRNSF=177766   XEIDR=177766   BUSY =177767
ECHOT=177767   HXERF=177767   PNOTR=177767   VDBLC=177767   XENOP=177767
SOAD =177770   BRKTA=177770   SCREE=177770   FINIS=177770   EFUNC=177770
X21TB=177770   PNOTP=177770   XEIRT=177770   LAST =177771   EMPTF=177771
ERROR=177771   CCTRL=177771   EMAXS=177771   VDNBL=177771   XEBNY=177771
KTMSU=177772   TMSUB=177772   DIVIS=177772   CMDAT=177772   EDEVN=177772
XEIBP=177772   TMR  =177773   5TMR =177773   CONTW=177773   IXSAC=177773
MSIZE=177773   EMSGS=177773   XENIM=177773   TTMR =177774   XHDEV=177774
[illegible]FPH=177774   ZDBPR=177774   OCTRL=177774   ANTME=177774   SCPRI=177774
```

```
ENBUF=177774   X21T1=177774   XETMM=177774   HDEV =177775   BYTS =177775
TDFLG=177775   NRDYF=177775   DCNRT=177775   XTEMI=177775   PORTN=177775
RPORT=177775   EEMTY=177775   X2DHD=177775   X21T3=177775   X21T4=177775
X21T5=177775   X21T6=177775   E2   =177776   INCR =177776   STDRI=177776
NRDTR=177776   XTEMO=177776   DBCOU=177776   PARTN=177776   X21T0=177776
X21T7=177776   X21TA=177776   XEIRM=177776
% :: Part One alphabetic Symbol List

=====================================================================
    SINTRAN III/VSX  VERSION I              16.15.41    23 SEPT  1984
=====================================================================

XXXXX    XXXX   XXXXX  XXXXXX        XXXX   XX  XX  XXXXX
XXXXXX  XXXXXX  XXXXXX XXXXXX       XXXXXX  XX  XX  XXXXX
XX  XX  XX  XX  XX  XX   XX         XX  XX  XXX XX  XX
XXXXXX  XX  XX  XXXXX    XX   XXXX  XX  XX  XXXXXX  XXXX
XXXXX   XXXXXX  XXXXX    XX   XXXX  XX  XX  XXXXXX  XXXX
XX      XXXXXX  XX  XX   XX         XX  XX  XX XXX  XX
XX      XX  XX  XX  XX   XX         XXXXXX  XX  XX  XXXXX
XX      XX  XX  XX  XX   XX          XXXX   XX  XX  XXXXX
=====================================================================
    SINTRAN III/VSX  VERSION I              16.15.41    23 SEPT  1984
=====================================================================


SYMBOL LIST IN ALPHABETIC ORDER .

=====================================================================


,DA  =000001   ,PRIN=000013   ,SA  =000002   ,SP  =000024   ,ST  =000007
,WRIT=000001

1OBIT=000014   1CFSI=123321   1CFST=123317   1DILB=000012   1DREA=021705
1DTU1=000012   1DTU2=000013   1DWRI=021716   1ILLC=000000   1IREA=021702
1IWRI=021713   1RREA=021710   1RWRI=021721   1TIMO=022107   1TIOO=022111
1TUSE=000012   1USER=000015   1XNDA=000035   1XNWD=000037   1XUST=023134

22BRW=023131   2ABOR=000105   2ABSE=000102   2ABST=000131   2ACM =000145
2ALTF=000034   2ALTN=000033   2ASSI=000154   2B64K=023115   2BALT=023133
2BBLC=023116   2BDBR=021262   2BDUM=021255   2BFIL=023114   2BLRE=023120
2BMCA=023117   2BMEM=023130   2BNWR=023125   2BOLD=023122   2BREC=021271
2BRKM=000004   2BRWP=023126   2BSEG=023123   2BSRE=021267   2BSST=023124
2BXAD=022057   2BXRE=023121   2CAMA=000147   2CFSI=123322   2CFST=123320
2CIBU=000013   2CLAD=000112   2CLOC=000113   2CLOS=000043   2CLSE=000043
2CMND=000070   2COBU=000014   2CONC=000106   2DABS=000127   2DESC=000071
2DIBA=000014   2DILB=000013   2DINT=000130   2DIW =000165   2DOLW=000166
2DOPE=000220   2DSCN=000107   2DSET=000126   2ECHO=000003   2EESC=000072
2ENTS=000157   2ERM =000064   2ERMS=000064   2ERRM=000142   2EXIO=000031
2FIX =000115   2FIXC=000160   2GETR=000030   2GL  =000150   2GRTD=000151
2GRTN=000152   2HOLD=000104   2INBT=000001   2INST=000161   2INTV=000103
2IOSE=000141   2IOUT=000035   2IOXN=000153   2ISIZ=000066   2LAMU=000315
2LAST=000026   2MAGT=000144   2MCAL=000132   2MEXI=000133   2MSG =000032
2MTER=000052   2NOPE=000050   2NOWA=000036   2OSIZ=000067   2OUTB=000002
2OUTS=000162   2PLOT=000155   2PREL=000125   2PRES=000124   2PRIO=000110
```

```
2PRLS=000125   2PRSR=000124   2QERM=000065   2RBYT=000075   2REEN=000167
2RELE=000123   2RESR=000122   2RFIL=000117   2RMAX=000062   2RPAG=000007
2RSIO=000143   2RT  =000100   2RTDS=000027   2RTEX=000134   2RTOF=000137
2RTON=000136   2RTWT=000135   2RTXT=000134   2SBLZ=000076   2SBYT=000074
2SET =000101   2SETO=000012   2SFAC=000237   2SIBA=000305   2SLRM=000316
2SMAX=000073   2STBC=000077   2SYCN=000261   2TIME=000011   2TIMO=022110
2TIOO=022112   2TRAC=000156   2TUSE=000114   2UNFI=000116   2UPDA=000111
2WAIT=000121   2WFIL=000120   2WHER=000140   2WPAG=000010   2WRQI=000163
2WSBC=000164   2WSEG=000164   2WSGB=000164   2XIBU=000066   2XMSG=000200
2XNDA=000036   2XNWD=000040

3BDUR=023135   3BFPA=000376   3DOUA=000015   3DOUB=000016   3ENTE=153727
3FILE=025233   3FLOP=000011   3FLTI=000013   3FRES=000014   3ILLF=000017
3LEAV=154011   3OUTT=153250   3STRE=000012

5OOMT=012712   5OORF=012715   5OOWF=012721   5SDFR=114045   5SDLR=113526
5SNSG=000021   5SSDS=000020   5ABJO=000004   5ABS =000013   5ACCS=000032
5ALEC=000006   5BACK=000017   5BAD =000004   5BADM=000036   5BADS=000600
5BCOM=000001   5BDSE=000601   5BESC=000003   5BFPA=000076   5BFUL=000017
5BITM=000010   5BLOC=000017   5BPAS=000000   5BREA=000001   5BREG=000030
5BRES=000013   5BRKF=000007   5BRST=000111   5BUFR=142134   5BUSE=000002
5BWLI=000002   5CAPI=000013   5CBUF=000150   5CFIL=000007   5CIMS=000002
5CLDV=000002   5CLOA=000501   5COBS=022134   5COM =000013   5CONC=000015
5CONT=000004   5CRDL=000011   5CTRL=000005   5DEMA=000001   5DFRE=113713
5DLRE=113715   5DMSE=001201   5DTU1=000014   5DTU2=000015   5ECHO=000000
5ECHS=000113   5ERRO=000004   5ERRS=000005   5ESC2=000002   5ESCF=000010
5ESCL=000010   5ESCO=000001   5FFGP=000014   5FILS=000006   5FIMO=000004
5FIU2=000026   5FIUS=000022   5FIX =000002   5FLOP=000010   5FRER=000015
5FYLL=000114   5HDMA=000007   5HDUP=000003   5HENT=000142   5HSEM=001203
5IBDV=000006   5IESC=000015   5IFS2=000012   5INHB=000003   5INNE=000016
5INT =000014   5INVR=000007   5IOBT=000017   5IPT3=000017   5IRTS=000013
5ISET=000014   5ISMS=000015   5ISPS=000042   5LBLO=000005   5LBRK=000016
5LCHA=000006   5LOC2=000013   5LOGI=000011   5LOGO=000003   5LOGR=000011
5LOGS=001202   5LSTA=000017   5MACD=000007   5MESS=000145   5MLGI=000015
5MLSE=001205   5MRST=000030   5MT  =000011   5N100=000014   5ND5E=000010
5NNET=000027   5NORE=000003   5NOSL=000011   5OK  =000000   5OP2S=000023
5OPCO=116227   5OPSE=000003   5OXON=000012   5PAER=000016   5PGU =000013
5PHOE=000007   5PRVT=000010   5PT3S=000041   5RCFI=000010   5RDEV=000017
5REG =000007   5REMO=000005   5REMP=000011   5REP =000016   5RERU=000006
5RFIL=000016   5RQI =000005   5RRUS=000040   5RT2S=000037   5RTFI=000010
5RTLS=000503   5RTOF=000016   5RTSG=000004   5RTSI=000017   5RWAI=000015
5SEGS=000005   5SEXI=000015   5SPEC=000002   5SPRF=000006   5SPRO=000005
5SPSI=000016   5SREE=000006   5SWWA=000015   5SYSE=000004   5TDUM=000000
5TERM=000005   5THIG=000002   5TLOW=000001   5TLRE=000000   5TMOU=000012
5TMR =177773   5TREA=000003   5TUSE=000014   5UBFP=000077   5UEAL=000013
5UECM=000006   5UEES=000010   5UEIE=000002   5UELO=000012   5UEMO=000000
5UEPR=000005   5UERU=000004   5UEST=000007   5UETE=000001   5UETL=000003
5UETM=000011   5USES=000007   5WAIT=000017   5WCBU=000064   5WCON=000005
5WECH=000006   5WESC=000011   5WIP =000014   5WLIN=000010   5WLOC=000012
5WORK=000060   5WPM =000017   5WRQI=000006   5XDEV=000011   5XOFF=000010
5XON =000007   5XRTD=000020   5XSG1=000033   5XSG2=000034

6ADRM=114463   6BFTY=114162   6BLCK=114362   6BLOC=114363   6CFIL=114364
6CFUP=114465   6CLOS=121015   6CTAB=114360   6DFTY=114165   6EIDT=114411
6ERRO=117223   6ERRP=117221   6FFHE=114307   6FFTR=114322   6FINA=114365
6IDMA=114401   6IDTA=114405   6ILLA=114446   6MACM=114424   6MOPE=120776
6OOPE=117240   6P3RE=114345   6PADD=114267   6PARE=114211   6PASS=023050
6PINO=114150   6PLOG=114256   6PRTF=114347   6PRTN=114170   6PSEG=114235
6PSPN=114275   6RPAR=114367   6RTLR=114352   6SINT=114415   6SOPE=120772
6STRN=114456   6TERR=114434   6TILB=114176   6TILL=114334   6TINC=114361
```

```
6TTYN=000023   6WPAR=114374   6XALT=114464   6XOFF=000014

7BADT=000034   7BDAT=000001   7BMMX=000004   7CERS=000041   7CESC=000016
7CONF=000005   7CORQ=000006   7CORS=000007   7CPCO=000372   7CUEL=000000
7CUTY=000012   7DATA=000000   7DBRE=000001   7DCON=000011   7DECO=000002
7DESC=000017   7DUMM=000030   7ECKM=000003   7EDRS=000051   7EOP =000377
7EPOF=000241   7ERRS=000373   7ESCA=000010   7ESRS=000040   7FBSI=000025
7IAM =000050   7IRQI=000004   7ISRQ=000042   7ISRS=000043   7KEYI=000033
7LUN =000013   7NCML=000000   7NOWT=000044   7NPAS=000004   7NWRE=000046
7OPSV=000037   7PASS=000021   7POLL=000375   7RECO=000027   7REJE=000376
7RESE=000026   7RFI =000002   7RLOC=000047   7RQIN=000003   7RTDL=000256
7SATY=000015   7STRQ=000031   7STRS=000032   7SYCN=000023   7SYSI=000012
7TMOD=000014   7TNOW=000045   7TREP=000052   7TTYP=000015   7USCN=000024
7USID=000020   7WHO =000374   7ZFPA=000003   7ZMEM=000001   7ZSCR=000002


8CLCN=000112

9CABL=123314   9CCDF=123323   9CFNO=123324   9CFSI=123321   9CFST=123317
9CLO0=000124   9CLO1=000125   9CLO2=000126   9CLO3=000127   9CLO4=000130
9CLO5=000131   9CLO6=000132   9CLOG=123315   9CUNI=123316   9EARE=017741
9EDRE=017742   9ERR =017752   9ERRA=017744   9ERRP=000206   9ESSA=017740
9ETRE=017740   9EXRE=017737   9FINF=123314   9HNTC=161153   9IHEN=161057
9IVAL=000010   9MCMX=000340   9MCTA=001040   9MOCO=161056   9OHEN=161115
9OUVA=000011   9TIM0=000111   9TIM1=000115   9TIM2=000117   9TIM3=000120
9TIM4=000121   9TIM5=000122   9TIM6=000123   9TMCT=000660

AACSE=000021   AASTC=027054   ABADM=024726   ABBA =164025   ABENT=126200
ABJOB=163746   ABL1 =151100   ABLOO=151007   ABLPA=000400   ABORT=006521
ABPRO=000213   ABRET=006415   ABS1 =010646   ABS6 =000122   ABSET=010575
ABSTR=013771   ABUFA=000140   ACCBU=177740   ACCFL=000216   ACCOU=154542
ACCRL=000036   ACCSE=021740   ACER =154377   ACHSU=147742   ACL7 =000133
ACQBH=000047   ACQFP=000045   ACQHP=000046   ACQU =000054   ACTES=020046
ACTOU=000030   ACTPA=022031   ACTPR=000012   ACTSE=000011   ADDD =140120
ADMIA=177720   ADMIS=177721   ADNST=177726   ADR2B=171730   ADRMA=000143
ADUMP=121626   ADVAN=000010   AERRB=177761   AFILN=126304   AFRET=177755
AGPAR=152236   AKMCH=000101   ALEV =000002   ALEVB=000010   ALEVL=000001
ALMSZ=000002   ALOAF=171726   ALOGN=000001   ALOGO=135002   ALTFO=000007
ALTMA=000204   ALTOF=001761   ALTON=001746   ALTPO=177722   AMBSU=115212
ANACS=177754   ANCHA=000011   ANI  =000001   ANTME=177774   ANTOR=177764
APPBA=162265   APREB=162244   ARG  =177747   ARSEG=000020   ARTFP=000264
ARTLP=000265   ASKAR=120365   ASTCK=027054   ATIM1=000133   ATIM2=000134
ATIME=000133   ATINT=177763   ATTNI=177756

B14  =012374   BABOR=021725   BACK =153227   BACKG=000004   BACKS=000015
BACTP=023112   BAD  =000005   BADAN=000042   BADDR=177723   BADIP=177724
BADTA=000021   BADTR=177725   BADTY=177757   BALTO=001736   BANUM=161776
BAPRO=025303   BATAB=022104   BATBL=025672   BATCH=161625   BATIN=022116
BAUSE=023007   BAUTI=172337   BBASP=000010   BBCAL=001516   BBCLB=001530
BBID =000001   BBLEN=000023   BBPRO=000000   BBRTW=020452   BBUFF=000003
BBYTC=000002   BCFLA=000017   BCHAI=000004   BCHFL=022063   BCHIS=000027
BCHNU=000020   BCHOS=000024   BCSEG=022047   BCSTA=110000   BDATA=000000
BDBIT=000011   BDBRK=024110   BDIRI=000000   BECHO=000000   BEG14=012217
BENA =000000   BERNR=025702   BERTY=025703   BEXQU=000013   BFAD =157273
BFDEV=177645   BFIEL=022200   BFSSP=000011   BGBRK=024125   BGERR=020672
BGFIE=022161   BGFPA=000040   BGLPA=000041   BGSYS=022000   BHEAD=000005
BHOLD=000016   BIFIL=000021   BIHDL=000016   BILCM=134575   BILPA=134577
BIML =000001   BINDV=022064   BINPE=134523   BISYM=000015   BITA =000002
BITC =000003   BITD =000004   BITER=000002   BITM1=000011   BITM2=000012
BITM3=000013   BITM4=000014   BITM5=000015   BITM6=000016   BITM7=000017
```

```
BITMA=000010   BITN =000005   BITR =000000   BITTA=136054   BITW =000001
BITX =000004   BLEV =000020   BLEVB=000040   BLEVL=000004   BLOCN=000002
BLOGO=134433   BLREG=025704   BLST =000030   BLSZ =177765   BMBYT=000003
BMECH=022145   BMEMO=000000   BNDC =000002   BNFIQ=000012   BOBOR=134625
BOTTE=000001   BOUPE=134535   BOXNO=000007   BPAGL=000001   BPART=000001
BPCFI=000001   BPLOG=000006   BPOPE=115031   BPRCL=000062   BPRFL=000002
BPRTM=000000   BPRTS=000016   BPSOK=000003   BPWAR=000002   BRECH=177756
BREGB=000030   BRELE=007772   BRESE=007745   BRESL=000013   BREST=022052
BRKM =015450   BRKMA=177763   BRKTA=177770   BRKTB=016327   BROAF=000001
BRTEX=007003   BRTWT=020456   BSAVE=000002   BSBRK=024153   BSEG =000003
BSEGL=000014   BSEND=000024   BSINI=000014   BSTAR=022051   BSTAT=000022
BSTDE=017027   BTBIN=000005   BTCLE=142032   BTIMQ=000012   BTLOA=142074
BTSTO=142011   BTYPR=000060   BUALL=000064   BUDIS=000004   BUFAD=000070
BUFAS=000145   BUFBA=000232   BUFER=000257   BUFFA=000054   BUFFE=000011
BUFFI=000014   BUFHA=000021   BUFMA=000201   BUFST=000014   BUFUL=000007
BUSFL=177743   BUSIS=000067   BUSIZ=000203   BUSTA=000064   BUSY =177767
BWLIN=000002   BXARG=000001   BXBAP=025627   BXBIT=000006   BXBRG=000002
BXLRG=000003   BXOTT=000003   BXTAD=000027   BXTRG=000000   BYPIN=000236
BYTEN=000020   BYTS =177775

CACCO=155223   CACHL=000151   CADRG=177753   CALIB=177654   CALLP=001437
CAMID=000110   CAOFF=150647   CAON =150642   CARG =177753   CBPTE=000001
CBSIZ=000200   CBUAD=000026   CCBRE=110462   CCBRS=110521   CCBWO=177644
CCCIN=147756   CCCOM=134274   CCDUM=141056   CCFPA=000044   CCLPA=000045
CCLR =000010   CCNN =011315   CCOLD=150714   CCTCO=152237   CCTRL=177771
CDAFI=131361   CDESC=146060   CDFIL=000012   CDISP=177726   CDRG =177754
CDRTC=132326   CDSPF=145550   CDVCO=171170   CEESC=146065   CENCT=134405
CERR1=120737   CERRC=177736   CESCP=177762   CEXIT=016724   CFLRG=177745
CFLUN=177622   CFREE=000020   CGTTS=160117   CGTTY=145433   CHARI=000027
CHBAT=164066   CHBME=014407   CHCLA=156373   CHFTR=162075   CHIOB=133404
CHIOP=162561   CHKO =000007   CHKRN=156347   CHMEM=025212   CHNR =022115
CHNST=000020   CHRTD=120434   CHSML=136252   CHTAC=150702   CHTER=140641
CHVAR=126417   CI4DF=022143   CIBUF=015100   CINBT=147765   CL7  =000025
CLACI=132604   CLADJ=011106   CLBUF=016717   CLCON=011304   CLEBA=164367
CLEDE=000020   CLFOR=153032   CLOCK=011024   CLOGD=000022   CLRG =177745
CM144=025712   CMAD1=000016   CMAD2=000017   CMADR=000016   CMAND=000002
CMCHT=170520   CMCPU=167577   CMDAT=177772   CMDFF=022147   CMLTS=156713
CMMCL=164746   CMRFI=135343   CMTRE=177721   CMWCN=177724   CMWFI=135345
CNAME=000000   CNOCH=000030   CNTRE=177765   CNTXL=000107   CNVRT=000313
COBST=022146   COBUF=015105   COFLA=000077   COMCO=000067   COMEN=023155
COMFL=177764   COMMO=013422   COMPA=000003   COMSB=116617   COMST=022527
COMTA=022075   CONCT=017065   CONTI=136216   CONTW=177773   CONTX=033774
COPCO=171275   COPYF=137433   CORAD=000027   CORCU=177722   CORES=006355
CORMB=000177   CORMS=000026   COTAB=000076   CPAG1=000005   CPAG2=000006
CPAGE=000005   CPARA=000001   CPAT1=177725   CPAT2=177727   CPLIS=022037
CPNT =022023   CPNUM=000357   CPROT=000003   CPSTA=000176   CPTSL=157150
CPUFI=171410   CPULO=000342   CPUNR=000032   CPUT5=025766   CRBIN=125316
CREAD=153154   CRESE=114646   CREVO=171410   CRLAM=143452   CRLF =153421
CRLRT=114633   CRSRT=114621   CRTRE=026006   CRTRF=000312   CSBSI=146400
CSGST=000226   CSPFN=000002   CSPJ1=000061   CSPJ2=000062   CSPJ3=000063
CSSL =021757   CSSLO=002034   CSSL1=021757   CSTAR=000203   CSTAT=137157
CSTCK=027056   CSTRI=022025   CSTTY=145360   CT500=025764   CTACN=177725
CTADR=177752   CTBIT=000017   CTIMU=154317   CTRCH=000041   CTRG =177752
CTRTI=016730   CTRTO=016746   CTTYP=177761   CUDBU=000017   CUIBU=000033
CUMSI=000012   CURBU=000017   CURID=000051   CURMA=000242   CURME=000047
CURPR=000010   CURUS=022103   CUSER=026001   CUUBU=000034   CWRIT=153167
CXFNA=000002   CXLOC=000547   CXNOC=000000   CXRG =177755   CXRTA=014762
CXSPJ=000061   CXSPM=000001   CXTST=000001   CXULO=000552   CXXRT=014760
```

```
D0    =000020   D1    =000021   D2    =000022   D3    =000023   D4    =000024
D5    =000025   D90   =000020   DABST=010657   DAC1C=000006   DAC1U=000007
DACCE=156614    DACCM=157045    DACT  =000004   DALCD=000044   DALCM=000042
DALFU=000041    DALOG=000010    DALTO=001731    DAREG=000003   DASAX=000101
DATAF=177635    DATCL=137334    DATIM=136052    DBACT=000041   DBADR=000025
DBCOU=177776    DBHCA=000064    DBLEN=000025    DBLO1=000013   DBLO2=000014
DBLOA=000013    DBLOC=000012    DBLOP=157122    DBLST=000030   DBLSZ=157101
DBPAG=000010    DBPCO=110161    DBPRE=000040    DBPRO=000024   DBREG=000007
DBSPR=127445    DBUFB=000010    DBYTP=157111    DCE19=000007   DCNRT=177775
DCOM  =022330   DCOMP=000063    DCONT=000003    DCORM=000177   DDASA=000101
DDAUF=000014    DDBLS=000030    DDD0  =000033   DDD1  =000034   DDD2  =000035
DDD3  =000036   DDD4  =000037   DDESD=157156    DDESF=157203   DDEVN=156440
DDIBA=000013    DDILF=000023    DDIRN=156376    DDR   =000000   DDREG=000004
DDRIV=017200    DDVNU=156707    DDW   =000001   DEC2  =153015   DECU  =153364
DEDFA=000027    DEFEL=156171    DEFES=145652    DEFHI=144574    DEFLI=150074
DEFLO=145771    DEFSG=124145    DEFSH=144570    DEFTE=147160    DELAM=143541
DELBE=163360    DELRE=163403    DEMAN=000005    DENTE=000017    DEPO  =150417
DERO  =000030   DER2  =000032   DERRC=177742    DERRO=000013    DFACC=156752
DFDCR=000126    DFDCW=000106    DFDEV=000023    DFHDL=135072    DFIAC=000051
DFLAG=177766    DFNAM=156642    DFNUM=157071    DFOPP=000012    DFPNT=000022
DFPRE=114054    DFPRM=123016    DFRFL=156472    DFRIN=156604    DFS1  =022220
DFS2  =022264   DFSNA=156651    DI1LA=000033    DI2LA=000034    DIECO=000040
DIERC=000000    DIERO=000010    DIERR=000030    DIERT=000020    DIESP=000044
DIEWC=000004    DIEWO=000014    DIEWR=000034    DIEWT=000024    DIFTC=000026
DIL1C=000000    DIL1D=000015    DIL1F=000031    DIL1L=000020    DIL1U=000001
DIL2D=000016    DIL2F=000032    DIL2L=000021    DILBA=000013    DILBO=000015
DILBP=000012    DILCO=000014    DILDA=000015    DILDE=000011    DILFA=000031
DILFL=000023    DILFU=000024    DILGF=000022    DILGL=000033    DILIP=150221
DILLA=000020    DILLI=000002    DILNS=000017    DILRA=000003    DILSM=000005
DILST=000010    DILWA=000004    DINCR=177625    DINTV=010521    DIOUT=016537
DIOUV=000010    DIRFL=000000    DISPN=000010    DISPR=132076    DITRM=147402
DIUET=000054    DIUEX=000050    DIVIS=177772    DKFUN=000015    DKNA1=000000
DKNA2=000001    DLALO=000027    DLAMD=000245    DLAUN=000030    DLDRI=000026
DLLOG=000025    DLOCK=000003    DLOGU=156450    DLREE=114653    DLREG=000005
DLRET=113601    DLSGF=124143    DMACN=115251    DMAIN=000016    DMAXB=000102
DMONI=017226    DMOUT=016543    DMRET=177741    DMSTR=010241    DNACO=000063
DNALT=001723    DNAME=000005    DNENT=157743    DNEWD=156424    DNFIL=156671
DNPAS=156573    DNUCO=157260    DNUMB=000004    DNUSE=156532    DOACC=157004
DOBJI=157034    DOLDD=156410    DOLDP=000016    DOORL=177624    DOPAS=156562
DOU0  =000000   DOU2  =000002   DOU4  =000004   DOU6  =000006   DOUTF=157133
DPACC=156720    DPAGE=156660    DPAGL=000000    DPAGP=000030    DPASS=156553
DPFNA=157243    DPGPR=000020    DPIN  =000001   DPNT0=000016    DPNT1=000017
DPNT2=000020    DPNT3=000021    DPPAW=157730    DPREE=114060    DPREG=000000
DRBIT=000016    DREAD=000060    DRESE=000004    DRFSG=026055    DRFUN=000026
DRG   =177750   DROPS=000015    DRSND=157701    DRT   =000033   DRTDE=117562
DRUMF=164737    DRUND=157715    DS0   =000000   DS1   =000002   DS2   =000004
DS3   =000006   DS4   =000010   DS5   =000012   DSCNT=017117    DSDEV=157216
DSET  =010545   DSGEN=157337    DSKTY=000002    DSOUD=157143    DSOUF=157173
DSPAC=156520    DSREB=000260    DSREG=000006    DST   =000002   DSTEN=120006
DTAPE=000015    DTAPT=017040    DTDEC=153360    DTIM1=000002    DTIM2=000003
DTIME=000002    DTIN1=000004    DTIN2=000005    DTINT=000004    DTLEN=000027
DTRCK=177640    DTREG=000002    DTRIG=000000    DTSLI=155643    DTUSE=000013
DTYPE=157351    DUBIM=000006    DUBM2=000010    DUFQU=157676    DUMCA=127113
DUMF  =134062   DUMP  =136322   DUMPX=024345    DUNIT=000001    DURPR=122655
DURSU=122603    DUSEI=156543    DUSEN=156511    DUSSU=122536    DV100=030237
DVNAM=000000    DVSTR=123011    DWONO=000073    DWORK=022701    DWRTE=000061
DXNDA=000035    DXNWD=000037    DXREG=000001    DY0   =000000   DY1   =000002
DY12  =000026   DY2   =000004   DY3   =000006   DY4   =000010   DY5   =000012
DY6   =000014   DY7   =000016   DY8   =000020
```

```
E2    =177776   ECBKF=000233   ECCBT=000012   ECCFL=177724   ECCR =000015
ECHOM=015541    ECHOT=177767   ECHTB=016337   ECL7 =000034   ECODS=000014
ECORM=000152    EDEVN=177772   EDIRS=022144   EDIRT=111521   EDIT =151341
EDOSG=022140    EDSVB=022137   EDSVL=022136   EEMTY=177775   EFILS=030005
EFUNC=177770    EGURS=021656   EILCO=136237   EMAXS=177771   EMOD =022135
EMPTF=177771    EMPTY=000000   EMSGS=177773   ENBUF=177774   ENDCO=000141
ENDPA=000157    ENDVT=164744   ENLIP=150217   ENTO =012273   ENT14=012204
ENTCT=022124    ENTER=022130   ENTOP=024017   ENTRM=147400   ENTRT=140664
EOPCA=027475    EOPSY=030011   EPAGP=000003   ER0  =000020   ER1  =000021
ER10 =000033    ER100=000163   ER101=000164   ER102=000165   ER103=000166
ER104=000167    ER105=000170   ER106=000171   ER107=000172   ER108=000173
ER109=000174    ER11 =000034   ER110=000175   ER111=000176   ER112=000177
ER113=000200    ER114=000201   ER115=000202   ER116=000203   ER117=000204
ER118=000205    ER119=000206   ER12 =000035   ER120=000207   ER121=000210
ER122=000211    ER123=000212   ER124=000213   ER125=000214   ER126=000215
ER127=000216    ER128=000217   ER129=000220   ER13 =000036   ER130=000221
ER131=000222    ER132=000223   ER133=000224   ER134=000225   ER135=000226
ER136=000227    ER137=000230   ER138=000231   ER139=000232   ER14 =000037
ER140=000233    ER141=000234   ER142=000235   ER143=000236   ER144=000237
ER145=000240    ER146=000241   ER147=000242   ER148=000243   ER149=000244
ER15 =000040    ER150=000245   ER151=000246   ER152=000247   ER153=000250
ER154=000251    ER155=000252   ER156=000253   ER157=000254   ER158=000255
ER159=000256    ER16 =000041   ER160=000257   ER161=000260   ER162=000261
ER163=000262    ER164=000263   ER165=000264   ER166=000265   ER167=000266
ER168=000267    ER169=000270   ER17 =000042   ER170=000271   ER171=000272
ER172=000273    ER173=000274   ER174=000275   ER175=000276   ER176=000277
ER177=000300    ER178=000301   ER179=000302   ER18 =000043   ER180=000303
ER181=000304    ER182=000305   ER183=000306   ER184=000307   ER185=000310
ER186=000311    ER187=000312   ER188=000313   ER189=000314   ER19 =000044
ER190=000315    ER191=000316   ER192=000317   ER193=000320   ER194=000321
ER195=000322    ER196=000323   ER197=000324   ER198=000325   ER199=000326
ER2  =000023    ER20 =000022   ER200=000327   ER201=000330   ER202=000331
ER203=000332    ER204=000333   ER205=000334   ER206=000335   ER207=000336
ER208=000337    ER209=000340   ER21 =000045   ER210=000341   ER211=000342
ER212=000343    ER22 =000046   ER23 =000047   ER24 =000050   ER25 =000051
ER26 =000052    ER27 =000053   ER28 =000054   ER29 =000055   ER3  =000024
ER30 =000056    ER31 =000057   ER32 =000060   ER33 =000061   ER34 =000062
ER35 =000063    ER36 =000064   ER37 =000065   ER38 =000066   ER39 =000067
ER4  =000025    ER40 =000070   ER41 =000071   ER42 =000072   ER43 =000073
ER44 =000074    ER45 =000075   ER46 =000076   ER47 =000077   ER48 =000100
ER49 =000101    ER5  =000026   ER50 =000102   ER51 =000103   ER52 =000104
ER53 =000105    ER54 =000106   ER55 =000107   ER56 =000110   ER57 =000111
ER58 =000112    ER59 =000113   ER6  =000027   ER60 =000114   ER61 =000115
ER62 =000116    ER63 =000117   ER64 =000120   ER65 =000121   ER66 =000122
ER67 =000123    ER68 =000124   ER69 =000125   ER7  =000030   ER70 =000126
ER71 =000003    ER72 =000127   ER73 =000130   ER74 =000131   ER75 =000132
ER76 =000133    ER77 =000134   ER78 =000135   ER79 =000136   ER8  =000031
ER80 =000137    ER81 =000140   ER82 =000141   ER83 =000142   ER84 =000143
ER85 =000144    ER86 =000145   ER87 =000146   ER88 =000147   ER89 =000150
ER9  =000032    ER90 =000151   ER91 =000152   ER92 =000153   ER93 =000154
ER94 =000155    ER95 =000156   ER96 =000157   ER97 =000160   ER98 =000161
ER99 =000162    ERASE=000014   ERCNT=177756   ERINI=150473   ERM  =000344
ERMSG=030143    EROPC=116447   ERPRI=150505   ERR14=012250   ERR22=011732
ERRC1=177733    ERRC2=177734   ERRFA=000406   ERRFL=110000   ERROR=177771
ERRS =116446    ERRSE=000005   ERSYS=027552   ESCAP=020054   ESCBL=022151
ESCBU=177755    ESCMA=170076   ESCOF=021012   ESCON=020735   ESCOP=023753
ESCQE=023736    ESGTA=000154   ESTCK=027045   ETCI =153564   EUSAD=000043
EXABS=017733    EXAM =150416   EXECC=154223   EXECI=145320   EXHEN=161417
EXIOX=016033    EXPAN=140404   EXSEC=000263   EXTDS=000066   EXUNA=110157
```

```
FACFL=026052   FBSIZ=177765   FCNT1=177632   FCNT2=177633   FCOMF=177644
FCPUL=167545   FCRCE=177756   FCST =000042   FDATA=015334   FDIFO=177662
FDIMO=177661   FDIST=177623   FDRIV=177657   FDTMS=014117   FENTL=026060
FFITA=120573   FFLOG=170370   FFMTA=120565   FFREL=000001   FFSTA=120601
FIACC=001777   FIBUF=022614   FILCA=027342   FILEN=000014   FILER=022127
FILNO=022066   FILSE=022073   FILSY=027561   FILUL=026057   FINBT=027121
FINIS=177770   FINRD=157364   FINRS=157412   FINST=012243   FIORE=014743
FIXCL=000106   FIXED=000002   FIXID=000052   FIXMA=000367   FIXPA=000340
FKICK=000002   FLAG =000004   FLAGB=000042   FLBGT=023045   FLLIP=023047
FLMAI=022122   FLMFA=177627   FLMQU=177626   FLOAD=000004   FLOGD=170345
FLOGO=134430   FLPT3=000105   FLQER=023046   FLRTT=000215   FMEMD=177646
FMEMH=177646   FMEML=177647   FMXRE=000002   FNABC=000347   FNAME=000072
FNO  =135445   FOBUF=022065   FORCE=000172   FORSP=000016   FOUTB=027175
FPAR1=000055   FPAR2=000056   FPAR3=000057   FPOFP=000237   FRACC=000001
FRECV=000001   FREEC=000167   FREES=113503   FREQU=002000   FRES =000000
FRETR=000030   FRETU=177660   FREXQ=010166   FRSCA=027702   FRSG1=000024
FRSG2=000025   FRTPR=177630   FRTRY=177624   FRUSE=136051   FRWQU=010173
FSABC=000344   FSEG =000006   FSEND=000000   FSPME=000071   FSPOO=000120
FSTA1=177652   FSTA2=177653   FSTAR=000006   FSTOP=000007   FSVCA=027655
FTIMQ=010354   FTPPR=115225   FULL =000006   FULLN=000002   FULLS=000003
FUNCT=145551   FUNLO=000005   FV0  =022000   FV11 =022011   FV3  =022003
FV6  =022006   FWFLA=026061   FWKIC=000003   FXSPF=026063   FYLLE=000021
FZERO=000100

GAPIT=002005   GBGSZ=117275   GCOM =151210   GDEVT=017707   GENDA=000055
GEPNE=145456   GET1L=025005   GETER=141407   GETOL=152150   GETPT=021151
GETRN=141042   GETSO=111617   GETVA=156304   GETW =016236   GETXM=017725
GGLOC=021056   GGSGM=115230   GLPAR=157774   GNLAM=000250   GNLPR=000251
GOOD =000004   GOTOU=136225   GPAR =152244   GPBFL=177765   GPDZI=177762
GPIIM=145477   GPRUN=177764   GPSEG=022046   GPUZI=177763   GPXTR=177761
GRI  =000002   GSI  =000000   GSO  =000011   GTMOD=017460   GURSE=027520
GVAL =145405   GZTRE=015305

HBRST=000110   HBUFA=000140   HDADR=022054   HDDAD=022056   HDEV =177775
HEADE=000011   HECHS=000112   HELP =142041   HENTE=000017   HENTF=000103
HINIF=000040   HMAXB=000102   HOLD =010713   HPEK =000016   HSTAT=000010
HTABL=000021   HUDV =002117   HWINF=000047   HXCC =000064   HXCOD=016531
HXERF=177767

IBACK=122200   IBLOA=000023   IBYTS=000050   ICCRT=000222   ICHRE=140303
ICLEP=140301   ICLK =010245   ICLNR=140302   ICORA=000022   ID10 =011477
ID11 =011540   ID12 =011574   ID13 =011635   IDADR=000103   IDBAD=000025
IDBST=000021   IDEVN=000100   IDLE =000016   IDNTS=000062   IEIDT=121256
IER  =025557   IEXI =164771   IFTRM=007063   IFUNC=000025   IGNAC=000010
IGTCH=016231   IICOR=000053   IIDTA=121251   IIFUN=000052   IIIOX=125604
IIMAX=000054   IINI =000013   IITAB=120504   ILLA =115103   ILLP =115113
ILSAV=000060   ILUTA=126737   IMASK=000137   IMAXB=000024   IMAXW=000024
IMBPR=000004   IMCBP=000003   IMTAB=120456   IMTFL=000056   INSMS=177754
INAMS=164647   INBRP=171010   INBT =014256   INCOM=141663   INCR =177776
INDA1=000031   INDA2=000053   INDAT=177761   INDB1=000014   INDB2=000013
INDEX=022020   INDX =000016   INDX1=000027   INDX2=000051   INFLG=000016
INHBT=000074   INIAD=000021   INIBU=141443   INIFL=000311   INIOS=016307
INITA=155001   INLOG=023054   INORO=121055   INSFG=026053   INTV =010512
IOB14=012207   IOLDP=000055   IOLOG=000020   IONIO=016653   IORES=014741
IOSET=015003   IOTRA=000007   IOUT =117221   IOXT =150415   IPFI =161175
IPTCH=016177   IQUEU=000061   IRETW=000026   IRI  =000003   IRWAI=000466
ISETP=140300   ISI  =000005   ISIZE=015173   ISO  =000017   ISPHE=127635
ISSRE=000051   ISTAB=120533   ISTAT=000004   ISTDV=016243   ISTR =000106
ISYSE=132537   ITIMT=120624   ITOM =000001   ITSLI=160155   IXSAC=177773
```

```
JAB2 =134677    JABOR=025616    JOBAB=134547

K1024=002000    K5ESC=000004    K5LOC=004000    K5NOS=001000    KABAC=105410
KALDR=010413    KALNX=010406    KBACT=105210    KGPAR=164340    KGPIB=145261
KICKP=000021    KLRC1=172117    KPROS=000011    KSETD=000011    KTMSU=177772
KTRVF=022117

L10LG=000077    L12LG=000074    L3EAV=153752    L4LGP=000075    LACTP=023052
LAINF=144156    LAMAC=000247    LAMAD=177654    LAMBA=000245    LAMCN=000000
LAMDT=000246    LAMFU=142334    LAMLP=000001    LAMNP=000001    LAMPP=000000
LAMPR=000002    LAPLI=142340    LAPRO=143612    LAREA=144010    LASBU=000020
LASMH=177654    LASML=177655    LAST =177771    LASTP=000015    LBUFA=000141
LCACH=000173    LCURB=000036    LCYLI=177741    LDADR=022053    LDATX=143300
LDBTX=143303    LDDAD=022055    LDDPI=021155    LDDTX=143302    LDEND=172205
LDILR=000050    LDITR=000046    LDIXR=000047    LDOLR=000031    LDOTR=000027
LDOXR=000030    LDPIO=021151    LDRAD=171733    LDTSZ=000003    LDVFC=150240
LDXTX=143301    LEAV2=154000    LEAVE=022131    LEAVX=153765    LEGSC=120411
LFADD=177666    LFCOU=000034    LGCOL=000227    LIADD=127575    LIBAT=162753
LIBEG=000014    LIBQU=163105    LIBSI=000064    LIDEV=140157    LIHDL=135076
LIICO=141720    LINR =000062    LIPBU=147746    LIPOI=000030    LIREE=111632
LIREQ=163062    LIRTL=156174    LISCO=125714    LISLI=150143    LISTR=152602
LISTS=137717    LISTT=137522    LISTX=137526    LITER=147264    LIVER=164476
LIVOL=171410    LLAMC=144461    LMCN =000000    LMDAT=000016    LMLP =000010
LMNP =000010    LMPP =000000    LMPR =000020    LNKSP=000061    LNUMB=000007
LOAD =136504    LOADI=000003    LOCAC=020235    LOCUS=026021    LODX =024410
LOGAD=000002    LOGDI=162027    LOGIN=126720    LOGOU=134731    LOGPH=007377
LOGSY=143302    LOOKA=150736    LOST =000075    LOUTT=136033    LPOFP=000240
LREMW=177657    LRESE=111725    LRESP=000202    LRSA =000021    LRTPR=140404
LSBUF=110160    LSTC =016110    LTSPR=157404    LUDV =002100    LUNIT=000002
LUSAD=000044    LUSER=000274    LUSTX=110035    LV10 =002000    LV10B=000120
LV11 =004000    LV11B=000130    LV12 =010000    LV12B=000140    LV13 =020000
LV13B=000150    LWBIT=000006    LWIAR=142355    LWPHY=000054

M144B=000012    M2UNT=000020    M4LRG=022142    M61RE=006423    M8RET=014615
MACOU=177740    MADR =000003    MAGTP=012725    MAIL =171535    MAILF=000050
MAINF=000010    MALTF=007167    MALTN=007124    MARGC=177742    MASKE=076033
MASSN=000370    MASSU=000374    MASTA=000014    MASTB=000041    MAX  =000015
MAXBH=000023    MAXBY=000103    MAXCT=022125    MAXCY=000003    MAXOP=000310
MAXP =000366    MAXUN=177737    MBDYN=150657    MBECH=117316    MBLEN=000020
MBOXH=000031    MBREA=000000    MBSYM=000175    MBTER=151530    MBZME=135130
MC5SP=000001    MCAL =001543    MCDES=017562    MCEES=017564    MCLR =016713
MCLRG=000034    MCMAG=000000    MCONT=000001    MCURB=000035    MDCUR=000035
MEMA1=000014    MEMA2=000015    MEMAD=000014    MEMLI=150603    MEMOR=136647
MENTR=023421    MESBU=136454    MESCA=020253    MESMA=135446    MESSF=000000
MESSI=000056    MFBBC=001523    MFUNC=000006    MGDAE=017611    MGTTY=015630
MILRG=000045    MINBH=000022    MINFR=000013    MINUS=023051    MIOIN=000004
MISTA=000031    MITRG=000044    MLBRO=140200    MLCTA=137051    MLDBR=140634
MLDLB=141350    MLDLM=141346    MLDSE=140455    MLEV =000010    MLEVB=000030
MLEVL=000003    MLFLA=000115    MLGRS=111152    MLHEL=137771    MLIAD=000035
MLICA=000032    MLICP=000031    MLIMS=000036    MLIMX=000033    MLINI=140042
MLINK=000005    MLIPA=000040    MLIST=000045    MLITE=000030    MLIWO=000034
MLLBR=141122    MLLME=141120    MLOAD=177744    MLRUN=140125    MLSEN=140176
MLSTO=140122    MLTTO=016656    MMEMO=000002    MMESI=000004    MMEXI=021614
MMREE=027723    MNCHK=157306    MND  =010403    MNTH1=010466    MOCOM=020634
MODE =161262    MODST=023013    MOFI2=012653    MOFIA=012647    MOFIB=012651
MOFIL=012623    MOFTY=023053    MOLRG=000026    MONBL=022170    MONDE=001557
MONED=151337    MONEN=000412    MONNO=000205    MONTH=010465    MOPEN=000003
MOSTK=022121    MOTRG=000025    MOVAA=143104    MOVAN=143103    MOVAP=143105
```

```
MOVB =140131    MOVBF=140132    MOVFL=167522    MOVNA=143101    MOVNN=143100
MOVNP=143102    MOVPA=143107    MOVPN=143106    MOVPP=143110    MOVSY=167527
MPIOC=000005    MQUEU=000011    MRECC=177722    MREMW=177656    MRET =001644
MRET1=012212    MRETU=177741    MRFI =012610    MRFO =012612    MRSTA=000030
MRTLA=000155    MSCIB=177646    MSDAE=017607    MSG  =024206    MSGN5=000001
MSIZE=177773    MSSTA=000014    MSTEN=023037    MSTOR=000400    MSTPN=022120
MSTTY=015626    MTBIT=000013    MTERM=017276    MTFLG=000027    MTIM1=000135
MTIM2=000136    MTIME=000135    MTLRG=177720    MTMRS=014121    MTOR =000006
MTRAN=000011    MTRNS=014056    MWCNT=177764    MWRIN=177742    MWSTA=177743
MXCHN=000102    MXMEL=000002    MXSET=012351    MXTIM=000017    MXTS0=164740
MXTS1=164741    MXTS2=164742    MXTS3=164743

N5RDF=000021    N5STA=000002    NAME1=000072    NAMSA=022135    NAMSR=164542
NAROU=022126    NBRTP=000220    NBSRT=000221    NBYTS=000060    NCBRK=177760
NCOMP=022141    ND500=000017    NDNCO=025740    NDNTO=025736    NDPRO=000022
NEWTR=177643    NEWUN=000065    NEXIN=141664    NFDIA=177726    NFLAD=142366
NFLAG=000013    NIMPL=155645    NINSZ=000024    NLOCK=000015    NLPAG=000027
NMATP=000343    NMTRE=177723    NMUSD=000026    NMUSM=000025    NNSWS=000024
NOBDI=000050    NOBJE=000400    NOBUF=177764    NOBYT=000103    NOCHA=000027
NOCHR=000104    NOCOP=000070    NOPGS=000140    NOSEC=177727    NOSIN=000032
NOWAI=015737    NOWFL=177744    NOWH =177750    NOWL =177751    NPFBU=000014
NPFUN=000001    NPHBU=000016    NPMAI=000024    NPOOL=030235    NPOPC=000001
NPTIG=000000    NRDTR=177776    NRDYF=177775    NSEGA=000347    NSEGB=000350
NSEGC=000351    NSLAR=142344    NSTAR=110357    NTERM=147155    NTLEN=000016
NTP  =000020    NULL =022067    NUMBE=000360    NUMIN=000033    NUSER=000400
NW2PA=111017    NWBPA=110646    NWLBA=177711    NWLBB=177710    NWORD=114366

O2SEG=000023    OABIT=000005    OACCE=000042    OBACK=000014    OBAER=025576
OBCOU=000044    OBFIL=000015    OBYTE=000061    OCBIT=000004    OCMD1=177712
OCMD2=177713    OCOMM=000013    OCOUN=000047    OCTRL=177774    OCTU =153337
ODATC=000051    ODATR=000053    ODATW=000055    ODEVN=000044    ODIRI=000001
ODUMP=121630    OFACC=000005    OFBLZ=000014    OFBUF=000010    OFCB =000012
OFDIR=000015    OFFLG=000007    OFFP =000025    OFFTP=000006    OFIND=000023
OFIOD=000023    OFIP =000017    OFIP1=000017    OFIP2=000020    OFLAG=000025
OFLCK=026007    OFLIB=000011    OFLOC=000011    OFNB =000013    OFNBR=000010
OFOBJ=000016    OFOP =000021    OFOP1=000021    OFOP2=000022    OFOUD=000024
OFPAG=000023    OFPDI=000011    OFSCR=000012    OFSET=000013    OFTYP=000043
OFWRT=000015    OIBIT=000003    OICOR=000034    OIFUN=000033    OILLP=153003
OIMAX=000035    OINDP=000003    OINDX=000005    OISYS=111302    OLBIT=000007
OLDPA=000017    OLDST=165233    OLDTR=177640    OLOCK=000000    OLSAV=000061
OMBIT=000006    OMEXI=021605    ON5MS=000025    ONAME=000026    ONDEX=000046
ONEXT=000040    OOLDP=000036    OOPEN=000050    OP2BG=110000    OP2EN=173774
OP2SE=022074    OP2SY=027547    OPAGE=000057    OPART=000002    OPBIT=000001
OPCAL=027423    OPCFI=116461    OPCHE=155537    OPCOM=116331    OPCOR=116307
OPEND=172205    OPERA=142426    OPFI =161205    OPFLG=177634    OPFTA=000277
OPNT =022024    OPOIN=000063    OPREV=000041    OPSEG=022071    OPSPO=030300
OPSYS=027544    OPTAB=030240    OPWCH=177650    OQUEU=000062    ORBIT=135451
OREAD=153161    ORESE=000015    OSBIT=000002    OSCPR=171457    OSEG =000003
OSIZE=015230    OSSRE=000032    OSTDV=016255    OSTRI=022026    OSVBA=177714
OSVBB=177715    OSVCO=177716    OSVWC=177717    OSVWK=177707    OTBIT=000000
OTMBI=000010    OTYPE=000036    OUSED=000017    OUSER=000045    OUTBT=014135
OUTTE=022132    OUTUS=126231    OWRIT=153174    OWRTE=000016

P0   =000000    P0PVL=012121    P1   =000001    P10PV=012157    P11PV=012162
P12PV=012165    P13PV=012170    P1PVL=012124    P2   =000002    P2PVL=012127
P3   =000003    P3PVL=012132    P3RET=022070    P4   =000004    P4PVL=012135
P5   =000005    P5PVL=012140    P6PVL=012143    P7PVL=012146    P8PVL=012151
P9PVL=012154    PAASF=157573    PACTA=000223    PAGEN=000016    PAGLI=000000
PAGPH=000002    PAGPR=000002    PANAM=022150    PANLI=157545    PAR1 =142335
```

```
PAR2 =142336   PAR3 =142337   PARA1=142341   PARA2=142342   PARA3=142343
PART1=000005   PARTN=177776   PASET=150426   PASST=022027   PASSW=126261
PAVA1=000010   PAVA2=000011   PAVAI=000010   PBBYT=000070   PBRK7=177744
PBRTE=007003   PBSIZ=000034   PCADR=000023   PCBNK=000022   PCNT =000004
PCORA=000022   PDATC=137310   PDDEB=000231   PDMOU=016641   PE1  =177746
PECH7=177734   PELIP=150033   PENTO=012271   PERCE=144325   PERMF=000017
PETEC=147033   PFLAM=143203   PFMSG=000016   PFOID=157655   PFULL=177750
PGC  =000014   PGNFL=000225   PIBSF=157557   PICKF=017234   PICKL=017241
PICKX=017250   PICKY=017255   PIKEY=121314   PILF =177754   PILSL=177752
PIMBH=000022   PIMBL=000023   PIMON=000255   PIMPG=000027   PINAP=157606
PINIT=000012   PINO =000031   PIOAC=000224   PIOCA=000013   PIOCI=000007
PIOCN=000030   PIOCO=000011   PIOF =150405   PION =150412   PIPOW=000013
PIPRO=000021   PISTT=000026   PKICK=000016   PLACE=136506   PLEFT=000023
PLMSG=000015   PLPRO=135637   PMFUN=000007   PMONE=000412   PN500=000230
PNBOX=000015   PNEXT=000022   PNI  =177745   PNLPA=157514   PNMAI=000023
PNOME=177747   PNOPC=000003   PNOTF=177744   PNOTP=177770   PNOTR=177767
PNOTY=177751   PNPAG=157533   PNUMB=000356   POK  =000001   POLSY=000004
PONN =150410   POOLL=000051   POOLP=000022   PORTN=177775   POTES=000104
PPIAC=157621   PPLOC=021074   PPREV=000023   PPROM=000002   PPRTD=000160
PRECP=177634   PREQU=000025   PRHIS=145043   PRINC=000171   PRIO =000014
PRIOR=006472   PRIRT=153112   PRJN =022113   PRJPA=000305   PRKEY=052163
PRLS =006715   PRNAM=025754   PROJ1=130263   PROJ3=130325   PROJN=126267
PROPE=115033   PRSCU=000060   PRSFN=157436   PRSR =000010   PRSRV=006647
PRTEX=007061   PRVFB=000005   PRVFL=000004   PRVMA=000003   PRVUE=000002
PRW  =000526   PRWAI=000455   PSBRT=007004   PSIZ =000030   PSLBS=177753
PSTDR=017266   PSTMR=017262   PSTSP=157460   PSTUP=000524   PSYSN=000302
PTERM=114136   PTLAM=142532   PTN  =000017   PTRNS=000014   PTSIN=110003
PUT1L=025010   PUTW =016214   PVEFU=000021   PVLAD=012103   PWCR =000007
PXT  =000037   PYERA=177654   PYRTL=137065   PZRTL=137072

QBSEM=000527   QERMS=030145

R1IMA=121102   R1MEM=121110   R1SAV=121126   RA   =000003   RALTO=001742
RAREG=000003   RBACK=122202   RBGUF=154171   RBPAR=161530   RBUFF=141763
RCALA=150114   RCOM =024315   RDATA=012306   RDHII=000026   RDLOO=000024
RDREG=000004   RDSC =007226   RDVDF=002400   RDVHI=002600   RDVLO=002500
READ =000000   READM=141576   REC80=001000   RECEI=000025   RECFI=135643
RECOv=135647   RECOX=024343   REEC =024262   REECE=112456   REECO=112002
REECS=112001   REECT=112457   REEDU=114064   REEPN=112000   REESS=022133
REETE=113503   REFOR=000005   REGPO=000005   REIDT=121257   RELEA=007643
RELES=006723   RELOA=171626   REMAI=000017   REMBY=000023   REMCM=134162
REMCO=162007   REMES=116576   REMLI=022114   REMRU=116600   REMSI=000046
REMTI=177757   REMUS=000271   REMWO=177656   RERNU=022076   RERPR=022077
RERRC=000026   RESCA=020070   RESCY=000006   RESER=007632   RESES=110176
RESI2=006355   RESIP=110151   RESLI=000000   RESOP=110154   RESRT=115063
RESRV=006654   RET  =006427   RET14=012212   RETBA=021476   RETRA=014101
RETRN=000100   RETRT=006411   RETRW=006421   RETSG=114346   RETST=006417
RETUA=114345   RETXI=006413   REUSE=142640   REVER=000011   REVLE=000054
REWIN=000013   REWUN=000017   REX  =150407   RFIEL=000067   RFIOX=125606
RFORM=000042   RGSI =000035   RHSTA=000022   RIDTA=121253   RIFIL=000026
RIMAG=120142   RING1=000000   RING2=000001   RINPA=110613   RLLSA=177763
RLTSA=177762   RLUTA=126741   RMEMO=120162   RMLNR=000027   RN1  =022100
RN2  =022101   RNACO=000037   ROFIL=000023   ROFIP=154044   ROUSP=177757
RPOOL=030234   RPORT=177775   RPREG=000000   RPRPA=026042   RPRS =000053
RRPOO=000267   RRTPR=022102   RSAVE=120202   RSCAL=027673   RSCLD=173730
RSCOL=135161   RSCUR=000060   RSEGM=000014   RSELR=177760   RSICH=000013
RSIO =117252   RSIST=177755   RSPHE=127637   RSPNU=000031   RSREG=000006
RSRET=000057   RSTAC=155736   RSTDE=017023   RSTOC=155740   RSYSN=026032
RT   =006463   RTACC=000217   RTACT=012317   RTBH =000024   RTBL =000025
RTBOX=000033   RTCHE=006360   RTCLD=000010   RTDCO=023622   RTDLG=000016
```

```
RTENT=007464   RTEXT=007022   RTFEL=164647   RTFPA=000042   RTIMT=120630
RTIN =000022   RTLPA=000043   RTLRM=115066   RTOUT=153057   RTRAN=170626
RTREF=000007   RTREG=000002   RTRES=000001   RTRSC=027711   RTSEG=022072
RTSTA=000024   RTSVC=027663   RTSVU=030070   RTTER=000214   RTUSU=030117
RTUT =000043   RTWT =006776   RTZFL=177730   RUSNA=026010   RW   =000461
RWAIT=000456   RWFIE=000004   RWPOF=000266   RXREG=000001

S0   =000000   S1   =000001   S10  =000010   S11  =000011   S12  =000012
S13  =000013   S14  =000014   S15  =000015   S16  =000016   S17  =000017
S2   =000002   S20  =000020   S21  =000021   S3   =000003   S3RET=022070
S4   =000004   S5   =000005   S6   =000006   S7   =000007   S90  =000000
S91  =000003   SABOR=000007   SACCO=155225   SAGPA=112110   SALMO=021747
SARDF=022105   SAVE =156332   SAVEF=000363   SAVFN=000062   SAVTA=022015
SAVUS=030045   SBHOL=000043   SBINB=027115   SBLST=000030   SBOUT=027171
SBRKD=024621   SBUFR=000070   SBYTS=000032   SC100=022774   SCAB =153234
SCADR=177741   SCDEV=157232   SCEDU=164172   SCHPR=130061   SCOMT=114532
SCOND=000026   SCPRI=177774   SCPRO=132620   SCREE=177770   SCRSI=022123
SCRWR=026062   SCSFS=132770   SCTIB=177637   SCTTR=177651   SCUSE=026004
SDFLA=030236   SDLOG=156460   SECSY=000002   SECT =177636   SECTO=000012
SECTP=000024   SECTR=000001   SECWO=000000   SEEKF=177732   SEGFN=164733
SEGLI=000000   SEGM =000007   SEGMA=000344   SEGMB=000345   SEGMC=000346
SEGPR=000355   SEGRE=000362   SEGST=000025   SELAR=120237   SELUN=000066
SENCT=134403   SEND =000023   SERRB=177757   SERVS=171603   SET  =010536
SETAV=141210   SETDV=000011   SETER=141255   SETME=136710   SETOL=116612
SETPA=016156   SETUN=141156   SEX  =150406   SFCOM=177623   SFILC=027337
SFILS=027556   SFORM=000040   SGCNU=000365   SGMAX=000015   SGPAR=152231
SGPBS=145304   SGPIB=144614   SGWPE=134670   SGWPR=134664   SHEAD=177734
SHSTA=177732   SIAD =177766   SIAD1=000021   SIAD2=000022   SIB50=000020
SICCO=000020   SIDRT=000021   SIM20=162531   SINCO=117446   SINIT=000013
SINSE=117036   SINVE=000052   SIZF =000252   SLAKK=000000   SLAMC=144560
SLCMO=154502   SLIPO=000020   SLMAX=000010   SLONG=177723   SLV10=011453
SLV11=011453   SLV12=011453   SLV13=011453   SM1AB=110544   SM2DE=113401
SM2LE=110321   SM2TC=113704   SM3LE=110273   SM3OC=113514   SMABL=110451
SMAGP=112122   SMARG=177737   SMBAC=113167   SMCAM=114517   SMCCL=113216
SMCHT=110373   SMCIL=114434   SMCRE=113120   SMCRL=113675   SMCWR=113133
SMDDE=113317   SMDEC=113540   SMDOC=113442   SMDTD=113534   SMDYN=114012
SMEDI=111020   SMEND=110205   SMENT=110250   SMFAM=114474   SMFFL=153330
SMFIB=114051   SMFIL=117235   SMGCO=110656   SMGFI=123327   SMGPA=112130
SMI  =000043   SMKGP=113061   SMLEA=110332   SMMAI=137165   SMO  =000044
SMOCT=113473   SMOOP=145522   SMORE=113125   SMOUT=113230   SMOWR=113140
SMPER=113600   SMSCA=113174   SMSGC=025742   SMSGP=112115   SMSPA=113663
SMSRC=111657   SMSTR=110232   SMTAC=114037   SMTCI=113723   SMTCO=113715
SMTIM=112007   SMTMT=111720   SMTRS=153672   SMTXF=114506   SMWIN=113735
SMXLE=110306   SMYES=110433   SNACO=000040   SNAMS=164533   SNLIN=000025
SNPAG=000024   SOAD =177770   SOFTA=177764   SOHSC=000020   SOUR1=000065
SOUR2=000066   SOURX=000065   SPACC=000235   SPACO=177720   SPAFL=177721
SPAGE=000031   SPAST=022030   SPERI=000002   SPFLA=000143   SPFNA=000035
SPINX=000010   SPJN1=000151   SPJN2=000152   SPJN3=000153   SPJNA=000151
SPL12=000143   SPLEN=000012   SPLMO=021743   SPMES=000161   SPMOD=000030
SPOOL=030233   SPOP =027103   SPROG=000001   SPRS =000052   SPSEG=000025
SPSGM=000043   SPUME=000071   SPUSH=027060   SQELS=000210   SQERR=000036
SQIOS=000005   SQSEM=000004   SQUEU=000003   SRCHI=153675   SRCHO=152162
SRCSH=111631   SREBA=000261   SREBB=000260   SREEC=024260   SRHIN=153675
SRMMO=154500   SRTCS=000004   SRTON=110177   SRTRE=000364   SRTRY=177735
SSBUF=142234   SSCLD=173652   SSDNU=127331   SSEC =177735   SSEG =000025
SSETE=141252   SSLEA=117154   SSMON=117057   SSPAL=014373   SSPAS=022124
SSPLE=000340   SSPOO=000270   SSREF=000031   SSSLE=117153   SSTEP=177736
SSTOP=000006   ST500=135564   STABA=154424   STACK=026145   STADR=000006
STAGP=110062   STAHI=144760   STAPL=143673   STATU=000001   STATX=143304
STBEG=022351   STCH =164711   STDEL=000013   STDEV=000010   STDPI=021145
```

```
STDRI=177776   STDTX=143306   STEND=022527   STEPR=177630   STERM=011363
STESC=020015   STFLG=153562   STMSV=135444   STOBA=154426   STOGP=110074
STOHI=144764   STOPL=144033   STOPS=110162   STOPT=140605   STORT=000063
STORX=000053   STPIO=021141   STPNT=022050   STRES=115135   STRFI=152767
STRHI=115147   STRLO=115160   STRNU=152776   STROF=115170   STRSE=000032
STRT =010377   STRTE=115200   STSLU=110562   STSTA=115124   STSUP=115337
STUPR=000446   STUSE=021620   STWOR=147747   STYPE=157327   STZTX=143305
SUBIN=000017   SURTM=000026   SURUS=000025   SUSDN=026005   SVBPR=000000
SVCAL=027644   SVERO=110016   SVER1=110020   SVER2=110022   SVER4=110026
SVER5=110030   SVERA=110003   SVLBA=177732   SVLBB=177731   SVLCA=177744
SVLCO=177740   SVLET=110014   SVLWC=177745   SVLWK=177706   SVTSL=000001
SVTXT=021674   SWAPP=001400   SWBUF=000053   SWDEN=126611   SWICH=000015
SWPFL=000341   SWPLO=166062   SWTRY=177736   SYCNT=017637   SYEND=025770
SYSNO=000046

T1P01=015224   T1P02=015260   T1P03=015607   T1P04=015707   T1P05=016015
T1P06=017427   T1P07=017601   T1P08=017622   T1P09=142337   T1P10=145730
T1P11=146144   T1P12=020264   T1P13=020766   T1P14=145430   T1P15=134026
TABLE=000072   TACNS=177762   TACOU=177763   TADRG=177746   TALRE=115053
TAPEF=147067   TBUSA=177761   TCI  =153457   TCNTI=012372   TCO  =153441
TCO2 =153430   TDBTP=000017   TDFBA=000244   TDFLG=177775   TDFPA=000243
TDFPH=177774   TDIND=000045   TDIPD=000044   TDISI=000112   TDOND=000010
TDOPD=000043   TDOSI=000054   TDRAD=000000   TDTAD=000015   TDTAF=000015
TDTAL=000016   TDVN =026000   TDXRE=000001   TECBU=146632   TERO0=000314
TERO1=000315   TERO2=000316   TERM =000026   TERMO=142254   TERMS=151675
TERWD=014706   TETTO=016461   TEXIA=016726   TEXIT=016725   TEXTA=022045
TEXTN=153311   TFILE=116456   TIME =011015   TIMER=012402   TIMON=022107
TIMOU=140040   TIMUS=154312   TINFO=177733   TIOOF=022111   TLBUF=172000
TLID =142433   TLINK=000000   TMLRE=177765   TMOUT=010761   TMPBU=177756
TMR  =177773   TMSUB=177772   TMTIM=154330   TNOWA=015742   TNPAG=142421
TODF =000142   TOEXQ=010112   TOOPC=154511   TOOPS=024337   TORTL=021543
TOUS =025320   TOWQU=010126   TPPAG=142405   TQCOP=023553   TRAIL=000012
TREPP=017717   TRFP =142441   TRG  =177746   TRGET=017016   TRGIN=177760
TRIG =000001   TRLRE=000007   TRNSF=177766   TRTER=171557   TRTPU=016522
TRTZ =177737   TSBAN=000404   TSEGA=000352   TSEGB=000353   TSEGC=000354
TSEGS=000355   TSGNA=111560   TSGNN=111552   TSLAN=110051   TSPEE=177764
TSPTR=000042   TSSIZ=000405   TSTAD=000045   TSTAT=000023   TTIFI=022022
TTIMQ=010310   TTIMR=016347   TTMCO=000007   TTMR =177774   TTNO =022021
TTOMR=016541   TTPUT=016514   TUSED=011352   TWT04=012017   TWT05=012015
TWT10=012013   TWT11=012011   TWT12=012007   TXAPP=023113   TXBBP=000000
TXBPR=000020   TXCBP=000010   TXFLL=000040   TXIBP=000040   TXICB=000030
TXSVB=000000   TXSVT=000010   TXUEF=000020   TXXSV=000060   TYPD =022771
TYPEC=177730   TYPRI=000003   TYPS =022766   TYREM=162021

UANTM=000015   UCACH=000174   UCLIN=177762   UCLOS=000007   UCOPA=000035
UCRAL=000017   UDATE=000037   UDD  =110061   UDD4 =110040   UDELP=000036
UDENT=000041   UDIRI=000001   UDROB=000010   UDSCN=000031   UDWOB=000012
UEADM=133137   UEBSZ=000037   UECHE=116525   UECMD=150747   UECMR=023107
UECOM=116625   UEDAT=023056   UEFLG=023055   UELGN=153454   UELOG=116653
UEMAX=000027   UENTE=000025   UEPRE=133043   UEXPF=000041   UEXRE=023106
UFDFD=000034   UFOBJ=000037   UFREE=000064   UFRIE=000055   UGDIE=000033
UGUIO=000016   UGUSN=000015   UINDP=000003   UINDX=000005   UINIT=000023
ULIOP=000021   ULOCK=000000   UMAGT=000003   UMDLF=000020   UMGFI=000030
UMROB=000011   UMRUS=000014   UMUID=000013   UNAFL=000104   UNAME=000026
UNDEX=000047   UNEXT=141057   UNRST=000064   UOFLG=000010   UOPEN=000006
UPART=000002   UPASS=000036   UPAVA=000043   UPDAT=011221   UPDSI=170763
UPUS1=000045   UPUS2=000046   UPUSE=000045   URBYP=000004   URENF=000042
UREST=000077   URFIL=000000   URLFE=000025   URPRO=000100   URSOF=000024
URTLI=000043   USAVU=030104   USBLP=000005   USBLS=000002   USBYP=000022
USCNT=017635   USDAD=000043   USDI =026002   USDVS=000010   USEGA=000142
```

```
USEGM=000016   USESC=020244   USET =025214   USFIL=000017   USIDX=000015
USLOG=000166   USNO =026003   USNXT=000053   USPAR=023040   USPEF=000027
USPER=000026   USPRV=000052   USRTB=000150   USRTW=020470   USTAR=000076
USTAT=177760   USTEM=000032   USTER=000040   USTX4=110036   USTX5=110034
USTXT=110060   UUELO=132654   UUSED=000017   UWFIL=000001   UWLOG=026056
UZERO=000144

V0   =022000   V12  =022014   V3   =022003   V6   =022006   VDABS=000020
VDBFR=177756   VDBLC=177767   VDBUF=177760   VDCDI=000040   VDCFU=000010
VDCHS=000070   VDCME=000020   VDCNR=000017   VDCNU=000060   VDCPF=000000
VDCRE=000372   VDCSE=000374   VDCST=000013   VDDFL=000234   VDDST=000014
VDFUN=177764   VDLIN=000012   VDMAD=177765   VDMB0=000015   VDMB1=000016
VDMBX=000015   VDMTT=177762   VDNBL=177771   VDPDI=000007   VDPFU=000004
VDPGL=000000   VDPGR=000001   VDPME=000005   VDPNU=000011   VDRTP=177763
VDUBS=152140   VDUNI=000017   VDUSB=111607   VDUST=111575   VDUTT=152126
VENTX=000026   VONA =157317

W1IMA=121114   W1MEM=121122   W1SAV=121161   W789P=022060   WAIF =134065
WAITF=013720   WAKEF=000063   WANKN=177726   WBUFF=141770   WCOUN=177651
WDATA=012300   WDCNT=177631   WDSCT=177706   WDX  =014640   WEOF =000012
WERRB=177760   WERRC=000036   WFLAG=000021   WFOPE=142534   WHERE=006745
WHOIS=140704   WHSTA=000032   WIMBA=121214   WINB =153510   WIND1=000361
WIND0=000015   WLINK=000010   WMESS=141672   WORKA=022701   WPECT=026054
WPROP=126313   WRTLO=137060   WSNLI=000032   WSNPA=000033   WT   =001657
WT10 =011501   WT11 =011542   WT12 =011576   WT13 =011637   WTRAN=170650
WXSAV=121201   WXTRA=170674

X21BL=000007   X21C1=000000   X21C2=000001   X21C3=000002   X21C4=000003
X21EN=000177   X21K0=000054   X21LO=000000   X21NL=000000   X21PL=000053
X21PM=000011   X21PP=000253   X21SA=000000   X21SB=000002   X21SC=000001
X21SL=000057   X21ST=000052   X21SY=000026   X21T0=177776   X21T1=177774
X21T2=177753   X21T3=177775   X21T4=177775   X21T5=177775   X21T6=177775
X21T7=177776   X21TA=177776   X21TB=177770   X2D00=000121   X2D02=000122
X2D05=000123   X2D06=000124   X2D0C=000116   X2D10=000125   X2D11=000126
X2D12=000127   X2D13=000130   X2DBC=000120   X2DBR=000065   X2DC1=000105
X2DC2=000106   X2DC3=000107   X2DC4=000110   X2DC5=000111   X2DC6=000112
X2DCC=000117   X2DCI=000104   X2DCN=000074   X2DDF=000076   X2DER=000100
X2DHD=177775   X2DL3=000131   X2DLA=000102   X2DLI=000103   X2DLS=000066
X2DMM=000113   X2DMP=000113   X2DPC=000077   X2DPI=000073   X2DPS=000101
X2DRL=000114   X2DSL=000070   X2DSP=000115   X2DSR=000067   X2DST=000075
X2DUI=000071   X2DUO=000072   X2F01=000010   X2F20=000032   X2F30=000033
X2F50=000035   X2FAB=000014   X2FAC=000017   X2FBR=000020   X2FCP=000015
X2FEC=000016   X2FFF=000012   X2FIC=000003   X2FIL=000002   X2FIS=000004
X2FNC=000011   X2FNI=000007   X2FNM=000005   X2FNP=000013   X2FNR=000006
X2FOK=000000   X2FSM=000001   X2FXX=000121   X2M1 =000006   X2M1B=000007
X2M2 =000010   X2M3 =000011   X2M4 =000012   X2M5 =000013   X2MCL=000011
X2MCP=000010   X2MFA=000007   X2MFU=000005   X2MS2=000013   X2MSL=000011
X2MSS=000012   X2MST=000006   X5DSK=164736   X5FUN=000050   X9HNT=161163
XALTO=025255   XALTS=000146   XAPPB=162334   XBADM=154430   XBBNK=000042
XBCHF=177663   XBEND=000044   XBLOG=110301   XBLPA=164735   XBMRE=023151
XBRTW=020473   XBSAV=000045   XBSTR=000043   XBTLO=142054   XCHAI=000000
XCSTR=022134   XDFOP=000001   XDGER=000006   XDGES=000005   XDINF=000001
XDPUR=000004   XDPUS=000003   XDSBP=000002   XEAIN=177762   XEBFC=177763
XEBNC=177761   XEBNY=177771   XECRA=177701   XEDRI=177744   XEIBP=177772
XEIDP=177741   XEIDR=177766   XEILF=177756   XEILM=177753   XEILR=177740
XEIMA=177755   XEIPN=177752   XEIRM=177776   XEIRT=177770   XEITL=177742
XEIXT=177727   XEMCH=177764   XEMFL=177754   XENDM=177765   XENDP=177743
XENIM=177773   XENOP=177767   XENOS=177737   XENRU=177733   XENSE=177736
XENTO=177732   XENUS=177731   XENVI=177757   XEPCL=177734   XEPRV=177751
XEPVR=177750   XERDE=000156   XEREJ=177730   XERNA=177747   XERND=177735
```

```
XEROP=023751   XEROV=177746   XETMM=177774   XEWNA=177760   XEXBF=177745
XFABR=000022   XFABW=000023   XFALM=000046   XFBNC=000014   XFCLS=000013
XFCRD=000031   XFDBK=000043   XFDCT=000001   XFDIB=000033   XFDMM=000045
XFDUB=000041   XFDUM=000000   XFFRM=000047   XFFWD=000013   XFGET=000002
XFGST=000017   XFHIP=000015   XFIPV=012062   XFM2P=000026   XFMLK=000024
XFMST=000011   XFMUL=000025   XFOPN=000012   XFOPS=000014   XFP2M=000027
XFPON=000015   XFPRV=000036   XFPST=000016   XFRCV=000015   XFREA=000006
XFREL=000003   XFRES=000014   XFRHD=000004   XFRIB=000034   XFRIN=000030
XFROU=000012   XFRRH=000040   XFRRO=000015   XFRTN=000037   XFSCM=000010
XFSEC=000011   XFSIN=000020   XFSMC=000044   XFSND=000014   XFSRL=000021
XFSTD=000032   XFSYS=000007   XFUSG=000015   XFWAK=000016   XFWDF=000042
XFWHD=000005   XFWIB=000035   XFWOK=000015   XFWRI=000007   XFWTF=000017
XGBRK=024550   XHDEV=177774   XINST=133771   XJABO=024054   XKXXX=041000
XLOCK=007662   XMACD=024463   XMCHK=134100   XMKIK=000005   XMLGR=136042
XMREM=134032   XMROU=000002   XMRW =013036   XMSGU=000161   XMSGX=000000
XMTHI=000003   XMTNO=000001   XMTPS=000006   XMTRE=000004   XMXSE=012346
XNDPR=000016   XNMAI=000022   XNOCH=000031   XNOWU=000063   XNTP =000006
XOFF =000023   XON  =000021   XOUTU=126255   XPIPR=000012   XPKIC=000000
XPMAI=000026   XPRHI=145064   XPTN =000002   XQCOP=023574   XRBLK=000031
XRBUS=000036   XRDDF=000003   XRETB=021473   XRFIL=013006   XRG  =177751
XRIIV=000015   XRILN=000020   XRIPT=000005   XRIRQ=000053   XRISE=000052
XRISN=000001   XRISY=000013   XRMFL=000042   XRMMP=000006   XRMTL=000010
XRNCO=000056   XRNEI=000016   XRNGA=000054   XRNLS=000033   XRNNA=000051
XRNRB=000045   XRNRO=000014   XRNSE=000037   XRNSP=000004   XRNTR=000023
XRNXD=000022   XRNXL=000021   XRNXM=000017   XRPAG=013212   XRPRV=000012
XRRDR=000000   XRRFU=000044   XRRNA=000035   XRRND=000050   XRRNL=000055
XRROV=000043   XRRPN=000040   XRRSR=000002   XRRTO=110452   XRRTS=000010
XRSA =000020   XRSMF=000011   XRSNR=000047   XRSOK=000000   XRSYD=000032
XRTAC=012333   XRTCH=006355   XRTFE=000026   XRTIS=000030   XRTLO=137046
XRTPR=126276   XRTRA=000024   XRTRE=000034   XRTRP=000025   XRTRT=000027
XRTSL=160157   XRTSR=000006   XRTTS=000012   XRUKS=000041   XRUNM=000007
XRUNN=000002   XRURT=000046   XRXXX=041100   XSAC =177765   XSBPR=110111
XSC10=023006   XSCNM=000103   XSCRS=000120   XSDLO=000123   XSDMC=000112
XSDRN=000111   XSDSY=000112   XSETU=012350   XSGFN=000147   XSGIN=000122
XSGMC=000113   XSGMG=000107   XSGNI=000105   XSGNM=000104   XSGSY=000113
XSLEK=000124   XSLET=000101   XSLKI=000114   XSMAX=000126   XSNAM=000102
XSNET=000125   XSNSP=000121   XSNUL=000100   XSPFM=026064   XSRTO=110441
XSSCI=000126   XSTCL=000116   XSTCX=027057   XSTDC=000117   XSTIN=000115
XSTOP=110206   XSWTP=171734   XTADR=000000   XTDAR=000030   XTDBR=000070
XTDDR=000040   XTDLR=000050   XTDPR=000000   XTDSR=000060   XTDTR=000020
XTDXR=000010   XTEMI=177775   XTEMO=177776   XTMRT=000153   XTRSE=000005
XTTIF=177622   XTTNO=177621   XUBI1=000007   XUBI2=000010   XUBI3=000011
XUBI4=000012   XUBI5=000013   XUBI6=000014   XUBI7=000015   XUBIM=000006
XUNIT=000057   XUNLO=007670   XURSE=000005   XUSEG=000004   XWAIT=000055
XWCHL=000004   XWFIL=013010   XWPAG=013214   XWPCR=000001   XWRTC=000011
XWSAR=000003   XWTCR=000007   XWTDR=000005   XWTTC=000013   XX100=000020
XXBER=000041   XXBIN=000003   XXCHE=000015   XXCLS=000014   XXEIE=000001
XXFBI=000023   XXHER=000027   XXHOM=000051   XXICM=000017   XXIEN=000005
XXIFL=000006   XXILL=000062   XXILN=000025   XXINP=000040   XXIOW=000002
XXIRT=000007   XXMCE=000004   XXMER=000032   XXMIN=000051   XXMMC=000022
XXMON=000021   XXN10=000046   XXN33=000054   XXNBF=000010   XXNCX=000056
XXNER=000033   XXNGA=000036   XXNI2=000050   XXNIM=000013   XXNLD=000044
XXNMM=000012   XXNOR=000016   XXNSG=000047   XXNXM=000043   XXPER=000024
XXRI2=000037   XXRIN=000011   XXRO2=000030   XXROU=000026   XXRUT=000042
XXSAV=022134   XXSBF=000050   XXSBK=000047   XXSPM=026065   XXTAI=000052
XXTAS=000031   XXUBF=000046   XXUSR=000045   XXVSX=000055   XXWS3=000053
XXWSY=000057   XZRTT=000262

YBCSE=164734   YBFIE=022200   YBRTW=020463   YESNO=142372   YFGET=013216
YFPUT=013220   YOUHA=137735   YRTLO=137077   YSVBP=000006   YSVTS=000007
```

```
ZAREG=000012    ZBREG=000016    ZCESC=000002    ZCNTR=000005    ZCTTY=000001
ZDBPR=177774    ZDREG=000013    ZLREG=000014    ZMEMO=136701    ZOARG=000035
ZOPRG=000032    ZOSRG=000040    ZPREG=000007    ZROUS=000000    ZRTLO=137053
ZSREG=000015    ZTADR=000011    ZTREG=000011    ZTSPE=000004    ZXREG=000010
                                                                % :: Filesystem numeric Symbol List

SINTRAN III/VSX  VERSION I                16.16.44    23 SEPT  1984

XXXXX  XX  XX    XXXXX  XXXXX  XX  XX  XXXXX  XXXXXX  XXXXX  XX   XX
XXXXX  XX  XX    XXXXX  XXXXX  XX  XX  XXXXX  XXXXXX  XXXXX  XX   XX
XX     XX  XX    XX     XX     XX  XX  XX       XX     XX     XXX XXX
XXXX   XX  XX    XXXX   XXXXX  XX  XX  XXXXX    XX     XXXX   XX X XX
XXXX   XX  XX    XXXX    XXXX   XXXX    XXXX    XX     XXXX   XX   XX
XX     XX  XX    XX        XX     XX      XX    XX     XX     XX   XX
XX     XX  XXXXX XXXXX  XXXXX    XX    XXXXX    XX     XXXXX  XX   XX
XX     XX  XXXXX XXXXX  XXXXX    XX    XXXXX    XX     XXXXX  XX   XX
SINTRAN III/VSX  VERSION I                16.16.45    23 SEPT  1984

SYMBOL LIST WITH VALUES IN ASCENDING ORDER.

P0    =000000   S0    =000000   S90   =000000   DS0   =000000   DDR   =000000
GSI   =000000   DY0   =000000   50K   =000000   DOUO  =000000   BENA  =000000
FRES  =000000   LMPP  =000000   LMCN  =000000   BITR  =000000   READ  =000000
XREG  =000000   RING1=000000   XMSGX=000000   TLINK=000000   DPREG=000000
XTDPR=000000   BXTRG=000000   XTADR=000000   RESLI=000000   BMEMO=000000
5ECHO=000000   5TLRE=000000   5BPAS=000000   5TDUM=000000   BECHO=000000
CNAME=000000   RPREG=000000   DIERC=000000   SECWO=000000   MCMAG=000000
7CUEL=000000   BBPRO=000000   BPRTM=000000   TXBBP=000000   SVBPR=000000
TXSVB=000000   DILIC=000000   7DATA=000000   EMPTY=000000   FSEND=000000
X21LO=000000   XCHAI=000000   X21C1=000000   X21SA=000000   X21NL=000000
XRRDR=000000   X2FOK=000000   NPTIG=000000   XPKIC=000000   DTRIG=000000
MBREA=000000   LAMPP=000000   LAMCN=000000   VDPGL=000000   VDCPF=000000
7NCML=000000   SEGLI=000000   PAGLI=000000   DPAGL=000000   DKNA1=000000
URFIL=000000   DIRFL=000000   DVNAM=000000   BDATA=000000   BDIRI=000000
ULOCK=000000   MESSF=000000   OLOCK=000000   OTBIT=000000   CXNOC=000000
RDATF=000000   P1    =000001   S1    =000001   DDW   =000001   ANI   =000001
.DA   =000001   POK   =000001   BIML  =000001   DPIN  =000001   ITOM  =000001
BBID  =000001   TRIG  =000001   BITW  =000001   TREG  =000001   ALEVL=000001
RING2=000001   STATU=000001   DXREG=000001   TDXRE=000001   BXARG=000001
RTRES=000001   2INBT=000001   5BREA=000001   5BCOM=000001   5ESCO=000001
5TLOW=000001   BOTTE=000001   CPARA=000001   RXREG=000001   SECTR=000001
MC5SP=000001   CBPTE=000001   BPCFI=000001   SVTSL=000001   DIL1U=000001
7DBRE=000001   7BDAT=000001   FRECV=000001   X21C2=000001   X21SC=000001
XWPCR=000001   X2FSM=000001   NPFUN=000001   NPOPC=000001   MCONT=000001
FFREL=000001   MSGN5=000001   LAMNP=000001   LAMLP=000001   VDPGR=000001
7ZMEM=000001   BPAGL=000001   5DEMA=000001   ALOGN=000001   DKNA2=000001
FRACC=000001   UWFIL=000001   .WRIT=000001   DUNIT=000001   BPART=000001
UDIRI=000001   BROAF=000001   ODIRI=000001   OPBIT=000001   SPROG=000001
```

```
CXSPM=000001   TADRE=000001   SFLOC=000001   P2    =000002   S2    =000002
DS1  =000002   DST  =000002   GRI  =000002   ,SA   =000002   DY1   =000002
ALEV =000002   DOU2 =000002   7RFI =000002   XPTN =000002   BNDC =000002
5FIX =000002   BITA =000002   AREG =000002   5CIMS=000002   FIXED=000002
ALMSZ=000002   DTIM1=000002   DTIME=000002   DTREG=000002   BXBRG=000002
BWLIN=000002   5BWLI=000002   5CLDV=000002   2OUTB=000002   BSAVE=000002
5SPEC=000002   5BUSE=000002   5ESC2=000002   5THIG=000002   BITER=000002
CMAND=000002   RTREG=000002   SECSY=000002   BPRFL=000002   BPWAR=000002
PRVUE=000002   DILLI=000002   7DECO=000002   FULLN=000002   MXMEL=000002
FMXRE=000002   BBYTC=000002   X21C3=000002   X21SB=000002   XRRSR=000002
X2FIL=000002   MMEMO=000002   PPROM=000002   FKICK=000002   N5STA=000002
LAMPR=000002   DSKTY=000002   7ZSCR=000002   LOGAD=000002   PAGPR=000002
PAGPH=000002   USBLS=000002   LUNIT=000002   BLOCN=000002   UPART=000002
OPART=000002   OSBIT=000002   SPERI=000002   CXFNA=000002   CSPFN=000002
ADREG=000002   QSGMT=000002   P3   =000003   S3   =000003   RA   =000003
S91  =000003   IRI  =000003   BSEG =000003   MADR =000003   OSEG =000003
BITC =000003   ER71 =000003   DREG =000003   FORM =000003   MLEVL=000003
5OPSE=000003   LDTSZ=000003   DTIM2=000003   DAREG=000003   BXLRG=000003
TYPRI=000003   5NORE=000003   2ECHO=000003   DCONT=000003   5HDUP=000003
5LOGO=000003   5BESC=000003   5TREA=000003   BXOTT=000003   CPROT=000003
RAREG=000003   MAXCY=000003   IMCBP=000003   BPSOK=000003   PRVMA=000003
DILRA=000003   7RQIN=000003   FULLS=000003   7ECKM=000003   BMBYT=000003
X21C4=000003   XWSAR=000003   X2FIC=000003   PNOPC=000003   MOPEN=000003
FWKIC=000003   LOADI=000003   7ZFPA=000003   5INHB=000003   EPAGP=000003
UMAGT=000003   COMPA=000003   DLOCK=000003   BBUFF=000003   UINDP=000003
OINDP=000003   OIBIT=000003   SQUEU=000003   P4   =000004   S4   =000004
DS2  =000004   DY2  =000004   5BAD =000004   DOU4 =000004   DACT =000004
GOOD =000004   PCNT =000004   FLAG =000004   BITD =000004   BITX =000004
LREG =000004   BLEVL=000004   5RTSG=000004   MMESI=000004   DTIN1=000004
DTINT=000004   DDREG=000004   XUSEG=000004   ISTAT=000004   2BRKM=000004
5FIMO=000004   5ABJO=000004   K5ESC=000004   5ERRO=000004   5CONT=000004
RDREG=000004   DIEWC=000004   POLSY=000004   IMBPR=000004   PRVFL=000004
DILWA=000004   7IRQI=000004   7BMMX=000004   BUDIS=000004   BCHAI=000004
XWCHL=000004   X2FIS=000004   MIOIN=000004   FLOAD=000004   SRTCS=000004
VDPFU=000004   BACKG=000004   7NPAS=000004   5SYSE=000004   URBYP=000004
DNUMB=000004   DRESE=000004   RWFIE=000004   OCBIT=000004   SQSEM=000004
P5   =000005   S5   =000005   ISI  =000005   BAD  =000005   5RQI =000005
BITN =000005   BREG =000005   5ERRS=000005   DTIN2=000005   DLREG=000005
XURSE=000005   XTRSE=000005   MLINK=000005   5TERM=000005   5CTRL=000005
5LBLO=000005   5REMO=000005   5WCON=000005   REFOR=000005   BTBIN=000005
PRVFB=000005   DILSM=000005   7CONF=000005   BHEAD=000005   X2MFU=000005
XWTDR=000005   X2FNM=000005   REGPO=000005   MPIOC=000005   FUNLO=000005
VDPME=000005   DEMAN=000005   5SPRO=000005   5SEGS=000005   ERRSE=000005
USBLP=000005   CPAG1=000005   CPAGE=000005   PART1=000005   DNAME=000005
OFACC=000005   UINDX=000005   OINDX=000005   OABIT=000005   SQIOS=000005
S6   =000006   DS3  =000006   DY3  =000006   DOU6 =000006   FULL =000006
X2M1 =000006   XNTP =000006   MTOR =000006   FSEG =000006   5FILS=000006
STADR=000006   DSREG=000006   XUBIM=000006   DUBIM=000006   BXBIT=000006
MFUNC=000006   5IBDV=000006   5WECH=000006   5WRQI=000006   5ALEC=000006
5LCHA=000006   5RERU=000006   RSREG=000006   RESCY=000006   5SPRF=000006
BPLOG=000006   VSVBP=000006   DAC1C=000006   7CORQ=000006   X2MST=000006
XRTSR=000006   X2FNR=000006   FSTAR=000006   5SREE=000006   UOPEN=000006
CPAG2=000006   LWBIT=000006   OFFTP=000006   OMBIT=000006   SSTOP=000006
S7   =000007   ,ST  =000007   SEGM =000007   5REG =000007   5XON =000007
CHKO =000007   PWCR =000007   5MACD=000007   DBREG=000007   XUBI1=000007
TRLRE=000007   5INVR=000007   5HDMA=000007   ZPREG=000007   2RPAG=000007
IOTRA=000007   5USES=000007   5CFIL=000007   ALTFO=000007   5BRKF=000007
TTMCO=000007   VSVTS=000007   DAC1U=000007   BOXNO=000007   BUFUL=000007
7CORS=000007   X2MFA=000007   X2M1B=000007   X21BL=000007   DCE19=000007
XWTCR=000007   X2FNI=000007   PIOCI=000007   FSTOP=000007   PMFUN=000007
```

```
VDPDI=000007   RTREF=000007   UCLOS=000007   LNUMB=000007   5PHOE=000007
OFFLG=000007   OLBIT=000007   SABOR=000007   S10  =000010   DS4  =000010
DY4  =000010   MLEV =000010   CCLR =000010   X2M2 =000010   PRSR =000010
LMLP =000010   LMNP =000010   SLAKK=000010   ALEVB=000010   5RTFI=000010
USDVS=000010   WLINK=000010   BITMA=000010   5WLIN=000010   5BITM=000010
XTDXR=000010   XUBI2=000010   DUBM2=000010   HSTAT=000010   5FLOP=000010
ZXREG=000010   2WPAG=000010   RTCLD=000010   STDEV=000010   5XOFF=000010
TDOND=000010   5ESCL=000010   5ND5E=000010   DIERO=000010   DISPN=000010
5ESCF=000010   9IVAL=000010   DIOUV=000010   TXCBP=000010   5PRVT=000010
TXSVT=000010   DILST=000010   MAINF=000010   IGNAC=000010   7ESCA=000010
X2MCP=000010   XRRTS=000010   X2F01=000010   SLMAX=000010   VDCFU=000010
CURPR=000010   DALOG=000010   UDROB=000010   ADVAN=000010   DBUFB=000010
DBPAG=000010   PAVA1=000010   PAVAI=000010   5RCFI=000010   OFBUF=000010
OFNBR=000010   UOFLG=000010   OTMBI=000010   BBASP=000010   SPINX=000010
5MT  =000011   S11  =000011   GSO  =000011   X2M3 =000011   5LOGR=000011
ACTSE=000011   BITM1=000011   XUBI3=000011   MTRAN=000011   5CRDL=000011
ZTREG=000011   ZTADR=000011   2TIME=000011   KSETD=000011   SETDV=000011
5XDEV=000011   5WESC=000011   5LOGI=000011   3FLOP=000011   5NOSL=000011
9OUVA=000011   DILDE=000011   ANCHA=000011   7DCON=000011   X2MSL=000011
X2MCL=000011   X21PM=000011   XWRTC=000011   X2FNC=000011   KPROS=000011
PIOCO=000011   VDPNU=000011   MQUEU=000011   UMROB=000011   REVER=000011
BUFFE=000011   PAVA2=000011   BDBIT=000011   5REMP=000011   OFLIB=000011
OFLOC=000011   OFPDI=000011   BFSSP=000011   HEADE=000011   S12  =000012
DS5  =000012   DY5  =000012   X2M4 =000012   WEOF =000012   OFCB =000012
5IFS2=000012   CUMSI=000012   ACTPR=000012   BITM2=000012   XUBI4=000012
5TMOU=000012   M144B=000012   ZAREG=000012   2SETO=000012   DFOPP=000012
5OXON=000012   5WLOC=000012   3STRE=000012   7CUTY=000012   1DTU1=000012
1TUSE=000012   DILBP=000012   1DILB=000012   CDFIL=000012   7SYSI=000012
X2MSS=000012   XRTTS=000012   X2FFF=000012   PINIT=000012   XPIPR=000012
VDLIN=000012   BTIMQ=000012   UDWOB=000012   DBLOC=000012   SECTO=000012
ECCBT=000012   OFSCR=000012   BNFIQ=000012   TRAIL=000012   SPLEN=000012
S13  =000013   5ABS =000013   5COM =000013   IINI =000013   7LUN =000013
X2M5 =000013   5PGU =000013   OFNB =000013   5IRTS=000013   BRESL=000013
BITM3=000013   5BRES=000013   XUBI5=000013   ZDREG=000013   2CIBU=000013
DERRO=000013   5CAPI=000013   5LOC2=000013   STDEL=000013   3FLTI=000013
1DTU2=000013   DILBA=000013   DDIBA=000013   2DILB=000013   RSICH=000013
SINIT=000013   MINFR=000013   OFSET=000013   X2MS2=000013   XWTTC=000013
X2FNP=000013   PIOCA=000013   PIPOW=000013   VDCST=000013   BEXQU=000013
UMUID=000013   REWIN=000013   DBLO1=000013   DBLOA=000013   DTUSE=000013
NFLAG=000013   MTBIT=000013   INDB2=000013   OCOMM=000013   ,PRIN=000013
PGC  =000014   S14  =000014   DY6  =000014   5INT =000014   PRIO =000014
5WIP =000014   ECODS=000014   5N100=000014   RSEGM=000014   BITM4=000014
XUBI6=000014   5ISET=000014   ZLREG=000014   2COBU=000014   BUFST=000014
6XOFF=000014   5FFGP=000014   MEMA1=000014   MEMAD=000014   DIEWO=000014
3FRES=000014   5DTU1=000014   5TUSE=000014   2DIBA=000014   DILCO=000014
LIBEG=000014   NPFBU=000014   BSINI=000014   MSSTA=000014   BUFFI=000014
7TMOD=000014   X2FAB=000014   MASTA=000014   VDDST=000014   BSEGL=000014
UMRUS=000014   ERASE=000014   DBLO2=000014   DDAUF=000014   10BIT=000014
PTRNS=000014   INDB1=000014   OFBLZ=000014   OBACK=000014   FILEN=000014
S15  =000015   MAX  =000015   ECCR =000015   5ISMS=000015   5SEXI=000015
WINDO=000015   BITM5=000015   XUBI7=000015   5RWAI=000015   5SWWA=000015
6CONC=000015   BISYM=000015   ZSREG=000015   5FRER=000015   5IESC=000015
USIDX=000015   5MLGI=000015   MEMA2=000015   3DOUA=000015   7SATY=000015
5DTU2=000015   DIL1D=000015   DILDA=000015   DILBO=000015   SWICH=000015
PLMSG=000015   UANTM=000015   TDTAF=000015   TDTAD=000015   7TTYP=000015
X2FCP=000015   PNBOX=000015   DROPS=000015   VDMBO=000015   VDMBX=000015
SGMAX=000015   UGUSN=000015   BACKS=000015   DKFUN=000015   DTAPE=000015
OBFIL=000015   1USER=000015   NLOCK=000015   OFWRT=000015   OFDIR=000015
ORESE=000015   LASTP=000015   S16  =000016   DY7  =000016   5REP =000016
IDLE =000016   HPEK =000016   INDX =000016   5INNE=000016   RTDLG=000016
```

```
BITM6=000016   5RTOF=000016   5RFIL=000016   BIHDL=000016   ZBREG=000016
DOLDP=000016   BHOLD=000016   5PAER=000016   5LBRK=000016   5SPSI=000016
CMAD1=000016   CMADR=000016   3DOUB=000016   BPRTS=000016   DIL2D=000016
NPHBU=000016   LMDAT=000016   PFMSG=000016   TDTAL=000016   7CESC=000016
X2FEC=000016   PKICK=000016   XNDPR=000016   VDMB1=000016   USEGM=000016
UGUIO=000016   FORSP=000016   DPNT0=000016   DMAIN=000016   DRBIT=000016
NTLEN=000016   INFLG=000016   OFOBJ=000016   OWRTE=000016   PAGEN=000016
S17  =000017   ISO  =000017   PTN  =000017   5WPM =000017   OFIP =000017
5IPT3=000017   BITM7=000017   5RTSI=000017   5WAIT=000017   5BACK=000017
5IOBT=000017   OLDPA=000017   HENTE=000017   5BLOC=000017   5BFUL=000017
5RDEV=000017   MXTIM=000017   5LSTA=000017   CMAD2=000017   3ILLF=000017
DILNS=000017   CUDBU=000017   CURBU=000017   TDBTP=000017   7DESC=000017
X2FAC=000017   BCFLA=000017   VDCNR=000017   VDUNI=000017   ND500=000017
UCRAL=000017   REWUN=000017   DPNT1=000017   DENTE=000017   USFIL=000017
CTBIT=000017   PERMF=000017   OFIP1=000017   UUSED=000017   OUSED=000017
SUBIN=000017   REMAI=000017   D0   =000020   D90  =000020   S20  =000020
NTP  =000020   DY8  =000020   ER0  =000020   BLEV =000020   XRSA =000020
LMPR =000020   55SDS=000020   ARSEG=000020   XTDTR=000020   5XRTD=000020
CFREE=000020   BCHNU=000020   CLEDE=000020   M2UNT=000020   DIERT=000020
IOLOG=000020   TXBPR=000020   TXUEF=000020   DIL1L=000020   DILLA=000020
LASBU=000020   SOHSC=000020   SLIPO=000020   CHNST=000020   7USID=000020
X2FBR=000020   SIB50=000020   SICCO=000020   VDABS=000020   VDCME=000020
DPGPR=000020   UMDLF=000020   DPNT2=000020   MBLEN=000020   OFIP2=000020
BYTEN=000020   D1   =000021   S21  =000021   XON  =000021   ER1  =000021
LRSA =000021   OFOP =000021   55NSG=000021   AACSE=000021   FYLLE=000021
HTABL=000021   PVEFU=000021   BADTA=000021   WFLAG=000021   DIL2L=000021
N5RDF=000021   INIAD=000021   IDBST=000021   7PASS=000021   PIPRO=000021
KICKP=000021   SIAD1=000021   SIDRT=000021   ULIOP=000021   DPNT3=000021
BIFIL=000021   OFOP1=000021   BUFHA=000021   D2   =000022   RTIN =000022
ER20 =000022   5FIUS=000022   MINBH=000022   BSTAT=000022   CLOGD=000022
RHSTA=000022   ICORA=000022   DILGF=000022   DFPNT=000022   POOLP=000022
NDPRO=000022   XNMAI=000022   PIMBH=000022   SIAD2=000022   USBYP=000022
PNEXT=000022   OFOP2=000022   PCBNK=000022   PCORA=000022   D3   =000023
ER2  =000023   XOFF =000023   SEND =000023   5OP2S=000023   MAXBH=000023
TSTAT=000023   6TTYN=000023   ROFIL=000023   DFDEV=000023   IBLOA=000023
DILFL=000023   DDILF=000023   REMBY=000023   7SYCN=000023   PNMAI=000023
PIMBL=000023   02SEG=000023   UINIT=000023   PPREV=000023   PLEFT=000023
OFIND=000023   OFIOD=000023   OFPAG=000023   BBLEN=000023   PCADR=000023
D4   =000024   ,SP  =000024   ER3  =000024   RTBH =000024   FRSG1=000024
NINSZ=000024   NNSWS=000024   RDLOO=000024   IMAXB=000024   DBPRO=000024
BCHOS=000024   DIEWT=000024   IMAXW=000024   DILFU=000024   BSEND=000024
7USCN=000024   NPMAI=000024   RTSTA=000024   URSOF=000024   SECTP=000024
OFOUD=000024   SNPAG=000024   D5   =000025   CL7  =000025   ER4  =000025
RTBL =000025   SSEG =000025   OFFP =000025   SPSEG=000025   FRSG2=000025
ON5MS=000025   DBADR=000025   IFUNC=000025   DLLOG=000025   IDBAD=000025
RECEI=000025   MOTRG=000025   7FBSI=000025   PREQU=000025   SEGST=000025
URLFE=000025   DBLEN=000025   NMUSM=000025   SURUS=000025   UENTE=000025
OFLAG=000025   SNLIN=000025   ER5  =000026   TERM =000026   DY12 =000026
5FIU2=000026   2LAST=000026   RDHII=000026   CBUAD=000026   RIFIL=000026
DIFTC=000026   RERRC=000026   IRETW=000026   DLDRI=000026   VENTX=000026
MOLRG=000026   DRFUN=000026   7RESE=000026   X21SY=000026   PISTT=000026
XPMAI=000026   CORMS=000026   USPER=000026   NMUSD=000026   SURTM=000026
UNAME=000026   ONAME=000026   SCOND=000026   ER6  =000027   5NNET=000027
2RTDS=000027   NOCHA=000027   CHARI=000027   BCHIS=000027   DEDFA=000027
MTFLG=000027   DLALO=000027   RMLNR=000027   LDOTR=000027   BXTAD=000027
7RECO=000027   PIMPG=000027   CORAD=000027   USPEF=000027   DTLEN=000027
INDX1=000027   NLPAG=000027   ER7  =000030   DERO =000030   PSIZ =000030
BLST =000030   MLEVB=000030   XTDAR=000030   2GETR=000030   CNOCH=000030
BREGB=000030   5BREG=000030   DIERR=000030   MRSTA=000030   5MRST=000030
TXICB=000030   DLAUN=000030   LIPOI=000030   FRETR=000030   LDOXR=000030
```

```
ACTOU=000030   7DUMM=000030   PIOCN=000030   MLITE=000030   SBLST=000030
DDBLS=000030   DBLST=000030   DPAGP=000030   UMGFI=000030   SPMOD=000030
ER8  =000031   PINO =000031   2EXIO=000031   XNOCH=000031   SSREF=000031
DIL1F=000031   DILFA=000031   MISTA=000031   LDOLR=000031   RSPNU=000031
7STRQ=000031   MBOXH=000031   MLICP=000031   UDSCN=000031   INDA1=000031
SPAGE=000031   ER9  =000032   2MSG =000032   DER2 =000032   5ACCS=000032
ZOPRG=000032   WHSTA=000032   STRSE=000032   DIL2F=000032   SBYTS=000032
OSSRE=000032   7STRS=000032   X2F20=000032   CPUNR=000032   MLICA=000032
USTEM=000032   WSNLI=000032   NOSIN=000032   DRT  =000033   DDDO =000033
ER10 =000033   5XSG1=000033   2ALTN=000033   DI1LA=000033   DILGL=000033
CUIBU=000033   OIFUN=000033   7KEYI=000033   X2F30=000033   RTBOX=000033
MLIMX=000033   UGDIE=000033   WSNPA=000033   NUMIN=000033   ECL7 =000034
DDD1 =000034   ER11 =000034   5XSG2=000034   2ALTF=000034   DIEWR=000034
MCLRG=000034   DI2LA=000034   CUUBU=000034   OICOR=000034   7BADT=000034
MLIWO=000034   PBSIZ=000034   UFDFD=000034   LFCOU=000034   RGSI =000035
DDD2 =000035   ER12 =000035   2IOUT=000035   ZOARG=000035   1XNDA=000035
DXNDA=000035   OIMAX=000035   X2F50=000035   MCURB=000035   MDCUR=000035
MLIAD=000035   UCOPA=000035   SPFNA=000035   DDD3 =000036   ER13 =000036
5BADM=000036   2NOWA=000036   WERRC=000036   2XNDA=000036   SQERR=000036
OOLDP=000036   LCURB=000036   ACCRL=000036   MLIMS=000036   UDELP=000036
UPASS=000036   OTYPE=000036   PXT  =000037   DDD4 =000037   ER14 =000037
5RT2S=000037   1XNWD=000037   DXNWD=000037   RNACO=000037   7OPSV=000037
UFOBJ=000037   UDATE=000037   ER15 =000040   BLEVB=000040   5RRUS=000040
XTDDR=000040   ZOSRG=000040   DBPRE=000040   DIECO=000040   TXIBP=000040
TXFLL=000040   2XNWD=000040   SNACO=000040   7ESRS=000040   HINIF=000040
MLIPA=000040   VDCDI=000040   BGFPA=000040   USTER=000040   SFORM=000040
ONEXT=000040   ER16 =000041   5PT3S=000041   DBACT=000041   DALFU=000041
CTRCH=000041   7CERS=000041   MASTB=000041   BGLPA=000041   UEXPF=000041
UDENT=000041   OPREV=000041   FCST =000042   ER17 =000042   5ISPS=000042
FLAGB=000042   DALCM=000042   BADAN=000042   7ISRQ=000042   XBBNK=000042
RTFPA=000042   URENF=000042   RFORM=000042   TSPTR=000042   OACCE=000042
SMI  =000043   RTUT =000043   ER18 =000043   SPSGM=000043   2CLOS=000043
2CLSE=000043   SBHOL=000043   TDOPD=000043   EUSAD=000043   USDAD=000043
7ISRS=000043   XBSTR=000043   RTLPA=000043   URTLI=000043   UPAVA=000043
OFTYP=000043   SMO  =000044   ER19 =000044   TDIPD=000044   LUSAD=000044
DIESP=000044   DALCD=000044   MITRG=000044   OBCOU=000044   7NOWT=000044
XBEND=000044   CCFPA=000044   ODEVN=000044   ER21 =000045   TDIND=000045
ACQFP=000045   MILRG=000045   TSTAD=000045   7TNOW=000045   XBSAV=000045
MLIST=000045   CCLPA=000045   UPUS1=000045   UPUSE=000045   OUSER=000045
ER22 =000046   ACQHP=000046   LDITR=000046   REMSI=000046   7NWRE=000046
XXUBF=000046   SYSNO=000046   UPUS2=000046   ONDEX=000046   ER23 =000047
ACQBH=000047   LDIXR=000047   CURME=000047   7RLOC=000047   XXSBK=000047
HWINF=000047   UNDEX=000047   OCOUN=000047   7IAM =000050   ER24 =000050
XTDLR=000050   2NOPE=000050   DIUEX=000050   IBYTS=000050   LDILR=000050
NOBDI=000050   XXSBF=000050   MAILF=000050   OOPEN=000050   ER25 =000051
CURID=000051   ISSRE=000051   POOLL=000051   7EDRS=000051   XXHOM=000051
INDX2=000051   DFIAC=000051   ODATC=000051   SPRS =000052   ER26 =000052
2MTER=000052   IIFUN=000052   7TREP=000052   FIXID=000052   X21ST=000052
SINVE=000052   USPRV=000052   RPRS =000053   ER27 =000053   IICOR=000053
SWBUF=000053   X21PL=000053   INDA2=000053   USNXT=000053   ODATR=000053
STORX=000053   ACQU =000054   ER28 =000054   TDOSI=000054   DIUET=000054
IIMAX=000054   LWPHY=000054   X21KO=000054   REVLE=000054   UFREE=000054
BUFFA=000054   ER29 =000055   IOLDP=000055   XWAIT=000055   GENDA=000055
UFRIE=000055   ODATW=000055   FPAR1=000055   ER30 =000056   IMTFL=000056
MESSI=000056   FPAR2=000056   ER31 =000057   XUNIT=000057   RSRET=000057
X21SL=000057   OPAGE=000057   FPAR3=000057   ER32 =000060   XTDSR=000060
5WORK=000060   BTYPR=000060   TXXSV=000060   ILSAV=000060   RSCUR=000060
PRSCU=000060   VDCNU=000060   DREAD=000060   NBYTS=000060   ER33 =000061
OLSAV=000061   IQUEU=000061   DWRTE=000061   OBYTE=000061   LNKSP=000061
CXSPJ=000061   CSPJ1=000061   LINR =000062   ER34 =000062   2RMAX=000062
```

```
BPRCL=000062   OQUEU=000062   IDNTS=000062   SAVFN=000062   CSPJ2=000062
ER35 =000063   XNOWU=000063   DNACO=000063   WAKEF=000063   DCOMP=000063
OPOIN=000063   STORT=000063   CSPJ3=000063   2ERM =000064   HXCC =000064
ER36 =000064   2ERMS=000064   5WCBU=000064   UNRST=000064   BUSTA=000064
BUALL=000064   DBHCA=000064   ER37 =000065   2QERM=000065   NEWUN=000065
X2DBR=000065   SOUR1=000065   SOURX=000065   ER38 =000066   2ISIZ=000066
2XIBU=000066   SELUN=000066   X2DLS=000066   EXTDS=000066   SOUR2=000066
ER39 =000067   2OSIZ=000067   COMCO=000067   BUSIS=000067   X2DSR=000067
RFIEL=000067   ER40 =000070   XTDBR=000070   2CMND=000070   BUFAD=000070
X2DSL=000070   PBBYT=000070   VDCHS=000070   SBUFR=000070   NOCOP=000070
ER41 =000071   2DESC=000071   X2DUI=000071   FSPME=000071   SPUME=000071
ER42 =000072   2EESC=000072   X2DUO=000072   TABLE=000072   FNAME=000072
NAME1=000072   ER43 =000073   2SMAX=000073   DWONO=000073   X2DPI=000073
ER44 =000074   2SBYT=000074   INHBT=000074   X2DCN=000074   LOST =000075
ER45 =000075   2RBYT=000075   X2DST=000075   ER46 =000076   2SBLZ=000076
COTAB=000076   X2DDF=000076   USTAR=000076   ER47 =000077   2STBC=000077
COFLA=000077   X2DPC=000077   UREST=000077   2RT  =000100   ER48 =000100
RETRN=000100   X2DER=000100   IDEVN=000100   URPRO=000100   FZERO=000100
2SET =000101   ER49 =000101   AKMCH=000101   X2DPS=000101   DASAX=000101
DDASA=000101   ER50 =000102   2ABSE=000102   MXCHN=000102   X2DLA=000102
HMAXB=000102   DMAXB=000102   ER51 =000103   2INTV=000103   IDADR=000103
X2DLI=000103   NOBYT=000103   MAXBY=000103   HENTF=000103   ER52 =000104
POTES=000104   2HOLD=000104   X2DCI=000104   NOCHR=000104   UNAFL=000104
ER53 =000105   2ABOR=000105   X2DC1=000105   FLPT3=000105   ISTR =000106
ER54 =000106   2CONC=000106   DFDCW=000106   X2DC2=000106   FIXCL=000106
ER55 =000107   2DSCN=000107   X2DC3=000107   CNTXL=000107   ER56 =000110
2PRIO=000110   X2DC4=000110   HBRST=000110   CAMID=000110   ER57 =000111
2UPDA=000111   X2DC5=000111   5BRST=000111   9TIM0=000111   ER58 =000112
2CLAD=000112   TDISI=000112   X2DC6=000112   HECHS=000112   8CLCN=000112
ER59 =000113   2CLOC=000113   X2DMP=000113   X2DMM=000113   5ECHS=000113
ER60 =000114   2TUSE=000114   X2DRL=000114   5FYLL=000114   2FIX =000115
ER61 =000115   X2DSP=000115   MLFLA=000115   9TIM1=000115   ER62 =000116
2UNFI=000116   X2DOC=000116   ER63 =000117   2RFIL=000117   X2DCC=000117
9TIM2=000117   MAXCH=000117   ER64 =000120   LV10B=000120   2WFIL=000120
X2DBC=000120   9TIM3=000120   FSPOO=000120   ER65 =000121   2WAIT=000121
X2DD0=000121   X2FXX=000121   9TIM4=000121   ER66 =000122   SIZE =000122
2RESR=000122   X2DD2=000122   9TIM5=000122   ER67 =000123   2RELE=000123
X2DD5=000123   9TIM6=000123   ER68 =000124   2PRSR=000124   X2D06=000124
9CLO0=000124   2PRES=000124   ER69 =000125   2PRLS=000125   X2D10=000125
9CLO1=000125   2PREL=000125   ER70 =000126   2DSET=000126   DFDCR=000126
X2D11=000126   9CLO2=000126   ER72 =000127   2DABS=000127   X2D12=000127
9CLO3=000127   ER73 =000130   LV11B=000130   2DINT=000130   X2D13=000130
9CLO4=000130   ER74 =000131   2ABST=000131   X2DL3=000131   9CLO5=000131
ER75 =000132   2MCAL=000132   9CLO6=000132   ACL7 =000133   ER76 =000133
2MEXI=000133   ATIM1=000133   ATIME=000133   ER77 =000134   2RTEX=000134
2RTXT=000134   ATIM2=000134   ER78 =000135   2RTWT=000135   MTIM1=000135
MTIME=000135   ER79 =000136   2RTON=000136   MTIM2=000136   ER80 =000137
2RTOF=000137   IMASK=000137   ER81 =000140   LV12B=000140   2WHER=000140
HBUFA=000140   ABUFA=000140   NOPGS=000140   ER82 =000141   2IOSE=000141
LBUFA=000141   ENDCO=000141   TODF =000142   ER83 =000142   2ERRM=000142
5HENT=000142   USEGA=000142   ER84 =000143   2RSIO=000143   SPL12=000143
SPFLA=000143   ADRMA=000143   ER85 =000144   2MAGT=000144   UZERO=000144
2ACM =000145   ER86 =000145   5MESS=000145   BUFAS=000145   ER87 =000146
XALTS=000146   ER88 =000147   2CAMA=000147   XSGFN=000147   2GL  =000150
ER89 =000150   LV13B=000150   5CBUF=000150   USRTB=000150   ER90 =000151
2GRTD=000151   CACHL=000151   SPJNA=000151   SPJN1=000151   ER91 =000152
2GRTN=000152   ECORM=000152   SPJN2=000152   ER92 =000153   2IOXN=000153
XTMRT=000153   SPJN3=000153   ER93 =000154   2ASSI=000154   ESGTA=000154
ER94 =000155   2PLOT=000155   MRTLA=000155   ER95 =000156   2TRAC=000156
XERDE=000156   ER96 =000157   2ENTS=000157   ENDPA=000157   ER97 =000160
```

```
2FIXC=000160   PPRTD=000160   ER98 =000161   2INST=000161   XMSGU=000161
SPMES=000161   ER99 =000162   2OUTS=000162   2WRQI=000163   ER100=000163
2WSEG=000164   2WSBC=000164   2WSGB=000164   ER101=000164   2DIW =000165
ER102=000165   2DOLW=000166   USLOG=000166   ER103=000166   2REEN=000167
FREEC=000167   ER104=000167   ER105=000170   PRINC=000171   ER106=000171
FORCE=000172   ER107=000172   LCACH=000173   ER108=000173   UCACH=000174
ER109=000174   MBSYM=000175   ER110=000175   CPSTA=000176   ER111=000176
X21EN=000177   CORMB=000177   DCORM=000177   ER112=000177   2XMSG=000200
ER113=000200   BUFMA=000201   ER114=000201   LRESP=000202   ER115=000202
BUSIZ=000203   CSTAR=000203   ER116=000203   ALTMA=000204   ER117=000204
MONNO=000205   ER118=000205   9ERRP=000206   ER119=000206   ER120=000207
ER121=000210   SQELS=000210   ER122=000211   ER123=000212   ABPRO=000213
ER124=000213   RTTER=000214   ER125=000214   FLRTT=000215   ER126=000215
ACCFL=000216   ER127=000216   RTACC=000217   ER128=000217   2DOPE=000220
NBRTP=000220   ER129=000220   NBSRT=000221   ER130=000221   ICCRT=000222
ER131=000222   PACTA=000223   ER132=000223   PIOAC=000224   ER133=000224
PGNFL=000225   ER134=000225   CSGST=000226   ER135=000226   LGCOL=000227
ER136=000227   PN500=000230   ER137=000230   PDDEB=000231   ER138=000231
BUFBA=000232   ER139=000232   ECBKF=000233   ER140=000233   VDDFL=000234
ER141=000234   SPACC=000235   ER142=000235   BYPIN=000236   ER143=000236
2SFAC=000237   FPOFP=000237   ER144=000237   LPOFP=000240   ER145=000240
7EPOF=000241   ER146=000241   CURMA=000242   ER147=000242   TDFPA=000243
ER148=000243   TDFBA=000244   ER149=000244   LAMBA=000245   DLAMD=000245
ER150=000245   LAMDT=000246   ER151=000246   LAMAC=000247   ER152=000247
GNLAM=000250   ER153=000250   GNLPR=000251   ER154=000251   SIZF =000252
ER155=000252   X21PP=000253   ER156=000253   ER157=000254   PIMON=000255
ER158=000255   7RTDL=000256   ER159=000256   BUFER=000257   ER160=000257
SREBB=000260   DSREB=000260   ER161=000260   2SYCN=000261   SREBA=000261
ER162=000261   XZRTT=000262   ER163=000262   EXSEC=000263   ER164=000263
ARTFP=000264   ER165=000264   ARTLP=000265   ER166=000265   RWPOF=000266
ER167=000266   RRPOO=000267   ER168=000267   SSPOO=000270   ER169=000270
REMUS=000271   ER170=000271   ER171=000272   ER172=000273   LUSER=000274
ER173=000274   ER174=000275   ER175=000276   OPFTA=000277   ER176=000277
ER177=000300   ER178=000301   PSYSN=000302   ER179=000302   ER180=000303
ER181=000304   2SIBA=000305   PRJPA=000305   ER182=000305   ER183=000306
ER184=000307   MAXOP=000310   ER185=000310   INIFL=000311   ER186=000311
CRTRF=000312   ER187=000312   CNVRT=000313   ER188=000313   TER00=000314
ER189=000314   2LAMU=000315   TER01=000315   ER190=000315   2SLRM=000316
TER02=000316   ER191=000316   ER192=000317   ER193=000320   ER194=000321
ER195=000322   ER196=000323   ER197=000324   ER198=000325   ER199=000326
ER200=000327   ER201=000330   ER202=000331   ER203=000332   ER204=000333
ER205=000334   ER206=000335   ER207=000336   ER208=000337   FIXPA=000340
ER209=000340   SSPLE=000340   SWPFL=000341   ER210=000341   CPULO=000342
ER211=000342   NMATP=000343   ER212=000343   ERM  =000344   SEGMA=000344
FSABC=000344   SEGMB=000345   SEGMC=000346   NSEGA=000347   FNABC=000347
NSEGB=000350   NSEGC=000351   TSEGA=000352   TSEGB=000353   TSEGC=000354
TSEGS=000355   SEGPR=000355   PNUMB=000356   CPNUM=000357   NUMBE=000360
WIND1=000361   SEGRE=000362   SAVEF=000363   SRTRE=000364   SGCNU=000365
MAXP =000366   FIXMA=000367   MASSN=000370   7CPCO=000372   VDCRE=000372
7ERRS=000373   7WHO =000374   VDCSE=000374   MASSU=000374   7POLL=000375
7REJE=000376   7EOP =000377   MSTOR=000400   ABLPA=000400   NUSER=000400
NOBJE=000400   TSBAN=000404   TSSIZ=000405   5CLOA=000501   5RTLS=000503
QBSEM=000527   5BADS=000600   5BDSE=000601   K5NOS=001000   REC80=001000
5DMSE=001201   5LOGS=001202   5HSEM=001203   5MLSE=001205   FIACC=001777
LV10 =002000   FREQU=002000   K1024=002000   CSSLO=002034   CORST=002044
UEBUF=002044   LUDV =002100   HUDV =002117   OEBUF=002131   MRUBU=002216
LRUBU=002217   GMBCH=002220   MAXBU=002221   G3IBU=002236   G3BUF=002246
G5BUF=002256   RDVDF=002400   RDVLO=002500   RDVHI=002600   R3IBU=002726
R3BUF=002736   R5BUF=002746   CLADB=003204   CL1DB=003213   CLRDB=003223
CLRDP=003341   RELRT=003425   RELBU=003452   GETCH=003505   PUTCH=003513
```

```
ACOPY=003517   COPYS=003521   SETBL=003610   GDIRA=003631   GNAMA=003641
FATAL=003651   LOCK =003654   UNLOC=003656   EULOC=003660   WHERE=003702
CABST=003721   BABST=003776   LV11 =004000   K5LOC=004000   MABST=004131
FDABS=004215   SINBT=004270   SOUTB=004320   SUCPB=004353   USCPB=004356
SUCPS=004361   USCPS=004364   FGET =004575   FPUT =004674   SETUP=005343
WCI  =005432   SETW =005556   RSPQE=005677   XCSGM=006036   CHSGM=006040
TSEGN=006121   CORES=006355   LV12 =010000   HXCOD=016531   LV13 =020000
PRKEY=052163   MASKE=076033   KBACT=105210   KABAC=105410   SPSTA=110000
SSPTA=110043   SSPR1=110043   SSPR2=110403   SSPR3=110743   SSPR4=111303
SSPR5=111643   SSPR6=112203   SSPR7=112543   SSPRB=113103   SSPR9=113443
MCSTA=114000   SPR10=114003   FBRES=114005   FBREL=114007   CDATF=114043
SDATF=114074   RESST=114204   CTOWQ=114244   SEMPL=114271   TDATA=114275
TSYMB=114300   TPERI=114303   USEVT=114304   POT10=114315   ACTBL=114322
RFILN=114334   SPR11=114343   RSSEG=114436   RFDEV=114437   RRTFL=114440
COLFI=114441   CLOUT=114516   MPYAT=114546   ATMUL=114560   DMULT=114601
DDMUL=114626   LOCKQ=114652   SPR12=114703   UNLCQ=114720   APPEQ=114760
WRITQ=114767   READQ=115062   INITQ=115155   FINDQ=115214   SPR13=115243
GSPQB=115264   RSPQB=115302   GDATE=115320   NMREA=115366   NMHIT=115367
NMWRI=115370   G3NWT=115371   GDEVB=115557   SPR14=115603   RBLOC=115633
RCBLO=116063   SPR15=116143   WBLOC=116311   WCBLO=116470   SPR16=116503
WTAPE=116622   COMPP=116727   SPR17=117043   SSIZE=117063   GSIZE=117204
RSSPG=117262   MOCTA=117367   SPR18=117403   OCTAL=117415   MDECI=117440
ORTDE=117517   DECIM=117527   MDDEC=117607   ORTDD=117674   DDECI=117704
MTWOD=117740   SPR19=117743   TWODE=117772   OUTRC=120016   OUTST=120100
SRTOU=120107   MDATE=120260   SPR20=120303   LDATE=120353   LACCW=120405
INSTR=120531   STRNG=120533   APPST=120640   SPR21=120643   COMPS=120671
BDUMP=121031   DUMP =121040   CHANG=121111   SPR22=121203   PIKEY=121314
SEPST=121345   SPR23=121543   SEPPA=121546   SEPFS=121730   SPR24=122103
SEPUS=122206   CLPAR=122323   FERRO=122347   CMMON=122364   DLPAR=122443
SPR25=122443   SPR26=123003   SPR27=123343   GNAMR=123372   SPACI=123450
GDIRI=123552   SPR28=123703   GNAMI=123706   GMAIN=124023   WDIRE=124066
GNEXM=124163   SPR29=124243   GDDRT=124257   GDIRT=124262   PDDRT=124265
PDIRT=124270   GDNMT=124356   GNAMT=124361   PDNMT=124364   PNAMT=124367
FBFBU=124461   RBFBL=124514   WBFBL=124603   SPR30=124603   WBFBU=124670
ALPAG=124732   XRLPA=124735   RLPAG=124740   TPAGF=125121   SPR31=125143
RSPAG=125214   TESTB=125442   CURR =125443   RESAR=125443   TESTP=125444
SENDS=125503   GP5IX=125522   RINDX=125524   FINDX=126130   WINDX=126516
TUSSY=127064   TUSRT=127116   TUSEN=127154   RUSPW=127177   FUSEB=127223
RUSER=127226   WUSER=127370   RUSEB=127676   GUSEI=127715   GMFKN=130115
GDFKN=130324   GMUSI=130506   COLUN=130720   GUSEN=131070   NGUSN=131072
CUSED=131165   GDEFD=131242   GUSAC=131365   GSYSI=131505   FOBJB=131530
ROBJE=131533   WOBJE=131711   ROBJB=132237   GOBJI=132256   VERSI=132422
TSTPR=132513   PUSTX=132534   SEPOB=132537   SPORT=132677   GFILI=133075
DIRUN=133414   GPREV=133436   GNEXV=133473   GVERS=133530   GFIAC=133672
DLSPA=134037   DLPAG=134041   R1PAG=134232   SETAL=134454   GFILN=134461
COPBF=134502   OPQUE=134573   SPFPR=134611   MDEAB=134707   DEABB=134716
UTTEX=135017   XUTTE=135023   XCRLF=135040   DO1SU=135050   SPRIN=135113
REMCH=135162   SSPRN=135177   COBJE=135213   WRBIX=135373   EPPRN=135404
PRIFF=135476   SHRIN=135510   FDRES=135513   FDREL=135515   SPQBU=135631
SPEBU=135632   HEAPR=135645   TRAPR=136076   APTXT=136213   MCCLD=136253
FORMS=136277   F1STA=136400   CNEWV=136403   CHIGV=136405   ACCSP=136444
CROBJ=136724   MDESI=136751   XRTDE=137030   MDDES=137102   DLOBJ=137123
XRTDD=137167   MTWRD=137215   XRTOU=137254   CRNEW=137363   MDATO=137424
GDATO=137535   APSTG=137603   GCFIL=137616   SPCLD=137634   FSSTA=140000
INIQ =140021   RELQ =140064   FFILE=140066   ADDD =140120   WLOCQ=140124
MOVB =140131   MOVBF=140132   LOCQ =140134   REAQ =140143   WRIQ =140146
APPQ =140152   XTAKQ=140155   TAKQ =140162   ISETP=140300   ICLEP=140301
ICLNR=140302   ICHRE=140303   UNLQ =140314   XUNLQ=140317   TBDRE=140347
MAXAD=140352   MAXNU=140353   QSEM =140354   IOSEM=140355   SAVDI=140357
FOFT =140760   SOFT =141011   COPDI=141252   BACDI=141255   RETDI=141260
```

```
TER2C=141362   OFRND=141563   FCON =141630   FOPEN=142271   SPCLO=142436
XFCLO=142446   FCLOS=142456   CHECP=142602   FCL1 =142636   FCL2 =142736
RELTU=142767   RESTU=142771   SRESD=143061   SRELD=143063   MOVNN=143100
MOVNA=143101   MOVNP=143102   MOVAN=143103   MOVAA=143104   MOVAP=143105
MOVPN=143106   MOVPA=143107   MOVPP=143110   LDATX=143300   LDXTX=143301
LDDTX=143302   LDBTX=143303   STATX=143304   STZTX=143305   STDTX=143306
CREDI=143417   CRLDI=143421   GDIRE=143475   XCOLD=143621   COLDE=143627
CRDIR=144111   GBUF =144315   GBUFS=144317   ALBIT=144646   RBUF =144665
RBYTE=144710   RMAXB=144712   SBLOP=144766   RNDIR=145057   DOPSC=145061
OPSCR=145063   REMOP=145250   ENDIR=145316   NBAVA=145710   SBLSZ=146055
RLDIR=146115   SDATA=146152   DATA =146162   SETPO=146171   ENTER=146200
LEAVE=146243   ISTCK=146267   SMAXB=146310   SBYTE=146312   SMDIR=146334
SDDIR=146336   MINBT=146345   MOUTB=146361   SPOOD=146375   CMDIR=146462
CDDIR=146464   CPFIL=146546   LIDIR=146616   DIRST=146620   INSQB=147166
GPUPI=147303   DFPAG=147347   GPCOP=147352   GPREA=147355   GPADR=147360
CLEAS=147514   DUDIR=147534   CHDIR=147640   JNALL=147754   CHNUS=150074
INSUS=150207   PIOF =150405   SEX  =150406   REX  =150407   WBACK=150407
PONN =150410   PION =150412   IOXT =150415   EXAM =150416   DEPO =150417
RMVAL=150451   RMUSF=150564   GPAGE=150603   CRUSE=150673   RNUSE=151272
REBUF=151377   WRBUF=151501   DLUSE=151542   FREA =151675   GIUSE=152065
FWRT =152230   TAUSE=152241   LIUSE=152420   USEST=152422   FOPTB=152637
RPAGE=152714   WPAGE=152716   RDISK=153024   WDISK=153026   RFILE=153125
WFILE=153127   DUUSE=153211   CHUSE=153336   FDREA=153472   CHANP=153473
FDWRT=153474   CLPAS=153640   CRFRI=153711   DOPEN=153745   SCROP=153750
OPFIL=153753   OLDOP=153756   DLFRI=154075   BCLOS=154213   SPERM=154216
SFRIA=154216   CLOFI=154220   SMAX =154257   SETBY=154270   SETBC=154304
SBSIZ=154320   RMAX =154334   REABT=154351   LIFRI=154354   MROBJ=154377
DROBJ=154401   TESDI=154620   REGDI=154622   DWOBJ=154772   MRUSE=155347
MUIDI=155351   GUSNA=155642   GUIOI=155763   SFACC=156074   EXPFI=156077
CRALN=156102   CRALF=156104   SETTF=156365   STEFI=156374   SPEFI=156377
MRNFI=156402   MDLFI=156405   APSPF=156534   SUSCN=156606   RUSCN=156721
GPAGA=156754   FDFDI=157066   FDINA=157070   WDIEN=157232   GDIEN=157235
GNAEN=157240   RESDI=157517   RELDI=157521   OUTFN=157544   RDPAG=157553
WDPAG=157555   LSFN =157704   DPAGA=157777   COPAG=160100   RSFAC=160271
RTSFA=160273   XRTRF=160451   CHLDN=160516   RLFAC=160526   RTLFA=160530
DUPAG=160605   DELPG=160661   CHPAG=160757   MGFIL=161041   DEABF=161043
DUBIT=161104   FOBJN=161214   FOPFN=161216   CHBIT=161276   DUOBJ=161455
SDRUS=161462   INCTS=161544   DECTS=161553   CHOBJ=161561   CPUST=161634
INITF=161655   SDFIA=162053   ENFUS=162104   MENSY=162114   MRENU=162117
ENSYS=162122   ENUSE=162125   RESFI=162160   STARS=162472   RLUSE=162517
DEENT=162711   SRDUS=163010   SDRUI=163043   RDRUI=163100   STSPL=163113
CRFIL=163206   MCRFI=163211   MCRNW=163214   CRNVE=163217   RESTS=163237
ABORS=163247   STOPR=163257   STAPR=163271   LSPOQ=163601   ALFIL=163675
MALFI=163700   MALNE=163703   ALNVE=163706   PRFIN=164137   MEXFI=164361
EXFIL=164363   MAPST=164561   APPES=164563   MRENF=165064   RNFIL=165066
RMSPF=165257   DELES=165261   CHNAM=165531   MODLF=165564   DLFIL=165566
MSTRM=165725   MSTMP=165730   STERF=165733   STMPF=165736   GIVES=166004
TAKES=166052   SPOPL=166130   MSPER=166147   SPERF=166151   MINPR=166176
INPER=166200   FPERI=166311   FFILI=166363   MSFLA=166415   SFLAC=166417
MSPQE=166601   DEUFI=166656   LIFIL=166666   FILST=166670   SNSPC=167231
FWSPR=167510   BSPRI=167517   DSCON=170003   LEASP=170055   ESCMA=170076
OPENF=170105   SSFOR=170177   CONNF=170220   CLOSF=170321   LSFOR=170330
LIOPF=170360   LIRTO=170362   SSCLD=170447   SBLOS=171036   SPERO=171102
SBYTP=171130   SBLOC=171175   RELFI=171241   WHEFI=171373   TLBUF=172000
OPRTF=172033   XFPEN=172202   CORTF=172235   XFCNN=172375   CLRTF=172431
OPENS=172524   COPFI=172640   FSCLD=173021   CFLUN=177622   FDIST=177623
SFCOM=177623   FRTRY=177624   DOORL=177624   DINCR=177625   FLMQU=177626
FLMFA=177627   FRTPR=177630   STEPR=177630   WDCNT=177631   FCNT1=177632
FCNT2=177633   OPFLG=177634   PRECP=177634   DATAF=177635   SECT =177636
SCTIB=177637   OLDTR=177640   DTRCK=177640   NEWTR=177643   CCBWO=177644
```

```
FCOMF=177644   BFDEV=177645   MSCIB=177646   FMEMH=177646   FMEMD=177646
FMEML=177647   OPWCH=177650   SCTTR=177651   WCOUN=177651   FSTA1=177652
FSTA2=177653   CALIB=177654   LASMH=177654   LAMAD=177654   PVERA=177654
LASML=177655   MREMW=177656   REMWO=177656   FDRIV=177657   LREMW=177657
FRETU=177660   FDIMO=177661   FDIFO=177662   LFADD=177666   SVLWK=177706
WDSCT=177706   OSVWK=177707   NWLBB=177710   NWLBA=177711   OCMD1=177712
OCMD2=177713   OSVBA=177714   OSVBB=177715   OSVCO=177716   OSVWC=177717
SPACO=177720   ADMIA=177720   MTLRG=177720   SPAFL=177721   ADMIS=177721
CMTRE=177721   CORCU=177722   ALTPO=177722   MRECC=177722   SLONG=177723
BADDR=177723   NMTRE=177723   ECCFL=177724   BADIP=177724   CMWCN=177724
CPAT1=177725   BADTR=177725   CTACN=177725   CDISP=177726   WANKN=177726
ADNST=177726   NFDIA=177726   CPAT2=177727   NOSEC=177727   TYPEC=177730
RTZFL=177730   SVLBB=177731   SVLBA=177732   SEEKF=177732   SHSTA=177732
TINFO=177733   ERRC1=177733   PECH7=177734   ERRC2=177734   SHEAD=177734
SSEC =177735   SRTRY=177735   SWTRY=177736   SSTEP=177736   CERRC=177736
TRTZ =177737   SMARG=177737   MAXUN=177737   SVLCO=177740   MACOU=177740
ACCBU=177740   SCADR=177741   LCYLI=177741   MRETU=177741   DMRET=177741
MARGC=177742   DERRC=177742   MWRIN=177742   BUSFL=177743   MWSTA=177743
PBRK7=177744   SVLCA=177744   MLOAD=177744   NOWFL=177744   PNOTF=177744
PNI  =177745   CLRG =177745   SVLWC=177745   CFLRG=177745   TRG  =177746
PE1  =177746   TADRG=177746   ARG  =177747   PNOME=177747   DRG  =177750
NOWH =177750   PFULL=177750   XRG  =177751   NOWL =177751   PNOTY=177751
CTRG =177752   CTADR=177752   PILSL=177752   CARG =177753   CADRG=177753
X21T2=177753   PSLBS=177753   CDRG =177754   PILF =177754   IN5MS=177754
ANACS=177754   CXRG =177755   RSIST=177755   AFRET=177755   ESCBU=177755
BRECH=177756   ERCNT=177756   ATTNI=177756   FCRCE=177756   TMPBU=177756
VDBFR=177756   ROUSP=177757   SERRB=177757   REMTI=177757   BADTY=177757
NCBRK=177760   WERRB=177760   TRGIN=177760   USTAT=177760   RSELR=177760
VDBUF=177760   CTTYP=177761   AERRB=177761   INDAT=177761   GPXTR=177761
TBUSA=177761   CESCP=177762   TACNS=177762   UCLIN=177762   GPDZI=177762
RLTSA=177762   VDMTT=177762   BRKMA=177763   TACOU=177763   ATINT=177763
GPUZI=177763   RLLSA=177763   VDRTP=177763   TSPEE=177764   COMFL=177764
MWCNT=177764   SOFTA=177764   GPRUN=177764   ANTOR=177764   NOBUF=177764
VDFUN=177764   BLSZ =177765   XSAC =177765   CNTRE=177765   GPBFL=177765
TMLRE=177765   FBSIZ=177765   VDMAD=177765   SIAD =177766   DFLAG=177766
TRNSF=177766   BUSY =177767   ECHOT=177767   HXERF=177767   PNOTR=177767
VDBLC=177767   SOAD =177770   BRKTA=177770   SCREE=177770   FINIS=177770
EFUNC=177770   X21TB=177770   PNOTP=177770   LAST =177771   EMPTF=177771
ERROR=177771   CCTRL=177771   EMAXS=177771   VDNBL=177771   KTMSU=177772
TMSUB=177772   DIVIS=177772   CMDAT=177772   EDEVN=177772   TMR  =177773
5TMR =177773   CONTW=177773   IXSAC=177773   MSIZE=177773   EMSGS=177773
TTMR =177774   XHDEV=177774   OCTRL=177774   ANTME=177774   SCPRI=177774
ENBUF=177774   X21T1=177774   HDEV =177775   BYTS =177775   NRDYF=177775
DCNRT=177775   XTEMI=177775   PORTN=177775   RPORT=177775   EEMTY=177775
X2DHD=177775   X21T3=177775   X21T4=177775   X21T5=177775   X21T6=177775
E2   =177776   INCR =177776   STDRI=177776   NRDTR=177776   XTEMO=177776
DBCOU=177776   PARTN=177776   X21T0=177776   X21T7=177776   X21TA=177776
                                                            % :: Filesystem alphabetic Symbol List
```

```
SINTRAN III/VSX  VERSION I                16.17.32    23 SEPT  1984

XXXXX  XX  XX     XXXXX  XXXXX  XX  XX  XXXXX  XXXXXX  XXXXX  XX   XX
XXXXX  XX  XX     XXXXX  XXXXX  XX  XX  XXXXX  XXXXXX  XXXXX  XX   XX
XX     XX  XX     XX     XX     XX  XX  XX       XX     XX     XXX XXX
XXXX   XX  XX     XXXX   XXXXX  XX  XX  XXXXX    XX     XXXX   XX X XX
XXXX   XX  XX     XXXX    XXXX   XXXX   XXXX     XX     XXXX   XX   XX
XX     XX  XX     XX        XX     XX     XX     XX     XX     XX   XX
XX     XX  XXXXX  XXXXX  XXXXX     XX  XXXXX     XX     XXXXX  XX   XX
XX     XX  XXXXX  XXXXX  XXXXX     XX  XXXXX     XX     XXXXX  XX   XX
SINTRAN III/VSX  VERSION I                16.17.32    23 SEPT  1984
```

SYMBOL LIST IN ALPHABETIC ORDER .

```
,DA  =000001   ,PRIN=000013   ,SA  =000002   ,SP  =000024   ,ST  =000007
,WRIT=000001

1OBIT=000014   1DILB=000012   1DTU1=000012   1DTU2=000013   1TUSE=000012
1USER=000015   1XNDA=000035   1XNWD=000037

2ABOR=000105   2ABSE=000102   2ABST=000131   2ACM =000145   2ALTF=000034
2ALTN=000033   2ASSI=000154   2BRKM=000004   2CAMA=000147   2CIBU=000013
2CLAD=000112   2CLOC=000113   2CLOS=000043   2CLSE=000043   2CMND=000070
2COBU=000014   2CONC=000106   2DABS=000127   2DESC=000071   2DIBA=000014
2DILB=000013   2DINT=000130   2DIW =000165   2DOLW=000166   2DOPE=000220
2DSCN=000107   2DSET=000126   2ECHO=000003   2EESC=000072   2ENTS=000157
2ERM =000064   2ERMS=000064   2ERRM=000142   2EXIO=000031   2FIX =000115
2FIXC=000160   2GETR=000030   2GL  =000150   2GRTD=000151   2GRTN=000152
2HOLD=000104   2INBT=000001   2INST=000161   2INTV=000103   2IOSE=000141
2IOUT=000035   2IOXN=000153   2ISIZ=000066   2LAMU=000315   2LAST=000026
2MAGT=000144   2MCAL=000132   2MEXI=000133   2MSG =000032   2MTER=000052
2NOPE=000050   2NOWA=000036   2OSIZ=000067   2OUTB=000002   2OUTS=000162
2PLOT=000155   2PREL=000125   2PRES=000124   2PRIO=000110   2PRLS=000125
2PRSR=000124   2QERM=000065   2RBYT=000075   2REEN=000167   2RELE=000123
2RESR=000122   2RFIL=000117   2RMAX=000062   2RPAG=000007   2RSIO=000143
2RT  =000100   2RTDS=000027   2RTEX=000134   2RTOF=000137   2RTON=000136
2RTWT=000135   2RTXT=000134   2SBLZ=000076   2SBYT=000074   2SET =000101
2SETO=000012   2SFAC=000237   2SIBA=000305   2SLRM=000316   2SMAX=000073
2STBC=000077   2SYCN=000261   2TIME=000011   2TRAC=000156   2TUSE=000114
2UNFI=000116   2UPDA=000111   2WAIT=000121   2WFIL=000120   2WHER=000140
2WPAG=000010   2WRQI=000163   2WSBC=000164   2WSEG=000164   2WSGB=000164
2XIBU=000066   2XMSG=000200   2XNDA=000036   2XNWD=000040

3DOUA=000015   3DOUB=000016   3FLOP=000011   3FLTI=000013   3FRES=000014
3ILLF=000017   3STRE=000012

55NSG=000021   55SDS=000020   5ABJO=000004   5ABS =000013   5ACCS=000032
```

```
5ALEC=000006   5BACK=000017   5BAD =000004   5BADM=000036   5BADS=000600
5BCOM=000001   5BDSE=000601   5BESC=000003   5BFUL=000017   5BITM=000010
5BLOC=000017   5BPAS=000000   5BREA=000001   5BREG=000030   5BRES=000013
5BRKF=000007   5BRST=000111   5BUSE=000002   5BWLI=000002   5CAPI=000013
5CBUF=000150   5CFIL=000007   5CIMS=000002   5CLDV=000002   5CLOA=000501
5COM =000013   5CONC=000015   5CONT=000004   5CRDL=000011   5CTRL=000005
5DEMA=000001   5DMSE=001201   5DTU1=000014   5DTU2=000015   5ECHO=000000
5ECHS=000113   5ERRO=000004   5ERRS=000005   5ESC2=000002   5ESCF=000010
5ESCL=000010   5ESCO=000001   5FFGP=000014   5FILS=000006   5FIMO=000004
5FIU2=000026   5FIUS=000022   5FIX =000002   5FLOP=000010   5FRER=000015
5FYLL=000114   5HDMA=000007   5HDUP=000003   5HENT=000142   5HSEM=001203
5IBDV=000006   5IESC=000015   5IFS2=000012   5INHB=000003   5INNE=000016
5INT =000014   5INVR=000007   5IOBT=000017   5IPT3=000017   5IRTS=000013
5ISET=000014   5ISMS=000015   5ISPS=000042   5LBLO=000005   5LBRK=000016
5LCHA=000006   5LOC2=000013   5LOGI=000011   5LOGO=000003   5LOGR=000011
5LOGS=001202   5LSTA=000017   5MACD=000007   5MESS=000145   5MLGI=000015
5MLSE=001205   5MRST=000030   5MT  =000011   5N100=000014   5ND5E=000010
5NNET=000027   5NORE=000003   5NOSL=000011   5OK  =000000   5OP2S=000023
5OPSE=000003   5OXON=000012   5PAER=000016   5PGU =000013   5PHOE=000007
5PRVT=000010   5PT3S=000041   5RCFI=000010   5RDEV=000017   5REG =000007
5REMO=000005   5REMP=000011   5REP =000016   5RERU=000006   5RFIL=000016
5RQI =000005   5RRUS=000040   5RT2S=000037   5RTFI=000010   5RTLS=000503
5RTOF=000016   5RTSG=000004   5RTSI=000017   5RWAI=000015   5SEGS=000005
5SEXI=000015   5SPEC=000002   5SPRF=000006   5SPRO=000005   5SPSI=000016
5SREE=000006   5SWWA=000015   5SYSE=000004   5TDUM=000000   5TERM=000005
5THIG=000002   5TLOW=000001   5TLRE=000000   5TMOU=000012   5TMR =177773
5TREA=000003   5TUSE=000014   5USES=000007   5WAIT=000017   5WCBU=000064
5WCON=000005   5WECH=000006   5WESC=000011   5WIP =000014   5WLIN=000010
5WLOC=000012   5WORK=000060   5WPM =000017   5WRQI=000006   5XDEV=000011
5XOFF=000010   5XON =000007   5XRTD=000020   5XSG1=000033   5XSG2=000034

6TTYN=000023   6XOFF=000014

7BADT=000034   7BDAT=000001   7BMMX=000004   7CERS=000041   7CESC=000016
7CONF=000005   7CORQ=000006   7CORS=000007   7CPCO=000372   7CUEL=000000
7CUTY=000012   7DATA=000000   7DBRE=000001   7DCON=000011   7DECO=000002
7DESC=000017   7DUMM=000030   7ECKM=000003   7EDRS=000051   7EOP =000377
7EPOF=000241   7ERRS=000373   7ESCA=000010   7ESRS=000040   7FBSI=000025
7IAM =000050   7IRQI=000004   7ISRQ=000042   7ISRS=000043   7KEYI=000033
7LUN =000013   7NCML=000000   7NOWT=000044   7NPAS=000004   7NWRE=000046
7OPSV=000037   7PASS=000021   7POLL=000375   7RECO=000027   7REJE=000376
7RESE=000026   7RFI =000002   7RLOC=000047   7RQIN=000003   7RTDL=000256
7SATY=000015   7STRQ=000031   7STRS=000032   7SYCN=000023   7SYSI=000012
7TMOD=000014   7TNOW=000045   7TREP=000052   7TTYP=000015   7USCN=000024
7USID=000020   7WHO =000374   7ZFPA=000003   7ZMEM=000001   7ZSCR=000002

8CLCN=000112

9CL00=000124   9CL01=000125   9CL02=000126   9CL03=000127   9CL04=000130
9CL05=000131   9CL06=000132   9ERRP=000206   9IVAL=000010   9OUVA=000011
9TIM0=000111   9TIM1=000115   9TIM2=000117   9TIM3=000120   9TIM4=000121
9TIM5=000122   9TIM6=000123

AACSE=000021   ABLPA=000400   ABORS=163247   ABPRO=000213   ABUFA=000140
ACCBU=177740   ACCFL=000216   ACCRL=000036   ACCSP=136444   ACL7 =000133
ACOPY=003517   ACQBH=000047   ACQFP=000045   ACQHP=000046   ACQU =000054
ACTBL=114322   ACTOU=000030   ACTPR=000012   ACTSE=000011   ADDD =140120
ADMIA=177720   ADMIS=177721   ADNST=177726   ADREG=000002   ADRMA=000143
```

```
ADVAN=000010   AERRB=177761   AFRET=177755   AKMCH=000101   ALBIT=144646
ALEV =000002   ALEVB=000010   ALEVL=000001   ALFIL=163675   ALMSZ=000002
ALNVE=163706   ALOGN=000001   ALPAG=124732   ALTFO=000007   ALTMA=000204
ALTPO=177722   ANACS=177754   ANCHA=000011   ANI  =000001   ANTME=177774
ANTOR=177764   APPEQ=114760   APPES=164563   APPQ =140152   APPST=120640
APSPF=156534   APSTG=137603   APTXT=136213   AREG =000002   ARG  =177747
ARSEG=000020   ARTFP=000264   ARTLP=000265   ATIM1=000133   ATIM2=000134
ATIME=000133   ATINT=177763   ATMUL=114560   ATTNI=177756

BABST=003776   BACDI=141255   BACKG=000004   BACKS=000015   BAD  =000005
BADAN=000042   BADDR=177723   BADIP=177724   BADTA=000021   BADTR=177725
BADTY=177757   BBASP=000010   BBID =000001   BBLEN=000023   BBPRO=000000
BBUFF=000003   BBYTC=000002   BCFLA=000017   BCHAI=000004   BCHIS=000027
BCHNU=000020   BCHOS=000024   BCLOS=154213   BDATA=000000   BDBIT=000011
BDIRI=000000   BDUMP=121031   BECHO=000000   BENA =000000   BEXQU=000013
BFDEV=177645   BFSSP=000011   BGFPA=000040   BGLPA=000041   BHEAD=000005
BHOLD=000016   BIFIL=000021   BIHDL=000016   BIML =000001   BISYM=000015
BITA =000002   BITC =000003   BITD =000004   BITER=000002   BITM1=000011
BITM2=000012   BITM3=000013   BITM4=000014   BITM5=000015   BITM6=000016
BITM7=000017   BITMA=000010   BITN =000005   BITR =000000   BITW =000001
BITX =000004   BLEV =000020   BLEVB=000040   BLEVL=000004   BLOCN=000002
BLST =000030   BLSZ =177765   BMBYT=000003   BMEMO=000000   BNDC =000002
BNFIQ=000012   BOTTE=000001   BOXNO=000007   BPAGL=000001   BPART=000001
BPCFI=000001   BPLOG=000006   BPRCL=000062   BPRFL=000002   BPRTM=000000
BPRTS=000016   BPSOK=000003   BPWAR=000002   BRECH=177756   BREG =000005
BREGB=000030   BRESL=000013   BRKMA=177763   BRKTA=177770   BROAF=000001
BSAVE=000002   BSEG =000003   BSEGL=000014   BSEND=000024   BSINI=000014
BSPRI=167517   BSTAT=000022   BTBIN=000005   BTIMQ=000012   BTYPR=000060
BUALL=000064   BUDIS=000004   BUFAD=000070   BUFAS=000145   BUFBA=000232
BUFER=000257   BUFFA=000054   BUFFE=000011   BUFFI=000014   BUFHA=000021
BUFMA=000201   BUFST=000014   BUFUL=000007   BUSFL=177743   BUSIS=000067
BUSIZ=000203   BUSTA=000064   BUSY =177767   BWLIN=000002   BXARG=000001
BXBIT=000006   BXBRG=000002   BXLRG=000003   BXOTT=000003   BXTAD=000027
BXTRG=000000   BYPIN=000236   BYTEN=000020   BYTS =177775

CABST=003721   CACHL=000151   CADRG=177753   CALIB=177654   CAMID=000110
CARG =177753   CBPTE=000001   CBUAD=000026   CCBWO=177644   CCFPA=000044
CCLPA=000045   CCLR =000010   CCTRL=177771   CDATF=114043   CDDIR=146464
CDFIL=000012   CDISP=177726   CDRG =177754   CERRC=177736   CESCP=177762
CFLRG=177745   CFLUN=177622   CFREE=000020   CHANG=121111   CHANP=153473
CHARI=000027   CHBIT=161275   CHDIR=147640   CHECP=142602   CHIGV=136405
CHKO =000007   CHLDN=160516   CHNAM=165531   CHNST=000020   CHNUS=150074
CHOBJ=161561   CHPAG=160757   CHSGM=006040   CHUSE=153336   CL1DB=003213
CL7  =000025   CLADB=003204   CLEAS=147514   CLEDE=000020   CLOFI=154220
CLOGD=000022   CLOSF=170321   CLOUT=114516   CLPAR=122323   CLPAS=153640
CLRDB=003223   CLRDP=003341   CLRG =177745   CLRTF=172431   CMAD1=000016
CMAD2=000017   CMADR=000016   CMAND=000002   CMDAT=177772   CMDIR=146462
CMMON=122364   CMTRE=177721   CMWCN=177724   CNAME=000000   CNEWV=136403
CNOCH=000030   CNTRE=177765   CNTXL=000107   CNVRT=000313   COBJE=135213
COFLA=000077   COLDE=143627   COLFI=114441   COLUN=130720   COMCO=000067
COMFL=177764   COMPA=000003   COMPP=116727   COMPS=120671   CONNF=170220
CONTW=177773   COPAG=160100   COPBF=134502   COPDI=141252   COPFI=172640
COPYS=003521   CORAD=000027   CORCU=177722   CORES=006355   CORMB=000177
CORMS=000026   CORST=002044   CORTF=172235   COTAB=000076   CPAG1=000005
CPAG2=000006   CPAGE=000005   CPARA=000001   CPAT1=177725   CPAT2=177727
CPFIL=146546   CPNUM=000357   CPROT=000003   CPSTA=000176   CPULO=000342
CPUNR=000032   CPUST=161634   CRALF=156104   CRALN=156102   CRDIR=144111
CREDI=143417   CRFIL=163206   CRFRI=153711   CRLDI=143421   CRNEW=137363
CRNVE=163217   CROBJ=136724   CRTRF=000312   CRUSE=150673   CSGST=000226
CSPFN=000002   CSPJ1=000061   CSPJ2=000062   CSPJ3=000063   CSSLO=002034
```

```
CSTAR=000203   CTACN=177725   CTADR=177752   CTBIT=000017   CTOWQ=114244
CTRCH=000041   CTRG =177752   CTTYP=177761   CUDBU=000017   CUIBU=000033
CUMSI=000012   CURBU=000017   CURID=000051   CURMA=000242   CURME=000047
CURPR=000010   CURR =125443   CUSED=131165   CUUBU=000034   CXFNA=000002
CXNOC=000000   CXRG =177755   CXSPJ=000061   CXSPM=000001

D0    =000020  D1    =000021  D2    =000022  D3    =000023  D4    =000024
D5    =000025  D90   =000020  DAC1C=000006  DAC1U=000007  DACT  =000004
DALCD=000044   DALCM=000042   DALFU=000041   DALOG=000010   DAREG=000003
DASAX=000101   DATA =146162   DATAF=177635   DBACT=000041   DBADR=000025
DBCOU=177776   DBHCA=000064   DBLEN=000025   DBLO1=000013   DBLO2=000014
DBLOA=000013   DBLOC=000012   DBLST=000030   DBPAG=000010   DBPRE=000040
DBPRO=000024   DBREG=000007   DBUFB=000010   DCE19=000007   DCNRT=177775
DCOMP=000063   DCONT=000003   DCORM=000177   DDASA=000101   DDAUF=000014
DDBLS=000030   DDD0 =000033   DDD1 =000034   DDD2 =000035   DDD3 =000036
DDD4 =000037   DDECI=117704   DDIBA=000013   DDILF=000023   DDMUL=114626
DDR  =000000   DDREG=000004   DDW  =000001   DEABB=134716   DEABF=161043
DECIM=117527   DECTS=161553   DEDFA=000027   DEENT=162711   DELES=165261
DELPG=160661   DEMAN=000005   DENTE=000017   DEPO =150417   DERO =000030
DER2 =000032   DERRC=177742   DERRO=000013   DEUFI=166656   DFDCR=000126
DFDCW=000106   DFDEV=000023   DFIAC=000051   DFLAG=177766   DFOPP=000012
DFPAG=147347   DFPNT=000022   DI1LA=000033   DI2LA=000034   DIECO=000040
DIERC=000000   DIERO=000010   DIERR=000030   DIERT=000020   DIESP=000044
DIEWC=000004   DIEWO=000014   DIEWR=000034   DIEWT=000024   DIFTC=000026
DIL1C=000000   DIL1D=000015   DIL1F=000031   DIL1L=000020   DIL1U=000001
DIL2D=000016   DIL2F=000032   DIL2L=000021   DILBA=000013   DILBO=000015
DILBP=000012   DILCO=000014   DILDA=000015   DILDE=000011   DILFA=000031
DILFL=000023   DILFU=000024   DILGF=000022   DILGL=000033   DILLA=000020
DILLI=000002   DILNS=000017   DILRA=000003   DILSM=000005   DILST=000010
DILWA=000004   DINCR=177625   DIOUV=000010   DIRFL=000000   DIRST=146620
DIRUN=133414   DISPN=000010   DIUET=000054   DIUEX=000050   DIVIS=177772
DKFUN=000015   DKNA1=000000   DKNA2=000001   DLALO=000027   DLAMD=000245
DLAUN=000030   DLDRI=000026   DLFIL=165566   DLFRI=154075   DLLOG=000025
DLOBJ=137123   DLOCK=000003   DLPAG=134041   DLPAR=122443   DLREG=000005
DLSPA=134037   DLUSE=151542   DMAIN=000016   DMAXB=000102   DMRET=177741
DMULT=114601   DNACO=000063   DNAME=000005   DNUMB=000004   DO1SU=135050
DOLDP=000016   DOORL=177624   DOPEN=153745   DOPSC=145061   DOU0 =000000
DOU2 =000002   DOU4 =000004   DOU6 =000006   DPAGA=157777   DPAGL=000000
DPAGP=000030   DPGPR=000020   DPIN =000001   DPNT0=000016   DPNT1=000017
DPNT2=000020   DPNT3=000021   DPREG=000000   DRBIT=000016   DREAD=000060
DREG =000003   DRESE=000004   DRFUN=000026   DRG  =177750   DROBJ=154401
DROPS=000015   DRT  =000033   DS0  =000000   DS1  =000002   DS2  =000004
DS3  =000006   DS4  =000010   DS5  =000012   DSCON=170003   DSKTY=000002
DSREB=000260   DSREG=000006   DST  =000002   DTAPE=000015   DTIM1=000002
DTIM2=000003   DTIME=000002   DTIN1=000004   DTIN2=000005   DTINT=000004
DTLEN=000027   DTRCK=177640   DTREG=000002   DTRIG=000000   DTUSE=000013
DUBIM=000006   DUBIT=161104   DUBM2=000010   DUDIR=147534   DUMP =121040
DUNIT=000001   DUOBJ=161455   DUPAG=160605   DUUSE=153211   DVNAM=000000
DWOBJ=154772   DWONO=000073   DWRTE=000061   DXNDA=000035   DXNWD=000037
DXREG=000001   DY0  =000000   DY1  =000002   DY12 =000026   DY2  =000004
DY3  =000006   DY4  =000010   DY5  =000012   DY6  =000014   DY7  =000016
DY8  =000020

E2   =177776   ECBKF=000233   ECCBT=000012   ECCFL=177724   ECCR =000015
ECHOT=177767   ECL7 =000034   ECODS=000014   ECORM=000152   EDEVN=177772
EEMTY=177775   EFUNC=177770   EMAXS=177771   EMPTF=177771   EMPTY=000000
EMSGS=177773   ENBUF=177774   ENDCO=000141   ENDIR=145316   ENDPA=000157
ENFUS=162104   ENSYS=162122   ENTER=146200   ENUSE=162125   EPAGP=000003
EPPRN=135404   ER0  =000020   ER1  =000021   ER10 =000033   ER100=000163
ER101=000164   ER102=000165   ER103=000166   ER104=000167   ER105=000170
```

```
ER106=000171   ER107=000172   ER108=000173   ER109=000174   ER11 =000034
ER110=000175   ER111=000176   ER112=000177   ER113=000200   ER114=000201
ER115=000202   ER116=000203   ER117=000204   ER118=000205   ER119=000206
ER12 =000035   ER120=000207   ER121=000210   ER122=000211   ER123=000212
ER124=000213   ER125=000214   ER126=000215   ER127=000216   ER128=000217
ER129=000220   ER13 =000036   ER130=000221   ER131=000222   ER132=000223
ER133=000224   ER134=000225   ER135=000226   ER136=000227   ER137=000230
ER138=000231   ER139=000232   ER14 =000037   ER140=000233   ER141=000234
ER142=000235   ER143=000236   ER144=000237   ER145=000240   ER146=000241
ER147=000242   ER148=000243   ER149=000244   ER15 =000040   ER150=000245
ER151=000246   ER152=000247   ER153=000250   ER154=000251   ER155=000252
ER156=000253   ER157=000254   ER158=000255   ER159=000256   ER16 =000041
ER160=000257   ER161=000260   ER162=000261   ER163=000262   ER164=000263
ER165=000264   ER166=000265   ER167=000266   ER168=000267   ER169=000270
ER17 =000042   ER170=000271   ER171=000272   ER172=000273   ER173=000274
ER174=000275   ER175=000276   ER176=000277   ER177=000300   ER178=000301
ER179=000302   ER18 =000043   ER180=000303   ER181=000304   ER182=000305
ER183=000306   ER184=000307   ER185=000310   ER186=000311   ER187=000312
ER188=000313   ER189=000314   ER19 =000044   ER190=000315   ER191=000316
ER192=000317   ER193=000320   ER194=000321   ER195=000322   ER196=000323
ER197=000324   ER198=000325   ER199=000326   ER2  =000023   ER20 =000022
ER200=000327   ER201=000330   ER202=000331   ER203=000332   ER204=000333
ER205=000334   ER206=000335   ER207=000336   ER208=000337   ER209=000340
ER21 =000045   ER210=000341   ER211=000342   ER212=000343   ER22 =000046
ER23 =000047   ER24 =000050   ER25 =000051   ER26 =000052   ER27 =000053
ER28 =000054   ER29 =000055   ER3  =000024   ER30 =000056   ER31 =000057
ER32 =000060   ER33 =000061   ER34 =000062   ER35 =000063   ER36 =000064
ER37 =000065   ER38 =000066   ER39 =000067   ER4  =000025   ER40 =000070
ER41 =000071   ER42 =000072   ER43 =000073   ER44 =000074   ER45 =000075
ER46 =000076   ER47 =000077   ER48 =000100   ER49 =000101   ER5  =000026
ER50 =000102   ER51 =000103   ER52 =000104   ER53 =000105   ER54 =000106
ER55 =000107   ER56 =000110   ER57 =000111   ER58 =000112   ER59 =000113
ER6  =000027   ER60 =000114   ER61 =000115   ER62 =000116   ER63 =000117
ER64 =000120   ER65 =000121   ER66 =000122   ER67 =000123   ER68 =000124
ER69 =000125   ER7  =000030   ER70 =000126   ER71 =000003   ER72 =000127
ER73 =000130   ER74 =000131   ER75 =000132   ER76 =000133   ER77 =000134
ER78 =000135   ER79 =000136   ER8  =000031   ER80 =000137   ER81 =000140
ER82 =000141   ER83 =000142   ER84 =000143   ER85 =000144   ER86 =000145
ER87 =000146   ER88 =000147   ER89 =000150   ER9  =000032   ER90 =000151
ER91 =000152   ER92 =000153   ER93 =000154   ER94 =000155   ER95 =000156
ER96 =000157   ER97 =000160   ER98 =000161   ER99 =000162   ERASE=000014
ERCNT=177756   ERM  =000344   ERRC1=177733   ERRC2=177734   ERROR=177771
ERRSE=000005   ESCBU=177755   ESCMA=170076   ESGTA=000154   EULOC=003660
EUSAD=000043   EXAM =150416   EXFIL=164363   EXPFI=156077   EXSEC=000263
EXTDS=000066

FISTA=136400   FATAL=003651   FBFBU=124461   FBREL=114007   FBRES=114005
FBSIZ=177765   FCL1 =142636   FCL2 =142736   FCLOS=142456   FCNT1=177632
FCNT2=177633   FCOMF=177644   FCON =141630   FCRCE=177756   FCST =000042
FDABS=004215   FDFDI=157066   FDIFO=177662   FDIMO=177661   FDINA=157070
FDIST=177623   FDREA=153472   FDREL=135515   FDRES=135513   FDRIV=177657
FDWRT=153474   FERRO=122347   FFILE=140066   FFILI=166363   FFREL=000001
FGET =004575   FIACC=001777   FILEN=000014   FILST=166670   FINDQ=115214
FINDX=126130   FINIS=177770   FIXCL=000106   FIXED=000002   FIXID=000052
FIXMA=000367   FIXPA=000340   FKICK=000002   FLAG =000004   FLAGB=000042
FLMFA=177627   FLMQU=177626   FLOAD=000004   FLPT3=000105   FLRTT=000215
FMEMD=177646   FMEMH=177646   FMEML=177647   FMXRE=000002   FNABC=000347
FNAME=000072   FOBJB=131530   FOBJN=161214   FOFT =140760   FOPEN=142271
FOPFN=161216   FOPTB=152637   FORCE=000172   FORM =000003   FORMS=136277
FORSP=000016   FPAR1=000055   FPAR2=000056   FPAR3=000057   FPERI=166311
```

```
FPOFP=000237   FPUT =004674   FRACC=000001   FREA =151675   FRECV=000001
FREEC=000167   FREQU=002000   FRES =000000   FRETR=000030   FRETU=177660
FRSG1=000024   FRSG2=000025   FRTPR=177630   FRTRY=177624   FSABC=000344
FSCLD=173021   FSEG =000006   FSEND=000000   FSPME=000071   FSPOO=000120
FSSTA=140000   FSTA1=177652   FSTA2=177653   FSTAR=000006   FSTOP=000007
FULL =000006   FULLN=000002   FULLS=000003   FUNLO=000005   FUSEB=127223
FWKIC=000003   FWRT =152230   FWSPR=167510   FYLLE=000021   FZERO=000100

G3BUF=002246   G3IBU=002236   G3NWT=115371   G5BUF=002256   GBUF =144315
GBUFS=144317   GCFIL=137616   GDATE=115320   GDATO=137535   GDDRT=124257
GDEFD=131242   GDEVB=115557   GDFKN=130324   GDIEN=157235   GDIRA=003631
GDIRE=143475   GDIRI=123552   GDIRT=124262   GDNMT=124356   GENDA=000055
GETCH=003505   GFIAC=133672   GFILI=133075   GFILN=134461   GIUSE=152065
GIVES=166004   GMAIN=124023   GMBCH=002220   GMFKN=130115   GMUSI=130506
GNAEN=157240   GNAMA=003641   GNAMI=123706   GNAMR=123372   GNAMT=124361
GNEXM=124163   GNEXV=133473   GNLAM=000250   GNLPR=000251   GOBJI=132256
GOOD =000004   GP5IX=125522   GPADR=147360   GPAGA=156754   GPAGE=150603
GPBFL=177765   GPCOP=147352   GPDZI=177762   GPREA=147355   GPREV=133436
GPRUN=177764   GPUPI=147303   GPUZI=177763   GPXTR=177761   GRI  =000002
GSI  =000000   GSIZE=117204   GSO  =000011   GSPQB=115264   GSYSI=131505
GUIOI=155763   GUSAC=131365   GUSEI=127715   GUSEN=131070   GUSNA=155642
GVERS=133530

HBRST=000110   HBUFA=000140   HDEV =177775   HEADE=000011   HEAPR=135645
HECHS=000112   HENTE=000017   HENTF=000103   HINIF=000040   HMAXB=000102
HPEK =000016   HSTAT=000010   HTABL=000021   HUDV =002117   HWINF=000047
HXCC =000064   HXCOD=016531   HXERF=177767

IBLOA=000023   IBYTS=000050   ICCRT=000222   ICHRE=140303   ICLEP=140301
ICLNR=140302   ICORA=000022   IDADR=000103   IDBAD=000025   IDBST=000021
IDEVN=000100   IDLE =000016   IDNTS=000062   IFUNC=000025   IGNAC=000010
IICOR=000053   IIFUN=000052   IIMAX=000054   IINI =000013   ILSAV=000060
IMASK=000137   IMAXB=000024   IMAXW=000024   IMBPR=000004   IMCBP=000003
IMTFL=000056   IN5MS=177754   INCR =177776   INCTS=161544   INDA1=000031
INDA2=000053   INDAT=177761   INDB1=000014   INDB2=000013   INDX =000016
INDX1=000027   INDX2=000051   INFLG=000016   INHBT=000074   INIAD=000021
INIFL=000311   INIQ =140021   INITF=161655   INITQ=115155   INPER=166200
INSQB=147166   INSTR=120531   INSUS=150207   IOLDP=000055   IOLOG=000020
IOSEM=140355   IOTRA=000007   IOXT =150415   IQUEU=000061   IRETW=000026
IRI  =000003   ISETP=140300   ISI  =000005   ISO  =000017   ISSRE=000051
ISTAT=000004   ISTCK=146267   ISTR =000106   ITOM =000001   IXSAC=177773

JNALL=147754

K1024=002000   K5ESC=000004   K5LOC=004000   K5NOS=001000   KABAC=105410
KBACT=105210   KICKP=000021   KPROS=000011   KSETD=000011   KTMSU=177772

LACCW=120405   LAMAC=000247   LAMAD=177654   LAMBA=000245   LAMCN=000000
LAMDT=000246   LAMLP=000001   LAMNP=000001   LAMPP=000000   LAMPR=000002
LASBU=000020   LASMH=177654   LASML=177655   LAST =177771   LASTP=000015
LBUFA=000141   LCACH=000173   LCURB=000036   LCYLI=177741   LDATE=120353
LDATX=143300   LDBTX=143303   LDDTX=143302   LDILR=000050   LDITR=000046
LDIXR=000047   LDOLR=000031   LDOTR=000027   LDOXR=000030   LDTSZ=000003
LDXTX=143301   LEASP=170055   LEAVE=146243   LFADD=177666   LFCOU=000034
LGCOL=000227   LIBEG=000014   LIDIR=146616   LIFIL=166666   LIFRI=154354
LINR =000062   LIOPF=170360   LIPOI=000030   LIRTO=170362   LIUSE=152420
LMCN =000000   LMDAT=000016   LMLP =000010   LMNP =000010   LMPP =000000
```

```
LMPR =000020    LNKSP=000061    LNUMB=000007    LOADI=000003    LOCK =003654
LOCKQ=114652    LOCQ =140134    LOGAD=000002    LOST =000075    LPOFP=000240
LREG =000004    LREMW=177657    LRESP=000202    LRSA =000021    LRUBU=002217
LSFN =157704    LSFOR=170330    LSPOQ=163601    LUDV =002100    LUNIT=000002
LUSAD=000044    LUSER=000274    LV10 =002000    LV10B=000120    LV11 =004000
LV11B=000130    LV12 =010000    LV12B=000140    LV13 =020000    LV13B=000150
LWBIT=000006    LWPHY=000054

M144B=000012    M2UNT=000020    MABST=004131    MACOU=177740    MADR =000003
MAILF=000050    MAINF=000010    MALFI=163700    MALNE=163703    MAPST=164561
MARGC=177742    MASKE=076033    MASSN=000370    MASSU=000374    MASTA=000014
MASTB=000041    MAX  =000015    MAXAD=140352    MAXBH=000023    MAXBU=002221
MAXBY=000103    MAXCH=000117    MAXCY=000003    MAXNU=140353    MAXOP=000310
MAXP =000366    MAXUN=177737    MBLEN=000020    MBOXH=000031    MBREA=000000
MBSYM=000175    MC5SP=000001    MCCLD=136253    MCLRG=000034    MCMAG=000000
MCONT=000001    MCRFI=163211    MCRNW=163214    MCSTA=114000    MCURB=000035
MDATE=120260    MDATO=137424    MDCUR=000035    MDDEC=117607    MDDES=137102
MDEAB=134707    MDECI=117440    MDESI=136751    MDLFI=156405    MEMA1=000014
MEMA2=000015    MEMAD=000014    MENSY=162114    MESSF=000000    MESSI=000056
MEXFI=164361    MFUNC=000006    MGFIL=161041    MILRG=000045    MINBH=000022
MINBT=146345    MINFR=000013    MINPR=166176    MIOIN=000004    MISTA=000031
MITRG=000044    MLEV =000010    MLEVB=000030    MLEVL=000003    MLFLA=000115
MLIAD=000035    MLICA=000032    MLICP=000031    MLIMS=000036    MLIMX=000033
MLINK=000005    MLIPA=000040    MLIST=000045    MLITE=000030    MLIWO=000034
MLOAD=177744    MMEMO=000002    MMESI=000004    MOCTA=117367    MODLF=165564
MOLRG=000026    MONNO=000205    MOPEN=000003    MOTRG=000025    MOUTB=146361
MOVAA=143104    MOVAN=143103    MOVAP=143105    MOVB =140131    MOVBF=140132
MOVNA=143101    MOVNN=143100    MOVNP=143102    MOVPA=143107    MOVPN=143106
MOVPP=143110    MPIOC=000005    MPYAT=114546    MQUEU=000011    MRECC=177722
MREMW=177656    MRENF=165064    MRENU=162117    MRETU=177741    MRNFI=156402
MROBJ=154377    MRSTA=000030    MRTLA=000155    MRUBU=002216    MRUSE=155347
MSCIB=177646    MSFLA=166415    MSGN5=000001    MSIZE=177773    MSPER=166147
MSPQE=166601    MSSTA=000014    MSTMP=165730    MSTOR=000400    MSTRM=165725
MTBIT=000013    MTFLG=000027    MTIM1=000135    MTIM2=000136    MTIME=000135
MTLRG=177720    MTOR =000006    MTRAN=000011    MTWOD=117740    MTWRD=137215
MUIDI=155351    MWCNT=177764    MWRIN=177742    MWSTA=177743    MXCHN=000102
MXMEL=000002    MXTIM=000017

N5RDF=000021    N5STA=000002    NAME1=000072    NBAVA=145710    NBRTP=000220
NBSRT=000221    NBYTS=000060    NCBRK=177760    ND500=000017    NDPRO=000022
NEWTR=177643    NEWUN=000065    NFDIA=177726    NFLAG=000013    NGUSN=131072
NINSZ=000024    NLOCK=000015    NLPAG=000027    NMATP=000343    NMHIT=115367
NMREA=115366    NMTRE=177723    NMUSD=000026    NMUSM=000025    NMWRI=115370
NNSWS=000024    NOBDI=000050    NOBJE=000400    NOBUF=177764    NOBYT=000103
NOCHA=000027    NOCHR=000104    NOCOP=000070    NOPGS=000140    NOSEC=177727
NOSIN=000032    NOWFL=177744    NOWH =177750    NOWL =177751    NPFBU=000014
NPFUN=000001    NPHBU=000016    NPMAI=000024    NPOPC=000001    NPTIG=000000
NRDTR=177776    NRDYF=177775    NSEGA=000347    NSEGB=000350    NSEGC=000351
NTLEN=000016    NTP  =000020    NUMBE=000360    NUMIN=000033    NUSER=000400
NWLBA=177711    NWLBB=177710

O2SEG=000023    OABIT=000005    OACCE=000042    OBACK=000014    OBCOU=000044
OBFIL=000015    OBYTE=000061    OCBIT=000004    OCMD1=177712    OCMD2=177713
OCOMM=000013    OCOUN=000047    OCTAL=117415    OCTRL=177774    ODATC=000051
ODATR=000053    ODATW=000055    ODEVN=000044    ODIRI=000001    OEBUF=002131
OFACC=000005    OFBLZ=000014    OFBUF=000010    OFCB =000012    OFDIR=000015
OFFLG=000007    OFFP =000025    OFFTP=000006    OFIND=000023    OFIOD=000023
OFIP =000017    OFIP1=000017    OFIP2=000020    OFLAG=000025    OFLIB=000011
OFLOC=000011    OFNB =000013    OFNBR=000010    OFOBJ=000016    OFOP =000021
OFOP1=000021    OFOP2=000022    OFOUD=000024    OFPAG=000023    OFPDI=000011
```

```
OFRND=141563   OFSCR=000012   OFSET=000013   OFTYP=000043   OFWRT=000015
OIBIT=000003   OICOR=000034   OIFUN=000033   OIMAX=000035   OINDP=000003
OINDX=000005   OLBIT=000007   OLDOP=153756   OLDPA=000017   OLDTR=177640
OLOCK=000000   OLSAV=000061   OMBIT=000006   ON5MS=000025   ONAME=000026
ONDEX=000046   ONEXT=000040   OOLDP=000036   OOPEN=000050   OPAGE=000057
OPART=000002   OPBIT=000001   OPENF=170105   OPENS=172524   OPFIL=153753
OPFLG=177634   OPFTA=000277   OPOIN=000063   OPQUE=134573   OPREV=000041
OPRTF=172033   OPSCR=145063   OPWCH=177650   OQUEU=000062   ORESE=000015
ORTDD=117674   ORTDE=117517   OSBIT=000002   OSEG =000003   OSSRE=000032
OSVBA=177714   OSVBB=177715   OSVCO=177716   OSVWC=177717   OSVWK=177707
OTBIT=000000   OTMBI=000010   OTYPE=000036   OUSED=000017   OUSER=000045
OUTFN=157544   OUTRC=120016   OUTST=120100   OWRTE=000016

P0  =000000    P1  =000001    P2  =000002    P3  =000003    P4  =000004
P5  =000005    PACTA=000223   PAGEN=000016   PAGLI=000000   PAGPH=000002
PAGPR=000002   PART1=000005   PARTN=177776   PAVA1=000010   PAVA2=000011
PAVAI=000010   PBBYT=000070   PBRK7=177744   PBSIZ=000034   PCADR=000023
PCBNK=000022   PCNT =000004   PCORA=000022   PDDEB=000231   PDDRT=124265
PDIRT=124270   PDNMT=124364   PE1 =177746    PECH7=177734   PERMF=000017
PFMSG=000016   PFULL=177750   PGC =000014    PGNFL=000225   PIKEY=121314
PILF =177754   PILSL=177752   PIMBH=000022   PIMBL=000023   PIMON=000255
PIMPG=000027   PINIT=000012   PINO =000031   PIOAC=000224   PIOCA=000013
PIOCI=000007   PIOCN=000030   PIOCO=000011   PIOF =150405   PION =150412
PIPOW=000013   PIPRO=000021   PISTT=000026   PKICK=000016   PLEFT=000023
PLMSG=000015   PMFUN=000007   PN500=000230   PNAMT=124367   PNBOX=000015
PNEXT=000022   PNI  =177745   PNMAI=000023   PNOME=177747   PNOPC=000003
PNOTF=177744   PNOTP=177770   PNOTR=177767   PNOTY=177751   PNUMB=000356
POK  =000001   POLSY=000004   PONN =150410   POOLL=000051   POOLP=000022
PORTN=177775   POT10=114315   POTES=000104   PPREV=000023   PPROM=000002
PPRTD=000160   PRECP=177634   PREQU=000025   PRFIN=164137   PRIFF=135476
PRINC=000171   PRIO =000014   PRJPA=000305   PRKEY=052163   PRSCU=000060
PRSR =000010   PRVFB=000005   PRVFL=000004   PRVMA=000003   PRVUE=000002
PSIZ =000030   PSLBS=177753   PSYSN=000302   PTN  =000017   PTRNS=000014
PUSTX=132534   PUTCH=003513   PVEFU=000021   PWCR =000007   PXT  =000037
PYERA=177654

QBSEM=000527   QSEM =140354   QSGMT=000002

R1PAG=134232   R3BUF=002736   R3IBU=002726   R5BUF=002746   RA  =000003
RAREG=000003   RBFBL=124514   RBLOC=115633   RBUF =144665   RBYTE=144710
RCBLO=116063   RDATF=000000   RDHII=000026   RDISK=153024   RDLOO=000024
RDPAG=157553   RDREG=000004   RDRUI=163100   RDVDF=002400   RDVHI=002600
RDVLO=002500   REABT=154351   READ =000000   READQ=115062   REAQ =140143
REBUF=151377   REC80=001000   RECEI=000025   REFOR=000005   REGDI=154622
REGPO=000005   RELBU=003452   RELDI=157521   RELFI=171241   RELQ =140064
RELRT=003425   RELTU=142767   REMAI=000017   REMBY=000023   REMCH=135162
REMOP=145250   REMSI=000046   REMTI=177757   REMUS=000271   REMWO=177656
RERRC=000026   RESAR=125443   RESCY=000006   RESDI=157517   RESFI=162160
RESLI=000000   RESST=114204   RESTS=163237   RESTU=142771   RETDI=141260
RETRN=000100   REVER=000011   REVLE=000054   REWIN=000013   REWUN=000017
REX  =150407   RFDEV=114437   RFIEL=000067   RFILE=153125   RFILN=114334
RFORM=000042   RGSI =000035   RHSTA=000022   RIFIL=000026   RINDX=125524
RING1=000000   RING2=000001   RLDIR=146115   RLFAC=160526   RLLSA=177763
RLPAG=124740   RLTSA=177762   RLUSE=162517   RMAX =154334   RMAXB=144712
RMLNR=000027   RMSPF=165257   RMUSF=150564   RMVAL=150451   RNACO=000037
RNDIR=145057   RNFIL=165066   RNUSE=151272   ROBJB=132237   ROBJE=131533
ROFIL=000023   ROUSP=177757   RPAGE=152714   RPORT=177775   RPREG=000000
RPRS =000053   RRPOO=000267   RRTFL=114440   RSCUR=000060   RSEGM=000014
RSELR=177760   RSFAC=160271   RSICH=000013   RSIST=177755   RSPAG=125214
RSPNU=000031   RSPQB=115302   RSPQE=005677   RSREG=000006   RSRET=000057
```

```
RSSEG=114436   RSSPG=117262   RTACC=000217   RTBH =000024   RTBL =000025
RTBOX=000033   RTCLD=000010   RTDLG=000016   RTFPA=000042   RTIN =000022
RTLFA=160530   RTLPA=000043   RTREF=000007   RTREG=000002   RTRES=000001
RTSFA=160273   RTSTA=000024   RTTER=000214   RTUT =000043   RTZFL=177730
RUSCN=156721   RUSEB=127676   RUSER=127226   RUSPW=127177   RWFIE=000004
RWPOF=000266   RXREG=000001

S0   =000000   S1   =000001   S10  =000010   S11  =000011   S12  =000012
S13  =000013   S14  =000014   S15  =000015   S16  =000016   S17  =000017
S2   =000002   S20  =000020   S21  =000021   S3   =000003   S4   =000004
S5   =000005   S6   =000006   S7   =000007   S90  =000000   S91  =000003
SABOR=000007   SAVDI=140357   SAVEF=000363   SAVFN=000062   SBHOL=000043
SBLOC=171175   SBLOP=144766   SBLOS=171036   SBLST=000030   SBLSZ=146055
SBSIZ=154320   SBUFR=000070   SBYTE=146312   SBYTP=171130   SBYTS=000032
SCADR=177741   SCOND=000026   SCPRI=177774   SCREE=177770   SCROP=153750
SCTIB=177637   SCTTR=177651   SDATA=146152   SDATF=114074   SDDIR=146336
SDFIA=162053   SDRUI=163043   SDRUS=161462   SECSY=000002   SECT =177636
SECTO=000012   SECTP=000024   SECTR=000001   SECWO=000000   SEEKF=177732
SEGLI=000000   SEGM =000007   SEGMA=000344   SEGMB=000345   SEGMC=000346
SEGPR=000355   SEGRE=000362   SEGST=000025   SELUN=000066   SEMPL=114271
SEND =000023   SENDS=125503   SEPFS=121730   SEPOB=132537   SEPPA=121546
SEPST=121345   SEPUS=122206   SERRB=177757   SETAL=134454   SETBC=154304
SETBL=003610   SETBY=154270   SETDV=000011   SETPO=146171   SETTF=156365
SETUP=005343   SETW =005556   SEX  =150406   SFACC=156074   SFCOM=177623
SFLAC=166417   SFLOC=000001   SFORM=000040   SFRIA=154216   SGCNU=000365
SGMAX=000015   SHEAD=177734   SHRIN=135510   SHSTA=177732   SIAD =177766
SIAD1=000021   SIAD2=000022   SIB50=000020   SICCO=000020   SIDRT=000021
SINBT=004270   SINIT=000013   SINVE=000052   SIZE =000122   SIZF =000252
SLAKK=000010   SLIPO=000020   SLMAX=000010   SLONG=177723   SMARG=177737
SMAX =154257   SMAXB=146310   SMDIR=146334   SMI  =000043   SMO  =000044
SNACO=000040   SNLIN=000025   SNPAG=000024   SNSPC=167231   SOAD =177770
SOFT =141011   SOFTA=177764   SOHSC=000020   SOUR1=000065   SOUR2=000066
SOURX=000065   SOUTB=004320   SPACC=000235   SPACI=123450   SPACO=177720
SPAFL=177721   SPAGE=000031   SPCLD=137634   SPCLO=142436   SPEBU=135632
SPEFI=156377   SPERF=166151   SPERI=000002   SPERM=154216   SPERO=171102
SPFLA=000143   SPFNA=000035   SPFPR=134611   SPINX=000010   SPJN1=000151
SPJN2=000152   SPJN3=000153   SPJNA=000151   SPL12=000143   SPLEN=000012
SPMES=000161   SPMOD=000030   SPOOD=146375   SPOPL=166130   SPORT=132677
SPQBU=135631   SPR10=114003   SPR11=114343   SPR12=114703   SPR13=115243
SPR14=115603   SPR15=116143   SPR16=116503   SPR17=117043   SPR18=117403
SPR19=117743   SPR20=120303   SPR21=120643   SPR22=121203   SPR23=121543
SPR24=122103   SPR25=122443   SPR26=123003   SPR27=123343   SPR28=123703
SPR29=124243   SPR30=124603   SPR31=125143   SPRIN=135113   SPROG=000001
SPRS =000052   SPSEG=000025   SPSGM=000043   SPSTA=110000   SPUME=000071
SQELS=000210   SQERR=000036   SQIOS=000005   SQSEM=000004   SQUEU=000003
SRDUS=163010   SREBA=000261   SREBB=000260   SRELD=143063   SRESD=143061
SRTCS=000004   SRTOU=120107   SRTRE=000364   SRTRY=177735   SSCLD=170447
SSEC =177735   SSEG =000025   SSFOR=170177   SSIZE=117063   SSPLE=000340
SSPOO=000270   SSPR1=110043   SSPR2=110403   SSPR3=110743   SSPR4=111303
SSPR5=111643   SSPR6=112203   SSPR7=112543   SSPR8=113103   SSPR9=113443
SSPRN=135177   SSPTA=110043   SSREF=000031   SSTEP=177736   SSTOP=000006
STADR=000006   STAPR=163271   STARS=162472   STATU=000001   STATX=143304
STDEL=000013   STDEV=000010   STDRI=177776   STDTX=143306   STEFI=156374
STEPR=177630   STERF=165733   STMPF=165736   STOPR=163257   STORT=000063
STORX=000053   STRNG=120533   STRSE=000032   STSPL=163113   STZTX=143305
SUBIN=000017   SUCPB=004353   SUCPS=004361   SURTM=000026   SURUS=000025
SUSCN=156606   SVBPR=000000   SVLBA=177732   SVLBB=177731   SVLCA=177744
SVLCO=177740   SVLWC=177745   SVLWK=177706   SVTSL=000001   SWBUF=000053
SWICH=000015   SWPFL=000341   SWTRY=177736   SYSNO=000046
```

```
TABLE=000072   TACNS=177762   TACOU=177763   TADRE=000001   TADRG=177746
TAKES=166052   TAKQ =140162   TAUSE=152241   TBDRE=140347   TBUSA=177761
TDATA=114275   TDBTP=000017   TDFBA=000244   TDFPA=000243   TDIND=000045
TDIPD=000044   TDISI=000112   TDOND=000010   TDOPD=000043   TDOSI=000054
TDTAD=000015   TDTAF=000015   TDTAL=000016   TDXRE=000001   TER00=000314
TER01=000315   TER02=000316   TER20=141362   TERM =000026   TESDI=154620
TESTB=125442   TESTP=125444   TINFO=177733   TLBUF=172000   TLINK=000000
TMLRE=177765   TMPBU=177756   TMR  =177773   TMSUB=177772   TODF =000142
TPAGF=125121   TPERI=114303   TRAIL=000012   TRAPR=136076   TREG =000001
TRG  =177746   TRGIN=177760   TRIG =000001   TRLRE=000007   TRNSF=177766
TRTZ =177737   TSBAN=000404   TSEGA=000352   TSEGB=000353   TSEGC=000354
TSEGN=006121   TSEGS=000355   TSPEE=177764   TSPTR=000042   TSSIZ=000405
TSTAD=000045   TSTAT=000023   TSTPR=132513   TSYMB=114300   TTMCO=000007
TTMR =177774   TUSEN=127154   TUSRT=127116   TUSSY=127064   TWODE=117772
TXBBP=000000   TXBPR=000020   TXCBP=000010   TXFLL=000040   TXIBP=000040
TXICB=000030   TXSVB=000000   TXSVT=000010   TXUEF=000020   TXXSV=000060
TYPEC=177730   TYPRI=000003

UANTM=000015   UCACH=000174   UCLIN=177762   UCLOS=000007   UCOPA=000035
UCRAL=000017   UDATE=000037   UDELP=000036   UDENT=000041   UDIRI=000001
UDROB=000010   UDSCN=000031   UDWOB=000012   UEBUF=002044   UENTE=000025
UEXPF=000041   UFDFD=000034   UFOBJ=000037   UFREE=000054   UFRIE=000055
UGDIE=000033   UGUIO=000016   UGUSN=000015   UINDP=000003   UINDX=000005
UINIT=000023   ULIOP=000021   ULOCK=000000   UMAGT=000003   UMDLF=000020
UMGFI=000030   UMROB=000011   UMRUS=000014   UMUID=000013   UNAFL=000104
UNAME=000026   UNDEX=000047   UNLCQ=114720   UNLOC=003656   UNLQ =140314
UNRST=000064   UOFLG=000010   UOPEN=000006   UPART=000002   UPASS=000036
UPAVA=000043   UPUS1=000045   UPUS2=000046   UPUSE=000045   URBYP=000004
URENF=000042   UREST=000077   URFIL=000000   URLFE=000026   URPRO=000100
URSOF=000024   URTLI=000043   USBLP=000005   USBLS=000002   USBYP=000022
USCPB=004356   USCPS=004364   USDAD=000043   USDVS=000010   USEGA=000142
USEGM=000016   USEST=152422   USEVT=114304   USFIL=000017   USIDX=000015
USLOG=000166   USNXT=000053   USPEF=000027   USPER=000026   USPRV=000052
USRTB=000150   USTAR=000076   USTAT=177760   USTEM=000032   USTER=000040
UTTEX=135017   UUSED=000017   UWFIL=000001   UZERO=000144

VDABS=000020   VDBFR=177756   VDBLC=177767   VDBUF=177760   VDCDI=000040
VDCFU=000010   VDCHS=000070   VDCME=000020   VDCNR=000017   VDCNU=000060
VDCPF=000000   VDCRE=000372   VDCSE=000374   VDCST=000013   VDDFL=000234
VDDST=000014   VDFUN=177764   VDLIN=000012   VDMAD=177765   VDMB0=000015
VDMB1=000016   VDMBX=000015   VDMTT=177762   VDNBL=177771   VDPDI=000007
VDPFU=000004   VDPGL=000000   VDPGR=000001   VDPME=000005   VDPNU=000011
VDRTP=177763   VDUNI=000017   VENTX=000026   VERSI=132422

WAKEF=000063   WANKN=177726   WBACK=150407   WBFBL=124603   WBFBU=124670
WBLOC=116311   WCBLO=116470   WCI  =005432   WCOUN=177651   WDCNT=177631
WDIEN=157232   WDIRE=124066   WDISK=153026   WDPAG=157555   WDSCT=177706
WEOF =000012   WERRB=177760   WERRC=000036   WFILE=153127   WFLAG=000021
WHEFI=171373   WHERE=003702   WHSTA=000032   WIND1=000361   WIND0=000015
WINDX=126516   WLINK=000010   WLOCQ=140124   WOBJE=131711   WPAGE=152716
WRBIX=135373   WRBUF=151501   WRIQ =140146   WRITQ=114767   WSNLI=000032
WSNPA=000033   WTAPE=116622   WUSER=127370

X21BL=000007   X21C1=000000   X21C2=000001   X21C3=000002   X21C4=000003
X21EN=000177   X21KO=000054   X21LO=000000   X21NL=000000   X21PL=000053
X21PM=000011   X21PP=000253   X21SA=000000   X21SB=000002   X21SC=000001
X21SL=000057   X21ST=000052   X21SY=000026   X21T0=177776   X21T1=177774
X21T2=177753   X21T3=177775   X21T4=177775   X21T5=177775   X21T6=177775
X21T7=177776   X21TA=177776   X21TB=177770   X2D00=000121   X2D02=000122
X2D05=000123   X2D06=000124   X2D0C=000116   X2D10=000125   X2D11=000126
```

```
X2D12=000127   X2D13=000130   X2DBC=000120   X2DBR=000065   X2DC1=000105
X2DC2=000106   X2DC3=000107   X2DC4=000110   X2DC5=000111   X2DC6=000112
X2DCC=000117   X2DCI=000104   X2DCN=000074   X2DDF=000076   X2DER=000100
X2DHD=177775   X2DL3=000131   X2DLA=000102   X2DLI=000103   X2DLS=000066
X2DMM=000113   X2DMP=000113   X2DPC=000077   X2DPI=000073   X2DPS=000101
X2DRL=000114   X2DSL=000070   X2DSP=000115   X2DSR=000067   X2DST=000075
X2DUI=000071   X2DUO=000072   X2F01=000010   X2F20=000032   X2F30=000033
X2F50=000035   X2FAB=000014   X2FAC=000017   X2FBR=000020   X2FCP=000015
X2FEC=000016   X2FFF=000012   X2FIC=000003   X2FIL=000002   X2FIS=000004
X2FNC=000011   X2FNI=000007   X2FNM=000005   X2FNP=000013   X2FNR=000006
X2FOK=000000   X2FSM=000001   X2FXX=000121   X2M1 =000006   X2M1B=000007
X2M2 =000010   X2M3 =000011   X2M4 =000012   X2M5 =000013   X2MCL=000011
X2MCP=000010   X2MFA=000007   X2MFU=000005   X2MS2=000013   X2MSL=000011
X2MSS=000012   X2MST=000006   XALTS=000146   XBBNK=000042   XBEND=000044
XBSAV=000045   XBSTR=000043   XCHAI=000000   XCOLD=143621   XCRLF=135040
XCSGM=006036   XERDE=000156   XFCLO=142446   XFCNN=172375   XFPEN=172202
XHDEV=177774   XMSGU=000161   XMSGX=000000   XNDPR=000016   XNMAI=000022
XNOCH=000031   XNOWU=000063   XNTP =000006   XOFF =000023   XON  =000021
XPIPR=000012   XPKIC=000000   XPMAI=000026   XPTN =000002   XREG =000000
XRG  =177751   XRLPA=124735   XRRDR=000000   XRRSR=000002   XRRTS=000010
XRSA =000020   XRTDD=137167   XRTDE=137030   XRTOU=137254   XRTRF=160451
XRTSR=000006   XRTTS=000012   XSAC =177765   XSGFN=000147   XTADR=000000
XTAKQ=140155   XTDAR=000030   XTDBR=000070   XTDDR=000040   XTDLR=000050
XTDPR=000000   XTDSR=000060   XTDTR=000020   XTDXR=000010   XTEMI=177775
XTEMO=177776   XTMRT=000153   XTRSE=000005   XUBI1=000007   XUBI2=000010
XUBI3=000011   XUBI4=000012   XUBI5=000013   XUBI6=000014   XUBI7=000015
XUBIM=000006   XUNIT=000057   XUNLQ=140317   XURSE=000005   XUSEG=000004
XUTTE=135023   XWAIT=000055   XWCHL=000004   XWPCR=000001   XWRTC=000011
XWSAR=000003   XWTCR=000007   XWTDR=000005   XWTTC=000013   XXHOM=000051
XXSBF=000050   XXSBK=000047   XXUBF=000046   XZRTT=000262

YSVBP=000006   YSVTS=000007

ZAREG=000012   ZBREG=000016   ZDREG=000013   ZLREG=000014   ZOARG=000035
ZOPRG=000032   ZOSRG=000040   ZPREG=000007   ZSREG=000015   ZTADR=000011
```

ZTREG=000011 ZXREG=000010

% :: RT-loader numeric Symbol List

```
SINTRAN III/VSX  VERSION I                    16.18.16    23 SEPT  1984

XXXXX  XXXXXX        XX      XXXX    XXXX    XXXX    XXXXX  XXXXX
XXXXXX XXXXXX        XX     XXXXXX  XXXXXX  XXXXX   XXXXX  XXXXXX
XX  XX    XX         XX     XX  XX  XX  XX  XX  XX  XX     XX  XX
XXXXX     XX    XXXX XX     XX  XX  XX  XX  XX  XX  XXXX   XXXXX
XXXXX     XX    XXXX XX     XX  XX  XXXXXX  XX  XX  XXXX   XXXXX
XX  XX    XX         XX     XX  XX  XXXXXX  XX  XX  XX     XX  XX
XX  XX    XX         XXXXX  XXXXXX  XX  XX  XXXXX   XXXX   XX  XX
XX  XX    XX         XXXXX   XXXX   XX  XX  XXXX    XXXX   XX  XX
SINTRAN III/VSX  VERSION I                    16.18.16    23 SEPT  1984
```

SYMBOL LIST WITH VALUES IN ASCENDING ORDER.

```
NL    =000000   ER0   =000000   BOK   =000000   POD   =000000   FNL   =000000
BIML  =000000   SXS0  =000000   BSGFS=000000   DPAGL=000000   TLINK=000000
SEGLI=000000    PAGLI=000000    P1    =000001   ER1   =000001   P1D   =000001
DP1   =000001   FP1   =000001   SXS1  =000001   2INBT=000001   BDEMA=000001
BBACK=000001    STATU=000001    BPAGL=000001   ALOGN=000001   P2    =000002
ER2   =000002   P2D   =000002   DP2   =000002   ,FIX  =000002   BCOB  =000002
SXS2  =000002   2OUTB=000002    BBOOT=000002   LOGAD=000002   PAGPH=000002
P3    =000003   ER3   =000003   P3D   =000003   DP3   =000003   FP3   =000003
BREL  =000003   RERTF=000003    BPRIN=000003   INHBT=000003   SXCLC=000003
OPSEG=000003    MASSA=000003    P4    =000004   ER4   =000004   P4D   =000004
DP4   =000004   FP4   =000004   FLAG  =000004   INRTF=000004   2BRKM=000004
5RTSG=000004    BNREL=000004    SXCL1=000004   FCONT=000004   RTLSE=000004
P5    =000005   ER5   =000005   P5D   =000005   DP5   =000005   TEMP  =000005
BXLO  =000005   SEGSI=000005   SPROT=000005   SXCL2=000005   FALLO=000005
P6    =000006   ER6   =000006   P6D   =000006   DP6   =000006   BPREP=000006
SREEP=000006    BFTNL=000006    STADR=000006   DSTAD=000006   OFTYP=000006
P7    =000007   ER7   =000007   P7D   =000007   FP7   =000007   SEGM  =000007
RFELS=000007    BDECP=000007    BALPH=000007   ER8   =000010   P8D   =000010
P10   =000010   MLEV  =000010   LITES=000010   BPADR=000010   RTFSE=000010
ALEVB=000010    DALOG=000010    WLINK=000010   ER9   =000011   P11   =000011
BDREL=000011    ACTSE=000011    P12   =000012   ER10  =000012   FP12  =000012
BPR1C=000012    5TMOU=000012    ACTPR=000012   P13   =000013   ER11  =000013
5ABS  =000013   BPR2C=000013   BRESL=000013   P14   =000014   ER12  =000014
BWIP  =000014   5INT  =000014   BTLIS=000014   5N100=000014   REESE=000014
FP    =000015   P15   =000015   ER13  =000015   STDEL=000015   BLOCM=000015
6SEXI=000015    5RWAI=000015    RP    =000016   ER14  =000016   5REP  =000016
BRTOF=000016    BSCDA=000016    WP    =000017   ER15  =000017   BEGBT=000017
RTDSI=000017    BCDAT=000017    5WAIT=000017   5BACK=000017   ER16  =000020
DPGPR=000020    ER17  =000021   ENDBT=000021   ER18  =000022   ER19  =000023
EOFBT=000023    ER20  =000024   OFPAG=000024   ER21  =000025   PRIBT=000025
MLEVB=000030    DPAGP=000030    WRERT=000032   INLBT=000034   5RT2S=000037
INCBT=000040    2CLOS=000043    5CBUF=000044   DBGBT=000047   2NOPE=000050
AXCB=000054     FTMP0=000055    FTMP1=000060   FTMP2=000063   64ERR=000064
```

```
FTMP3=000066   FTMP4=000071   SAVTA=000074   TREG =000077   TAREG=000077
TADRE=000077   AREG =000100   DREG =000101   XREG =000102   SAVX =000103
LTREG=000104   9TADR=000104   LAREG=000105   2ABOR=000105   LDREG=000106
9LREG=000106   LXREG=000107   5MXCH=000107   PWMCA=000110   WMCSE=000111
PRMCA=000112   RMCSE=000113   AMOUN=000114   BATCH=000115   BEGFL=000116
BLCKI=000117   2RFIL=000117   BOOTS=000120   2WFIL=000120   BRFDE=000121
BRFTP=000122   2RESE=000122   BRTDS=000123   2RELE=000123   BTABC=000124
BUFAD=000125   BUFFR=000126   BUFS =000127   CBCLC=000130   CBCUR=000131
CBLLI=000132   2MCAL=000132   CBULI=000133   2MEXI=000133   CCSGF=000134
CDIC =000135   CHCI =000136   CHSUM=000137   CIMSE=000140   CLC  =000141
CLCO =000142   CLC1 =000143   2RSIO=000143   CLC2 =000144   CLITE=000145
CLITS=000146   CNZER=000147   COBCL=000150   COBDA=000151   COBSE=000152
COMFL=000153   COM2F=000154   COMIN=000155   COMOU=000156   CPNT =000157
CRLFC=000160   CRTFS=000161   CSGFN=000162   CSTRI=000163   CURLI=000164
2WSGB=000164   CURSL=000165   CURTP=000166   CURSU=000167   CUR1S=000170
CUR2S=000171   COCOM=000172   C1COM=000173   C2COM=000174   DICRE=000175
IDICR=000175   DICP =000176   DICPP=000177   DGPAR=000200   DLMOD=000201
ELRTF=000202   ENTDI=000203   FACCO=000204   FCURT=000205   FEIRT=000206
FREEP=000207   F100F=000210   FPNT =000211   FPNT1=000211   FPNT2=000212
GSGFL=000213   GSGFP=000214   IBLCK=000215   ILREG=000216   IGMAI=000217
INCOX=000220   INITZ=000221   INPFP=000222   IPTAB=000223   KEYDE=000224
KGPFL=000225   LCOBC=000226   LEIRT=000227   LESGT=000230   LFILT=000231
LFIXC=000232   LIBNO=000233   LONGF=000234   LOWPR=000235   LRTDS=000236
LRTFP=000237   LRTLP=000240   LSEGT=000241   LTBP =000242   LTBPP=000243
LTKO =000244   LTK1 =000245   LUSEG=000246   LUSFA=000247   MAXPA=000250
MK   =000251   MXPRW=000252   NBYT1=000253   NCOST=000254   BVAL =000255
NELRT=000255   NEWPA=000256   NOBS =000257   NOBSW=000260   NULL =000261
OLDCH=000262   OPNT =000263   OSTRI=000264   OUPFP=000265   OUTDE=000266
OUTFI=000267   PACCE=000270   PARDE=000271   PARNO=000272   PB   =000273
PCIMS=000274   PNOBS=000275   PPB  =000276   PRIAD=000277   PR1CO=000300
PR2CO=000301   PSY1 =000302   DPSY1=000302   FPSYM=000302   PSY2 =000303
DPSY2=000303   PSY3 =000304   PWORK=000305   RTFBU=000306   SCRFI=000307
SEGBL=000310   SEGLL=000311   SEGML=000312   SEGNP=000313   SEGN1=000314
SEGN2=000315   SEGUL=000316   SEGOL=000317   SEGOU=000320   SEG1L=000321
SEG1T=000322   SEG1U=000323   SEG2L=000324   SEG2T=000325   SEG2U=000326
SERRF=000327   SGFIL=000330   SG1AD=000331   SKIPF=000332   SPACE=000333
SRING=000334   STACK=000335   STPNT=000336   SYMBP=000337   SY1  =000340
FSYM =000340   DFSY1=000340   SY2  =000341   DFSY2=000341   SY3  =000342
S1LOG=000343   S1LON=000344   S1MAD=000345   S2LOG=000346   S2LON=000347
S2MAD=000350   TGARB=000351   UCOBC=000352   WAREG=000353   WBRTF=000354
WXREG=000355   W1   =000356   DW1  =000356   FW1  =000356   W2   =000357
DW2  =000357   W3   =000360   DW3  =000360   5CHCO=000361   9ERW1=000362
9ERW2=000363   LINK1=000364   LINK2=000365   LINK3=000366   ESCLR=000367
ADREF=000370   ADSUM=000371   BACKS=000372   BTR  =000373   CMMEX=000374
CREAD=000375   CRLF =000376   CWRIT=000377   DGET =000400   DGETY=000401
DELTB=000402   ENTER=000403   FERRF=000404   FFSEG=000405   FNSYM=000406
FNSYT=000407   FRTDE=000410   FSYMB=000411   FSYTY=000412   GETEL=000413
GPAR =000414   ILLCH=000415   ILLFI=000416   ILLPA=000417   KGPAR=000420
LEAVE=000421   LEAVX=000422   LEGSE=000423   LENTE=000424   LERRF=000425
LLEAV=000426   LMON =000427   LTBCH=000430   LXLEA=000431   OREAD=000432
OWRIT=000433   PCHAR=000434   PNSPA=000435   RCB  =000436   READ =000437
READ1=000440   RFILN=000441   SCAB =000442   SCLC =000443   SEGFT=000444
SEGTT=000445   SERRO=000446   SFREE=000447   SILLC=000450   SRCHI=000451
ENTTR=000452   ENTTA=000452   ENTAR=000453   ENTDR=000454   ENTXR=000455
ENTLR=000456   ENTBR=000457   COMST=000460   LSTAS=000500   5NASE=000517
RWORK=000524   XWORK=000570   PARDG=000634   SEGFP=000641   RIOPA=000646
PRPAR=000653   RTFRE=000660   SIZF =000664   SYTYP=000670   MASSN=000700
SEGTB=000704   RTFIL=000711   RTFBL=000712   NWRTF=000713   PARRT=000714
SGFTA=000721   XSGF0=000731   XSGF1=000771   XSGF2=001031   XSGF3=001071
BUFLI=001131   ERRTX=001401   FRTFT=001427   FFTNF=001442   F1BAN=001454
```

```
SRTFI=001471   SSTEN=001503   SSTLE=001506   NOIOB=001511   IRTDS=001512
TAXN =001513   COBSY=001570   ECOBS=001615   FIDDD=001615   BMSCR=001616
CBOOT=001626   CBOO2=001633   MASK2=001647   RE16 =001650   IOXIN=001660
AQR1 =001670   EBOOT=001671   COMTA=001672   DSTAC=002000   CN0  =002250
CN1  =002303   CN2  =002343   CN3  =002405   CN4  =002424   CN5  =002515
CN6  =002532   CN7  =002575   CN7A =002634   CN8  =002707   CN8A =002745
CN10 =002774   CN11 =003026   CN12 =003050   CN13 =003063   CN13A=003104
CN13B=003133   CN14 =003155   CN15 =003171   CN16 =003226   CN17 =003233
CN17A=003241   CN18 =003255   CN18A=003276   CN19 =003312   CN19A=003364
CN19B=003404   CN20 =003431   CN20A=003453   CN20B=003471   CN21 =003513
CN22 =003545   CN22B=003601   CN23 =003656   CN24 =003725   CN25 =003756
CN26 =004021   CN27 =004052   CN27A=004071   CN28 =004121   CN28A=004145
CN29 =004200   CN30 =004241   CN30A=004255   CN31 =004273   CN32 =004320
CN32A=004347   CN32F=004354   CN33 =004371   CN34 =004400   CN33A=004410
CN34A=004427   CN35C=004457   CN38C=004507   CN35 =004526   CN35A=004532
CN35B=004555   CN35F=004573   CN36 =004607   CN37 =004636   CN35D=004645
CN38 =004670   CN38B=004711   CN39 =004727   CN40 =004751   CN40A=004774
CN40B=005021   CN40C=005036   CN41 =005047   CN42 =005072   CN43 =005112
CN44 =005135   CN45 =005155   CN46 =005176   CN47 =005223   CN48 =005251
CN49 =005271   CN50 =005310   CN51 =005345   CN52 =005363   REGBL=005374
FATLT=005404   REGNA=005424   FATXR=005431   DELIM=005432   DESCR=005436
ALPN =005445   ERRTY=005452   TXERR=005513   XERR0=005532   XERR1=005546
XERR2=005561   XERR3=005575   XERR4=005610   XERR5=005620   XERR6=005672
XERR7=005677   XERR8=005707   TNREE=005715   TTEND=006174   TTEOF=006202
TTTEX=006210   TPUNT=006217   YES  =006223   NO   =006225   TYESN=006227
LVERS=006236   TXDEF=006262   TNOTP=006315   TCOC1=006331   TLCHE=006354
TIMC1=006373   TIMC2=006416   XTLAD=006450   XTUAD=006454   XTCLC=006461
XTLDA=006466   XTUDA=006477   TLO1 =006511   TLO2 =006525   TLO11=006541
TLO22=006561   TXSIZ=006601   TNSBO=006606   TWARN=006625   TCRSE=006652
TERME=006677   RSMES=006715   AMBIG=006733   TXILC=006744   TXCOM=006754
ETX0 =006762   TILCO=007001   WROPN=007022   WROPA=007031   ETX1 =007041
ETX3 =007054   ETX4 =007074   ETX5 =007107   ETX11=007123   ETX12=007135
ETX13=007151   ETX14=007162   ETX15=007201   ETX16=007215   ETX17=007226
ETX6 =007261   ETX19=007274   ETX18=007315   ETX10=007325   ETX9 =007360
TNREF=007372   PMESS=007406   RPLME=007417   PSEGN=007426   PNPIT=007434
PAPIT=007450   RINGP=007466   SEGTP=007473   PWIPA=007504   TPROT=007511
TAREA=007523   NSMES=007550   PINPF=007560   PBRFT=007570   PSYMP=007572
TFL0 =007575   TFL1 =007605   TFL2 =007617   TFL3 =007632   TFL4 =007646
TFL5 =007662   TFL6 =007675   TFL7 =007710   SEGCO=007722   CORCO=007733
POPFP=007741   IMERR=007751   RTFUL=007763   TCIMA=007771   BINER=007777
BITBF=010005   TSNUS=010020   LOSEG=010032   LISEG=010043   SCIUS=010055
TILLF=010071   TSNO =010103   TLADR=010107   TUADR=010113   TMADR=010117
TDMAN=010123   TNDEM=010127   TCAEX=010135   TCLAB=010155   TPSGF=010166
TSFNU=010201   T2SFN=010212   TSFNO=010221   TXXSZ=010232   TFRPA=010253
TLASE=010274   PARAD=010315   TCHCC=010323   TILLC=010344   TPRAC=010356
TNSPR=010372   TSNFO=010407   TSDEF=010421   TSNAM=010436   QUEST=010446
TPRTP=010452   PSGNA=010462   TRTAD=010473   TLABE=010512   ILLSE=010515
TCCUS=010532   CLABN=010545   TPTAB=010562   SGF1E=010574   TINIZ=010615
TILAD=010644   CHAEX=010655   PLADR=010674   PUADR=010705   TDCCO=010716
ILBOU=010730   TPASI=010745   TCLCO=010754   TRTFI=010766   TXADR=010771
TXCLA=010776   TXCAR=011006   TPPRI=011015   TP1SE=011024   TP2SE=011034
TPCOS=011044   TPSTA=011057   TPRTD=011070   TSGEX=011106   TRLCC=011130
TRYEX=011143   NRTDA=011161   TSREF=011200   PSTKL=011216   NCCOM=011227
PSYSE=011241   TOSNA=011255   TNSNA=011265   TLFNA=011275   PIMFI=011310
PRTDS=011320   PBOOT=011334   PBINF=011352   PTDAT=011362   SNCON=011365
TSEMP=011410   SIZCC=011417   TFPCC=011444   TNPCC=011471   TXSEG=011516
TFIXE=011525   TPROS=011540   TSPRO=011552   TREES=011557   TYSEG=011564
TSCER=011571   ILLST=011600   ILLPT=011610   TBPUN=011625   NSCRF=011630
PSYVA=011656   PCILA=011667   PCIL1=011724   TSECH=011747   ETX21=011752
TNSPA=012006   TDSSE=012026   TCART=012050   TRTDE=012070   TXCOB=012105
```

```
TNSW1=012124   TNSW2=012147   TNRTF=012202   TNOIO=012220   TNCAW=012235
TNOTI=012270   RTFNN=012302   TXSSN=012336   TISNA=012346   TSGNA=012362
TNSGS=012402   TAPRF=012425   TASGF=012447   SNNDE=012472   TSNDC=012510
TWSE1=012524   TWSE2=012563   TSYER=012622   DISPL=012645   PCLAD=012661
TXSYA=012676   TPROG=012715   TONLY=012720   TPRPT=012756   TILPT=012776
BITBU=013014   BITBE=017014   IOBUF=017014   STBEG=021014   LSTAC=021014
STEND=021514   SCRBU=021514   BSSGT=021514   ESSGT=021724   BSYPT=021724
ESYPT=031204   RTLOA=110003   RRTLO=110012   RRRTL=110021   RRRRT=110030
RRPRD=110037   RRPRL=110046   RSSWA=110067   XSWAP=110074   SWAPO=110101
RTPAC=110413   ACTCH=110437   REORG=110464   7ENTR=111120   7LEAX=111150
7LEAV=111151   7LENT=111207   7LXLE=111212   7LLEA=111213   RBRFB=111220
7RBRF=111220   FINBT=111233   IOINI=111260   7READ=111332   7REA1=111343
SYMB =111360   7ADSU=111401   7FSYM=111404   7FSYT=111524   7FNST=111530
7FNSY=111535   TOLTB=111542   7DELT=111572   7ADRE=111650   DAREF=111707
DXREF=112015   DACHR=112133   7BTR =112237   7GETE=112302   TREFE=112347
ZEROE=112375   7LTBC=112422   7SFRE=112440   GARBT=112455   SUBGA=112654
GETTA=112735   7FRTD=113067   RL1  =113115   RTDCH=113202   WTEMP=113222
RTEMP=113246   SOVER=113265   7SOVE=113265   7RCB =113470   5RCBE=113530
RTPB =113537   LF   =113546   LR   =113552   LRP  =113556   LRD  =113571
LC   =113604   AFR  =113700   ARR  =113712   SFL  =113725   AFL  =113764
SRL  =113775   LNF  =114017   5RUT =114033   INHB =114043   LSM  =114046
ASF  =114051   ADS  =114325   LIBR =114422   LIB  =114442   MAIN =114462
REF  =114665   ENTR =114714   PMO  =114776   DMO  =115104   DIC  =115113
BEG  =115316   END  =115364   EOF  =115520   INC  =116000   CXC  =116056
INL  =116076   CXL  =116134   DWL  =116141   DWC  =116163   DBG  =116223
CHBIO=116237   7SCLC=116274   7DGET=116321   FETCB=116370   EMTYB=116445
7DGEY=116467   TDGET=116476   TDGEY=116536   CSRFI=116563   CSWFI=116602
SDLAD=116627   CHCOB=116631   NCOBS=116703   MESS =117037   7MESS=117037
ERRME=117060   7ERRM=117060   7CRLF=117065   SYMW =117073   DECOD=117152
SHSYL=117155   THISS=117177   RSAPR=117263   PWORD=117274   7PWOR=117274
7PCHA=117304   7PNSP=117332   DOCTU=117342   OCTUT=117362   7OCTU=117362
OCT1 =117406   OCTO =117427   7OCTO=117427   OCTOX=117440   NOCTU=117444
DECUT=117476   7DECU=117476   7LEGS=117551   7SC1 =117560   7FFSE=117602
IDGBU=117634   RESET=117662   7RESE=117662   BRSET=120126   7RFIL=120155
RSFIL=120241   RBRFI=120252   7SEGT=120272   7SEGF=120274   SSEGM=120476
RSEGM=120541   7RSEG=120541   ISMAP=120545   XSSEG=120555   XRSEG=120562
MAPR =120577   RESPA=120673   7RESP=120673   RRESP=120777   RERES=121003
PRRES=121015   SSEGT=121022   RSEGT=121136   7RSTA=121136   SXSEG=121161
CWIMA=121204   SRTD =121226   RRTD =121403   WIMAG=121515   RIMAG=121517
7RIMA=121517   FOPEN=121630   WBCIM=121747   WSEGB=121750   INITR=121755
RIRTF=122002   RSRTF=122036   TOLFR=122063   WRTFI=122125   WSRTF=122233
WARTF=122266   DELRT=122303   DSRTF=122403   FSYRT=122434   RRTEL=122540
RTFAC=122557   CLRTF=122624   FLTOR=122643   TORTF=122730   GRTFI=122747
ABLOO=122770   ABL1 =123057   GCOM =123165   INFO =123216   NOYES=123241
TCI  =123255   XTCI =123303   ZEROP=123332   DPUN2=123355   DPUN1=123362
DPUN0=123366   7KGPA=123376   7GPAR=123402   CR   =123424   MDIGI=123511
MSGNA=123746   3CREA=124141   3OREA=124146   3CWRI=124154   3OWRI=124161
3BACK=124200   3SCAB=124205   3SRHI=124222   ONESC=124247   OFFES=124256
OPENF=124267   7LOAD=124323   7LMON=124461   IOERR=124564   7IOER=124564
7SERR=124570   7FATA=124664   SCRFE=124727   SGF2E=124737   SGFER=124752
7FERR=124770   7SILL=125003   7ILLF=125010   7ILCH=125015   7ILLP=125022
PARIL=125037   7PARI=125037   INBDR=125047   STDGB=125113   NBSCO=125127
NSCOM=125132   XLOCO=125450   LOCOM=125454   NRLCO=125744   RELCO=125750
ASCOX=126057   RSCOM=126116   EXCOM=126121   EXRTL=126126   TABLI=126152
WLSEQ=126155   TLOCO=126503   ENCOM=126510   ENDRU=127373   COBER=127726
WSEGN=130674   WPSEG=130761   RELSC=131277   CRSCO=131302   SSCOM=131646
SCCOM=131650   NEWPC=131707   NNEWP=131711   LIRTF=131714   DUBIC=132125
SLADC=132346   CHLOC=132611   RDIGI=133016   HELPC=133113   DNREC=133172
DNREF=133174   WRNRE=133242   DEFSC=133254   DELSC=133451   WSYCO=133475
COLSN=134011   WPRCO=134013   WCLCO=134015   WRSCO=134017   WRSAC=134021
```

```
PRADR=134067   DECPR=134172   DELPR=134306   CDRTP=134327   LDELP=134364
WLACO=134405   DELCL=134573   SSFIL=134642   SPTCO=134672   RBINC=134761
RBINX=135613   BINDU=135631   ODUMP=136165   PCTCO=136500   PCRCO=136631
AAREA=137103   CHPRC=137400   MOVB =140131   LFSEG=140211   DEFPR=140237
REFSY=140423   DELRS=140450   RENAM=140543   IMAGL=140663   COMPC=141065
WHATC=141445   BACKU=141711   SC5CO=141716   RC5CO=141721   LBACO=141723
LIALP=141730   PRSCO=141736   FIXCO=142025   UNFIX=142054   DUMPR=142126
RDUMP=142230   TABDC=142304   SPELE=142461   PELEM=142520   TGPEL=142554
LFRTD=142562   SFRSY=142630   SFRS1=142670   CDALO=142725   CDASE=142750
IGMCO=142775   CLRSE=143024   MOVNA=143101   MOVAN=143103   MOVAA=143104
CHBSE=143120   SNOIO=143251   SPROF=143267   LDATX=143300   LDXTX=143301
LDDTX=143302   LDBTX=143303   RPROF=143303   STATX=143304   STZTX=143305
STDTX=143306   CODSN=143345   CODLS=143347   COSED=143506   CODCB=143516
PRDUM=143665   PRLOA=143665   RPRCO=144221   RPRFI=144615   WTSGF=144632
RPRAR=144663   SFORT=145026   RFORT=145031   DEFSY=145034   LLRTL=145236
DSEGT=160000   POULI=173777   MAXLT=176170   BURTF=176200   MCURT=177770
M6    =177772  M5    =177773  M4    =177774  M3    =177775  M2    =177776
                                                            % :: RT-loader alphabetic Symbol List
```

```
SINTRAN III/VSX  VERSION I                    16.18.37    23 SEPT  1984

XXXXX  XXXXXX        XX       XXXX     XXXX    XXXX     XXXXX  XXXXX
XXXXXX XXXXXX        XX      XXXXXX   XXXXXX   XXXXX    XXXXX  XXXXXX
XX  XX   XX          XX      XX  XX   XX  XX   XX  XX   XX     XX  XX
XXXXX    XX   XXXX   XX      XX  XX   XX  XX   XX  XX   XXXX   XXXXX
XXXXX    XX   XXXX   XX      XX  XX   XXXXXX   XX  XX   XXXX   XXXXX
XX  XX   XX          XX      XX  XX   XXXXXX   XX  XX   XX     XX  XX
XX  XX   XX          XXXXX   XXXXXX   XX  XX   XXXXX    XXXX   XX  XX
XX  XX   XX          XXXXX    XXXX    XX  XX   XXXX     XXXX   XX  XX
SINTRAN III/VSX  VERSION I                    16.18.37    23 SEPT  1984
```

SYMBOL LIST IN ALPHABETIC ORDER .

```
,FIX =000002

2ABOR=000105   2BRKM=000004   2CLOS=000043   2INBT=000001   2MCAL=000132
2MEXI=000133   2NOPE=000050   2OUTB=000002   2RELE=000123   2RESE=000122
2RFIL=000117   2RSIO=000143   2WFIL=000120   2WSGB=000164

3BACK=124200   3CREA=124141   3CWRI=124154   3OREA=124146   3OWRI=124161
3SCAB=124205   3SRHI=124222

5ABS =000013   5BACK=000017   5CBUF=000044   5CHCO=000361   5INT =000014
5MXCH=000107   5N100=000014   5NASE=000517   5RCBE=113530   5REP =000016
5RT2S=000037   5RTSG=000004   5RUT =114033   5RWAI=000015   5SEXI=000015
5TMOU=000012   5WAIT=000017

64ERR=000064

7ADRE=111650   7ADSU=111401   7BTR =112237   7CRLF=117065   7DECU=117476
7DELT=111572   7DGET=116321   7DGEY=116467   7ENTR=111120   7ERRM=117060
7FATA=124664   7FERR=124770   7FFSE=117602   7FNST=111530   7FNSY=111535
7FRTD=113067   7FSYM=111404   7FSYT=111524   7GETE=112302   7GPAR=123402
7ILCH=125015   7ILLF=125010   7ILLP=125022   7IOER=124564   7KGPA=123376
7LEAV=111151   7LEAX=111150   7LEGS=117551   7LENT=111207   7LLEA=111213
7LMON=124461   7LOAD=124323   7LTBC=112422   7LXLE=111212   7MESS=117037
7OCTO=117427   7OCTU=117362   7PARI=125037   7PCHA=117304   7PNSP=117332
7PWOR=117274   7RBRF=111220   7RCB =113470   7REA1=111343   7READ=111332
7RESE=117662   7RESP=120673   7RFIL=120155   7RIMA=121517   7RSEG=120541
7RSTA=121136   7SC1 =117560   7SCLC=116274   7SEGF=120274   7SEGT=120272
7SERR=124570   7SFRE=112440   7SILL=125003   7SOVE=113265

9ERW1=000362   9ERW2=000363   9LREG=000106   9TADR=000104

AAREA=137103   ABL1 =123057   ABLOO=122770   ACTCH=110437   ACTPR=000012
ACTSE=000011   ADREF=000370   ADS  =114325   ADSUM=000371   AFL  =113764
```

```
AFR  =113700   ALEVB=000010   ALOGN=000001   ALPN =005445   AMBIG=006733
AMOUN=000114   AQR1 =001670   AREG =000100   ARR  =113712   ASCOX=126057
ASF  =114051

BACKS=000372   BACKU=141711   BALPH=000007   BATCH=000115   BBACK=000001
BBOOT=000002   BCDAT=000017   BCOB =000002   BDECP=000007   BDEMA=000001
BDREL=000011   BEG  =115316   BEGBT=000017   BEGFL=000116   BFTNL=000006
BIML =000000   BINDU=135631   BINER=007777   BITBE=017014   BITBF=010005
BITBU=013014   BLCKI=000117   BLOCM=000015   BMSCR=001616   BNREL=000004
BOK  =000000   BOOTS=000120   BPADR=000010   BPAGL=000001   BPR1C=000012
BPR2C=000013   BPREP=000006   BPRIN=000003   BREL =000003   BRESL=000013
BRFDE=000121   BRFTP=000122   BRSET=120126   BRTDS=000123   BRTOF=000016
BSCDA=000016   BSGFS=000000   BSSGT=021514   BSYPT=021724   BTABC=000124
BTLIS=000014   BTR  =000373   BUFAD=000125   BUFFR=000126   BUFLI=001131
BUFS =000127   BURTF=176200   BVAL =000255   BWIP =000014   BXLO =000005

C0COM=000172   C1COM=000173   C2COM=000174   CBCLC=000130   CBCUR=000131
CBLLI=000132   CBOO2=001633   CBOOT=001626   CBULI=000133   CCSGF=000134
CDALO=142725   CDASE=142750   CDIC =000135   CDRTP=134327   CHAEX=010655
CHBIO=116237   CHBSE=143120   CHCI =000136   CHCOB=116631   CHLOC=132611
CHPRC=137400   CHSUM=000137   CIMSE=000140   CLABN=010545   CLC  =000141
CLC0 =000142   CLC1 =000143   CLC2 =000144   CLITE=000145   CLITS=000146
CLRSE=143024   CLRTF=122624   CMMEX=000374   CN0  =002250   CN1  =002303
CN10 =002774   CN11 =003026   CN12 =003050   CN13 =003063   CN13A=003104
CN13B=003133   CN14 =003155   CN15 =003171   CN16 =003226   CN17 =003233
CN17A=003241   CN18 =003255   CN18A=003276   CN19 =003312   CN19A=003364
CN19B=003404   CN2  =002343   CN20 =003431   CN20A=003453   CN20B=003471
CN21 =003513   CN22 =003545   CN22B=003601   CN23 =003656   CN24 =003725
CN25 =003756   CN26 =004021   CN27 =004052   CN27A=004071   CN28 =004121
CN28A=004145   CN29 =004200   CN3  =002405   CN30 =004241   CN30A=004255
CN31 =004273   CN32 =004320   CN32A=004347   CN32F=004354   CN33 =004371
CN33A=004410   CN34 =004400   CN34A=004427   CN35 =004526   CN35A=004532
CN35B=004555   CN35C=004457   CN35D=004645   CN35F=004573   CN36 =004607
CN37 =004636   CN38 =004670   CN38B=004711   CN38C=004507   CN39 =004727
CN4  =002424   CN40 =004751   CN40A=004774   CN40B=005021   CN40C=005036
CN41 =005047   CN42 =005072   CN43 =005112   CN44 =005135   CN45 =005155
CN46 =005176   CN47 =005223   CN48 =005251   CN49 =005271   CN5  =002515
CN50 =005310   CN51 =005345   CN52 =005363   CN6  =002532   CN7  =002575
CN7A =002634   CN8  =002707   CN8A =002745   CNZER=000147   COBCL=000150
COBDA=000151   COBER=127726   COBSE=000152   COBSY=001570   CODCB=143516
CODLS=143347   CODSN=143345   COLSN=134011   COM2F=000154   COMFL=000153
COMIN=000155   COMOU=000156   COMPC=141065   COMST=000460   COMTA=001672
CORCO=007733   COSED=143506   CPNT =000157   CR   =123424   CREAD=000375
CRLF =000376   CRLFC=000160   CRSCO=131302   CRTFS=000161   CSGFN=000162
CSRFI=116563   CSTRI=000163   CSWFI=116602   CUR1S=000170   CUR2S=000171
CURLI=000164   CURSL=000165   CURSU=000167   CURTP=000166   CWIMA=121204
CWRIT=000377   CXC  =116056   CXL  =116134

DACHR=112133   DALOG=000010   DAREF=111707   DBG  =116223   DBGBT=000047
DECOD=117152   DECPR=134172   DECUT=117476   DEFPR=140237   DEFSC=133254
DEFSY=145034   DELCL=134573   DELIM=005432   DELPR=134306   DELRS=140450
DELRT=122303   DELSC=133451   DELTB=000402   DESCR=005436   DFSY1=000340
DFSY2=000341   DGET =000400   DGETY=000401   DGPAR=000200   DIC  =115113
DICP =000176   DICPP=000177   DICRE=000175   DISPL=012645   DLMOD=000201
DMO  =115104   DNREC=133172   DNREF=133174   DOCTU=117342   DP1  =000001
DP2  =000002   DP3  =000003   DP4  =000004   DP5  =000005   DP6  =000006
DPAGL=000000   DPAGP=000030   DPGPR=000020   DPSY1=000302   DPSY2=000303
DPUN0=123366   DPUN1=123362   DPUN2=123355   DREG =000101   DSEGT=160000
DSRTF=122403   DSTAC=002000   DSTAD=000006   DUBIC=132125   DUMPR=142126
```

```
DW1  =000356   DW2  =000357   DW3  =000360   DWC  =116163   DWL  =116141
DXREF=112015

EBOOT=001671   ECOBS=001615   ELRTF=000202   EMTYB=116445   ENCOM=126510
END  =115364   ENDBT=000021   ENDRU=127373   ENTAR=000453   ENTBR=000457
ENTDI=000203   ENTDR=000454   ENTER=000403   ENTLR=000456   ENTR =114714
ENTTA=000452   ENTTR=000452   ENTXR=000455   EOF  =115520   EOFBT=000023
ER0  =000000   ER1  =000001   ER10 =000012   ER11 =000013   ER12 =000014
ER13 =000015   ER14 =000016   ER15 =000017   ER16 =000020   ER17 =000021
ER18 =000022   ER19 =000023   ER2  =000002   ER20 =000024   ER21 =000025
ER3  =000003   ER4  =000004   ER5  =000005   ER6  =000006   ER7  =000007
ER8  =000010   ER9  =000011   ERRME=117060   ERRTX=001401   ERRTY=005452
ESCLR=000367   ESSGT=021724   ESYPT=031204   ETX0 =006762   ETX1 =007041
ETX10=007325   ETX11=007123   ETX12=007135   ETX13=007151   ETX14=007162
ETX15=007201   ETX16=007215   ETX17=007226   ETX18=007315   ETX19=007274
ETX21=011752   ETX3 =007054   ETX4 =007074   ETX5 =007107   ETX6 =007261
ETX9 =007360   EXCOM=126121   EXRTL=126126

F100F=000210   F1BAN=001454   FACCO=000204   FALLO=000005   FATLT=005404
FATXR=005431   FCONT=000004   FCURT=000205   FEIRT=000206   FERRF=000404
FETCB=116370   FFSEG=000405   FFTNF=001442   FIDDD=001615   FINBT=111233
FIXCO=142025   FLAG =000004   FLTOR=122643   FNL  =000000   FNSYM=000406
FNSYT=000407   FOPEN=121630   FP   =000015   FP1  =000001   FP12 =000012
FP3  =000003   FP4  =000004   FP7  =000007   FPNT =000211   FPNT1=000211
FPNT2=000212   FPSYM=000302   FREEP=000207   FRTDE=000410   FRTFT=001427
FSYM =000340   FSYMB=000411   FSYRT=122434   FSYTY=000412   FTMP0=000055
FTMP1=000060   FTMP2=000063   FTMP3=000066   FTMP4=000071   FW1  =000356

GARBT=112455   GCOM =123165   GETEL=000413   GETTA=112735   GPAR =000414
GRTFI=122747   GSGFL=000213   GSGFP=000214

HELPC=133113

IBLCK=000215   IDGBU=117634   IDICR=000175   IGMAI=000217   IGMCO=142775
ILBOU=010730   ILLCH=000415   ILLFI=000416   ILLPA=000417   ILLPT=011610
ILLSE=010515   ILLST=011600   ILREG=000216   IMAGL=140663   IMERR=007751
INBDR=125047   INC  =116000   INCBT=000040   INCOX=000220   INFO =123216
INHB =114043   INHBT=000003   INITR=121755   INITZ=000221   INL  =116076
INLBT=000034   INPFP=000222   INRTF=000004   IOBUF=017014   IOERR=124564
IOINI=111260   IOXIN=001660   IPTAB=000223   IRTDS=001512   ISMAP=120545

KEYDE=000224   KGPAR=000420   KGPFL=000225

LAREG=000105   LBACO=141723   LC   =113604   LCOBC=000226   LDATX=143300
LDBTX=143303   LDDTX=143302   LDELP=134364   LDREG=000106   LDXTX=143301
LEAVE=000421   LEAVX=000422   LEGSE=000423   LEIRT=000227   LENTE=000424
LERRF=000425   LESGT=000230   LF   =113546   LFILT=000231   LFIXC=000232
LFRTD=142562   LFSEG=140211   LIALP=141730   LIB  =114442   LIBNO=000233
LIBR =114422   LINK1=000364   LINK2=000365   LINK3=000366   LIRTF=131714
LISEG=010043   LITES=000010   LLEAV=000426   LLRTL=145236   LMON =000427
LNF  =114017   LOCOM=125454   LOGAD=000002   LONGF=000234   LOSEG=010032
LOWPR=000235   LR   =113552   LRD  =113571   LRP  =113556   LRTDS=000236
LRTFP=000237   LRTLP=000240   LSEGT=000241   LSM  =114046   LSTAC=021014
LSTAS=000500   LTBCH=000430   LTBP =000242   LTBPP=000243   LTK0 =000244
LTK1 =000245   LTREG=000104   LUSEG=000246   LUSFA=000247   LVERS=006236
LXLEA=000431   LXREG=000107

M2   =177776   M3   =177775   M4   =177774   M5   =177773   M6   =177772
```

```
MAIN =114462    MAPR =120577    MASK2=001647    MASSA=000003    MASSN=000700
MAXCB=000054    MAXLT=176170    MAXPA=000250    MCURT=177770    MDIGI=123511
MESS =117037    MK   =000251    MLEV =000010    MLEVB=000030    MOVAA=143104
MOVAN=143103    MOVB =140131    MOVNA=143101    MSGNA=123746    MXPRW=000252

NBSCO=125127    NBYT1=000253    NCCOM=011227    NCOBS=116703    NCOST=000254
NELRT=000255    NEWPA=000256    NEWPC=131707    NL   =000000    NNEWP=131711
NO   =006225    NOBS =000257    NOBSW=000260    NOCTU=117444    NOIOB=001511
NOYES=123241    NRLCO=125744    NRTDA=011161    NSCOM=125132    NSCRF=011630
NSMES=007550    NULL =000261    NWRTF=000713

OCT1 =117406    OCTO =117427    OCTOX=117440    OCTUT=117362    ODUMP=136165
OFFES=124256    OFPAG=000024    OFTYP=000006    OLDCH=000262    ONESC=124247
OPENF=124267    OPNT =000263    OPSEG=000003    OREAD=000432    OSTRI=000264
OUPFP=000265    OUTDE=000266    OUTFI=000267    OWRIT=000433

P0D  =000000    POULI=173777    P1   =000001    P10  =000010    P11  =000011
P12  =000012    P13  =000013    P14  =000014    P15  =000015    P1D  =000001
P2   =000002    P2D  =000002    P3   =000003    P3D  =000003    P4   =000004
P4D  =000004    P5   =000005    P5D  =000005    P6   =000006    P6D  =000006
P7   =000007    P7D  =000007    P8D  =000010    PACCE=000270    PAGLI=000000
PAGPH=000002    PAPIT=007450    PARAD=010315    PARDE=000271    PARDG=000634
PARIL=125037    PARNO=000272    PARRT=000714    PB   =000273    PBINF=011352
PBOOT=011334    PBRFT=007570    PCHAR=000434    PCIL1=011724    PCILA=011667
PCIMS=000274    PCLAD=012661    PCRCO=136631    PCTCO=136500    PELEM=142520
PIMFI=011310    PINPF=007560    PLADR=010674    PMESS=007406    PMO  =114776
PNOBS=000275    PNPIT=007434    PNSPA=000435    POPFP=007741    PPB  =000276
PR1CO=000300    PR2CO=000301    PRADR=134067    PRDUM=143665    PRIAD=000277
PRIBT=000025    PRLOA=143665    PRMCA=000112    PRPAR=000653    PRRES=121015
PRSCO=141736    PRTDS=011320    PSEGN=007426    PSGNA=010462    PSTKL=011216
PSY1 =000302    PSY2 =000303    PSY3 =000304    PSYMP=007572    PSYSE=011241
PSYVA=011656    PTDAT=011362    PUADR=010705    PWIPA=007504    PWMCA=000110
PWORD=117274    PWORK=000305

QUEST=010446

RBINC=134761    RBINX=135613    RBRFB=111220    RBRFI=120252    RC5CO=141721
RCB  =000436    RDIGI=133016    RDUMP=142230    RE16 =001650    READ =000437
READ1=000440    REESE=000014    REF  =114665    REFSY=140423    REGBL=005374
REGNA=005424    RELCO=125750    RELSC=131277    RENAM=140543    REORG=110464
RERES=121003    RERTF=000003    RESET=117662    RESPA=120673    RFELS=000007
RFILN=000441    RFORT=145031    RIMAG=121517    RINGP=007466    RIOPA=000646
RIRTF=122002    RL1  =113115    RMCSE=000113    RP   =000016    RPLME=007417
RPRAR=144663    RPRCO=144221    RPRFI=144615    RPROF=143303    RRESP=120777
RRPRD=110037    RRPRL=110046    RRRRT=110030    RRRTL=110021    RRTD =121403
RRTEL=122540    RRTLO=110012    RSAPR=117263    RSCOM=126116    RSEGM=120541
RSEGT=121136    RSFIL=120241    RSMES=006715    RSRTF=122036    RSSWA=110067
RTDCH=113202    RTDSI=000017    RTEMP=113246    RTFAC=122557    RTFBL=000712
RTFBU=000306    RTFIL=000711    RTFNN=012302    RTFRE=000660    RTFSE=000010
RTFUL=007763    RTLOA=110003    RTLSE=000004    RTPAC=110413    RTPB =113537
RWORK=000524

S1LOG=000343    S1LON=000344    S1MAD=000345    S2LOG=000346    S2LON=000347
S2MAD=000350    SAVTA=000074    SAVX =000103    SC5CO=141716    SCAB =000442
SCCOM=131650    SCIUS=010055    SCLC =000443    SCRBU=021514    SCRFE=124727
SCRFI=000307    SDLAD=116627    SEG0L=000317    SEG0U=000320    SEG1L=000321
SEG1T=000322    SEG1U=000323    SEG2L=000324    SEG2T=000325    SEG2U=000326
SEGBL=000310    SEGCO=007722    SEGFP=000641    SEGFT=000444    SEGLI=000000
SEGLL=000311    SEGM =000007    SEGML=000312    SEGN1=000314    SEGN2=000315
```

```
SEGNP=000313   SEGSI=000005   SEGTB=000704   SEGTP=007473   SEGTT=000445
SEGUL=000316   SERRF=000327   SERRO=000446   SFL  =113725   SFORT=145026
SFREE=000447   SFRS1=142670   SFRSY=142630   SG1AD=000331   SGF1E=010574
SGF2E=124737   SGFER=124752   SGFIL=000330   SGFTA=000721   SHSYL=117155
SILLC=000450   SIZCC=011417   SIZF =000664   SKIPF=000332   SLADC=132346
SNCON=011365   SNNDE=012472   SNOIO=143251   SOVER=113265   SPACE=000333
SPELE=142461   SPROF=143267   SPROT=000005   SPTCO=134672   SRCHI=000451
SREEP=000006   SRING=000334   SRL  =113775   SRTD =121226   SRTFI=001471
SSCOM=131646   SSEGM=120476   SSEGT=121022   SSFIL=134642   SSTEN=001503
SSTLE=001506   STACK=000335   STADR=000006   STATU=000001   STATX=143304
STBEG=021014   STDEL=000015   STDGB=125113   STDTX=143306   STEND=021514
STPNT=000336   STZTX=143305   SUBGA=112654   SWAPO=110101   SXCL1=000004
SXCL2=000005   SXCLC=000003   SXS0 =000000   SXS1 =000001   SXS2 =000002
SXSEG=121161   SY1  =000340   SY2  =000341   SY3  =000342   SYMB =111360
SYMBP=000337   SYMW =117073   SYTYP=000670

T2SFN=010212   TABDC=142304   TABLI=126152   TADRE=000077   TAPRF=012425
TAREA=007523   TAREG=000077   TASGF=012447   TAXN =001513   TBPUN=011625
TCAEX=010135   TCART=012050   TCCUS=010532   TCHCC=010323   TCI  =123255
TCIMA=007771   TCLAB=010155   TCLCO=010754   TCOC1=006331   TCRSE=006652
TDCCO=010716   TDGET=116476   TDGEY=116536   TDMAN=010123   TDSSE=012026
TEMP =000005   TERME=006677   TFIXE=011525   TFL0 =007575   TFL1 =007605
TFL2 =007617   TFL3 =007632   TFL4 =007646   TFL5 =007662   TFL6 =007675
TFL7 =007710   TFPCC=011444   TFRPA=010253   TGARB=000351   TGPEL=142554
THISS=117177   TILAD=010644   TILCO=007001   TILLC=010344   TILLF=010071
TILPT=012776   TIMC1=006373   TIMC2=006416   TINIZ=010615   TISNA=012346
TLABE=010512   TLADR=010107   TLASE=010274   TLCHE=006354   TLFNA=011275
TLINK=000000   TLO1 =006511   TLO11=006541   TLO2 =006525   TLO22=006561
TLOCO=126503   TMADR=010117   TNCAW=012235   TNDEM=010127   TNOIO=012220
TNOTI=012270   TNOTP=006315   TNPCC=011471   TNREE=005715   TNREF=007372
TNRTF=012202   TNSBO=006606   TNSGS=012402   TNSNA=011265   TNSPA=012006
TNSPR=010372   TNSW1=012124   TNSW2=012147   TOLFR=122063   TOLTB=111542
TONLY=012720   TORTF=122730   TOSNA=011255   TP1SE=011024   TP2SE=011034
TPASI=010745   TPCOS=011044   TPPRI=011015   TPRAC=010356   TPROG=012715
TPROS=011540   TPROT=007511   TPRPT=012756   TPRTD=011070   TPRTP=010452
TPSGF=010166   TPSTA=011057   TPTAB=010562   TPUNT=006217   TREES=011557
TREFE=112347   TREG =000077   TRLCC=011130   TRTAD=010473   TRTDE=012070
TRTFI=010766   TRYEX=011143   TSCER=011571   TSDEF=010421   TSECH=011747
TSEMP=011410   TSFNO=010221   TSFNU=010201   TSGEX=011106   TSGNA=012362
TSNAM=010436   TSNDC=012510   TSNFO=010407   TSNO =010103   TSNUS=010020
TSPRO=011552   TSREF=011200   TSYER=012622   TTEND=006174   TTEOF=006202
TTTEX=006210   TUADR=010113   TWARN=006625   TWSE1=012524   TWSE2=012563
TXADR=010771   TXCAR=011006   TXCLA=010776   TXCOB=012105   TXCOM=006754
TXDEF=006262   TXERR=005513   TXILC=006744   TXSEG=011516   TXSIZ=006601
TXSSN=012336   TXSYA=012676   TXXSZ=010232   TYESN=006227   TYSEG=011564

UÇOBC=000352   UNFIX=142054

W1   =000356   W2   =000357   W3   =000360   WAREG=000353   WARTF=122266
WBCIM=121747   WBRTF=000354   WCLCO=134015   WHATC=141445   WIMAG=121515
WLACO=134405   WLINK=000010   WLSEQ=126155   WMCSE=000111   WP   =000017
WPRCO=134013   WPSEG=130761   WRERT=000032   WRNRE=133242   WROPA=007031
WROPN=007022   WRSAC=134021   WRSCO=134017   WRTFI=122125   WSEGB=121750
WSEGN=130674   WSRTF=122233   WSYCO=133475   WTEMP=113222   WTSGF=144632
WXREG=000355

XERR0=005532   XERR1=005546   XERR2=005561   XERR3=005575   XERR4=005610
XERR5=005620   XERR6=005672   XERR7=005677   XERR8=005707   XLOCO=125450
XREG =000102   XRSEG=120562   XSGF0=000731   XSGF1=000771   XSGF2=001031
```

```
XSGF3=001071   XSSEG=120555   XSWAP=110074   XTCI =123303   XTCLC=006461
XTLAD=006450   XTLDA=006466   XTUAD=006454   XTUDA=006477   XWORK=000570

YES  =006223

ZEROE=112375   ZEROP=123332

% :: Part One Listing
000002
000002   %********************  S I N - D A T A  ***************************
000002   %**&*****************************************************************
000002   %**&*****************************************************************
000002   %
000002   %
000002   %     C O M M O N   P A R T   O F :
000002   %
000002   %     S I N T R A N  I I I -  V S E /  V S X / V S X - 5 0 0
000002   %    ==================================================
000002   %
000002   %         V E R S I O N   N D 1 0 1 7 4 J
000002   %                         N D 1 0 1 7 5 J
000002   %         ------------------------------
000002   %         O P E R A T I N G - S Y S T E M
000002   %
000002   %      FOR THE ND-100, ND-100/CX, ND-100/500 COMPUTERS
000002   %
000002   %
000002   %**&*****************************************************************
000002   %**&*****************************************************************
```

```
000002
000002   *0/
000000
000000   %==========================================================================
000000   %
000000   %  1.0        G L O B A L   D A T A
000000   %
000000   %==========================================================================
000000   % 1.1         D E F I N I T I O N S
000000   %
000000
000000   % DEFINITION SYMBOLS
000000
000000   SYMBOL ALEVL=1,ALEV=2,ALEVB=10          % RT LEVEL
000000   SYMBOL MLEVL=3,MLEV=10,MLEVB=30         % MONITOR LEVEL
000000   SYMBOL BLEVL=4,BLEV=20,BLEVB=40         % INBT-OUTBT LEVEL
000000   SYMBOL LV10=2000,LV10B=120              % PIO/OUTPUT
000000   SYMBOL LV11=4000,LV11B=130              % DMA/INPUT & OUTPUT
000000   SYMBOL LV12=10000,LV12B=140             % PIO/INPUT & HDLC OUTPUT
000000   SYMBOL LV13=20000,LV13B=150             % RT-CLOCK & HDLC OUTPUT
000000
000000   SYMBOL MASKE=76033            % PIE-REGISTER
000000
000000   %==========================================================================
000000   %       S E G M E N T   N A M E S
000000
000000   SYMBOL 5CIMSEG=2              % CORE IMAGE
000000   SYMBOL 5OPSEG=3               % COMMAND SEGMENT
000000   SYMBOL 5RTSG=4                % RT-LOADER SEGMENT
000000   SYMBOL 5ERRSEG=5              % SYSTEM SEGMENT FOR ERROR PROGRAM
000000   SYMBOL 5FILSEG=6              % FILE SYSTEM SEGMENT
000000   SYMBOL 5MACDSEG=7             % DMAC SEGMENT
000000   SYMBOL 5RTFIL=10              % RTFIL SEGMENT
000000   SYMBOL 5LOGRSEG=11            % ERROR LOG SEGMENT
000000   SYMBOL 5IFS2=12               % INITIAL REENTRANT FILE SYS. SEG #2
000000   SYMBOL 5IRTS=13               % INITIAL RT-LOADER SEGMENT
000000   SYMBOL ECODSEG=14             % ERROR ROUTINE SEGMENT
000000   SYMBOL 5FIUS=22               % REENTRANT FILE USER SEGMENT #1
000000   SYMBOL 5FIU2=26               % REENTRANT FILE USER SEGMENT #2
000000   SYMBOL 5RRUS=40               % FILE USER DATA SEGMENT FOR RT PROGRAMS
000000   SYMBOL 55SDS=20               % ND-500 STANDARD DOMAINS SEGMENT
000000   SYMBOL 55NSG=21               % ND-500 NAME TABLES SEGMENT
000000   SYMBOL 5ISMS=15               % INITIAL SERVICE-PROGR. AND MAIL SEGMENT
000000   SYMBOL 5OP2SEG=23             % SIN-SERV-PROG AND MAIL SYSTEM SEGMENT
000000   SYMBOL SPSEG=25               % SPOOLING COMMAND SEGMENT
000000   SYMBOL FRSG1=24               % REENTRANT FILE SYSTEM SEGMENT #1
000000   SYMBOL FRSG2=25               % REENTRANT FILE SYSTEM SEGMENT #2
000000   SYMBOL 5INNE=16               % INITIAL NORD-NET
000000   SYMBOL 5NNET=27               % NORD-NET SEGMENT
000000   SYMBOL 5ACCS=32               % ACCRT (RT ACCOUNTING) SEGMENT
000000   SYMBOL 5XSG1=33               % XMSG POF SEGMENT
000000   SYMBOL 5XSG2=34               % XMSG DEMAND SEGMENT (XROUT)
000000   SYMBOL 5BADM=36               % TADADM SEGMENT
000000   SYMBOL 5RT2SG=37              % RT-LOADER DATA SEGMENT
000000   SYMBOL 5PT3S=41               % PIT3-SEGMENT
000000   SYMBOL 5ISPS=42               % INITIAL SPOOLING SEGMENT
000000   SYMBOL SPSGM=43               % SPOOLING SEGMENT
000000   SYMBOL 5IPT3S=17              % INITIAL PIT3-SEGMENT.
000000
000000
```

```
000000    SYMBOL RING1=0,RING2            % PROTECTION RINGS (SEGMENT TABLE ENTRY)
000000    SYMBOL KBACTPRI=105210          % PCR FOR BACKGROUND (ALT=NORM=1)
000000    SYMBOL KABACTPRI=105410         % ALTERNATIVE PCR FOR BACKGROUND (ALT=2, NORM=1)
000000    SYMBOL XMSGX=0                  % DUMMY TO FILL HOLE IN MON SWITCH TABLE LEV 1
000000    SYMBOL 5MESSIZE=145             % ND-500 MESSAGE LENGTH
000000    SYMBOL POTESIZE=104             % PAGE-OWNER-TABLE-ELEMENT SIZE
000000    SYMBOL FIXEDEV=2                % LOGICAL DEVICE FOR ERROR-DEVICE
000000    SYMBOL ESCMASK=170076           % MASK USED TO RESET FLAGB
000000
000000    INTEGER ARRAY ERSTB=?,ERTAB=? % ERROR MESSAGE POINTERS IN ERLIST IN ERROR-PROGR.
000000    INTEGER ARRAY SFTIM=?           % FLOPPY FUNCTION DESCRIPTION TABLE
000000    INTEGER 1NBPP=?                 % NUMBER OF BACKGROUND PROCESSES
000000    INTEGER 2NBPP=?                 % NUMBER OF DYNAMIC ALLOCATED BACKGROUND PROCESSES
000000    INTEGER 5MBBANK=?               % MEMORY BANK OF ND-500 MESSAGES
000000    INTEGER 5FXTBL=?                % 16 LEAST SIGNIFICANT BITS IN ADDR OF "UDMA-ND500-FIX" TABLE
000000    INTEGER 5DSPS=?                 % 16 LEAST SIGNIFICANT BITS IN ADDR OF "UDMA-ND500-CAPABILITY" TABLE
000000    INTEGER 5FXBNK=?                % MEMORY BANK OF "UDMA-ND500" TABLES
000000    INTEGER XDILF=?                 % FUNCTION CODE ADDRESS FOR MON ABSTR TO DISC-ACCESS-LOG
000000    INTEGER ERUCL=?                 % PANEL CLOCK ERROR (INCORRECT TIME)
000000    INTEGER ARRAY TBASE=?           % USED BY READCLOCK
000000    INTEGER ARRAY DAYEAR=?          % USED BY READCLOCK
000000    INTEGER ARRAY DAMONTH=?         % USED BY READCLOCK
000000    INTEGER ADDYEAR=?               % USED BY READCLOCK
000000    INTEGER ADDMONTH=?              % USED BY READCLOCK
000000    INTEGER HDAYES=?                % USED BY READCLOCK
000000    INTEGER DAYES=?                 % USED BY READCLOCK
000000    INTEGER HOURS=?                 % USED BY READCLOCK
000000    INTEGER SECOND=?                % USED BY READCLOCK
000000    INTEGER HDAY2=?                 % USED BY READCLOCK
000000    INTEGER SEC2=?                  % USED BY READCLOCK
000000    INTEGER PERIOD=?                % USED BY READCLOCK
000000    INTEGER M1TMRTAB=?              % IN TIMER TABLE
000000    INTEGER ARRAY NDEMFIELD=?       % WORKING AREA FOR MONITOR CALLS
000000    INTEGER BCHTAB=?                % BATCH TABLE
000000    INTEGER MDSEG=?                 % DMAC BREAKPOINT
000000    INTEGER ARRAY REFBP=?           % DMAC BREAKPOINT
000000    DOUBLE ARRAY NINIT=?            % TABLE OF NOT INITIALIZED MEMORY PAGES
000000    DOUBLE ARRAY CUMTABLE=?         % TABLE OF MEMORY RESERVED FOR THE CUSTOMER
000000    DOUBLE ARRAY USIOXTAB=?         % TABLE OF USER-RESERVED PHYSICAL DEVICE NUMBERS
000000    INTEGER ARRAY SVTXT=?           % VERSION STRING ON OPCOM. SEGMENT
000000    INTEGER BACTPRI=?               % ON SYSTEM SEGMENT
000000    INTEGER UEFLG=?                 % ON SYSTEM SEGMENT
000000    INTEGER MXSIBAS=?               % NUMBER OF SIBAS PROCESSES
000000    INTEGER PSCCTAB=?               % POINTER TO RTCOMMON-SEGMENT IN POF
000000    DOUBLE DFPTSLICE=?              % FIRST AND LAST PROGRAM TO TIMESLICE
000000    INTEGER TADFPHPAGE=?            % FIRST PHYSICAL PAGE FOR TAD-STACK
000000    INTEGER TADLPHPAGE=?            % LAST PHYSICAL PAGE FOR TAD-STACK
000000    INTEGER MBPRVTTABLE=?           % MEMORY BANK OF PRVTTABLE
000000    INTEGER MBSPRTAB=?              % MEMORY BANK OF BPRTAB
000000    INTEGER APRVTTABLE=?            % 16 LEAST SIGNIFICANT BITS IN PHYSICAL ADDR OF PRVTTABLE
000000    INTEGER AEPRVTTABL=?            % 16 LEAST SIGNIFICANT BITS IN PHYSICAL ADDR OF END OF PRVTTABLE
000000    INTEGER ASBPRTAB=?              % 16 LEAST SIGNIFICANT BITS IN PHYSICAL ADDR OF SBPRTAB
000000    INTEGER AEBPRTAB=?              % 16 LEAST SIGNIFICANT BITS IN PHYSICAL ADDR OF END OF SBPRTAB
000000    INTEGER ONTTMCOUNT=?            % TIME BEFORE LOGGED OUT IN BACKGROUND ALLOCATION SYSTEM
000000    INTEGER TTMWARNING=?            % TIME BEFORE WARNING MESSAGE FROM BACKGROUND ALLOCATION SYSTEM
000000    INTEGER ARRAY DISPE=?           % ARRAY OF DISC-LAYOUT ELEMENT POINTERS
000000
000000    % MONITOR CALL LOG AND SWAPPING LOG VARIABLES IN POF
000000    INTEGER CMCLG=?,MCLGBANK=?,TNMCALL=?,MCLGFLG=?,MCLGPAGE=?,MCLGOWNER=?
000000    INTEGER CSWLG=?,TFPF1=?,TFPF2=?  CPF1=?,CFPF2=?,TPFR1=?,TPFR2=?
```

```
000000    INTEGER CPFR1=?,CPFR2=?,TPF41=?,TPF42=?,CPF41=?,CPF42=?,CCSWLG=?
000000    INTEGER TPF11=?,TPF12=?,CPF11=?,CPF12=?,SNWP1=?,SNWP2=?,SWLGOWNER=?
000000    DOUBLE TFPFS=?,CFPFS=?,TPFRS=?,TPFL4=?,CPFL4=?,TPFL1=?,CPFL1=?,SNWPS=?
000000
000000
000000    *PIOF=POF+1; PION=PON+2; PONN=PON
000000
000000    % SPECIAL INSTRUCTIONS FOR NORD 100
000000    SYMBOL 5N100=14                % NORD-100 INDICATOR IN STATUS REGISTER
000000    SYMBOL 5SEXI=15                % N-100 EXTENDED ADDRESS MODE IN STATUS REGISTER
000000    SYMBOL CCLR=10                 % CLEAR CACHE
000000    SYMBOL ECCR=15                 % ERROR CORRECTION CONTROL REGISTER
000000    SYMBOL EXAM=150416
000000    SYMBOL DEPO=150417
000000    SYMBOL SEX=150406
000000    SYMBOL REX=150407
000000    SYMBOL MOVB=140131             % MOVE BYTE STRING FORWARDS / BACKWARDS
000000    SYMBOL MOVBF=140132            % MOVE BYTE STRING FORWARDS
000000    SYMBOL LDATX=143300
000000    SYMBOL LDXTX=143301
000000    SYMBOL LDDTX=143302
000000    SYMBOL LDBTX=143303
000000    SYMBOL STATX=143304
000000    SYMBOL STZTX=143305
000000    SYMBOL STDTX=143306
000000    SYMBOL IOXT=150415
000000    SYMBOL ADDD=140120
000000    SYMBOL ISETP=140300            % INSTRUCTION FOR ROUTINE SETPT
000000    SYMBOL ICLEP=140301            % INSTRUCTION FOR ROUTINE CLEPT
000000    SYMBOL ICLNR=140302            % INSTRUCTION FOR ROUTINE CLNREENTR
000000    SYMBOL ICHRE=140303            % INSTRUCTION FOR ROUTINE CHREENTPAGES
000000    SYMBOL MOVNN=143100            % MOVEW NORMAL PAGE TABLE TO NORMAL PAGE TABLE
000000    SYMBOL MOVNA=143101            % MOVEW NORMAL PAGE TABLE TO ALTERNATIVE PAGE TABLE
000000    SYMBOL MOVNP=143102            % MOVEW NORMAL PAGE TABLE TO PHYSICAL MEMORY
000000    SYMBOL MOVAN=143103            % MOVEW ALTERNATIVE PAGE TABLE TO NORMAL PAGE TABLE
000000    SYMBOL MOVAA=143104            % MOVEW ALTERNATIVE PAGE TABLE TO ALTERNATIVE PAGE TABLE
000000    SYMBOL MOVAP=143105            % MOVEW ALTERNATIVE PAGE TABLE TO PHYSICAL MEMORY
000000    SYMBOL MOVPN=143106            % MOVEW PHYSICAL MEMORY TO NORMAL PAGE TABLE
000000    SYMBOL MOVPA=143107            % MOVEW PHYSICAL MEMORY TO ALTERNATIVE PAGE TABLE
000000    SYMBOL MOVPP=143110            % MOVEW PHYSICAL MEMORY TO PHYSICAL MEMORY
000000    SYMBOL PGC=14                  % READ PAGING CONTROL REGISTER (TRA PGC)
000000
000000    SYMBOL LDTSZ=3                 % SIZE OF LAMU DESCRIPTION ELEMENT
000000    SYMBOL ALMSZ=2                 % SIZE OF ACTIVE LAMU TABLE ELEMENT
000000    SYMBOL NINSZ=24                % NUMBER OF ELEMENTS IN NINITPAGE
000000    SYMBOL NNSWSZ=24               % NUMBER OF ELEMENTS IN NSWPAGE
000000    SYMBOL MMESIZE=4               % MEMORY MAP ELEMENT SIZE
000000    SYMBOL CUMSIZE=12              % NUMBER OF ELEMENTS IN CUMTABLE
000000    SYMBOL USDVSIZE=10             % NUMBER OF ELEMENTS IN USIOXTAB
000000
000000    %RT DESCRIPTION:
000000    DISP 0
000000            INTEGER TLINK          % TIME QUEUE
000000            INTEGER STATUS         % STATUS BITS (SEE BELOW) AND PRIORITY
000000            INTEGER DTIM1,DTIM2    % START TIME
000000                  DOUBLE DTIME=DTIM1
000000            INTEGER DTIN1,DTIN2    % INTERVAL - TIME USED FOR BACKGROUND
000000                  DOUBLE DTINT=DTIN1
000000            INTEGER STADR          % START ADRESS
000000            INTEGER SEGM           % INITIAL SEGMENTS
000000
```

```
000000          INTEGER WLINK          % WAITING QUEUE
000000          INTEGER ACTSEG         % ACTUAL SEGMENTS
000000          INTEGER ACTPRI         % ACTUAL PAGE TABLES AND RING
000000          INTEGER BRESLINK       % BEGINNING OF RESERVATION LINK
000000          INTEGER RSEGM          % REENTRANT SEGMENT
000000          INTEGER WINDOW         % LEFT BYTE: BUFFER WINDOW,RIGHT BYTE: USER W.
000000          INTEGER RTDLGADDR      % LOGICAL ADDR OF REGISTER BLOCK
000000   PSID
000000   DISP 0
000000   INTEGER DPREG,DXREG,DTREG,DAREG,DDREG,DLREG,DSREG,DBREG     % REGISTERS
000000   INTEGER BITMAP,BITM1,BITM2,BITM3,BITM4,BITM5,BITM6,BITM7    % SAVED REENTRANT BITMAP
000000   INTEGER ARSEGM,AACSEGM                                      % SAVED SEGMENT IN ABSTR
000000          TRIPLE TDXREG=DXREG
000000   PSID
000000   SYMBOL 5RTSIZE=17,5BRESLINK=13,5WLINK=10,5BITM=10     % POS. IN RT-DESCR.
000000   SYMBOL XTDPR=0,XTDXR=10,XTDTR=20,XTDAR=30,XTDDR=40,XTDLR=50,XTDSR=60,XTDBR=70
000000   SYMBOL 5XRTDSIZE=20                          % SIZE OF RT-DESCRIPTION IN POF
000000
000000   % DATA-DEFINITIONS FOR SAVE/UNSAVE OF RT-DESCRIPTION'S REENTRANT
000000   % SEGMENT INFORMATION
000000
000000   DISP 0
000000   INTEGER BXTRG,BXARG,BXBRG,BXLRG
000000   INTEGER XUSEGM,XURSEGM,XUBIM,XUBI1,XUBI2,XUBI3
000000   INTEGER XUBI4,XUBI5,XUBI6,XUBI7
000000   TRIPLE XTADR=BXTRG,XTRSEGM=XURSEGM
000000   DOUBLE DUBIM=XUBIM,DUBM2=XUBI2
000000   PSID
000000   SYMBOL BXBITMAP=6
000000
000000
000000
000000   %STATUS BITS:
000000   SYMBOL 5WAIT=17                % I/O-WAIT
000000   SYMBOL 5REP=16                 % REPEAT EXECUTION
000000   SYMBOL 5RWAIT=15               % RTWT OR HOLD HAS BEEN USED
000000   SYMBOL 5INT=14                 % INTV HAS BEEN USED
000000   SYMBOL 5ABS=13                 % ABSET HAS BEEN USED
000000   SYMBOL 5TMOUT=12               % TMOUT HAS BEEN USED
000000   %IN ACTPRI:
000000   SYMBOL 5BACKGR=17              % BACKGROUND PROGRAM
000000   SYMBOL 5RTOFF=16               % START INHIBITED
000000   SYMBOL 5SWWAIT=15              % PROCESS IN SWAPPING QUEUE
000000
000000   %DATAFIELD, STANDARD PART:
000000   DISP 0
000000          INTEGER RESLINK        % RESERVATION LINK
000000          INTEGER RTRES          % RESERVING RT-PROGRAM
000000          INTEGER BWLINK         % BEGINNING OF WAITING QUEUE
000000          INTEGER TYPRING        % DEV. TYPE BITS (SEE BELOW) AND RING
000000          INTEGER ISTATE         % 0=IDLE, 1=BUSY, -1=NOWAIT MODE
000000          INTEGER MLINK          % MONITOR QUEUE
000000          INTEGER MFUNC          % MONITOR LEVEL FUNCTION ADDRESS
000000
000000   % FOR DMA-DEVICES ONLY:
000000          INTEGER POINTER TRLREG % RETURN ADDRESS ON MON. LEVEL AFTER TRANSF.
000000          INTEGER HSTAT          % HARDWARE STATUS FROM DEVICE
000000          INTEGER POINTER MTRANS % MON. LEVEL ROUTINE TO ACTIVATE DRIVER
000000   PSID
000000   SYMBOL 5BWLINK=2               % POS. IN DATAFIELD
000000   %ACCESS BITS IN TYPRING:
```

```
000000   SYMBOL 5IOBT=17                % INBT/OUTBT ALLOWED
000000   SYMBOL 5RFILE=16               % FILE
000000   SYMBOL 5CONCT=15               % CONCT ALLOWED
000000   SYMBOL 5ISET=14                % IOSET ALLOWED
000000   SYMBOL 5COM=13                 % COMMUNICATION CHANNEL
000000   SYMBOL M144B=12                % BLOCK CALLS ALLOWED
000000   SYMBOL 5CRDLY=11               % CARRIAGE RETURN DELAY IN SOFTWARE
000000   % TYPRING BITS  9-6 ARE USED TO DELAY CARRIAGE RETURN IN DRIVER CRDLY
000000   SYMBOL 5MT=11                  % MAGTAPE
000000   SYMBOL 5FLOP=10                % FLOPPY DISC
000000   SYMBOL 5INVRT=7                % INVERT DIGITAL IO
000000   SYMBOL 5IBDV=6                 % INTERNAL BLOCK DEVICES
000000   SYMBOL 5TERM=5                 % TERMINAL
000000   SYMBOL 5BAD=4                  % BACKGROUND-ACCESS-DEVICE (BAD)
000000   SYMBOL 5NORES=3                % NO RESERVATION NECESSARY (TERM. OUTPUT)
000000   SYMBOL 5CLDV=2                 % CLEAR-DEVICE ROUTINE AVAILABLE
000000   % BIT 0,1=RING
000000
000000   SYMBOL 5HDMA=5INVRT            % BSC ON DMA, HDLC
000000   SYMBOL BIHDLC=16               % MARKING HDLC IN IOBUTAB
000000   SYMBOL BISYMO=15               % MARKING SYNC.MODEM IN IOBUTAB
000000
000000   %MONITOR CALL FIELD, WORKING AREA:
000000   SYMBOL 5REG=7
000000   DISP 5REG
000000          INTEGER ZPREG,ZXREG,ZTREG,ZAREG,ZDREG,ZLREG,ZSREG,ZBREG  % REG. BLOCK
000000          INTEGER OLDPAG         % ACTPRI OF CALLING PROGRAM
000000          INTEGER D0,D1,D2,D3,D4; REAL D90=D0  %PARAM. VALUES
000000          INTEGER ARRAY CL7(7),ECL7(0)  % USED BY MON. CALL CLOCK
000000          INTEGER D5=CL7
000000          TRIPLE ZTADREG=ZTREG
000000          DOUBLE DOLDPAGE=ZBREG % USED FOR LOADING OLDPAGE DIRECTLY INTO D-REG
000000   PSID
000000
000000   %PARAMETER LIST:
000000   DISP 0
000000          INTEGER POINTER P0,P1,P2,P3,P4,P5
000000   PSID
000000
000000   %GENERAL DATA ELEMENT:
000000   DISP 0
000000          INTEGER S0,S1,S2,S3,S4,S5,S6,S7,S10,S11,S12,S13,S14,S15,S16,S17,S20,S21
000000          REAL S90=S0,S91=S3
000000   PSID
000000
000000   DISP 0
000000          DOUBLE DS0,DS1,DS2,DS3,DS4,DS5
000000   PSID
000000
000000   DISP 0
000000          DOUBLE DOU0,DOU2,DOU4,DOU6
000000   PSID
000000
000000   %       MONITOR CALL NUMBERS:
000000   @ICR
000000   SYMBOL 2INBT=1,    2OUTB,     2ECHO,     2BRKM,     2RPAG=7,   2WPAG,     2TIME,     2SETOLD,
000000          2CIBU,     2COBU,     2LASTC=26, 2RTDSC,    2GETRT,    2EXIOX,    2MSG,      2ALTN,
000000          2ALTF,     2IOUT,     2NOWAIT,   2CLOS=43,  2NOPE=50,  2MTERM=52, 2RMAX=62,
000000          2ERMS=64,  2QERM,     2ISIZ,     2OSIZ,     2CMND,     2DESC,     2EESC,     2SMAX,
000000          2SBYT,     2RBYT,     2SBLZ,     2STBC,
```

```
000000          2RT,        2SET,       2ABSE,      2INTV,      2HOLD,      2ABORT,     2CONCT,     2DSCNT,
000000          2PRIOR,     2UPDAT,     2CLADJ,     2CLOCK,     2TUSED,     2FIX,       2UNFIX,     2RFILE,
000000          2WFILE,     2WAITF,     2RESRV,     2RELES,     2PRSRV,     2PRLS,      2DSET,      2DABST,
000000          2DINTV,     2ABSTR,     2MCALL,     2MEXIT,     2RTEXT,     2RTWT,      2RTON,      2RTOF,
000000          2WHERE,     2IOSET,     2ERRMON,    2RSIO,      2MAGTP,     2ACM,       2CAMAC=147,
000000          2GL,        2GRTD,      2GRTN,      2IOXN,      2ASSIG,     2PLOT,      2TRACB,     2ENTSG,
000000          2FIXC,      2INSTR,     2OUTST,     2WRQI,      2WSEG,      2DIW,       2DOLW,      2REEN,
000000          2XMSG=200,  2DOPEN=220, 2SFAC=237,  2SYCN=261,  2SIBA=305,  2LAMU=315,  2SLRM;
000000
000000    @CR;
000000    SYMBOL 2XIBU=2ISIZ,2WSBC=2WSEG,2WSGB=2WSEG,2RTXT=2RTEXT
000000    SYMBOL 2ERM=2ERMS,2CLSE=2CLOS
000000
000000    % AREA BITS FOR SINTRAN-SERVICE-PROGRAM
000000
000000    SYMBOL BMEMO,BIML,BSAVE,BSEG
000000
000000
```

```
000000
000000   %==========================================================================
000000   %
000000   %            I / O - S Y S T E M    D A T A  S T R U C T U R E
000000   %
000000   %==========================================================================
000000   % 1.3          D A T A F I E L D  D E F I N I T I O N S
000000
000000   SYMBOL RTCLD=10, FREQU=2000                          % RT CLOCK
000000   SYMBOL DDR,DDW,DST,DCONT                             % DATA READ, DATA WRITE, STATUS, CONTROL IN IOX
000000   SYMBOL DPIN=1,DACT=4                                 % BITS IN CONTROL WORD
000000   INTEGER MAX10=?,MAX11=?,MAX12=?,MAX13=?              % SIZE OF IDENT TABLES
000000   INTEGER ARRAY DV000=?,DV200=?,DV300=?                % LOG. NO. TABLE PARTS
000000   INTEGER ARRAY ITB10=?,ITB11=?,ITB12=?,ITB13=?        % IDENT TABLES
000000   INTEGER ARRAY SIBBDEVS=?,SIBAPDEVS=?
000000
000000   % DEVICE NUMBER RANGE FOR REMOTE OPENED PERIPHERAL DEVICES:
000000   SYMBOL RDVLO=2500              % DEV.NO IN THE RANGE 2500-2577
000000   SYMBOL RDVHI=2600
000000   SYMBOL RDVDF=2400              % TO BE SUBTRACTED TO GET OPEN FILE NUMBER
000000   SYMBOL RDLOO=24                % THE SAME VALUES USED IN LOGPH
000000   SYMBOL RDHII=26
000000
000000   % - -  DATAFIELDS - -
000000
000000   % CHARACTER DEVICES, CALL PART:
000000   DISP 7
000000           INTEGER POINTER IOTRANS          % CALLED FROM INBT/OUTBT TO TRANSFER
000000           INTEGER POINTER STDEV            % START DEVICE
000000           INTEGER KSETDV                   % ADDRESS OF IOSET ROUTINE
000000           INTEGER POINTER SETDV=KSETDV     % IOSET ROUTINE
000000           INTEGER DFOPP          % OPPOSITE DATAFIELD FOR TWO-WAY DEVICES
000000           INTEGER DERROR         % ERROR CODE
000000           INTEGER BUFST          % START OF RING BUFFER
000000           INTEGER MAX            % BUFFER CAPACITY
000000           INTEGER BHOLD          % NUMBER OF CHARACTERS IN BUFFER
000000           INTEGER HENTE          % FETCH POINTER
000000           INTEGER CFREE          % FREE POSITIONS
000000           INTEGER FYLLE          % STORE POINTER
000000           INTEGER MINBHOLD       % LOWER LIMIT FOR BREAK
000000           INTEGER MAXBHOLD       % UPPER LIMIT FOR BREAK
000000   PSID
000000   SYMBOL 5BLOC=17                % TERMINAL OUTPUT BUFFER-LOCK  FOR OUTSTRING
000000                                  % (TERMINAL MINBHOLD BIT 0-16, BIT 17 IS LOCK)
000000
000000   % TERMINAL DRIVER FIELD:
000000   DISP -45
000000           INTEGER TINFO          % VARIOUS INFORMATION BITS FOR TERMINAL
000000           INTEGER PECH7(10)      % ECHO TABLE 7
000000           INTEGER PBRK7(10)      % BREAK TABLE 7
000000           INTEGER IN5MSG         % ADDRESS OF ND-500 MESSAGE WHEN DOING
000000                                  % QUICK INSTRING TO THE ND-500
000000           INTEGER RSISTE         % ECHO POINTER
000000           INTEGER BRECHOFL       % BREAK & ECHO FLAG
000000           INTEGER POINTER ROUSPEC  % ADDRESS OF SPECIAL SUBROUTINE
000000           INTEGER NCBRK          % NO. OF CHARS AFTER LAST BREAK
000000           INTEGER CTTYP          % TERMINAL TYPE
000000           INTEGER CESCP          % DISCONNECT(LEFT BYTE) AND/OR ESCAPE CHARACTER(RIGHT BYTE)
000000           INTEGER BRKMAX         % MAX BHOLD BEFORE BREAK
000000           INTEGER TSPEED         % TERMINAL SPEED
```

```
000000          INTEGER CNTREG          % CONTROL REGISTER
000000          INTEGER DFLAG           % FLAG BITS (SEE BELOW)
000000          INTEGER ECHOTAB         % ECHO TABLE
000000          INTEGER BRKTAB          % BREAK TABLE
000000          INTEGER LAST            % LAST TYPED CHARACTER
000000          INTEGER KTMSUB
000000          INTEGER POINTER TMSUB=KTMSUB % TIMER SUBROUTINE
000000          INTEGER TMR             % TIMER COUNTER
000000          INTEGER TTMR            % START VALUE OF TMR
000000          INTEGER HDEV            % IOX INSTRUCTION
000000          INTEGER POINTER STDRIV% START POINT OF DRIVER
000000          INTEGER POINTER DRIVER% RESTART AFTER INTERRUPT
000000          INTEGER EMPTFLAG=LAST       %TT OUTPUT; BUFFER EMPTY
000000          INTEGER SCREEN=BRKTAB % COUNTER FOR STOP ON FULL PAGE
000000          INTEGER DIVISOR=TMSUB % CLOCK COUNTER VALUE    %%%
000000          INTEGER CONTW=TMR     % CONTROLWORD            %%% SPECIAL DISP FOR CONCT
000000          INTEGER XHDEV=TTMR    % HDEV                   %%% CLOCK
000000    PSID
000000    SYMBOL 5TMR=-5              % TMR
000000    SYMBOL 5WECH=6              % BIT IN BRECHOFL FOR WAITING ECHO WHEN OUTPUT-
000000                                % BUFFER IS FULL
000000    % TINFO BITS:
000000    SYMBOL 5BFUL=17             % 17: CHARACTERS LOST ON INPUT DUE TO FULL INPUT BUFFER
000000    SYMBOL 5PAER=16             % 16: PARITY ERROR ON INPUT
000000    SYMBOL 5FRER=15             % 15: FRAMING ERROR ON INPUT
000000
000000    %DFLAG BITS:
000000    @ICR
000000    SYMBOL 5ECHO,               %  0: THE DRIVER MAY GIVE ECHO
000000           5BREAK,              %  1: BREAK FLAG
000000           5SPEC,               %  2: SPECIAL BREAK CHAR. NO ECHO ON NEXT CHAR.
000000           5HDUP,               %  3: HALF DUPLEX
000000           5FIMO,               %  4: FIXED LINE, MODEM INTERFACE
000000           5RQI,                %  5: S III COMM.
000000           5CTRLO=5RQI,         %  5: (CTRL)-O PRESSED IN TERMINAL INPUT
000000           5WRQI,               %  6: S III COMM.
000000           5ALEC=5WRQI,         %  6: ALL CHARS. IN BUFFER ECHOED BY TERMINAL INPUT DRIVER
000000           5XON,                %  7: SEND 'XON'
000000           5XOFF,               % 10: SEND 'XOFF'
000000           5XDEVICE,            % 11: 'XON/XOFF' ON INPUT TURNS ON/OFF OUTPUT.
000000           5OXON,               % 12: XON/XOFF STATUS ON OUTPUT
000000           5CAPITAL,            % 13: CONVERT TO CAPITAL LETTERS
000000           6XOFF,               % 14: XON/XOFF STATUS ON INPUT
000000           5IESC,               % 15: INHIBIT ESCAPE
000000           5LBRK,               % 16: MISSING CARRIER
000000           5RDEVICE;            % 17: OUTPUT 'XON/XOFF' TO CONTROL INPUT
000000    @CR;
000000
000000    % TERMINAL OUTPUT DATAFIELD
000000    DISP 25
000000          INTEGER ON5MSG        % ADDRESS FOR ND-500 MESSAGE WHEN DOING
000000                                % QUICK OUTSTRING FROM ND-500
000000          INTEGER CBUADR        % CURRENT USER BUFFER ADDRESS (OUTSTRING)
000000          INTEGER NOCHAR        % NUMBER OF BYTES IN OUTSTRING MON.CALL
000000          INTEGER CNOCHAR       % NUMBER OF WORDS LEFT TO TRANSFER IN OUTST.
000000          INTEGER XNOCHAR       % WORKING LOCATION FOR OUTSTRING
000000          REAL ZOPRG            % P,X,T REGISTES IN OUTSTRING
000000          REAL ZOARG            % A,D,L REGISTES IN OUTSTRING
000000          REAL ZOSRG            % S,B REEGISTERS + OLDPAGE IN OUTSTRING
000000          INTEGER SBHOLD        % SAVED BHOLD IN OUTSTRING
```

```
000000   PSID
000000
000000   @LIB CXCPU
000000   % RESIDENT TERMINAL DATA FIELD FOR SINTRAN III VSX
000000
000000   DISP -4; INTEGER   TDFPHPAGE;       PSID  % PHYSICAL PAGE OF DATAFIELD
000000   DISP -4; INTEGER   ZDBPROG;         PSID  % SAVED DBPROG
000000   DISP -3; INTEGER   TDFLGADDR;       PSID  % LOGICAL ADDR WHITHIN A PAGE OF X-DATAF
000000   DISP -1; INTEGER   ZDFLAG;          PSID  % SAVED DFLAG
000000   DISP  0; INTEGER   ZROUSPEC;        PSID  % SAVED ROUSPEC
000000   DISP  1; INTEGER   ZCTTYP;          PSID  % SAVED CCTYP
000000   DISP  2; INTEGER   ZCESCP;          PSID  % SAVED CESCP
000000   DISP  4; INTEGER   ZTSPEED;         PSID  % SAVED TSPEED
000000   DISP  5; INTEGER   ZCNTREG;         PSID  % SAVED CNTREG
000000
000000   % THE REST OF THE RESIDENT DATAFIELD IN THE DISPLACEMENT RANGE -4 TO 6,
000000   % WHICH IS THE SIZE OF THE TERMINAL DATA FIELD IN RESIDENT, IS EQUAL
000000   % TO THE STANDARD TERMINAL DATA FIELD
000000
000000   % DISPLACEMENT IN TERMINAL DATAFIELD OUTSIDE RESIDENT
000000
000000   DISP  0; INTEGER TDRADDR;           PSID; % ADDR OF DATAFIELD IN RESIDENT
000000   DISP  1; INTEGER XDFOPP;            PSID; % ADDR OF DFOPP IN RESIDENT
000000
000000   DISP  1; INTEGER CXTSTATE;          PSID; % IN BACKTAB (OLD TSTATE)
000000   @ELIB
000000
000000   SYMBOL TDINDI=45                 % LAST NEGATIVE DISPLACEMENT IN TERM INPUT
000000   SYMBOL TDIPDI=44                 % LAST POSITIVE DISPLACEMENT IN TERM INPUT
000000   SYMBOL TDONDI=10                 % LAST NEGATIVE DISPLACEMENT IN TERM OUTPUT
000000   SYMBOL TDOPDI=43                 % LAST POSITIVE DISPLACEMENT IN TERM OUTPUT
000000   SYMBOL TDISIZ=TDINDI+TDIPDI+1          % INPUT  DATAFIELD SIZE
000000   SYMBOL TDOSIZ=TDONDI+TDOPDI+1          % OUTPUT DATAFIELD SIZE
000000
000000   % INTERNAL DEVICES
000000   DISP 24
000000   INTEGER IMAXBHOLD(2)             % MAXBHOLD
000000   INTEGER TERM % TERMINATOR CHARACTER
000000   INTEGER CHARI                    % CURRENT CHARACTER
000000   PSID
000000
000000   %==========================================================================
000000   %      B A C K G R O U N D  S Y S T E M  D A T A  D E F I N I T I O N S
000000
000000   % BACKGROUND DATA FIELD ELEMENTS
000000   DISP 14; INTEGER PRIO,USIDX,IDLE,MXTIME,BCHNUM; PSID   % FOR BATCH
000000   DISP 22; INTEGER BSTATE          % BACKGROUND PROGRAM STATE
000000          INTEGER TSTATE           % TIMESLICE STATE
000000          INTEGER 6TTYNO=TSTATE    % LOGICAL NO. OF TTY FOR BACKGROUND (RT-SINTRAN)
000000          INTEGER DBPROG           % BACKGROUND RT-PROGRAM
000000          INTEGER DBADR            % SAVED P-REG ON ESCAPE; +FILESY. MON. CALLS
000000                                   % B-FIELD FOR RT-SINTRAN BACKGROUND.
000000            INTEGER RIFIL,BCHISTS          % FOR BATCH
000000            DOUBLE DER0,DER2(3)            % ERROR INFORMATION
000000            INTEGER ARRAY BREGBLOCK=DER0   % REGISTER SAVE AT ESCAPE
000000            INTEGER DBPREG,DBACTPRI        % FOR PAGEFAULT ON IOBT LEVEL
000000            INTEGER FLAGB                  % BACKGROUND FLAGS
000000            INTEGER EUSADD                 % USER ESCAPE HANDLING ADDRESS
000000            INTEGER LUSADD                 % USER LOCAL  HANDLING ADDRESS
000000   PSID
```

```
000000   SYMBOL 5BCOM=1,5BUSER          % IN BSTATE
000000
000000   DISP 23; INTEGER ROFIL,BCHOSTS; PSID   % BATCH OUTPUT
000000   SYMBOL 5BREGBLOCK=30
000000   % FLAGB BITS:
000000   @ICR
000000   SYMBOL 5TLREP=0,                % 0: TERMINAL LINE REPPORT ENABLED/DISABLED
000000          5ESCON,                  % 1: ESCAPE ALLOWED IN COMMAND MODE
000000          5ESC2SET,                % 2: ESCAPE HAS BEEN TYPED BUT NOT SERVED
000000          5LOGOUT,                 % 3: LOG OUT; MISSING CARRIER!
000000          5ABJOB,                  % 4: ABORT JOB.
000000          5LBLOG,                  % 5: LOGOUT ON CARRIER MISSING.
000000          5LCHAR,                  % 6: LOCAL FUNCTION ON
000000          5USESC,                  % 7: USER ESCAPE HANDLING ON
000000          5ESCLOFF,                % 10: DELAYED ESCAPE/LOCAL ACTION
000000          5WESC,                   % 11: ESCAPE WAITING
000000          5WLOC,                   % 12: LOCAL WAITING
000000          5LOC2SET,                % 13: LOCAL AWAITING SYSTEM ESCON
000000          5FFGPAR,                 % 14: GLPAR CALLED FROM FILESYSTEM
000000          5MLGIN,                  % 15: STARTED BY MON LOGIN
000000          5SPSIO,                  % 16: INPUT FROM "MON-LOGIN-BUFFER", NO OUTPUT
000000          5LSTA;                   % 17: LINE STATUS  0 - OK, 1 - DEAD
000000   @CR;
000000   SYMBOL K5ESC2SET=4
000000   SYMBOL K5LOC2SET=4000
000000
000000   SYMBOL 5WCBUFSI=64,5CBUFSI=5WCBUFSI*2  % COMMAND BUFFER SIZE(WORDS AND BYTES)
000000   SYMBOL 5WORKSIZE=60            % SIZE OF STRING WORKA (WORDS)
000000   SYMBOL 5RTLSEM=503             % RT-LOADER SEMAPHORE
000000   SYMBOL 5DMSEM=1201             % DMAC
000000   SYMBOL 5LOGSEM=1202            % RT-PROGRAM-LOG
000000   SYMBOL 5HSEM=1203              % HISTOGRAM
000000   SYMBOL 5MLSEM=1205             % MAIL
000000   SYMBOL 5BADSEM=600             % BADMIN SEMAPHORE
000000   SYMBOL 5BDSEM=601              %
000000   SYMBOL STDELTA=13              % STACK NEED FOR EACH SUBROUTINE NESTING
000000
000000   % VALUES OF BSTATE IN TERM. DATAFIELD:
000000   @ICR
000000   SYMBOL 5BPASSIVE,              % NOT LOGGED IN   0
000000   %      5BCOMM,                 % COMMAND MODE    1
000000   %      5BUSER,                 % USER MODE       2
000000          5BESC=3,                % ESCAPE IS PUSHED
000000          5ERROR,                 % ERROR PRINTOUT
000000          5REMOT,
000000          5RERUB,
000000          5CFILTRA,               % REMOTE FILE TRANSFER
000000          5ND5ESC,                % ESCAPE WHEN CALLING ND-500 ST.DOM FROM S3 OP.COM.
000000          5LOGIN;                 % LOGGING IN
000000   % VALUES OF TSTATE IN TERM.DATAFIELD:
000000   SYMBOL 5TDUM,                  % OUTSIDE TIMESLICE
000000          5TLOW,                  % INSIDE TIMESLICE
000000          5THIGH,                 % AFTER BREAK CHARACTER.
000000          5TREACT,                % AFTER BREAK CHARACTER (PRIORITY NOT YET RAISED)
000000          5CONT,                  % MEDIUM PRIORITY
000000          5WCONT;                 % WAITING FOR MEDIUM PRIORITY
000000
000000   % BITS IN BMECHO (INPUT/OUTPUT CONTROL IN MODE/BATCH)
000000
000000   SYMBOL BECHOFF,                % NO ECHO
```

```
000000          BOTTERM,                % OUTPUT ON THE TERMINAL
000000          BITERM,                 % INPUT FROM THE TERMINAL
000000          BXOTTERM;               % ECHO ON THE MODE OUTPUT FILE
000000
000000   %COMMAND TABLE ELEMENT:
000000   DISP 0;
000000          INTEGER CNAME,          % POINTER TO NAME STRING
000000                  CPARAM,         % POINTER TO PARAMETER LIST
000000                  CMAND,          % COMMAND ROUTINE
000000                  CPROTECT;       % 0=PUBLIC,1=RT,2=SYSTEM,-1=FILESYSTEM
000000   PSID;
000000   @CR
000000
000000   % REGISTER BLOCK:
000000   DISP 0; INTEGER RPREG,RXREG,RTREG,RAREG,RDREG(2),RSREG(2); PSID
000000
000000
000000   % CHAR. I/O FOR BLOCK DEVICES ( INBT/OUTBT FOR MAGTAPE ETC.)
000000   DISP 22; INTEGER CLOGDV,DFDEV; PSID
000000
000000   % MASS STORAGE DRIVER FIELD
000000   DISP -72
000000          INTEGER SVLWK,OSVWK                % SAVE AREA FOR DRIVER
000000          INTEGER NWLBB,NWLBA                % DEVICE BUFFER. SPARE TRACK ADDRESSES
000000          INTEGER OCMD1,OCMD2,OSVBA,OSVBB,OSVCO,OSVWC          % SAVE AREA
000000          INTEGER SPACO,SPAFL                % SPARE TRACK COUNTER AND FLAG
000000          INTEGER CORCU,SLONG,ECCFL,CPAT1,CDISP,CPAT2,TYPEC,SVLBB
000000          INTEGER SVLBA,ERRC1,ERRC2,SRTRY,SWTRY,SMARG,SVLCO,SCADR,MARGC
000000          INTEGER BUSFL,SVLCA,SVLWC
000000          INTEGER TRG,ARG,DRG,XRG            % REGISTERS WHEN CALLING DRIVER
000000          INTEGER CTRG,CARG,CDRG,CXRG        % WHEN CALLING DRIVER FIRST TIME
000000                  DOUBLE CADRG=CARG
000000                  REAL TADRG=TRG,CTADRG=CTRG
000000          INTEGER ERCNT          % NUMBER OF ERROR RETURNS FROM DRIVER
000000          INTEGER SERRB          % SERIOUS ERROR BITS (NO RETRY)
000000          INTEGER WERRB          % WRITE BACK BITS
000000          INTEGER AERRB          % ACCUMULATED ERROR BITS
000000          INTEGER TACNS          % NUMBER OF RETRIES
000000          INTEGER TACOUNT        % RETRY COUNTER
000000          INTEGER COMFL          % COMPARE FLAG
000000          INTEGER BLSZ           % BLOCK SIZE
000000          INTEGER POINTER TRNSF % DRIVER ADDRESS
000000          INTEGER POINTER BUSY  % BUSY RETURN
000000          INTEGER POINTER FINISH %TRANSFER FINISHED
000000          INTEGER POINTER ERROR % ERROR RETURN FROM DRIVER
000000          INTEGER TRGINI=WERRB  % INITIAL FUNCTION
000000          INTEGER DERRC=MARGC
000000   PSID
000000   DISP 14
000000          INTEGER MEMA1,MEMA2       % INITIAL MEMORY ADDRESS (24 BITS)
000000          INTEGER CMAD1,CMAD2       % CURRENT MEMORY ADDRESS (24 BITS)
000000          INTEGER POINTER CLEDEV    % ADDRESS OF CLEAR-DEVICE-ROUTINE
000000          INTEGER ARRAY HTABL(5)    % DISC LAYOUT TABLE POINTERS (1 LOC IS DUMMY)
000000          INTEGER DIFTCOUNT         % TIME COUNTER AFTER POWER FAIL RESTART
000000          INTEGER DEDFADDR          % ADDRESS OF DISC-ERROR INFORMATION FIELD
000000          INTEGER POINTER PVEFUNC=HTABL  % ADDRESS OF PRINT/PLOT MODUS IN I/O DATAF.
000000          DOUBLE MEMAD=MEMA1,CMADR=CMAD1
000000          INTEGER M2UNTYP=CLEDEV    % CARTRIDGE DISC FLAGS
000000   PSID
000000
```

```
000000   % EXTENSION FOR THE WINCHESTER DATA FIELD
000000
000000   DISP -60; INTEGER ADMIA,ADMIS,ALTPO,BADDR,BADIP,BADTR,WANKN,NOSEC,RTZFL; PSID
000000                                                                            PSID
000000   DISP -46; INTEGER SEEKF;                                                 PSID
000000   DISP -44; INTEGER SHEAD,SSEC,SSTEP,TRTZ;                                 PSID
000000   DISP -37; INTEGER ARRAY LCYLI(2);
000000
000000   % DISC-ERROR INFORMATION FIELD
000000
000000   DISP 0
000000          INTEGER ARRAY DIERCOUNT(4)      % READ-ERROR COUNTER FOR EACH DISC DRIVE
000000          INTEGER ARRAY DIEWCOUNT(4)      % WRITE-ERROR COUNTER FOR EACH DISC DRIVE
000000          INTEGER ARRAY DIEROR(4)         % ACCUMULATED READ-ERROR BITS FOR EACH DISC DRIVE
000000          INTEGER ARRAY DIEWOR(4)         % ACCUMILATED WRITE-ERROR BITS FOR EACH DISC DRIVE
000000          INTEGER ARRAY DIERTMOUT(4)      % NUMBER OF TIMEOUTS ON READ-OPERATIONS
000000          INTEGER ARRAY DIEWTMOUT(4)      % NUMBER OF TIMEOUTS ON WRITE-OPERATIONS
000000          INTEGER ARRAY DIERRTRY(4)       % NUMBER OF DRIVER RETRIES ON READ-OPERATIONS
000000          INTEGER ARRAY DIEWRTRY(4)       % NUMBER OF DRIVER RETRIES ON WRITE-OPERATIONS
000000          INTEGER ARRAY DIECORCU(4)       % NUMBER OF ERROR-CORRECTIONS IN DRIVER
000000          INTEGER ARRAY DIESPACO(4)       % NUMBER OF TIMES REALLOCATED TRACKS ARE USED
000000          INTEGER ARRAY DIUEXRG(4)        % SAVED X-REG FOR EACH DRIVE ON ERROR RETURN FROM DRIVER
000000          INTEGER ARRAY DIUETRG(4)        % SAVED T-REG FOR EACH DRIVE ON ERROR RETURN FROM DRIVER
000000
000000   PSID
000000
000000   % DISC SPECIFICATION TABLE
000000
000000   DISP 0
000000          INTEGER SECWO                   % WORDS / SECTOR
000000          INTEGER SECTR                   % SECTORS / TRACK
000000          INTEGER SECSY                   % SECTORS / CYLINDER
000000          INTEGER MAXCY                   % VALUE OF MAX CYLINDER
000000          INTEGER POLSY                   % VALUE OF FIRST CYLINDER IN POOL
000000          INTEGER REFOR                   % FORMAT TYPE
000000                % ECC CONTROL BITS:
000000                % 5:  4:  3:  CODE:
000000                % 0   0   0   000        CHECK FIRST ADDRESS
000000                % 0   0   1   010        CHECK ALL ADDRESSES
000000                % 0   1   0   020        DO. + 1.0KB SECTOR REALLOCATING
000000                % 0   1   1   030        DO. + 0.5KB SECTOR REALLOCATING
000000                % 1   0   0   100        READ CDC MEDIA INFO
000000          INTEGER RESCY                   % VALUE OF FIRST RESERVED CYLINDER
000000          INTEGER ALTFO                   % POINTER TO ALTERNATIVE FORMAT
000000          INTEGER DISPN                   % INDEX NUMBER OF THIS ENTRY
000000
000000   PSID
000000
000000
000000   % EXTENSIONS TO MASS STORAGE DATAFIELD (CTRMAGT E.A).
000000
000000   DISP -60
000000          INTEGER POINTER MTLRG           % WORKING LOCATION FOR CTRMAGT
000000          INTEGER CMTRECSIZE              % SIZE OF RECORDS IN MULTIPLE REA/WRITE FUNCS
000000          INTEGER MRECCOUNTER             % NUMBER OF RECORDS READ/WRITTEN
000000          INTEGER NMTRECS                 % NUMBER OF RECORDS TO READ/WRITE IN MULTIPLE READ/WRITE FUNCS.
000000          INTEGER CMWCNT                  % NUMBER OF ERASE-GAPS TO WRITE WHEN ERROR IN WRITE OPERATION
000000          INTEGER CTACNS                  % RETRY COUNTER
000000          INTEGER ARRAY ADNSTY(4)         % ACTUAL DENSITY/PARITY (1 LOC. PER UNIT)
000000          INTEGER ARRAY SHSTAT(4)         % SAVE HARDWARE STATUS (1 LOC. PER UNIT)
000000          INTEGER CERRCODE                % CURRENT ERROR CODE
000000          INTEGER MAXUNIT                 % HIGHEST UNIT NO. IMPLEMENTED
000000          INTEGER MACOU                   % ERASE COUNTER
```

```
000000          INTEGER POINTER MRETURN          % ADDRESS FOR RETURNING READ WORDS
000000          INTEGER MWRING                   % WRITE RING BIT
000000          INTEGER MWSTAT                   % STATUS WHEN WRITE RING PRESNT
000000          INTEGER MLOAD                    % LOAD POINT STATUS
000000          INTEGER POINTER CLRG             % SAVED L-REG
000000          INTEGER MWCNT=COMFL              % NUMBER OF TIMES TO ERASE TAPE
000000          DOUBLE POINTER DMRETURN=MRETURN
000000     PSID
000000     DISP 21
000000          INTEGER BADTAPE                  % STATUS BITS FOR BAD TAPE
000000          INTEGER ARRAY RHSTAT(4)          % ERROR STATUS IN READ OPERATIONS
000000          INTEGER ARRAY RERRCOUNT(4)      % READ-ERROR COUNTER
000000          INTEGER ARRAY WHSTAT(4)          % ERROR STATUS IN WRITE OPERATIONS
000000          INTEGER ARRAY WERRCOUNT(4)      % WRITE-ERROR COUNTER
000000     PSID
000000
000000    %%      EXTENSION TO DMA-FIELD FOR STC-MAGTAPE
000000
000000    DISP 42
000000            INTEGER ARRAY FCST (15)          % FULL STC-STATUS
000000            INTEGER ARRAY XUNIT (4)          % POINTERS TO INBT-DATAFIELDS
000000            INTEGER XNOWUNIT                 % CURRENT INBT-DATAFIELD
000000            INTEGER UNRSTATUS                % LAST RETURN FLAG
000000            INTEGER NEWUN                    % NEW UNIT TO BE SELECTED
000000            INTEGER SELUN                    % SELECTED UNIT
000000            INTEGER ARRAY COMCO (4)          % COMAND-CODE.
000000            INTEGER DWONO                    % BITS 24-16 IN WORD-COUNT.
000000    PSID
000000
000000    %=====================================================================
000000    %       BITS IN FLOPPY FUNCTION DESCRIPTION TABLE
000000    %
000000    SYMBOL  3FLOP = 11 % LEGAL ON FLOPPY
000000    SYMBOL  3STRE = 12 % LEGAL ON STREAMER
000000    SYMBOL  3FLTI = 13 % SHORT FLOPPY TIMEOUT (SFTIM (101)).
000000    SYMBOL  3FRES = 14 % LEGAL ONLY FROM RT-PROGS ON RING 2
000000    SYMBOL  3DOUA = 15 % DOUBLE ADDRESS; BITS 0-5 GIVES NEW FUNCTION TO USE
000000    SYMBOL  3DOUB = 16 % DOUBLE AMOUNT; BITS 0-5 GIVES NEW FUNCTION TO USE
000000    SYMBOL  3ILLF = 17 % ILLEGAL FUNCTION
000000
000000    %=========================================================================
000000    %%  DATA FIELD FOR THE OLD FLOPPY CONTROLLER
000000
000000    DISP -156
000000
000000            INTEGER CFLUN          % CURRENT ACTIVE FLOPPY UNIT
000000            INTEGER FDIST          % TEMPORARY HARDWARE STATUS
000000            INTEGER FRTRY          % RETRY COUNTER FOR ADDRESS MISMATCH
000000            INTEGER DINCR          % NO. OF SECTORS TRANSFERRED
000000            INTEGER FLMQU          % LINK FOR MONITOR QUEUE (MLINK)
000000            INTEGER FLMFA          % MONITOR LEVEL FUNCTION ADDR (MFUNC)
000000            INTEGER FRTPR          % RT PROGRAM TO FILL OR EMPTY THE BUFFER
000000            INTEGER WDCNT          % WORD AND/OR SECTOR COUNT
000000            INTEGER FCNT1          % COUNTER
000000            INTEGER FCNT2          % COUNTER
000000            INTEGER OPFLG          % ROUTINE SWITCH IN DRIVER
000000            INTEGER DATAF          % CURRENT FORMAT NO
000000            INTEGER SECT           % SECTOR ADDR (STARTS AT 1)
000000            INTEGER SCTIB          % NO. OF SECTORS IN THE BUFFER
000000            INTEGER OLDTR(3)       % OLD TRACK NO. (3 UNITS)
```

```
000000          INTEGER NEWTR(3)        % NEW TRACK NO. (3 UNITS)
000000          INTEGER MSCIB(3)        % MAX. NO. OF SECTORS ALLOWED IN THE BUFFER (3 FORMATS)
000000          INTEGER SCTTR(3)        % SECTORS PER TRACK (3 FORMATS)
000000          INTEGER CALIB(3)        % CALIBRATION FLAG (3 UNITS)
000000          INTEGER FDRIV           % CURRENT UNIT NO.
000000   PSID
000000
000000   %==========================================================================
000000   %%  DATA FIELD FOR THE NEW FLOPPY CONTROLLERS
000000
000000   DISP -155
000000   %%     INTEGER CFLUN                   % CURRENT ACTIVE FLOPPY UNIT
000000          INTEGER SFCOM
000000
000000          INTEGER ARRAY DOORL(4)          % DOOR LOCK (4 UNITS)
000000          INTEGER ARRAY STEPR(4)          % STEP RATE (4 UNITS). FOR NEW DRIVE AND NEW CONTROLLER, IT COULD BE 0.
000000          INTEGER ARRAY PRECP(4)          % DISABLE PRECOMP (4 UNITS)
000000          INTEGER ARRAY DTRCK(4)          % DUAL TRACK DENSITY (FUTURE)
000000
000000   %      THE STRUCTURE OF THE CCB (COMMAND CONTROL BLOCK )
000000   %      (ALSO CALLED COMMAND FIELD)
000000
000000          INTEGER CCBWO                   % COMMAND WORD
000000          INTEGER BFDEV                   % DEVICE ADDRESS
000000          INTEGER POINTER FMEMH           % MEMORY ADDRESS ( HIGH )
000000          INTEGER POINTER FMEML           % MEMORY ADDRESS ( LOW )
000000          DOUBLE  FMEMD=FMEMH             %
000000          INTEGER OPWCH                   % OPTIONS AND WORD COUNT ( HIGH )
000000          INTEGER WCOUN                   % WORD/RECORD COUNT ( LOW )
000000          INTEGER FSTA1                   % STATUS 1
000000          INTEGER FSTA2                   % STATUS 2
000000          INTEGER LASMH                   % LAST MEMORY ADDRESS ( HIGH )
000000          DOUBLE  LAMAD=LASMH             %
000000          INTEGER PYERA=LASMH             % PHYSICAL ADDRESS OF ERROR IN CHECK FLOPPY
000000          INTEGER LASML                   % LAST MEMORY ADDRESS ( LOW )
000000          INTEGER MREMW                   % MOUST REMAINING WORDS
000000          DOUBLE  REMWO=MREMW             %
000000          INTEGER LREMW                   % LEAST REMAINING WORDS
000000
000000   %      END OF THE COMMAND CONTROL BLOCK
000000
000000          INTEGER ARRAY FCOMF=CCBWO       % COMMAND FIELD/FLOPPY DMA:
000000                                          %    0: COMMAND WORD
000000                                          %    1: DISK ADDR
000000                                          %    2: UNUSED / MEM ADDR HI BYTE
000000                                          %    3: MEM ADDR
000000                                          %    4: UNUSED / WORD COUNT HI
000000                                          %    5: WORD COUNT LO / SECTOR COUNT
000000                                          %    6: STATUS 1
000000                                          %    7: STATUS 2
000000                                          %   10: UNUSED / LAST MEM ADDR HI
000000                                          %   11: LAST MEM ADDR
000000                                          %   12: UNUSED
000000                                          %   13: UNUSED
000000
000000   PSID
000000
000000   %%  COMMON PART OF FLOPPY DATA FIELD
000000
000000   DISP -120
```

```
000000
000000          INTEGER POINTER FRETURN       % ADDR OF THE THIRD PARAMETER (FORMAT NO)
000000          INTEGER FDIMOD                % DISK ADDR MODIFIER
000000          INTEGER ARRAY FDIFORM(4)      % HARDWARE FORMAT FOR EACH UNIT
000000          INTEGER ARRAY LFADDR(20)      % LAST DISK ADDR FOR EACH FORMAT
000000          INTEGER ARRAY WDSCT(20)       % WORDS PER SECTOR (20 FORMATS)
000000          INTEGER ARRAY NFDIADR(4)      % NEXT DISK ADDR FOR EACH UNIT
000000   PSID
000000
000000
000000   DISP -33
000000
000000          INTEGER POINTER CFLRG         % RETURN ADDRESS INSIDE TFDIS
000000   PSID
000000
000000   %%     EXTENSION TO INBT-DATAFIELD FOR STC-MAGTAPE
000000
000000   DISP -3
000000          INTEGER NRDYF                   % FLAG FOR DRIVE REWINDING
000000          INTEGER NRDTR                   % TIMEOUT-COUNTER FOR DO.
000000   PSID
000000
000000   %CONCT FIELD:
000000   DISP -3; INTEGER  DCNRT; PSID
000000
000000   % DATA FIELD FOR BLOCK I/O TRANSFER (DF-DATAFIELD)
000000   DISP 20
000000   INTEGER IOLOG                 % LOGICAL NO.
000000   INTEGER WFLAG                 % ORIGINATOR WAITING FLAG
000000   INTEGER ICORAD                % MEMORY ADDRESS
000000   INTEGER IBLOAD                % BLOCK ADDRESS
000000   INTEGER IMAXW                 % NUMBER OF WORDS
000000   INTEGER IFUNC                 % FUMCTION CODE
000000   INTEGER POINTER IRETW         % RETURNED RECORD LENGTH(MAGTAPE)
000000   INTEGER MTFLG                 % FLAG WORD
000000          SYMBOL MCMAGTP=0       % SET IF MON 144
000000          SYMBOL MC5SPRT=1       % SET IF RFILE/WFILE AND MAGTP FROM ND-500 RT.PROCESS
000000   INTEGER POINTER MRSTA         % START ADDRESS OF ROUTINE
000000   INTEGER SSREF                 % ORIGINATOR
000000   INTEGER STRSEG                % SEGMENTS OF ORIGINATOR
000000   INTEGER DRT  % PROCESSING RT-PROGRAM
000000   INTEGER POINTER MCLRG                % RETURN ADDRESS FOR MC144
000000   PSID
000000   SYMBOL 5MRSTA=30
000000
000000
000000   % DF-DATAFIELDS FOR NORD NET           (8 RESP. CHANNEL DATAFIELD)
000000   DISP 126; INTEGER DFDCR; PSID          % POINTS TO DF-DATAF. INPUT
000000   DISP 106; INTEGER DFDCW; PSID           % POINTS TO DF-DATAF. OUTPUT
000000
000000
000000   SYMBOL XON=21                 % ASCII-CHAR FOR RESTARTING DATA-STREAM.
000000   SYMBOL XOFF=23                % ASCII-CHAR FOR STOPPING DATA-STREAM
000000
000000   INTEGER DVTAB=?,MAXCA=?       % CAMAC
000000
000000   %=====================================================================
000000   %            T I M E S L I C E R   D E F I N I T I O N S
000000   %
000000
```

```
000000   SYMBOL 7CUEL=0                  % BITS 0-4: CURRENT TIMESLICE ELEMENT
000000   SYMBOL 5SPRF=6                  % SPECIAL SYSTEM PRIORITY (ANTIJAM)
000000   SYMBOL 5BRKF=7                  % BREAK PRIORITY REQUESTED
000000   SYMBOL 5ESCF=10                 % ESCAPE PRIORITY REQUESTED
000000   SYMBOL 5NOSLICE=11              % PROCESS NOT TIMESLICED
000000          SYMBOL K5NOSLICE=1000    % 5NOSLICE
000000   SYMBOL 7CUTY=12                 % BITS 12-14 IS CURRENT SLICE TYPE
000000   SYMBOL 7SATY=15                 % BITS 15-17 IS SAVED SLICE TYPE
000000
000000   %=====================================================================
000000   %    B A C K R O U N D   P R O G R A M   A L L O C A T I O N   S Y S T E M
000000   %
000000   %=====================================================================
000000
000000   % DESCRIPTION OF AN ELEMENT IN SBPRTAB
000000
000000   DISP 0
000000   INTEGER BBPROC                  % ADDRESS OF RT-DESCRIPTION OF BACKGROUND PROGRAM
000000   INTEGER CBPTERM                 % ADDRESS OF TERMINAL (TAD) DATAFIELD CONNECTED TO THIS BACKGROUND PROGRAM
000000   INTEGER BPRFLG                  % FLAG WORD
000000   INTEGER IMCBPTERM               % IMAGE OF CBPTERM
000000   INTEGER IMBPRFLG                % IMAGE OF BPRFLG
000000   INTEGER BTBINDX                 % INDEX FOR THIS TERMINAL (TAD) IN BACKGROUND TABLE
000000   INTEGER BPLOGDV                 % LOGICAL DEVICE NUMBER OF TERMINAL (TAD) CONNECTED TO THIS BACKGROUND PROGRA
000000   INTEGER TTMCOUNT                % TIMEOUT COUNTER
000000   INTEGER 9IVAL                   % SUM OF HENTE AND FYLLE IN INPUT DATAFIELD
000000   INTEGER 9OUVAL                  % SUM OF HENTE AND FYLLE IN OUTPUT DATAFIELD
000000          DOUBLE DIOUVAL=9IVAL
000000   INTEGER 1DTU1,1DTU2             % ND-100 TIME USED
000000          DOUBLE 1TUSED=1DTU1
000000   INTEGER 5DTU1,5DTU2             % ND-500 TIME USED
000000          DOUBLE 5TUSED=5DTU1
000000   PSID
000000   % BITS IN BPRFLG
000000   SYMBOL BPRTMOUT=0               % TIMEOUT ALLOWED
000000   SYMBOL BPCFIXED=1               % PERMANENT CONNECTION BETWEEN TERMINAL (TAD) AND BACKG.PROG.
000000   SYMBOL BPWARNING=2              % FLAG THAT WARNING MESSAGE IS GIVEN
000000   SYMBOL BPSOK=3                  % FLAG MARKING THAT THE BACKGROUND PROGRAM IS INITIALIZED
000000   SYMBOL BTYPRMASK=60             % BACKG.PROG. ALLOCATION FOR TAD AND TERMINAL
000000   SYMBOL BPRTSIZE=16              % ELEMENT SIZE
000000   SYMBOL BPRCLMSK=62              % MASKE FOR BPRFLG
000000
000000   % SYMBOLS USED IN "LDATX" INSTRUCTIONS
000000   * TXBBP=BBPRO@3
000000   * TXCBP=CBPTE@3
000000   * TXBPR=BPRFL@3
000000   * TXICB=IMCBP@3
000000   * TXIBP=IMBPR@3
000000
000000   % DESCRIPTION OF ELEMENT IN SAVED-SEGMENT-INFO TABLE P R V T T A B L E
000000   %
000000   % THERE IS ONE ELEMENT IN PRVTTABLE FOR EACH ELEMENT IN BACKGROUND TABLE (BACKT)
000000   %
000000   DISP 0
000000   INTEGER SVBPRFLG                % PRIVATE BPRFLG
000000   INTEGER SVTSLSTATUS             % TIMESLICE STATUS
000000   INTEGER PRVUEFLG                % SAVED UEFLG
000000   INTEGER PRVMAXCT                % SAVED MAXCT (MAX ENTER COUNT)
000000   INTEGER PRVFLLIPCOM             % SAVED FLLIPCOM (DEFAULT SUBSYSTEM ENABLE/DISABLE)
000000   INTEGER PRVFBGTERM              % SAVED FLBGTERM (TERMINATION HANDLING ENABLE/DISABLE)
```

```
000000   INTEGER VSVBPRFLG              % SAVED SVBPRFLG
000000   INTEGER VSVTSLSTATUS           % SAVED SVTSLSTATUS
000000   PSID
000000
000000   SYMBOL 5PRVTSIZE=10            % ELEMENT SIZE
000000
000000   % SYMBOLS USED IN "LDATX" INSTRUCTIONS
000000
000000   * TXUEF=PRVUE@3
000000   * TXFLL=PRVFL@3
000000   * TXXSV=VSVBP@3
000000   * TXSVB=SVBPR@3
000000   * TXSVT=SVTSL@3
000000
000000   %==========================================================================
000000   %      D I S C   L O G   D A T A F I E L D
000000   %
000000   DISP 12
000000   INTEGER DILBPNT                % BUFFER POINTER FOR DISC-LOG
000000   INTEGER DILBANK                % MEMORY BANK FOR DISC-LOG BUFFER
000000   INTEGER 2DIBADDR               % 16 LOWER PHYSICAL MEMORY ADDR BITS OF DISC-LOG BUFFER
000000   DOUBLE DDIBADDR=DILBANK
000000   INTEGER DIL1DADDR,DIL2DADDR
000000   DOUBLE DILDADDR=DIL1DADDR      % START ADDR OF DISK LOG FILE (IN DISK ADDR)
000000   INTEGER DILNSEC                % NUMBER OF SECTORS PER PAGE ON "DISK LOG FILE" DISK
000000   INTEGER DIL1LDADDR,DIL2LDADDR
000000   DOUBLE DILLADDR=DIL1LDADDR     % LAST LEGAL DISK ADDR ON DISK LOG FILE
000000   INTEGER DILGFLAG               % DISK LOG FLAGS
000000   INTEGER DILFLOG                % LOGICAL DEVICE NUMBER OF DISK LOG FILE DISK
000000   INTEGER DILFUNIT               % DRIVE NUMBER FOR DISK LOG FILE DISK DRIVE
000000   DOUBLE DDILFLOG=DILFLOG
000000   INTEGER DLLOGDV                % LOGICAL UNIT NUMBER OF DISC TO LOG
000000   INTEGER DLDRIVE                % DRIVE NUMBER OF DISC TO LOG
000000   INTEGER DLALOGDV               % DISC CONTROLLER TO COUNT DISC ACCESSES ON
000000   INTEGER DLAUNIT                % DISC UNIT NUMBER TO COUNT DISC ACCESSES ON
000000   INTEGER DIL1FADDR,DIL2FADDR
000000   DOUBLE DILFADDR=DIL1FADDR      % FIRST DISC ADDRESS TO LOG
000000   INTEGER DI1LADDR,DI2LADDR
000000   DOUBLE DILGLADDR=DI1LADDR      % LAST DISC ADDRESS TO LOG
000000   INTEGER 1XNDACCESS,2XNDACCESS
000000   DOUBLE DXNDACCESS=1XNDACCESS    % DISC ACCESS COUNTER
000000   INTEGER 1XNWDACCESS,2XNWDACCESS
000000   DOUBLE DXNWDACCESS=1XNWDACCESS % WRITE IDSC ACCESS COUNTER
000000   INTEGER DALFUNC                % FUNCTION CODE IN MON ABSTR FOR DISC-ACCESS-LOG PROG
000000   DOUBLE DALCMADDR               % MEMORY ADDR IN MON ABSTR
000000   DOUBLE DALCDADDR               % DISC ADDR IN MON ABSTR
000000   PSID
000000
000000   % SYMBOLS IN DILGFLAG
000000   @ICR;
000000   SYMBOL DIL1CONTROLLER, % BIT 0: LOG ONLY FOR ONE CONTROLLER
000000          DIL1UNIT,       % BIT 1: LOG ONLY FOR ONE UNIT (DRIVE) NUMBER
000000          DILLIMIT,       % BIT 2: LOG ONLY A PART OF THE DISC
000000          DILRACCESS,     % BIT 3: LOG ONLY READ ACCESSES
000000          DILWACCESS,     % BIT 4: LOG ONLY WRITE ACCESSES
000000          DILSMALL,       % BIT 5: WRITE SMALL RECORD (4 WORDS) ON DISC LOG FILE
000000          DAC1CONTROLLER, % BIT 6: DISC ACCESS COUNTER FOR ONLY ONE CONTROLLER
000000          DAC1UNIT,       % BIT 7: DISC ACCESS COUNTER FOR ONLY ONE  UNIT NUMBER.
000000          DILSTART,       % BIT 10: DISC LOG STARTED
000000          DILDEFINED,     % BIT 11: DISC LOG FILE DEFINED
```

```
000000          1DILBFULL,      % BIT 12: DISC LOG FILE BUFFER #1 IS FULL
000000          2DILBFULL,      % BIT 13: DISC LOG FILE BUFFER #2 IS FULL
000000          DILCOUNT,       % BIT 14: COUNT DISC ACCESSES
000000          DILBOK;         % BIT 15: DISC LOG BUFFER IS FIXED
000000   @CR;
000000
000000   %=================================================================
000000   %% UDMA INPUT DATA FIELD LAYOUT
000000   %
000000
000000   SYMBOL LUDV=2100      % LOGICAL DEVICE NUMBER OF FIRST UNIVERSAL DMA (OR VICOM)
000000   SYMBOL HUDV=2117      % LOGICAL DEVICE NUMBER OF LAST UNIVERSAL DMA (OR VICOM)
000000
000000   % DISP -37; INTEGER  MRETURN;  PSID          % ADDRESS FOR RETUR PARAMETER
000000   % DISP -35; INTEGER  BUSFL;    PSID          % 0= IDLE, 1=DMA , 2=PIO IN ,3=PIO OUT
000000                                                 % 3= WAIT ON INTERUPT
000000   %%
000000     DISP -34; INTEGER  NOWFL;    PSID          % -1 = NO WAIT (FUNC 2 AND 3)
000000     DISP -30; INTEGER  NOWH;     PSID          % NO OF WORD TRANSFERD UPPER BITS
000000     DISP -27; INTEGER  NOWL;     PSID          % NO OF WORD TRANSFERD LOWER BITS
000000   % DISP -26; INTEGER  CTRG;     PSID          % FUNCTION CODE
000000   % DISP -24; INTEGER  CDRG;     PSID          % WORDCOUNT UPPER BIT OR PIO UCLIN MASK
000000   % DISP -23; INTEGER  CXRG;     PSID          % WORDCOUNT LOWER BIT OR PIO OUTPUT DATA
000000     DISP -22; INTEGER  ATTNI;    PSID          % =1 ATTENTION INTERUPT ENABLET
000000     DISP -20; INTEGER  USTAT;    PSID          % UNMASKED STATUS REG.
000000     DISP -17; INTEGER  INDAT;    PSID          % PIO INPUT DATA
000000     DISP -16; INTEGER  UCLIN;    PSID          % USER CONTROLL LINES
000000     DISP -15; INTEGER  ATINT;    PSID          % >< 0 IF ATTENTION INTERUPT HAS OCCURED
000000                                                 % BUT NOT RECIVED BY USER
000000   %%
000000     DISP -14; INTEGER  SOFTACT;  PSID          % SOFTWARE ACTIVATED FLAG
000000
000000   %% UDMA OUTPUT DATA FIELD LAYOUT
000000     DISP 21; INTEGER N5RDF; PSID    % POINTER TO DATAFIELD FOR UDR01  (RWRT. PROG)
000000     DISP 22; INTEGER DFPNT; PSID    % POINTER TO WORKING AREA FOR MON 333
000000
000000
000000   %=========================================================================
000000   % DATAFIELD LAYOUT FOR GPIB DRIVER  COMMON PART
000000
000000   DISP -17
000000     INTEGER POINTER GPXTR        % START ON DRIVER LEVEL FOR XMSG TRANSFER ROUTINE
000000     INTEGER GPDZI                % DMA BUFFER SIZE
000000     INTEGER GPUZI                % COMMAND-MESSAGE SIZE
000000     INTEGER GPRUN                % GPIB RUN FLAG
000000     INTEGER GPBFL                % BUSY FLAG
000000   PSID
000000
000000   %=========================================================================
000000   % DATAFIELD FOR OCTOBUS
000000   DISP  7
000000          INTEGER BOXNO                % POST BOX NUMBER FOR THIS SLOT (LOG-DEV)
000000          INTEGER MAINF                % ADDRESS OF OCTOBUS DEVICE FIELD
000000   PSID
000000
000000
```

```
000000
000000   %=================================================================================================
000000   %      C O M M U N I C A T I O N   D A T A F I E L D   D E F I N I T I O N S
000000   %=================================================================================================
000000
000000   %=================================================================================================
000000   %      L I N E   D A T A F I E L D   I N P U T :   T T I L (N)
000000
000000   DISP -  7; INTEGER                CCTRL      ;PSID  % HARDWARE CONTROL WORD
000000   %%%% -  6  INTEGER POINTER        TMSUB             % TIME-OUT SUBROUTINE
000000   %%%% -  5  INTEGER                TMR               % TIME-OUT COUNTER
000000   %%%% -  4  INTEGER                TTMR              % TIME-OUT TIME (-SECONDS)
000000   %%%% -  3  INTEGER                HDEV              % IOX INSTRUCTION
000000   %%%% -  2  INTEGER                STDRIV            % START ADDRESS OF DRIVER
000000   %%%% -  1  INTEGER                DRIVER            % RESTART ADDRESS OF DRIVER
000000
000000   %%%%    T T I L (N)     E N T R Y
000000
000000   %%%%    0  INTEGER                RESLINK           % RESERVATION LINK
000000   %%%%    1  INTEGER                RTRES             % RESERVING RT-PROGRAM
000000   %%%%    2  INTEGER                BWLINK            % BEGINNING OF WAITING QUEUE
000000   %%%%    3  INTEGER                TYPRING           % DEVICE TYPE BITS AND RING
000000   %%%%    4  INTEGER                ISTATE            % 0=IDLE, 1=BUSY, -1 NO WAIT MODE
000000   %%%%    5  INTEGER                MLINK             % MONITOR QUEUE LINK
000000   %%%%    6  INTEGER                MFUNC             % MONITOR LEVEL FUNCTION ADDRESS
000000   DISP    7; INTEGER                CHKO       ;PSID  % CHARACTER POINTER FOR INPUT DRIVER
000000   %%%%   10  INTEGER POINTER        STDEV             % START DEVICE ROUTINE
000000   DISP   11; INTEGER                ANCHA      ;PSID  % CHARACTER COUNTER FOR INPUT DRIVER
000000   %%%%   12  INTEGER                DFOPP             % POINTER TO OUTPUT DATAFIELD
000000   DISP   13; INTEGER POINTER        IINI       ;PSID  % I/O INITIATING ROUTINE
000000   DISP   14; INTEGER                LIBEG      ;PSID  % START OF BUFFER DESCRIPTOR LIST                -- (HD
000000   DISP   14; INTEGER                NPFBUF     ;PSID  % CURRENT BUFFER CHAIN USED BY INPUT DRIVER       -- (E
000000   DISP   15; INTEGER                SWICH      ;PSID  % GO SWITCH FOR INPUT DRIVER
000000   DISP   16; INTEGER POINTER        HPEK       ;PSID  % FETCH POINTER TO BUFFER DESCRIPTOR LIST        -- (H
000000   DISP   16; INTEGER                NPHBUF     ;PSID  % CURRENT BUFFER CHAIN READ BY RECEIVE RT PROGRAM -- (E
000000   DISP   17; INTEGER                CUDBU      ;PSID  % CURRENT BUFFER FILLED BY DRIVER
000000   DISP   20; INTEGER                LASBU      ;PSID  % LAST BUFFER IN FRAME WHEN FETCHING FROM LIST    -- (H
000000   DISP   20; INTEGER                SOHSC      ;PSID  % SEARCH COUNTER FOR START OF HEADER             -- (S
000000   DISP   21; INTEGER ARRAY          INIAD(7)   ;PSID  % INITIALIZING PARAMETERS FOR INTERFACE          -- (H
000000   DISP   21; INTEGER ARRAY          IDBSTI(4)  ;PSID  % STATUS OF BUFFER CHAIN N (0=EMPTY, 6=FILLED)   -- (E
000000   DISP   25; INTEGER ARRAY          IDBAD(4)   ;PSID  % ADDRESS OF FIRST BUFFER IN CHAIN N             -- (E
000000   DISP   30; INTEGER ARRAY          LIPOI(124);PSID   % BUFFER DESCRIPTOR LIST                         -- (H
```

```
000000
000000   %=================================================================================================
000000   %      L I N E   D A T A F I E L D   O U T P U T :    T T U L (N)
000000
000000   DISP -  4; INTEGER                  OCTRL      ;PSID  % OUTPUT HARDWARE CONTROL WORD
000000   %%%% -  3  INTEGER                  HDEV              % IOX INSTRUCTION
000000   %%%% -  2  INTEGER                  STDRIV            % START ADDRESS OF OUTPUT DRIVER
000000   %%%% -  1  INTEGER                  DRIVER            % RESTART ADDRESS OF OUTPUT DRIVER
000000
000000   %%%%    T T U L (N)      E N T R Y
000000
000000   %%%%    0  INTEGER                  RESLINK           % RESERVATION LINK
000000   %%%%    1  INTEGER                  RTRES             % RESERVING RT-PROGRAM
000000   %%%%    2  INTEGER                  BWLINK            % BEGINNING OF WAITING QUEUE
000000   %%%%    3  INTEGER                  TYPRING           % DEVICE TYPE BITS AND RING
000000   %%%%    4  INTEGER                  ISTATE            % 0=IDLE, 1=BUSY, -1=NO WAIT MODE
000000   %%%%    5  INTEGER                  MLINK             % MONITOR QUEUE LINK
000000   %%%%    6  INTEGER                  MFUNC             % MONITOR LEVEL FUNCTION ADDRESS
000000   %%%%    7  INTEGER                  CHKO              % CHARACTER POINTER IN OUTPUT BUFFER
000000   %%%%   10  INTEGER POINTER          STDEV             % START DEVICE ROUTINE
000000   %%%%   11  INTEGER                  ANCHA             % CHARACTER COUNTER FOR OUTPUT DRIVER
000000   %%%%   12  INTEGER                  DFOPP             % POINTER TO INPUT DATAFIELD
000000   DISP   13; INTEGER POINTER          RSICH      ;PSID  % RESET DEVICE ROUTINE
000000   %%%%   14  INTEGER                  BUFST             % POINTER TO FIRST BUFFER OF CURRENT OUTPUT BUFFER CHAI
000000   %%%%   15  INTEGER                  SWICH             % GO SWITCH FOR OUTPUT DRIVER
000000   DISP   16; INTEGER                  LMDAT      ;PSID  % POINTER TO CMO DATAFIELD
000000   %%%%   17  INTEGER                  CUDBU             % CURRENT OUTPUT BUFFER
000000   DISP   20; INTEGER ARRAY            SLIPO(25) ;PSID   % BUFFER DESCRIPTOR FOR HDLC          -- (HDLC-DMA)
```

```
000000
000000  %=================================================================================================
000000  %       L I N E   D A T A F I E L D :       C M O (N)
000000
000000  DISP - 24; INTEGER                    ANACS     ;PSID  % NUMBER OF CONSECUTIVE NACKS SENT BEFORE COM DEAD
000000  DISP - 23; INTEGER                    AFRET     ;PSID  % NUMBER OF CONSECUTIVE RETRANSMISSIONS BEFORE COM DEAD
000000  DISP - 22; INTEGER                    FCRCE     ;PSID  % NUMBER OF FRAMES WITH CRC ERROR RECEIVED
000000  DISP - 21; INTEGER                    REMTI     ;PSID  % REMAINING TIME OF TIMEOUT
000000  DISP - 20; INTEGER                    RSELR     ;PSID  % TEMPORARY SAVE OF L-REG IN RSEND
000000  DISP - 17; INTEGER                    TBUSA     ;PSID  % TEMPORARY SAVE OF BUFFER-ADDRESS
000000  DISP - 16; INTEGER                    RLTSAV    ;PSID  % SAVED SEGMENT NUMBERS IN REMOTE-LOAD
000000  DISP - 15; INTEGER POINTER            RLLSAV    ;PSID  % SAVED L-REG IN REMOTE-LOAD
000000  %%%% - 14  INTEGER                    TSPEED           % LINE-SPEED CODE
000000  DISP - 13; INTEGER POINTER            TMLRE     ;PSID  % SAVED L-REG IN TMSUB
000000  DISP - 12; DOUBLE                     SIAD      ;PSID  % SAVE AD-REG WHILE COMPUTING CRC ON INPUT
000000  DISP - 10; DOUBLE                     SOAD      ;PSID  % SAVE AD-REG WHILE COMPUTING CRC ON OUTPUT
000000  %%%% -  6  INTEGER POINTER            TMSUB            % TIME-OUT ROUTINE
000000  %%%% -  5  INTEGER                    TMR              % COUNTER FOR TIME-OUT
000000  %%%% -  4  INTEGER                    TTMR             % TIME-OUT TIME (-SECONDS)
000000  DISP -  3; INTEGER                    XTEMI     ;PSID  % SAVE X-REGISTER WHILE COMPUTING CRC ON INPUT
000000  DISP -  2; INTEGER                    XTEMO     ;PSID  % SAVE X-REGISTER WHILE COMPUTING CRC ON OUTPUT
000000  DISP -  1; INTEGER                    ISIT      ;PSID  % FRAME PHASE BIT ON INPUT
000000
000000  %%      C M O (N)    E N T R Y
000000
000000  DISP    0; INTEGER                    GSI       ;PSID  % GROUP NUMBER ON INPUT
000000  DISP    1; INTEGER                    ANI       ;PSID  % STATUS INFORMATION IN INPUT FRAME
000000  DISP    2; INTEGER                    GRI       ;PSID  % GROUP NUMBER FOR ANI
000000  DISP    3; INTEGER                    IRI       ;PSID  % PHASE BIT FOR GRI
000000  %%%%    4  INTEGER                    ISTATE           % SEND STATUS
000000  DISP    5; INTEGER ARRAY              ISI(4)    ;PSID  % LAST RECEIVED PHASE BIT ON GROUP N
000000  DISP   11; INTEGER                    GSO       ;PSID  % CURRENT GROUP NUMBER OUTPUT
000000  DISP   12; INTEGER                    CDFILD    ;PSID  % POINTER TO LINE DATAFIELD INPUT
000000  DISP   13; INTEGER POINTER            SINIT     ;PSID  % INITIALIZING ROUTINE
000000  DISP   14; INTEGER POINTER            BSINIT    ;PSID  % INITIALIZING ROUTINE FOR REMOTE-LOAD
000000  DISP   15; INTEGER                    PLMSG     ;PSID  % LAST FRAME OF SEND QUEUE
000000  DISP   16; INTEGER                    PFMSG     ;PSID  % FIRST FRAME OF SEND QUEUE
000000  DISP   17; INTEGER ARRAY              ISO(4)    ;PSID  % LAST SENT PHASE BIT ON GROUP N
000000  DISP   23; INTEGER POINTER            SEND      ;PSID  % SEND ROUTINE
000000  DISP   24; INTEGER POINTER            BSEND     ;PSID  % SEND BOOTSTRAP ROUTINE FOR REMOTE-LOAD
000000  DISP   25; INTEGER POINTER            RECEIVE   ;PSID  % RECEIVE ROUTINE
000000  DISP   26; INTEGER POINTER            VENTX     ;PSID  % WAIT ROUTINE (HALF DUPLEX)
000000  DISP   27; INTEGER                    RMLNR     ;PSID  % REMOTE LINE NUMBER
000000  DISP   30; INTEGER                    FRETR     ;PSID  % FRAMES RETRANSMITTED
000000  DISP   31; INTEGER                    MISTART   ;PSID  % START OF CURRENT INPUT FRAME
000000  DISP   32; INTEGER                    SBYTS     ;PSID  % NUMBER OF IFIELD BYTES IN OUTPUT FRAME
000000  DISP   33; INTEGER                    CUIBU     ;PSID  % CURRENT INPUT BUFFER
000000  DISP   34; INTEGER                    CUUBU     ;PSID  % CURRENT OUTPUT BUFFER
000000  DISP   35; INTEGER                    RGSI      ;PSID  % TEMPORARY VARIABLE TO HOLD LAST RECEIVED GROUP NO
000000  DISP   36; INTEGER                    SQERR     ;PSID  % NUMBER OF SEQUENCE ERRORS
000000  DISP   37; INTEGER                    RNACO     ;PSID  % NUMBER OF NAKS RECEIVED
000000  DISP   40; INTEGER                    SNACO     ;PSID  % NUMBER OF NAKS SENT
000000  DISP   41; INTEGER                    CTRCH     ;PSID  %.CONTROLLER AND CHANNEL OF LAST RECEIVED FRAME
000000  DISP   42; INTEGER                    BADANT    ;PSID  % RETRANSMIT FLAG
000000  DISP   43; INTEGER                    SMI       ;PSID  % CRC ON INPUT
000000  DISP   44; INTEGER                    SMO       ;PSID  % CRC ON OUTPUT
000000  DISP   45; INTEGER                    ACQFP     ;PSID  % WRITE POINTER IN STATUS INFORMATION BUFFER
000000  DISP   46; INTEGER                    ACQHP     ;PSID  % READ  POINTER IN STATUS INFORMATION BUFFER
000000  DISP   47; INTEGER                    ACQBH     ;PSID  % NUMBER OF ITEMS IN STATUS INFORMATION BUFFER
000000  DISP   50; INTEGER                    IBYTS     ;PSID  % NUMBER OF IFIELD BYTES IN INPUT FRAME
```

```
000000   DISP   51;  INTEGER                          CURID       ;PSID  % ADDRESS OF CURRENT INPUT BUFFER
000000   DISP   52;  INTEGER                          SPRS        ;PSID  % SEND RT PROGRAM
000000   DISP   53;  INTEGER                          RPRS        ;PSID  % RECEIVE RT PROGRAM
000000   DISP   54;  INTEGER ARRAY                    ACQU(4)     ;PSID  % STATUS INFORMATION BUFFER
000000   DISP   60;  INTEGER POINTER                  ILSAV       ;PSID  % TEMPORARY SAVED L REGISTER INPUT
000000   DISP   61;  INTEGER POINTER                  OLSAV       ;PSID  % TEMPORARY SAVED L REGISTER OUTPUT
000000   DISP   62;  INTEGER                          LINR        ;PSID  % LINE NUMBER
000000   DISP   63;  INTEGER                          DNACO       ;PSID  % NUMBER OF NACKS SENT SINCE LAST ACK SENT
000000   DISP   64;  INTEGER ARRAY                    BUSTA(4)    ;PSID  % OUTPUT STATUS FOR LAST SENT FRAME IN GROUP N
000000   DISP   64;  TRIPLE                           BUALL       ;PSID  % ALTERNATIVE DISP FOR 3 FIRST LOCATIONS IN BUSTA
000000   DISP   67;  INTEGER                          BUSIST      ;PSID  % ALTERNATIVE DISP FOR LAST LOCATION IN BUSTA
000000   DISP   70;  INTEGER ARRAY                    BUFADD(4)   ;PSID  % ADDRESS OF FIRST BUFFER OF OUTPUT FRAME IN GROUP N
000000   DISP   74;  INTEGER                          INHBT       ;PSID  % ADDRESS OF PROTECTED BUFFER (RETRANSMITT)
000000   DISP   75;  INTEGER                          LOST        ;PSID  % GROUP NUMBER FRAME TO RETRANSMITT SET BY RECEPTION OF
000000   DISP   76;  INTEGER ARRAY POINTER            COTAB       ;PSID  % POINTER TO CONFIGURATION TABLE
000000   DISP   77;  INTEGER                          COFLAG      ;PSID  % LINE STATUS 0=COM RUNNING ><0 COM DEAD
000000   DISP  100;  INTEGER                          RETRN       ;PSID  % COUNTER FOR CONSECUTIVE RETRANSMISSIONS
000000   DISP  101;  INTEGER                          AKMCH       ;PSID  % NO OF CHANNELS WITH LOWER NUMBER THAN FIRST CHANNEL 0
000000   DISP  102;  INTEGER                          MXCHN       ;PSID  % MAXIMUM CHANNEL NUMBER ON THIS LINE
000000   DISP  103;  INTEGER ARRAY                    IDADR(1)    ;PSID  % POINTER ARRAY TO THE CHANNEL INPUT DATA FIELDS OF THI
```

```
000000
000000  %=================================================================================================
000000  %       C H A N N E L   D A T A F I E L D   I N P U T :      S (M) (N) R
000000
000000
000000  %%%%  - 17  INTEGER                    CTTYP                % TERMINAL TYPE
000000  %%%%  - 16  INTEGER                    CESCP                % DISCONNECT AND SCAPE CHARACTER
000000  %%%%  - 15  INTEGER                    BRKMAX               % MAXIMUM BHOLD BEFORE BREAK
000000  DISP  - 14; INTEGER                    ANTORD     ;PSID     % NUMBER OF WORDS TO READ FOR MAGTP
000000  DISP  - 13; INTEGER                    XSAC       ;PSID     % SAVE X-REGISTER IN IOTRANS
000000  %%%%  - 12  INTEGER                    DFLAG                % FLAG BITS
000000  %%%%  - 11  INTEGER                    ECHOTAB              % ECHO TABLE
000000  %%%%  - 10  INTEGER                    BRKTAB               % BREAK TABLE
000000  %%%%  -  7  INTEGER                    LAST                 % LAST TYPED CHARACTER
000000  DISP  -  6; INTEGER                    CMDAT      ;PSID     % POINTER TO CMO DATAFIELD
000000  DISP  -  5; INTEGER                    IXSAC      ;PSID     % SAVE DF-DATAFIELD FOR MAGTP
000000  DISP  -  4; INTEGER                    ANTMEL     ;PSID     % NUMBER OF FRAMES IN INPUT QUEUE
000000  DISP  -  3; INTEGER                    BYTS       ;PSID     % BYTE COUNTER IN FRAME
000000  DISP  -  2; INTEGER                    INCR       ;PSID     % BYTE COUNTER IN BUFFER
000000  DISP  -  1; INTEGER                    CHAN       ;PSID     % CHANNEL NUMBER (0-37)
000000
000000  %%%%    S (M) (N) R      E N T R Y
000000
000000  %%%%    0  INTEGER                    RESLINK              % RESREVATION LINK
000000  %%%%    1  INTEGER                    RTRES                % RESERVING RT PROGRAM
000000  %%%%    2  INTEGER                    BWLINK               % BEGINNING OF WAITING QUEUE
000000  %%%%    3  INTEGER                    TYPRING              % DEVICE TYPE BITS AND RING
000000  %%%%    4  INTEGER                    ISTATE               % 0=IDLE, 1=BUSY, -1=NO WAIT MODE
000000  %%%%    5  INTEGER                    MLINK                % MONITOR QUEUE LINK
000000  %%%%    6  INTEGER                    MFUNC                % MONITOR LEVEL FUNCTION ADDRESS
000000  %%%%    7  INTEGER POINTER            IOTRANS              % INPUT IOTRANS ROUTINE
000000  %%%%   10  INTEGER POINTER            STDEV                % START DEVICE ROUTINE
000000  %%%%   11  INTEGER POINTER            SETDV                % IOSET ROUTINE
000000  %%%%   12  INTEGER                    DFOPP                % POINTER TO OUTPUT CHANNEL DATAFIELD
000000  %%%%   13  INTEGER                    DERROR               % ERROR CODE
000000  DISP   14; INTEGER                    MSSTART    ;PSID     % POINTER TO FIRST BUFFER OF FRAME
000000  %%%%   15  INTEGER                    PLMSG                % LAST FRAME OF INPUT QUEUE
000000  %%%%   16  INTEGER                    PFMSG                % FIRST FRAME OF INPUT QUEUE
000000  DISP   17; INTEGER                    CURBU      ;PSID     % CURRENT INPUT BUFFER
000000  DISP   20; INTEGER                    CHNST      ;PSID     % CHANNEL INPUT STATUS
000000  DISP   21; INTEGER POINTER            LRSA       ;PSID     % SAVE L-REGISTER IN IOTRANS
000000  %%%%   22  INTEGER                    BSTATE               % BACKGROUND PROGRAM STATE
000000  %%%%   23  INTEGER                    TSTATE               % TIME SLICE STATE
000000  %%%%   24  INTEGER                    DBPROG               % BACKGROUND RT PROGRAM
000000  %%%%   25  INTEGER                    DBADDR               % SAVED P-REGISTER ON ESCAPE + FILE SYSTEM MONITOR CALL
000000  %%%%   26  INTEGER                    RIFIL                % FOR MODE INPUT FILE NUMBER
000000  %%%%   27  INTEGER                    BCHISTS              % FOR MODE INPUT STATUS
000000  %%%%   30  DOUBLE                     DERO                 % ERROR INFORMATION
000000  %%%%   30  INTEGER ARRAY              BREGBLOCK            % REGISTER SAVE AT ESCAPE
000000  %%%%   32  DOUBLE                     DER2(3)              % ERROR INFORMATION
000000  %%%%   40  INTEGER                    DBPREG               % P-REGISTER ON PAGE FAULT ON IOBT LEVEL
000000  %%%%   41  INTEGER                    DBACTPRI             % ACTPRI ON PAGE FAULT ON IOBT LEVEL
000000  %%%%   42  INTEGER                    FLAGB                % BACKGROUNG FLAG
000000  DISP   43; INTEGER                    RTUT       ;PSID     %.RT DESCRIPTION OF FILE TRANSFER PROGRAM
000000  DISP   44; INTEGER                    MITRG      ;PSID     % SAVE T-REG IN BLOCK INPUT
000000  DISP   45; INTEGER                    MILRG      ;PSID     % SAVE L-REG IN BLOCK INPUT
000000  DISP   46; INTEGER                    LDITR      ;PSID     % SAVE T-REG IN SUBROUTINES CALLED FROM BLOCK INPUT
000000  DISP   47; INTEGER                    LDIXR      ;PSID     % SAVE X-REG IN SUBROUTINES CALLED FROM BLOCK INPUT
000000  DISP   50; INTEGER POINTER            LDILR      ;PSID     % SAVE L-REG IN SUBROUTINES CALLED FROM BLOCK INPUT
000000  DISP   51; INTEGER                    ISSREF     ;PSID     % SSREF  FROM SYSTEM SEGMENT USED BY BLOCK INPUT
000000  DISP   52; INTEGER                    IIFUNC     ;PSID     % IFUNC  FROM SYSTEM SEGMENT USED BY BLOCK INPUT
```

```
000000   DISP   53; INTEGER                    IICORAD   ;PSID  % ICORAD FROM SYSTEM SEGMENT USED BY BLOCK INPUT
000000   DISP   54; INTEGER                    IIMAXW    ;PSID  % IMAXW  FROM SYSTEM SEGMENT USED BY BLOCK INPUT
000000   DISP   55; INTEGER                    IOLDPAG   ;PSID  % OLDPAG FROM SYSTEM SEGMENT USED BY BLOCK INPUT
000000   DISP   56; INTEGER                    IMTFLG    ;PSID  % MTFLG  FROM SYSTEM SEGMENT USED BY BLOCK INPUT
000000
000000   %=================================================================================================
000000   %       C H A N N E L   D A T A F I E L D   O U T P U T :      S (M) (N) W
000000
000000   %%%% - 14  INTEGER                    ANTORD            % NUMBER OF WORDS TO WRITE FOR MAGTP
000000   %%%% - 13  INTEGER                    XSAC              % SAVE X-REGISTER IN IOTRANS
000000   %%%% - 12  INTEGER                    DFLAG             % FLAG WORD
000000   %%%% - 11  INTEGER                    ECHOTAB           % POINTER TO ECHO TABLE
000000   %%%% - 10  INTEGER                    BRKTAB            % POINTER TO BREAK TABLE
000000   %%%% -  7  INTEGER                    LAST              % SAVE LAST BYTE
000000   %%%% -  6  INTEGER                    CMDAT             % POINTER TO CMO DATAFIELD
000000   DISP -  5; INTEGER                    MSIZE     ;PSID   % MAXIMUM NUMBER OF BYTES IN A FRAME
000000   DISP -  4; INTEGER                    SCPRI     ;PSID   % CHANNEL PRIORITY
000000   %%%% -  3  INTEGER                    BYTS              % BYTE COUNTER IN FRAME
000000   %%%% -  2  INTEGER                    INCR              % BYTE COUNTER IN BUFFER
000000   %%%% -  1  INTEGER                    CHAN              % CHANNEL NUMBER (0-37)
000000
000000   %%%%    S (M) (N) W       E N T R Y
000000
000000   %%%%    0  INTEGER                    RESLINK           % RESERVATION LINK
000000   %%%%    1  INTEGER                    RTRES             % RESERVING RT-PROGRAM
000000   %%%%    2  INTEGER                    BWLINK            % BEGINNING OF WAITING QUEUE
000000   %%%%    3  INTEGER                    TYPRING           % DEVICE TYPE BITS AND RING
000000   %%%%    4  INTEGER                    ISTATE            % 0=IDLE, 1=BUSY, -1 NO WAIT MODE
000000   %%%%    5  INTEGER                    MLINK             % MONITOR QUEUE LINK
000000   %%%%    6  INTEGER                    MFUNC             % MONITOR LEVEL FUNCTION ADDRESS
000000   %%%%    7  INTEGER POINTER            IOTRANS           % OUTPUT IOTRANS ROUTINE
000000   %%%%   10  INTEGER POINTER            STDEV             % START DEVICE ROUTINE
000000   %%%%   11  INTEGER POINTER            SETDV             % IOSET ROUTINE
000000   %%%%   12  INTEGER                    DFOPP             % POINTER TO INPUT CHANNEL DATAFIELD
000000   %%%%   13  INTEGER                    DERROR            % ERROR CODE
000000   %%%%   14  INTEGER                    MSSTART           % POINTER TO FIRST BUFFER OF FRAME
000000   DISP   15; INTEGER                    UANTMEL   ;PSID   % NUMBER OF FRAMES IN SEND QUEUE FOR THIS CHANNEL
000000   %%%%   16                                               % NOT USED
000000   %%%%   17  INTEGER                    CURBU             % CURRENT OUTPUT BUFFER
000000   %%%%   20  INTEGER                    CHNST             % CHANNEL OUTPUT STATUS
000000   %%%%   21  INTEGER POINTER            LRSA              % SAVE L-REGISTER IN IOTRANS
000000   DISP   22; INTEGER                    RTIN      ;PSID   % RT DESCRIPTION OF FILE TRANSFER PROGRAM
000000   %%%%   23  INTEGER                    ROFIL             % FOR MODE OUTPUT FILE NUMBER
000000   %%%%   24  INTEGER                    BCHOSTS           % FOR MODE OUTPUT STATUS
000000   DISP   25; INTEGER                    MOTRG     ;PSID   % SAVE T-REGISTER IN BLOCK OUTPUT
000000   DISP   26; INTEGER                    MOLRG     ;PSID   % SAVE L-REGISTER IN BLOCK OUTPUT
000000   DISP   27; INTEGER                    LDOTR     ;PSID   % SAVE T-REG IN SUBROUTINES CALLED FROM BLOCK OUTPUT
000000   DISP   30; INTEGER                    LDOXR     ;PSID   % SAVE X-REG IN SUBROUTINES CALLED FROM BLOCK OUTPUT
000000   DISP   31; INTEGER POINTER            LDOLR     ;PSID   % SAVE L-REG IN SUBROUTINES CALLED FROM BLOCK OUTPUT
000000   DISP   32; INTEGER                    OSSREF    ;PSID   % SSREF  FROM SYSTEM SEGMENT USED BY BLOCK OUTPUT
000000   DISP   33; INTEGER                    OIFUNC    ;PSID   % IFUNC  FROM SYSTEM SEGMENT USED BY BLOCK OUTPUT
000000   DISP   34; INTEGER                    OICORAD   ;PSID   % ICORAD FROM SYSTEM SEGMENT USED BY BLOCK OUTPUT
000000   DISP   35; INTEGER                    OIMAXW    ;PSID   % IMAXW  FROM SYSTEM SEGMENT USED BY BLOCK OUTPUT
000000   DISP   36; INTEGER                    OOLDPAG   ;PSID   % OLDPAG FROM SYSTEM SEGMENT USED BY BLOCK OUTPUT
000000
000000
000000   %=================================================================
000000   %       SYMBOL DEFENITIONS FOR NORD-NET
000000
000000   INTEGER MXLIN=?; INTEGER ARRAY LINARR=?
```

```
000000
000000   % COMMUNICATION BUFFER TYPES
000000   SYMBOL 7DATA,7DBREK,7DECO,7RQIN,7IRQI,7CONF
000000
000000
000000   SYMBOL EMPTY=0,ITOM,FULLN,FULLS,GOOD,BAD,FULL,BUFUL,IGNACK
000000   SYMBOL MINFREE=13,MXMEL=2                % MAX QUEUE LENGTHS
000000   SYMBOL BUSIZ=203,MSTOR=400                                % MAX MESSAGE LENGTH
000000
000000   %=================================================================================
000000   %     T E R M I N A L - A C C E S S - D E V I C E     D A T A F I E L D S
000000   %=================================================================================
000000   %
000000   %      T A D   D A T A F I E L D   I N P U T:      B D (N) R
000000
000000   %%%% - 45  INTEGER                  TINFO                % VARIOUS INFO BITS
000000   %%%% - 44  INTEGER                  PECH7(10)            % ECHO  TABLE 7
000000   %%%% - 34  INTEGER                  PBRK7(10)            % BREAK TABLE 7
000000   %%%% - 24  INTEGER                  IN5MSG               % ADDRESS OF ND-500 MESSAGE IN FAST INSTRING
000000   DISP - 23; INTEGER                  ESCBUF     ;PSID     % BUFFER FOR ESCAPE-RESPONSE
000000   DISP - 22; INTEGER                  TMPBUF     ;PSID     % TEMPORARY BUFFID
000000   DISP - 21; INTEGER                  BADTYP     ;PSID     % TAD-TYPE
000000   %%%% - 20  INTEGER                  NCBRK                % NUMBER OF CHARACTERS AFTER LAST BREAK
000000   %%%% - 17  INTEGER                  CTTYP                % TERMINAL TYPE
000000   %%%% - 16  INTEGER                  CESCP                % DISCONNECT AND ESCAPE CHARACTER
000000   %%%% - 15  INTEGER                  BRKMAX               % MAXIMUM BHOLD BEFORE BREAK
000000   DISP - 14; INTEGER                  NOBUFF     ;PSID     % NUMBER OF XMSG BUFFERS TO USE
000000   DISP - 13; INTEGER                  FBSIZ      ;PSID     % SIZE   OF XMSG BUFFERS TO USE
000000   %%%% - 12  INTEGER                  DFLAG                % FLAG BITS
000000   %%%% - 11  INTEGER                  ECHOTAB              % ECHO TABLE
000000   %%%% - 10  INTEGER                  BRKTAB               % BREAK TABLE
000000   %%%% -  7  INTEGER                  LAST                 % LAST TYPED CHARACTER
000000   %%%% -  6; INTEGER                  TMSUB                % TIME OUT SUBROUTINE
000000   %%%% -  5; INTEGER                  TMR                  % TIME OUT COUNTER
000000   %%%% -  4; INTEGER                  TTMR                 % START VALUE OF TMR
000000   DISP -  3; INTEGER                  PORTNO     ;PSID     % PORT NUMBER OF OPENED PORT
000000   DISP -  2; INTEGER                  DBCOU      ;PSID     % DATA BYTE COUNTER IN INPUT CALLS
000000   %%%% -  1; INTEGER POINTER          DRIVER               % SAVED L-REG IN INPUT DRIVER
000000
000000   %%%%     B D (N) R        E N T R Y
000000
000000   %%%%     0  INTEGER                RESLINK              % RESREVATION LINK
000000   %%%%     1  INTEGER                RTRES                % RESERVING RT PROGRAM
000000   %%%%     2  INTEGER                BWLINK               % BEGINNING OF WAITING QUEUE
000000   %%%%     3  INTEGER                TYPRING              % DEVICE TYPE BITS AND RING
000000   %%%%     4  INTEGER                ISTATE               % 0=IDLE, 1=BUSY, -1=NO WAIT MODE
000000   %%%%     5  INTEGER                MLINK                % MONITOR QUEUE LINK
000000   %%%%     6  INTEGER                MFUNC                % MONITOR LEVEL FUNCTION ADDRESS
000000   %%%%     7  INTEGER POINTER        IOTRANS              % INPUT IOTRANS ROUTINE
000000   %%%%    10  INTEGER POINTER        STDEV                % START DEVICE ROUTINE
000000   %%%%    11  INTEGER POINTER        SETDV                % IOSET ROUTINE
000000   %%%%    12  INTEGER                DFOPP                % POINTER TO OUTPUT CHANNEL DATAFIELD
000000   %%%%    13  INTEGER                DERROR               % ERROR CODE
000000   DISP    14; INTEGER                BUFFID     ;PSID     % XMSG BUFFER IDENTIFIER
000000   DISP    15; INTEGER                TDTAFI     ;PSID     % FIRST PART OF BUFFER ADDRESS
000000   DISP    16; INTEGER                TDTALA     ;PSID     % LAST  PART OF BUFFER ADDRESS
000000   DISP    15; DOUBLE                 TDTADD     ;PSID     % ADDRES OF BUFFER DATA-PART
000000   DISP    17; INTEGER                TDBTPT     ;PSID     % BYTE POINTER IN XMSG BUFFER
000000   DISP    20; INTEGER                XRSA       ;PSID     % SAVE X-REG IN IOTRANS AND INIBDR
000000   %%%%    21  INTEGER POINTER        LRSA                 % SAVE L-REG IN IOTRANS
```

```
000000   %%%%   22  INTEGER                BSTATE            % BACKGROUND PROGRAM STATE
000000   DISP   23; INTEGER                REMBYT     ;PSID  % REMAINING BYTES IN CURMES (NEG. +1)
000000   %%%%   24  INTEGER                DBPROG            % BACKGROUND RT PROGRAM
000000   %%%%   25  INTEGER                DBADDR            % SAVED P-REGISTER ON ESCAPE + FILE SYSTEM MONITOR CALL
000000   %%%%   26  INTEGER                RIFIL             % FOR MODE INPUT FILE NUMBER
000000   %%%%   27  INTEGER                BCHISTS           % FOR MODE INPUT STATUS
000000   %%%%   30  DOUBLE                 DERO              % ERROR INFORMATION
000000   %%%%   30  INTEGER ARRAY          BREGBLOCK         % REGISTER SAVE AT ESCAPE
000000   %%%%   32  DOUBLE                 DER2(3)           % ERROR INFORMATION
000000   %%%%   40  INTEGER                DBPREG            % P-REGISTER ON PAGE FAULT ON IOBT LEVEL
000000   %%%%   41  INTEGER                DBACTPRI          % ACTPRI ON PAGE FAULT ON IOBT LEVEL
000000   %%%%   42  INTEGER                FLAGB             % BACKGROUNG FLAG
000000   %%%%   43; INTEGER                EUSADD            % ADDRESS FOR USER-ESCAPE HANDLING
000000   %%%%   44; INTEGER                LUSADD            % ADDRESS FOR LOCAL-FUNCTION HANDLING
000000   DISP   45; INTEGER                TSTADD     ;PSID  % TAD INPUT-STACK ADRESS (BANK IN TSBANK)
000000   DISP   46; INTEGER                REMSIZ     ;PSID  % REMAINING BYTES IN XMSG BUFFER
000000   DISP   47; INTEGER                CURMES     ;PSID  % CURRENT MESSAGE TYPE
```

```
000000
000000   %=================================================================================================
000000   %       T A D   D A T A F I E L D   O U T P U T :   B D (N) W
000000
000000   %%%%  - 10; INTEGER                 SCREEN            % COUNTER FOR STOP ON FULL PAGE
000000   %%%%  -  7; INTEGER                 LAST              % LAST CHARACTER OUTPUTTED
000000   %%%%  -  6; INTEGER                 TMSUB             % TIME OUT SUBROUTINE
000000   %%%%  -  5; INTEGER                 TMR               % TIME OUT COUNTER
000000   %%%%  -  4; INTEGER                 TTMR              % START VALUE OF TMR
000000   DISP  -  3  INTEGER                 RPORT     ;PSID   % FUNNY-NUMBER OF COMM. PARTNER
000000   DISP  -  2; DOUBLE                  PARTNER   ;PSID   % MAGIC-NUMBER OF COMM. PARTNER
000000
000000   %%%%    B D (N) W          E N T R Y
000000
000000   %%%%    0  INTEGER                 RESLINK           % RESERVATION LINK
000000   %%%%    1  INTEGER                 RTRES             % RESERVING RT-PROGRAM
000000   %%%%    2  INTEGER                 BWLINK            % BEGINNING OF WAITING QUEUE
000000   %%%%    3  INTEGER                 TYPRING           % DEVICE TYPE BITS AND RING
000000   %%%%    4  INTEGER                 ISTATE            % 0=IDLE, 1=BUSY, -1 NO WAIT MODE
000000   %%%%    5  INTEGER                 MLINK             % MONITOR QUEUE LINK
000000   %%%%    6  INTEGER                 MFUNC             % MONITOR LEVEL FUNCTION ADDRESS
000000   %%%%    7  INTEGER POINTER         IOTRANS           % OUTPUT IOTRANS ROUTINE
000000   %%%%   10  INTEGER POINTER         STDEV             % START DEVICE ROUTINE
000000   %%%%   11  INTEGER POINTER         SETDV             % IOSET ROUTINE
000000   %%%%   12  INTEGER                 DFOPP             % POINTER TO INPUT CHANNEL DATAFIELD
000000   %%%%   13  INTEGER                 DERROR            % ERROR CODE
000000   %%%%   14  INTEGER                 BUFFID            % XMSG BUFFER IDENTIFIER
000000   %%%%   15  INTEGER                 TDTAFI            % FIRST PART OF BUFFER ADDRESS
000000   %%%%   16  INTEGER                 TDTALA            % LAST  PART OF BUFFER ADDRESS
000000   %%%%   15  DOUBLE                  TDTADD            % ADDRES OF BUFFER DATA-PART
000000   %%%%   17  INTEGER                 TDBTPT            % BYTE POINTER IN XMSG BUFFER
000000   %%%%   20  INTEGER                 XRSA              % SAVE X-REG IN IOTRANS
000000   %%%%   21  INTEGER POINTER         LRSA              % SAVE L-REG IN IOTRANS
000000   DISP   22; INTEGER                 POOLP     ;PSID   % BUFFER-ID OF FIRST FREE BUFFER (CBM B-VERS)
000000   %%%%   23  INTEGER                 ROFIL             % FOR MODE OUTPUT FILE NUMBER
000000   %%%%   24  INTEGER                 BCHOSTS           % FOR MODE OUTPUT STATUS
000000   %%%%   25  INTEGER                 ON5MSG            % ADDRESS OF ND-500 MESSAGE IN FAST OUTSTRING
000000   DISP   26; INTEGER                 DRFUNC    ;PSID   % FUNCTION TO BE PERFORMED BY DRIVER
000000   DISP   27; INTEGER                 BXTADD    ;PSID   % ADDRESS OF XTBLOCK FOR DRIVER
000000   DISP   30; INTEGER POINTER         ACTOUT    ;PSID   % ACTIVATE ADDRESS FOR OUTPUT-DRIVER
000000   DISP   31; INTEGER                 RSPNUM    ;PSID   % RESPONSE NUMBER AWAITED ON INPUT
000000   %%%%   32  REAL                    ZOPRG             % P,X,T  REGISTERS IN OUTSTRING
000000   %%%%   35  REAL                    ZOARG             % A,D,L  REGISTERS IN OUTSTRING
000000   %%%%   40  REAL                    ZOSRG             % S,B  REGISTERS + SAVED OLDPAGE IN OUTSTRING
000000   DISP   43; INTEGER                 USDADR    ;PSID   % ADRESS OF USERS DATA IN OUTSTRING
000000   DISP   44; INTEGER                 OBCOU     ;PSID   % BYTE COUNTER IN OUTSTRING
000000   %%%%   45  INTEGER                 TSTADD            % TAD OUTPUT-STACK ADRESS (BANK IN TSBANK)
000000   %%%%   46  INTEGER                 REMSIZ            % REMAINING BYTES IN XMSG BUFFER
000000   %%%%   47  INTEGER                 CURMES            % CURRENT MESSAGE TYPE
000000   DISP   50; INTEGER                 NOBDIS    ;PSID   % MESSAGE BYTE COUNTER IN XMSG BUFFER
000000   DISP   51; DOUBLE                  POOLLI    ;PSID   % BUFFER ADDRESS OF FIRST BUFFER IN POOL (CBM C-VERS)
000000
000000   %=================================================================
000000   %       T A D   P R O T O C O L L   E L E M E N T   T Y P E S
000000   @ICR
000000   SYMBOL 7BDAT=1,   7RFI,     7ECKM,     7BMMX,     7CORQ=6,   7CORS,
000000          7ESCA,    7DCON,    7SYSI,     7LUN,      7TMOD,     7TTYP,
000000          7CESC,    7DESC,    7USID,     7PASS,     7SYCN=23,  7USCN,
000000          7FBSI,    7RESE,    7RECO,     7DUMM,     7STRQ,     7STRS,
000000          7KEYI,    7BADT,    7OPSV=37,  7ESRS,     7CERS,     7ISRQ,
```

163654
100003
702

(link)
bank no
word disp

```
000000          7ISRS,      7NOWT,      7TNOW,      7NWRE,      7RLOC,      7IAM,
000000          7EDRS,      7TREP,      7CPCO=372,  7ERRS,      7WHO,       7POLL,
000000          7REJE,      7EOP;
000000    @CR;
000000
000000    SYMBOL BUDIS=4                % DATA DISPLACEMENT IN BUFFER
000000
000000    %=================================================================
000000    %       E R R O R   C O D E S   G I V E N   F R O M   T A D
000000
000000    SYMBOL TER00=314              % INPUT WHILE ESCAPE/LOCAL OFF IS ILLEGAL
000000    SYMBOL TER01=315              % T.A.D. PROTOCOLL ERROR, ILLEGAL OR INCONSISTENT MESSAGE
000000    SYMBOL TER02=316              % TERMINAL LINE IS NOT CONNECTED
```

```
000000
000000   %=================================================================
000000   %
000000   %       M O N I T O R   C A L L   H D L C
000000   %       TABLE AND SYMBOL DEFENITIONS
000000   %
000000   %=================================================================
000000   %
000000   % FUNCTIONS VALUES
000000   %
000000   @1CR
000000     SYMBOL FSEND=0,              %SEND MESSAGE
000000   FRECV,                         %RECEIVE MESSAGE
000000   FMXRECV;                       % RECEIVE MESSAGE WITH USER SPECIFIED MAX. LENGTH
000000   @CR;
000000   SYMBOL HXCOD = 16531                    % PASSWORD FOR HDLC/X21 MON CALL *81F*
000000   SYMBOL X21LO = 0                        % HDLC INTERFACE LOCKED, SET IN HXDOK
000000   SYMBOL X21OP = 177777                   % HDLC INTERFACE OPEN, SET IN HXDOK
000000   %
000000   %
000000   % SYSTEM GENERATION PARAMS
000000   %
000000   %TEMPORARY
000000   SYMBOL ST = 7
000000   SYMBOL SP = 24
000000   SYMBOL OFSET = 13
000000   SYMBOL PCNT = 4                         % # OF CALLING PARAMETERS
000000   %
000000   % ERROR CODES
000000   %
000000   SYMBOL E1    = -1                       % DEVICE NOT RESERVED BY YOU
000000   SYMBOL E2    = -2                       % UNKNOWN DEVICE
000000   SYMBOL EEMTY = -3                       %NO ELEMENT IN USER RECV. QUEUE
000000   SYMBOL ENBUF = -4                       %NO VACANT BUFFER
000000   SYMBOL EMSGS = -5                       %ILLEGAL MESSAGE SIZE. SIZE < 0
000000   SYMBOL EDEVN = -6                       %ILLEGAL DEVICE NUMBER
000000   SYMBOL EMAXS = -7                       %MAX BUFFERSIZE < USED BUFFERSIZE
000000   SYMBOL EFUNC = -10                      %ILLEGAL FUNCTION
000000   SYMBOL HXERF = -11                      %FATAL ERROR (INCOSTINCY)
000000   %
000000   % DEVICE DATAFIELD DECRIPTION
000000   %
000000   % ** STANDARD SINTRAN PART (-7 TO +21)
000000   %
000000   % ** FROM DEMANDFIELD
000000   DISP HARAL=33
000000   INTEGER DDD0
000000   INTEGER DDD1
000000   INTEGER DDD2
000000   INTEGER DDD3
000000   INTEGER DDD4
000000   PSID
000000   %
000000   %       LOCATIONS USED FOR XMSG-SIMULATION
000000   %
000000   DISP HARAL
000000   INTEGER HINIF                           % SET TO -1 IF BUFFERPOOL INITIATED
000000   INTEGER MASTB                           % PHYSICAL BANK (USED IN LDATX STATX....)
000000   INTEGER XBBNK        % BUFFERSYS BLOCK  % BANK NO
000000   INTEGER XBSTR                           % ADDR FIRST WORD IN POOL
```

```
000000   INTEGER XBEND                                  % ADDR LAST WORD
000000   INTEGER XBSAV          % END BLOCK              % TEMP LOCATION FOR BUFFERSYSTEM
000000   INTEGER XXUBF          % COPYROUT BLOCK         % BUFFER ADDRESS IN USERS LOGICAL SPACE
000000   INTEGER XXSBK,XXSBF                            % PHYSICAL BUFFER ADDRESS (BANK, DISPLACEMENT)
000000   INTEGER POINTER XXHOM %END BLOCK               % TEMP LOCATION FOR COPY ROUTINE
000000   INTEGER FIXID                                  % IIDENT FOR BUFFERS IN SWAP-AREA
000000   INTEGER SWBUF                                  % SET TO 1 IF BUFFER IN SWAPPING RAEA
000000   INTEGER LWPHY                                  % LOW PHYSICAL PAGE FOR BUFFER ABOVE
000000   INTEGER XWAITF                                 % SET IN I/O-WAIT
000000   INTEGER MESSID                                 %CURRENT MESSAGE IDENTIFICATOR
000000   INTEGER POINTER RSRET                          %RETURN ADDRESS TO RESIDENT PART
000000   INTEGER RSCUR                                  %CURRENT MESSAGE SENT OR RESEIVED
000000   INTEGER ARRAY POINTER PRSCU=RSCUR
000000   INTEGER IQUEU                                  %HEAD OF INPUT QUEUE
000000   INTEGER OQUEU                                  %HEAD OF OUTPUT QUEUE (FROM DRIVER)
000000   INTEGER WAKEF                                  %SET IF DRIVER WAITING FOR MESSAGES
000000   INTEGER HXCC                                   % PASSWORD FOR HDLC/X21 MONITOR CALL *81F*
000000   PSID
000000   %
000000   % BUFFER (MESSAGE) HEADER DECRIPTION
000000   %
000000   SYMBOL BHEAD = 5                               % WORDS USED FOR DCB TRANSFER *81F*
000000   DISP 0
000000    INTEGER XCHAIN                                % ROUTINE ADDRESS TO HANDLE DCB TRANSFER *81F*
000000    INTEGER BBID                                  %MESSAGE ID
000000    INTEGER BBYTC                                 %BYTECOUNT
000000    INTEGER BMBYTE                                %MAX MESSAGE SIZE
000000    INTEGER BCHAIN                                %CHAIN-WORD
000000   PSID
000000   %
000000
000000
000000   %*******************************
000000   %
000000   %       X 2 1  DEFINITIONS
000000   %
000000   %*******************************
000000
000000
000000
000000   %       MESSAGE DECRIPTION
000000
000000
000000   DISP BHEAD
000000           INTEGER X2MFU                                  % FUNCTION
000000           INTEGER X2MST                                  % STATUS FROM X21
000000           INTEGER X2MFA                                  % FACILITY REQUEST
000000           INTEGER X2MCP                                  % CALL PROGRESS SIGNALS
000000           INTEGER X2MSL                                  % SELECTION SIGNALS
000000           INTEGER X2MCL=X2MSL                            % CALLED/CALLING-LINE IDENT.
000000           INTEGER X2MSS                                  % 2. WORD OF SELECTION SIGNALS
000000           INTEGER X2MS2                                  % 3. WORD OF SELECTION SIGNALS
000000   PSID
000000
000000   SYMBOL X2M1=BHEAD+1
000000   SYMBOL X2M1B=BHEAD+2
000000   SYMBOL X2M2=BHEAD+3
000000   SYMBOL X2M3=BHEAD+4
000000   SYMBOL X2M4=BHEAD+5
000000   SYMBOL X2M5=BHEAD+6
```

```
000000
000000   %       X2MFA BITS
000000
000000   SYMBOL X21C1 = 0                          % CAHRGING REQUEST
000000   SYMBOL X21C2 = 1                          % CALLED LINE ID
000000   SYMBOL X21C3 = 2                          % DIRECT CALL
000000   SYMBOL X21C4 = 3                          % ABRIVATED ADDRESSS
000000
000000   %       STATUS SET IN "GET STATUS COMMAND"
000000
000000   SYMBOL X21SA = 0                          % NOT CONNECTED
000000   SYMBOL X21SB = 2                          % DATA PHASE
000000   SYMBOL X21SC = 1                          % CONNECTED
000000
000000   %       USEFUL SYMBOLS
000000
000000   SYMBOL X21PM =  11                                % X2MFU MAX VALUE
000000   SYMBOL X21SY = 026                                % SYNC CHAR.
000000   SYMBOL X21PL = 053                        % "+"
000000   SYMBOL X21PP = 253                        % "+" WITH ODD PARITY
000000   SYMBOL X21KO = 054                        % ","
000000   SYMBOL X21SL = 057                        % "/"
000000   SYMBOL X21ST = 052                        % "*"
000000   SYMBOL X21BL = 007                        % "BEL" I.E TUUUUT!
000000   SYMBOL X21NL = 000                        % MUST BE DEFINED LATER
000000   SYMBOL X21EN = 177                        % MUST BE DEFINED LATER
000000   SYMBOL DA = 1
000000   SYMBOL SA = 2
000000   SYMBOL RA = 3
000000   SYMBOL DCE19 = 7                          % BIT SET IN RRSR WHEN DCE CLEAR REQUEST
000000
000000   %       DATAFIELD DECRIPTION
000000
000000   DISP -3
000000           INTEGER X2DHD                             % DEVICE NUMBER
000000   PSID
000000
000000   DISP DIX21=HARAL
000000           INTEGER X2DBR                             % ADDRESS OF BREAK MESSAGE OR 0 IF NONE
000000           INTEGER X2DLS                             % SAVE L-REG IN X2GET
000000           INTEGER X2DSR                             % SAVE L-REG IN STATE ROUT
000000           INTEGER X2DSL                             % SAVE L-REG IN GET CHARGING ROUTINE
000000           INTEGER X2DUI                             % USER DATAFIELD ADDRESS (INP)
000000           INTEGER X2DUO                             % USER DATAFIELD ADDRESS (OUT)
000000           INTEGER X2DPI                             % 1 IF PINNED FOR INPUT
000000           INTEGER X2DCN                             % 1 IF CONNECTED ELSE 0
000000           INTEGER X2DST                             % STATE VARIABLE (CCITT REC)
000000           INTEGER X2DDF                             % =1 WHEN  DATA TRANSFER STATE
000000           INTEGER X2DPC                             % GENERAL BYTE COUNTER
000000           INTEGER X2DER                             % ERROR CODE FOR USER
000000           INTEGER X2DPS                             % CALL PROGRESS SIGNALS
000000           INTEGER X2DLA                             % ADDRESS OF LINE ID
000000           INTEGER X2DLI                             % CALLED/CALLING LINE
000000           INTEGER X2DCI                             % ADDRESS OF CHARGING INFO
000000           INTEGER X2DC1                             % CHARGING INFORMATION
000000           INTEGER X2DC2
000000           INTEGER X2DC3
000000           INTEGER X2DC4
000000           INTEGER X2DC5
000000           INTEGER X2DC6
```

```
000000          INTEGER X2DMP                                   % MESSAGE POINTER
000000          INTEGER ARRAY POINTER X2DMM=X2DMP
000000          INTEGER X2DRL                                   % RUNNING LEVEL  12 OR 13
000000          INTEGER X2DSP                                   % SAVE LEVEL 13 P-REG WHEN
000000                                                          % RUNNING LEV 12 AT LEV 13
000000          INTEGER X2D0C                                   % USED IN X2GET AS COUNTER
000000          INTEGER X2DCC          % 1 IF CHARG REQUEST, 2 IF CHARG RECEIVED, ELSE 0
000000          INTEGER X2DBC          % NEGATIV # OF UNUSED BYTES IN CURRENT MESSAGE
000000          INTEGER X2D00                                   % LAST RECEIVER DATA REG
000000          INTEGER X2D02                                   % LAST RECEIVER STATUS
000000          INTEGER X2D05
000000          INTEGER X2D06
000000          INTEGER X2D10
000000          INTEGER X2D11
000000          INTEGER X2D12
000000          INTEGER X2D13                                   % CONT ADDR AFTER INTERRUPT
000000          INTEGER X2DL3
000000    PSID
000000
000000    %
000000    %
000000    %I O X DEFINITIONS
000000
000000    @ICR;                                                 % READ REC.DATA REG.
000000    SYMBOL XRRDR=0,                                       % WRITE PARAM. CONTR. REG.
000000    XWPCR,                                                % READ REC STATUS
000000    XRRSR,                                                % WRITE SYNC/ADDRESS REG.
000000    XWSAR,                                                % WRITE CHAR. LENGTH
000000    XWCHL,                                                % WRITE TRANSM. DATA REG.
000000    XWTDR,                                                % READ TRANSM. STATUS REG.
000000    XRTSR,                                                % WRITE TRANSM. CONTROLE REG.
000000    XWTCR,                                                % READ REC. TRANSFER STATUS
000000    XRRTS,                                                % WRITE REC. TRANSFER CONTROLE
000000    XWRTC,                                                % READ TRANSM. TRANSFER STATUS
000000    XRTTS,                                                % WRITE TRANSM. TRANSFER CONTRO
000000    XWTTC;
000000    @CR;
000000
000000    %
000000    %
000000    %       X21 DTE TIMEOUTS  (CCITT RECOMANDATION)
000000                                                          % STATE 2 TO 3
000000    SYMBOL X21T1 = -4                                     % STATE 4,5 TO 7,10,11,12
000000    SYMBOL X21T2 = -25                                    % STATE 7 TO 12,19
000000    SYMBOL X21T3 = -3                                     % STATE 9 TO 12,19
000000    SYMBOL X21T4 = -3                                     % STATE 16 TO 21
000000    SYMBOL X21T5 = -3                                     % STATE 20 TO 21
000000    SYMBOL X21T6 = -3                                     % USED IN STATE 1, NETWORK OK ?
000000    SYMBOL X21T0 = -2                                     % WAITING FOR CHARGING INFO (200MS)
000000    SYMBOL X21T7 = -2
000000
000000    %
000000    %       X21 DTE TIMOUTS  (SELF COMPOSED)
000000                                                          % USED IN STATE 1, NETWORK OK?
000000    SYMBOL X21TA = -2                                     % USED IN STATE 4, OUTPUT TIMOUT
000000    SYMBOL X21TB = -10
000000
000000
000000    %
000000    %
```

```
000000   %
000000   %       ERROR CODES
000000
000000   SYMBOL X2FXX = 99                    % *** NOT YET DEFINED ***
000000   SYMBOL X2FOK = 0             % OPERATION SUCESSFULLY COMPLETD
000000   SYMBOL X2FSM = 1             % MESSAGE TOO SMALL
000000   SYMBOL X2FIL = 2             % ILLEGAL LDN IN CONNECT (EXIST)
000000   SYMBOL X2FIC = 3             % ILLEGAL COMMAND
000000   SYMBOL X2FIS = 4             % ILLEGAL COMMAND IN DATAPHASE
000000   SYMBOL X2FNM = 5             % UNEQUAL HDEV IN LDN'S
000000   SYMBOL X2FNR = 6             % DATA LDN NOT RESERVED BY YOU
000000   SYMBOL X2FNI = 7             % NO IDENT ENTRY FOR CONNECTED LDN
000000   SYMBOL X2F01 = 10            % NETWORK ERROR, MODEM POWER ON?
000000   SYMBOL X2FNC = 11            % NO INCOMMING CALL, READY STATE FORCED TO TERMINAT
000000   SYMBOL X2FFF = 12            % NO LDN PREVIOUSLY CONNECTED
000000   SYMBOL X2FNP = 13            % MISSING TERMINATOR IN SELECTION SIGNALS
000000   SYMBOL X2FAB = 14            % NO CHARGING RECEIVED FOR LAST CALL
000000   SYMBOL X2F20 = 32            % NETWORK TIMOUT (STATE 2)
000000   SYMBOL X2F30 = 33            % NETWORK TIMOUT (STATE 3) NO "+" FROM NETTWORK
000000   SYMBOL X2F50 = 35            % NETWORK TIMOUT (STATE 5)
000000   SYMBOL X2FCP = 15            % CALL PROGRESS SIGN.
000000   SYMBOL X2FEC = 16            % MULTIPLE CP SIGNALS
000000   SYMBOL X2FAC = 17            % X21-ALLREADY CONNECTED
000000   SYMBOL X2FBR = 20            % CALL TERMINATED
```

```
000000
000000
000000   %*********************************************
000000   %
000000   %      P I O C    D E F E N I T I O N S
000000   %
000000   %*********************************************
000000
000000
000000   %      DESCRIPTION OF PIOC DATAFIELD
000000
000000   DISP 7
000000          INTEGER PWCR                                   % COPY OF PIOC CONTROLE REGISTER
000000          INTEGER PRSR                                   % COPY OF PIOC STATUS REGISTER
000000          INTEGER KPROS                                  % CONTAINS RT-ADDRESS OF PROCESS TO B
000000          INTEGER PINIT                                  % SET TO ONE IF TABLES BELOW INITIATE
000000          INTEGER PIOCA                                  % PIOC PAGE ADDRESS
000000          INTEGER MASTA                                  % POINTS TO NORD TO PIOC OPCOM BOX
000000          INTEGER PNBOX                                  % SET IF KICK TO MONITOR
000000          INTEGER PKICK                                  % SET IF KICK WANTED IN NORD
000000          INTEGER POINTER PTN                            % TRIGGERS PIOC TO NORD
000000          INTEGER POINTER NTP                            % TRIGGERS NORD TO PIOC
000000          INTEGER POINTER PIPRO                          % PIOC PROCESS SIDE OF SLOT
000000          INTEGER POINTER NDPRO                          % NORD PROCESS SIDE OF SLOT
000000          INTEGER POINTER PNMAIL                         % MAIL FOR NORD
000000          INTEGER POINTER NPMAIL                         % MAIL FOR PIOC
000000          INTEGER PREQU                                  % START=: 3, LOAD =:1
000000          INTEGER PISTT
000000          INTEGER PIMPG
000000          INTEGER PIOCN
000000   %      XMSG PART
000000          DOUBLE MBOXH                   % START OF XMSG QUEUE
000000          DOUBLE RTBOX                   % START OF RT-ACT QUEUE
000000          INTEGER MCURB                  % BANK INFO          CURRENT ELEMENT
000000          INTEGER LCURB                  % DISP WITHIN BANK   CURRENT ELEMENT
000000          DOUBLE MDCUR = MCURB
000000          INTEGER ARRAY PXT(20)          % ARRAY OF XT BLOCKS   PROCESS NUMBER IS INDEX
000000
000000   PSID
000000
000000   %      DESCRIPTION OF NORD TO PIOC MAILBOX
000000   DISP 0
000000          INTEGER NPTIG
000000          INTEGER NPFUN
000000   PSID
000000
000000   %      DESCRIPTION OF KICKTAB
000000
000000          SYMBOL  XPKIC =  0
000000          SYMBOL  XPTN  =  2
000000          SYMBOL  XNTP  =  6
000000          SYMBOL  XPIPR = 12
000000          SYMBOL  XNDPR = 16
000000          SYMBOL  XNMAI = 22
000000          SYMBOL  XPMAI = 26
000000
000000   %      DESCRIPTION OF PIOC POINTER TABLE
000000
000000   DISP 0
000000          INTEGER DYO
```

```
000000          INTEGER NPOPC
000000          INTEGER DY1
000000          INTEGER PNOPC
000000          INTEGER DY2
000000          INTEGER REGPO
000000          INTEGER DY3
000000          INTEGER PIOCI
000000          INTEGER DY4
000000          INTEGER PIOCO
000000          INTEGER DY5
000000          INTEGER PIPOW
000000          INTEGER DY6
000000          INTEGER DROPS
000000          INTEGER DY7
000000          INTEGER BCFLA
000000          INTEGER DY8
000000          INTEGER KICKP
000000          INTEGER PIMBH
000000          INTEGER PIMBL
000000          INTEGER RTBH
000000          INTEGER RTBL
000000          DOUBLE  DY12                    % POINTER PROCESS TABLE
000000          INTEGER PSIZ                    % PIOC SIZE
000000          INTEGER PINO                    % PIOC NUMBER
000000          INTEGER CPUNR                   % ND100 CPU NUMBER
000000   PSID
000000
000000   SYMBOL PIMON = 255
000000   SYMBOL TRIG  =  1
000000   SYMBOL DTRIG =  0
000000   SYMBOL PRKEY = 52163
000000
000000   %      BITS FOR PIOC WRITE CONTROLE REG
000000
000000   SYMBOL BENA  =  0                                          % ENABLE FOR PIOC INTERRUPT
000000   SYMBOL BNDC  =  2                                          % NORD CALLING
000000
000000   %      VALUES FOR NORD TO PIOC FUNCTION
000000
000000   SYMBOL MBREAK = 0
000000   SYMBOL MCONT  = 1
000000   SYMBOL MMEMO  = 2
000000   SYMBOL MOPEN  = 3
000000   SYMBOL MIOIN  = 4
000000   SYMBOL MPIOC  = 5                                          % START
000000   %
000000   %      ERROR CODES FOR CALLER
000000
000000   SYMBOL POK   =  1                                          % SUCCESS
000000   SYMBOL PPROM =  2                                          % NO ANSWER FROM PIOC PROM
000000   SYMBOL PNOTP =  -10                                        % NO PRIVILEGE
000000   SYMBOL PNOTR =  -11                                        % PIOC NOT STARTED AFTER LOAD
000000   SYMBOL PILF  =  -24                                        % ILLEGAL FUNCTION
000000   SYMBOL PSLBS =  -25                                        % SLOT OCCUPIED
000000   SYMBOL PILSL =  -26                                        % ILLEGAL SLOT
000000   SYMBOL PNOTY =  -27                                        % SLOT NOT RESERVED BY YOU
000000   SYMBOL PFULL =  -30                                        % BOX NOT EMTY
000000   SYMBOL PNOME =  -31                                        % NO MESSAGE
000000   SYMBOL PE1   =  -32                                        % ILLEGAL LDN
000000   SYMBOL PNI   =  -33                                        % PIOC NOT INITIATED
```

```
000000   SYMBOL PNOTFIX = -34                    % SEGMENTS NOT FIXED
000000   %  40 - 50                              % ERRORS FROM FIXC IN LOAD
000000
000000   %
000000   %       CALL FUNCTION CODES
000000   SYMBOL FRES = 0
000000   SYMBOL FFREL = 1
000000   SYMBOL FKICK = 2
000000   SYMBOL FWKICK = 3
000000   SYMBOL FLOAD = 4
000000   SYMBOL FUNLOAD = 5
000000   SYMBOL FSTART = 6
000000   SYMBOL FSTOP = 7
000000   SYMBOL PMFUN = FSTOP                    % TOTAL NUMBER OF LEGAL FUNCTIONS
000000   SYMBOL PIKEY = 121314                   % B-REG CONTENT WHEN CALLING PIOC SPE
000000
000000   SYMBOL SLMAX = 8                        % SIZE OF SLOT TABLE
000000
000000   SYMBOL K1024 = 2000
000000
000000
```

```
000000
000000   %
000000   % ND-500 SYMBOLS USED IN TERMINAL INPUT/OUTPUT ROUTINES
000000   %
000000   SYMBOL MSGN500=1                % MESSAGE TO ND-500
000000   DISP 2; INTEGER N5STATUS; PSID  % STATUS IN MAILBOX
000000   DISP 103; INTEGER NOBYT; PSID   % NUMBER OF BYTES IN DVOUTST (MON 504)
000000   DISP 140
000000          INTEGER                 HBUFADR, LBUFADR
000000                DOUBLE            ABUFADR=HBUFADR     % ADDRESS OF 2KB BUFFER
000000          INTEGER                 5HENTE               % POINTER WITHIN ABUFADR
000000                                                       % USED QUICK OUTSTRING
000000                INTEGER           TODF=5HENTE          % TERMINAL OUTBUF DATA FIELD
000000          INTEGER POINTER         SPL12                % SPECIAL FUNCTION EXECUTED
000000                                                       % ON LEVEL 12
000000                INTEGER           SPFLAG=SPL12
000000   PSID
000000
000000   DISP 100
000000          DOUBLE                  IDEVNO               % INPUT DEVIC NO.
000000          INTEGER                 HMAXBYT, MAXBYT      % MAXIMAL NO. BYTES RETURNED
000000                DOUBLE            DMAXBYT=HMAXBYT
000000          DOUBLE                  NOCHRET              % NO. OF CHARACTERS RETURNED
000000          DOUBLE                  ISTR
000000          INTEGER                 HBRST, 5BRST         % NEW BREAK STRATEGY
000000          INTEGER                 HECHST, 5ECHST       % NEW ECHO STRATEGY
000000          INTEGER                 5FYLLE
000000          INTEGER                 MLFLAG               % MONITOR LEVEL ACTIVE
000000   PSID
000000
000000   % DATA FIELD FOR SPECIAL SIBAS MONITOR CALL
000000
000000   % - DF RESERVED BY SIBAS DURING MONITOR CALL EXECUTION (MON 305 MSIBB)
000000
000000   DISP 20
000000          INTEGER SIB500          % =0: SIBAS IN ND-100; =1: SIBAS IN ND-500
000000   PSID
000000
000000   % - DF RESERVED BY APPLICATION DURING MONITOR CALL EXECUTION (MON 304 MAPSIB)
000000
000000   DISP SMCDF=20                   % SIBAS MONITOR CALL DATA FIELD
000000          INTEGER SICCOUNT
000000          INTEGER SIAD1,SIAD2             % ADDRESS OF MESSAGE IN RT-COMMON
000000          DOUBLE SIDRTC=SIAD1
000000          INTEGER SRTCSTAT=ISTAT          % =0: RTC FREE; ><0: RTC OCCUPIED
000000   PSID
000000
000000
000000   %      DEFINITIONS FOR THE ACCOUNTING SYSTEM.
000000
000000   SYMBOL ACCRL=36                         % ACCOUNTS:DATA RECORD LENGTH.
000000
000000
000000   %  DEFINITIONS FOR MON LOGIN
000000
000000   SYMBOL MLIPASET=40                      % PASET VALUE
000000   SYMBOL MLISTRN=MLIPASET+5               % PARAMETER STRING
000000
000000   DISP 30
000000   INTEGER MLITERM                         % ADDR OF TERMINAL INPUT DATAFIELD
```

```
000000    INTEGER MLICPNT                          % BYTE POINTER IN MLISTRN
000000    INTEGER MLICADDR
000000    INTEGER MLIMXADDR
000000    INTEGER MLIWORD                          % ADDR OF RETURNED STATUS
000000    INTEGER MLIADDR                          % STATUS VALUE
000000    INTEGER MLIMSTATUS
000000    PSID
000000
000000    % BUFFER SIZE IN BUFFER POOL
000000    SYMBOL PBSIZ=34,PBBYT=70
000000
000000    %
000000    %       GLOBAL DEFINITIONS FOR LAMU
000000    %                                          % ELEMENT OF LAMU DESCRIPTION TABLE
000000    DISP 0                                     % FIRST PHYSICAL PAGE OF LAMU
000000            INTEGER LAMPP                      % NUMBER OF PAGES IN LAMU
000000            INTEGER LAMNP                      % LAMU PROTECTION
000000            INTEGER LAMPR
000000    PSID
000000                                               % ELEMENT OF ACTIVE LAMU TABLE
000000    DISP 0                                     % CONNECTED LAMU
000000            INTEGER LAMCN                      % FIRST LOGICAL PAGE OF LAMU
000000            INTEGER LAMLP
000000    PSID
000000
000000    %       THE SAME DEFINITIONS FOR T X INSTRUCTIONS
000000            SYMBOL LMLP=10, LMPP=0, LMNP=10, LMPR=20, LMCN=0
000000
```

```
000000
000000   % ******************* VIRTUAL DISK DRIVER SYMBOLS *********************
000000
000000   %======LAYOUT OF VIRTUAL DISK DRIVER DATA FIELD (VDDXY):
000000
000000   DISP -22
000000          DOUBLE  VDBFREMOTE                  % MEMORY ADDRESS OF SPECIAL ...
000000                                              % ... BUFFER SEEN FROM VDP SIDE
000000
000000          DOUBLE  VDBUF                       % SPECIAL TRANSFER BUFFER
000000          INTEGER POINTER VDMTT               % ADDRESS OF MEMORY TRANSL. TABLE
000000          INTEGER VDRTP                       % COMM. RT-PROG. TO VDP
000000          INTEGER VDFUN                       % TRANSFER FUNCTION CODE
000000          DOUBLE  VDMAD                       % MEMORY ADDRESS
000000          DOUBLE  VDBLC                       % BLOCK ADDRESS ON DISK
000000          INTEGER VDNBL                       % NO. OF BLOCKS TO TRANSFER
000000   %      INTEGER POINTER TMSUB               % TIME-OUT SUBROUTINE (VDTMR)
000000   %      INTEGER TMR                         % TIME-OUT COUNTER
000000   %      INTEGER TTMR                        % INITIAL VALUE OF TMR
000000   %      INTEGER HDEV                        % NOT USED
000000   %      INTEGER POINTER STDRIV              % USED BY VDD TO SAVE L-REG
000000   %      INTEGER POINTER DRIVER              % VDC RESTART ADDRESS
000000   PSID
000000   DISP 0 % STANDARD PART:
000000   %      INTEGER RESLINK        % RESERVATION LINK
000000   %      INTEGER RTRES          % RESERVING RT-PROGRAM
000000   %      INTEGER BWLINK         % BEGINNING OF WAITING QUEUE
000000   %      INTEGER TYPRING        % DEV. TYPE BITS AND RING
000000   %      INTEGER ISTATE         % 0=IDLE, 1=BUSY, -1=NOWAIT MODE
000000   %      INTEGER MLINK          % MONITOR QUEUE
000000   %      INTEGER MFUNC          % MONITOR LEVEL FUNCTION ADDRESS
000000   %      INTEGER POINTER TRLREG  % RETURN ADDRESS ON MON. LEVEL AFTER TRANSF.
000000   %      INTEGER HSTAT          % HARDWARE STATUS FROM DEVICE
000000   %      INTEGER POINTER MTRANS  % MON. LEVEL ROUTINE TO ACTIVATE DRIVER
000000   PSID
000000   DISP 12 % IDENTICAL TO VDP:
000000   %      INTEGER VDLINE                      % VDC LINE NUMBER
000000   %      INTEGER VDCST                       % COMMUNICATION STATUS
000000   %      INTEGER VDDST                       % DISK STATUS
000000   %      INTEGER VDMB0, VDMB1                % PHYS. ADDRESS OF MAIL-BOX
000000   %      DOUBLE  VDMBX = VDMB0               % PHYS. ADDRESS OF MAIL-BOX
000000   PSID
000000   DISP 17
000000          INTEGER VDCNRTOUT                   % NUMBER OF TIMEOUTS
000000   PSID
000000
000000   %======LAYOUT OF VIRTUAL DISK PROCESSOR DATAFIELD (VDPXY):
000000
000000   DISP 0 % STANDARD PART:
000000   %      INTEGER RESLINK        % RESERVATION LINK
000000   %      INTEGER RTRES          % RESERVING RT-PROGRAM
000000   %      INTEGER BWLINK         % BEGINNING OF WAITING QUEUE
000000   %      INTEGER TYPRING        % DEV. TYPE BITS AND RING
000000   PSID
000000   DISP 4
000000          INTEGER VDPFUN                      % TRANSFER FUNCTION
000000          DOUBLE  VDPMEM                      % MEMORY ADDRESS
000000          DOUBLE  VDPDISK                     % BLOCK ADDRESS ON DISK
000000          INTEGER VDPNUMB                     % NO. OF BLOCKS TO TRANSFER
000000          INTEGER VDLINE                      % VDC LINE NUMBER
000000          INTEGER VDCST                       % COMMUNICATION STATUS
```

```
000000          INTEGER VDDST                       % DISK STATUS
000000          INTEGER VDMB0, VDMB1                % PHYS. ADDRESS OF MAIL-BOX
000000          DOUBLE  VDMBX = VDMB0               % PHYS. ADDRESS OF MAIL-BOX
000000          INTEGER VDUNIT                      % REAL DISK LDN.
000000          INTEGER ARRAY VDABSTR (4)           % ABSTR PAR. POINTERS
000000   PSID
000000
000000   %======LAYOUT OF ENTRY IN MEMORY TRANSLATION TABLE:
000000
000000   % NB! ENTRIES MUST BE INSERTED IN ASCENDING ORDER OF VDPGLOCAL
000000
000000   DISP 0
000000          INTEGER VDPGLOCAL                   % FIRST PAGE IN THIS AREA
000000          INTEGER VDPGREMOTE                  % PAGE NO IN REMOTE COMPUTER ...
000000                                              % ... (WITH DISK CONNECTED) ...
000000                                              % ... -2 = NOT SHARED ...
000000                                              % ... -1 = END OF TABLE
000000   PSID
000000
000000   %======LAYOUT OF VDC MAIL-BOX (FOR PHYSICAL ADDRESSING):
000000
000000   SYMBOL VDCPFLAG = 00    % INTEGER     % PACKET FLAG
000000   SYMBOL VDCFUN   = 10    % INTEGER     % TRANSFER FUNCTION CODE
000000   SYMBOL VDCMEM   = 20    % DOUBLE      % MEMORY ADDRESS
000000   SYMBOL VDCDISK  = 40    % DOUBLE      % BLOCK ADDRESS ON DISK
000000   SYMBOL VDCNUMB  = 60    % INTEGER     % NO. OF BLOCKS TO TRANSFER
000000   SYMBOL VDCHSTAT = 70    % INTEGER     % DISK STATUS
000000
000000   %======CHECK VALUES USED BY VDC:
000000
000000   SYMBOL VDCSEND     = 374
000000   SYMBOL VDCRECEIVE  = 372
000000
```

```
000000
000000   @DEV 1
000000   @DEV (S-S-J)FILSYS-DEF
000000   *
000000   %
000000   %==========================================================================
000000   % ND-100, ND-500  F I L E   S Y S T E M
000000
000000   %        1. J U N E .   1 9 8 3
000000
000000   *        1. J U N E .   1 9 8 3

000000
000000   %==========================================================================
000000   % 1   D A T A   A N D   M A C R O   D E F I N I T I O N S
000000
000000   %==========================================================================
000000   % 1.1   A U X I L I A R Y
000000
000000   % NAME INDEXES
000000
000000   SYMBOL DKNA1=0                  % DISC NAME INDEX 1
000000   SYMBOL DKNA2=1                  % DISC NAME INDEX 2
000000
000000   % DEVICE NUMBERS
000000
000000   INTEGER GLDN=?                  % SEMAPHORE FOR GENERAL LOCK
000000   INTEGER ULDN=?                  % SEMAPHORE FOR USER FILE BUFFER
000000   INTEGER OLDN=?                  % SEMAPHORE FOR OBJECT FILE BUFFER
000000   INTEGER BLDN=?                  % SEMAPHORE FOR DEVICE BUFFER ALLOCATION
000000   INTEGER OFLDN=?                 % SEMAPHORE FOR RT-OPEN FILE TABLE
000000   INTEGER OFLCK=?
000000   INTEGER MOLDN=?                 % SEMAPHORE FOR OPEN,CLOSE MONITOR CALLS
000000
000000   %
000000   %       MONITOR CALL NUMBERS WHERE A DIFFERENT SYMBOL IS USED IN THE FILE SYSTEM AND SINTRAN
000000   %
000000   SYMBOL 2PRES=2PRSRV, 2PREL=2PRLS
000000
000000   SYMBOL OSEG=3                   % OPERATOR COMM. SEGMENT
000000   SYMBOL ERRSEG=5                 % ERROR SEGMENT NUMBER
000000   SYMBOL FSEG=6                   % FILE SYSTEM SEGMENT NUMBER
000000   SYMBOL O2SEG=23                 % SINTRAN SERVICE AND MAIL SYSTEM SEGMENT
000000   SYMBOL SSEG=25                  % SPOOLING SEGMENT NUMBER
000000
000000   INTEGER SCRSIZE=?
000000
000000   INTEGER CURUS=?
000000   INTEGER PASST=?
000000   INTEGER SPAST=?
000000   INTEGER SSPAS=?
000000   INTEGER TTNO=?
000000   INTEGER ARRAY RUSNAM=?          % ON FILSYS-SYS
000000   INTEGER ARRAY LOCUS=?           % ON FILSYS-SYS
000000   INTEGER ARRAY RSYSNA=?          % ON FILSYS-SYS
000000   INTEGER ARRAY RPRPAS=?          % ON FILSYS-SYS
000000   INTEGER FACFLAG=?               % ON FILSYS-SYS
000000   INTEGER INSFG=?                 % ON FILSYS-SYS
000000   INTEGER WPECT=?                 % ON FILSYS-SYS
000000   INTEGER DRFSG=?                 % ON FILSYS-SYS
```

```
000000   INTEGER FWFLAG=?                % ON FILSYS-SYS
000000   INTEGER SCRWRITTEN=?            % ON FILSYS-SYS
000000   INTEGER INLOGGED=?
000000   INTEGER CSTCK=?
000000   INTEGER SPOOL=?
000000   INTEGER RPOOL=?
000000   INTEGER DRFSG=?
000000
000000   INTEGER REMLIN=?,CHNR=?    % WFLDN=?
000000
000000   INTEGER ARRAY RPASS=?
000000   %
000000   %       ACCESS WORD BITS
000000   %
000000   SYMBOL BITR=0
000000   SYMBOL BITW=1
000000   SYMBOL BITA=2
000000   SYMBOL BITC=3
000000   SYMBOL BITD=4
000000   SYMBOL BITX=4
000000   SYMBOL BITN=5
000000
000000   SYMBOL FRACC=1                  % DEFAULT FRIEND ACCESS (READ)
000000   SYMBOL FIACC=1777               % DEFAULT FILE ACCESS ( PUBLIC: NONE
000000                                   %                      FRIEND AND OWN: RWACD )
000000
000000
000000   % ERROR CODES
000000
000000   @ICR
000000   SYMBOL ER0=20,ER1,  ER20, ER2,  ER3,  ER4,  ER5,  ER6,  ER7,  ER8, ER9,
000000          ER10,  ER11, ER12, ER13, ER14, ER15, ER16, ER17, ER18, ER19,
000000                 ER21, ER22, ER23, ER24, ER25, ER26, ER27, ER28, ER29,
000000          ER30,  ER31, ER32, ER33, ER34, ER35, ER36, ER37, ER38, ER39,
000000          ER40,  ER41, ER42, ER43, ER44, ER45, ER46, ER47, ER48, ER49,
000000          ER50,  ER51, ER52, ER53, ER54, ER55, ER56, ER57, ER58, ER59,
000000          ER60,  ER61, ER62, ER63, ER64, ER65, ER66, ER67, ER68, ER69,
000000          ER70,        ER72, ER73, ER74, ER75, ER76, ER77, ER78, ER79,
000000          ER80,  ER81, ER82, ER83, ER84, ER85, ER86, ER87, ER88, ER89,
000000          ER90,  ER91, ER92, ER93, ER94, ER95, ER96, ER97, ER98, ER99,
000000          ER100, ER101,ER102,ER103,ER104,ER105,ER106,ER107,ER108,ER109,
000000          ER110, ER111,ER112,ER113,ER114,ER115,ER116,ER117,ER118,ER119,
000000          ER120, ER121,ER122,ER123,ER124,ER125,ER126,ER127,ER128,ER129,
000000          ER130, ER131,ER132,ER133,ER134,ER135,ER136,ER137,ER138,ER139,
000000          ER140, ER141,ER142,ER143,ER144,ER145,ER146,ER147,ER148,ER149,
000000          ER150, ER151,ER152,ER153,ER154,ER155,ER156,ER157,ER158,ER159,
000000          ER160, ER161,ER162,ER163,ER164,ER165,ER166,ER167,ER168,ER169,
000000          ER170, ER171,ER172,ER173,ER174,ER175,ER176,ER177,ER178,ER179,
000000          ER180, ER181,ER182,ER183,ER184,ER185,ER186,ER187,ER188,ER189,
000000          ER190, ER191,ER192,ER193,ER194,ER195,ER196,ER197,ER198,ER199,
000000          ER200, ER201,ER202,ER203,ER204,ER205,ER206,ER207,ER208,ER209,
000000          ER210, ER211,ER212,
000000          ERM;
000000   SYMBOL ER71=3;
000000   @CR;
000000
000000   % ==================================================================
000000   % SUBROUTINES ON FILE USER SEGMENT CALLED WHEN REMOTE COSMOS FILE:
000000   % SYMBOL VALUE IS USED AS AN INDEX INTO A SUBROUTINE ADDRESS TABLE:
000000
000000   SYMBOL URFILE=0                 % READ FILE          **
```

```
000000   SYMBOL UWFILE=1              % WRITE FILE          **  ON REENTRANT SEGMENT 2
000000   SYMBOL USBLSZ=2              % SET BLOCK SIZE      **
000000   SYMBOL UMAGTP=3              % MAGTP               **
000000   SYMBOL URBYPO=4              % READ BYTE POINTER   **
000000   SYMBOL USBLPO=5              % SET BLOCK POINTER   **
000000   SYMBOL UOPEN=6               % OPEN REMOTE FILE    **
000000   SYMBOL UCLOS=7               % CLOSE REMOTE FILE   **
000000   SYMBOL UDROBJ=10             % READ OBJECT ENTRY
000000   SYMBOL UMROBJ=11             % READ OBJECT ENTRY
000000   SYMBOL UDWOBJ=12             % WRITE OBJECT ENTRY
000000   SYMBOL UMUIDI=13             % GET DIRECTORY AND USER INDEX
000000   SYMBOL UMRUSE=14             % READ USER ENTRY
000000   SYMBOL UGUSNA=15             % GET USER NAME
000000   SYMBOL UGUIOI=16             % GET DIR., USER AND OBJECT INDEX
000000   SYMBOL UCRALF=17             % CREATE, ALLOCATE FILE OR NEW VERSION
000000   SYMBOL UMDLFI=20             % DELETE FILE
000000   SYMBOL ULIOP=21              % LIST (RT)OPENED FILES
000000   SYMBOL USBYPO=22             % SET BYTE POINTER
000000   SYMBOL UINIT=23              % INIT FILE USER DATA SEGMENT  (LOG IN)
000000   SYMBOL URSOFE=24             % RESERVE OPEN FILE ENTRIE(S)
000000   SYMBOL URLFES=25             % RELEASE OPEN FILE ENTRIES
000000   SYMBOL USPERM=26             % SET PERMANENT OPEN
000000   SYMBOL USPEFI=27             % SET PERIPHERAL FILE
000000   SYMBOL UMGFIL=30             % GET FILE NAME
000000   SYMBOL UDSCNT=31             % DISCONNECT ALL FILE ACCESS CONNECTIONS (LOG OUT)
000000   SYMBOL USTEMP=32             % SET TEMPORARY FILE
000000   SYMBOL UGDIEN=33             % GET DIRECTORY ENTRY
000000   SYMBOL UFDFDI=34             % FIND USERS DEFAULT DIRECTORY
000000   SYMBOL UCOPAG=35             % COPY PAGES BETWEEN TWO OPENED FILES
000000   SYMBOL UDELPG=36             % DELETE PAGES OF A FILE
000000   SYMBOL UFOBJN=37             % FIND DIRECTORY USER AND OBJECT INDEX
000000   SYMBOL USTERM=40             % SET TERMINAL FILE
000000   SYMBOL UEXPFI=41             % EXPAND FILE
000000   SYMBOL URENFI=42             % RENAME FILE
000000   SYMBOL URTLIO=43             % LIST RTOPEN FILES
000000
000000   SYMBOL TLBUF=172000          % BUFFER PAGE ON FILE USER DATA SEGM
000000                                % USED IN FREA AND FWRT
000000
000000   % ==================================================================
000000
000000   SYMBOL READ=0, WRITE=1                  % TRANSFER FUNCTIONS
000000   SYMBOL COMPARE=3
000000   SYMBOL ADVANCE=10,REVERSE=11
000000   SYMBOL WEOF=12, REWIND=13
000000   SYMBOL ERASE=14
000000   SYMBOL BACKSPACE=15,FORSPACE=16,REWUN=17
000000   SYMBOL SFORM=40,RFORM=42
000000   SYMBOL TSPTR=42                         % TEST DISK PACK FOR SPARE TRACKS.
000000   SYMBOL DREAD=60, DWRTE=61, DCOMP=63     % AS FOR READ, WRITE AND COMPARE BUT WITH DOUBLE WORD BLOCK ADDRESS
000000
000000   SYMBOL REC80=1000                       % READ/WRITE 80 BYTES RECORD
000000
000000   %=====================================================================
000000   % 1.2   M A C R O S
000000
000000   % REGISTER DEFINITIONS ARE ON FILE FILSYS-RES !!
000000
000000   % MACROS ARE FOUND ON SEPARATE FILE !!
000000
000000   %=====================================================================
```

```
000000   % 1.3   D E V I C E   B U F F E R S
000000
000000   %       EACH DEVICE BUFFER HAS ROOM FOR ONE PAGE (1K).
000000
000000   DISP 4
000000
000000   INTEGER DNUMB                  % DIRECTORY INDEX; 0-377: FILE SYSTEM DIRECTORIES
000000                                  %                  =400 : FOR SPECIAL USE
000000                                  %              BIT 17=1 : NOT USED
000000   INTEGER CPAG1,CPAG2            % CPAG1;  BIT 14-17:PHYSICAL UNIT
000000                                  %         CARTRIDGE OR PHOENIX, BIT 14: FIXED OR REMOVEABLE
000000                                  %                                BIT 15-17: PHYSICAL UNIT
000000   DOUBLE CPAGE=CPAG1             % CURRENT PAGE IN BUFFER
000000   INTEGER LNUMB                  % LOCK NUMBER
000000   INTEGER DBUFB                  % MEMORY BANK IN WHICH DEVICE BUFFERS LIE
000000   INTEGER BUFFER                 % BUFFER ADDRESS RELATIVE TO START OF BANK
000000   DOUBLE DBPAG=DBUFB             % PHYSICAL BUFFER ADDRESS.
000000   INTEGER DBLOC                  % NUMBER OF SECTORS (WORDS)
000000   INTEGER DBLO1                  % FIRST WORD OF BLOCK ADDRESS; THIS IS ONLY USED FOR BIG DISKS AND MAG TAPE.
000000   INTEGER DBLO2                  % FOR FLOPPY, CARTRIDGE AND PHOENIX DISKS ONLY THE SECOND WORD IS USED.
000000   DOUBLE  DBLOA=DBLO1
000000   INTEGER DKFUN                  % TRANSFER FUNCTION + MOST SIG. BITS OF DISK ADDRESS
000000   INTEGER DPNT0,DPNT1,DPNT2,DPNT3    % PARAMETER POINTERS FOR ABSTR
000000   INTEGER PNEXT                  % ADDRESS OF NEXT DBH IN CHAIN. (-1 INDICATES THE LAST)
000000   INTEGER PPREV                  % ADDRESS OF PREVIOUS DBH IN CHAIN. (-1 INDICATES THE FIRST)
000000   INTEGER SECTP                  % NUMBER OF SECTORS IN PAGE (FOR DIRECT TRANSFER)
000000                                  % (IN LEFT PART OF WORD !)
000000   % FOR SPECIAL BUFFER USE
000000
000000   INTEGER PART1=CPAG1
000000
000000   PSID
000000
000000   SYMBOL DBLEN=25                % DEVICE BUFFER HEADER LENGTH
000000
000000
000000   INTEGER DEVBU=?                % START OF DEVICE BUFFER HEADERS
000000   INTEGER ENDBU=?                % END OF DEVICE BUFFER HEADERS
000000
000000   % BUFFERS DEFINED ON SIN-GEN FILES
000000
000000   %==========================================================================
000000   % 1.4   D I R E C T O R Y   T A B L E
000000
000000   %       THERE IS ONE DIRECTORY ENTRY FOR EACH FILE UNIT IN THE SYSTEM.
000000
000000   DISP 0
000000
000000   INTEGER DIRFL                  % FLAG WORD
000000      SYMBOL  DENTE=17            % DIRECTORY ENTERED BIT
000000      SYMBOL  DMAIN=16            % MAIN DEVICE BIT
000000      SYMBOL  DTAPE=15            % TAPE FLAG BIT
000000      SYMBOL  DDAUF=14            % DEFAULT DIRECTORY BIT
000000      SYMBOL  DTUSE=13            % DIRECTORY RESERVED FOR SPECIAL USE BIT.
000000                                  % BIT 12-0: FILE OPEN COUNT
000000   INTEGER DUNIT                  % PHYSICAL/LOGICAL UNIT AND DEVICE NUMBER
000000                                  %                                BIT 17-14: UNIT NUMBER
000000                                  %                                BIT 13- 0: LOGICAL DEVICE NUMBER
000000   INTEGER LUNIT                  % LOGICAL SUB-UNIT AND NAME INDEX
000000                                  %                                BIT 7-0: NAME INDEX
```

```
000000                                  %          IF SMD DISK:          BIT 17-11: LOGICAL SUB-UNIT.
000000                                  %          IF CARTRIDGE DISK:
000000                                  %                                BIT 17-11: SUB-UNIT
000000                                  %                                BIT 10: 0:REMOVEABLE, 1:FIXED
000000                                  %          IF FLOPPY:
000000                                  %                                BIT 17-14: SECTOR SIZE MULTPLY FACTOR
000000                                  %                                BIT 13-10: DEVICE SIZE MULTIPLY FACTOR
000000   INTEGER DLOCK                  % LOCK NUMBER FOR DIRECTORY
000000   INTEGER DRESE                  % RESERVE NUMBER FOR DIRECTORY
000000   INTEGER ARRAY DNAME(10)        % DIRECTORY NAME                .
000000   DOUBLE OBFIL                   % OBJECT FILE POINTER           .
000000   DOUBLE USFIL                   % USER FILE POINTER             . MASTER BLOCK FROM DISK.
000000   DOUBLE BIFIL                   % BIT FILE POINTER   .          .
000000   DOUBLE PLEFT                   % PAGES LEFT (NOT RESERVED)     .
000000
000000   INTEGER NMUSM                  % NUMBER OF USERS LOGGED IN WITH THIS DIRECTORY AS THEIR MAIN DIRECTORY.
000000   INTEGER SURUS=NMUSM            % CUSER OF USER RESERVING DIRECTORY FOR SPECIAL USE.
000000   INTEGER NMUSD                  % NUMBER OF USERS LOGGED IN WITH THIS DIRECTORY AS THEIR DEFAULT DIRECTORY.
000000   INTEGER SURTM=NMUSD            % TERMINAL NUMBER OF USER RESERVING DIRECTORY FOR SPECIAL USE.
000000
000000   PSID
000000
000000   SYMBOL DTLEN=27                % DIRECTORY TABLE LENGTH
000000
000000   SYMBOL MBLEN=20                % LENTH OF MASTER BLOCK ENTRY
000000
000000
000000   INTEGER DIRTAB=?                          % START OF THE DIRECTORY TABLE
000000   INTEGER ENDDTAB=?                         % END OF THE DIRECTORY TABLE
000000   INTEGER ENDHD=?                           % END OF HARD DISK DIRECTORIES.
000000   INTEGER ENDDD=?                           % END OF DISK DIRECTORIES.
000000
000000   %==========================================================================
000000   % 1.5   N A M E   T A B L E
000000
000000   %      THERE IS ONE NAME TABLE ENTRY FOR EACH DEVICE TYPE IN THE SYSTEM.
000000
000000   DISP 0
000000
000000   INTEGER ARRAY DVNAM(10)        % DEVICE NAME (0 MEANS SAME NAME AS PREVIOUS ENTRY,
000000                                  % NFLAG BIT 6-3 AND PAVAI IS THEN DIFFERENT)
000000   INTEGER PAVA1,PAVA2
000000   DOUBLE  PAVAI=PAVA1            % PAGES AVAILABLE ON DEVICE
000000   INTEGER SECTO                  % SECTOR SIZE IN WORDS.
000000   INTEGER NFLAG                  % FLAG WORD
000000      SYMBOL CTBIT=17             % CARTRIDGE DISC BIT
000000      SYMBOL DRBIT=16             % DRUM DEVICE BIT
000000      SYMBOL 1USER=15             % SINGLE USER DEVICE BIT (FLOPP. AND MAG-TAPE)
000000      SYMBOL 10BIT=14             % 10 MBYTE CARTRIDGE DISK BIT.
000000      SYMBOL MTBIT=13             % MAG TAPE BIT.
000000      SYMBOL ECCBT=12             % ECC DISK BIT: I.E. 30/60/90, 38,75,288,150 MBYTE DISK DRIVES.
000000      SYMBOL BDBIT=11             % BIG DISK BIT: I.E. 33 AND 66 MBYTE DISKS.
000000   %          5FLOP=10            % FLOPPY DISK BIT (DEFINED IN SIN-DATA)
000000      SYMBOL 5PHOE=7              % PHOENIX DISK BIT
000000      SYMBOL LWBIT=6              % LITTLE WINCHESTER DISK: I.E. 14, 21,23 AND 45 MBYTE.
000000                                  % NFLAG BIT 0-2: NO. OF DIR. PRESENT IN
000000                                  % DIRECTORY TABLE FOR EACH UNIT.
000000   INTEGER PTRNS                  % ADDRESS OF DRIVER RROUTINE
000000   INTEGER NLOCK                  % LOCK NUMBER FOR MONITOR CALL PROGRAM
000000
```

2000
64000
74000
132000
156000

54164
54211

4 55
5 DD

```
000000   PSID
000000
000000   SYMBOL NTLEN=16                % NAME ENTRY LENGTH
000000
000000   INTEGER NAMTAB=?
000000   INTEGER ENDNT=?
000000
000000   % FLOPPY FORMAT TABLE.
000000
000000   % IN CONNECTION WITH THE NAME TABLE, THERE IS A FLOPPY FORMAT TABLE.
000000   % THE TABLE HAS ONE WORD FOR EACH POSSIBLE FLOPPY FORMAT:
000000
000000   % BIT 17-14 : SECTOR SIZE (MULTIPLY FACTOR FOR FORMAT 0 SIZE)
000000   % BIT 14-10 : DEVICE SIZE (MULTIPLY FACTOR FOR FORMAT 0 SIZE)
000000   % BIT 7-0 :   FORMAT NUMBER
000000
000000   % -1 MARKS END OF TABLE
000000
000000   INTEGER FFTAB=?
000000
000000   %==============================================================================
000000   % 1.6   S Y S T E M   S E G M E N T
000000
000000   %==============================================================================
000000   % 1.6.1   A U X I L I A R Y
000000
000000   INTEGER TDVN=?
000000   INTEGER CUSER=?
000000   INTEGER CRTREF=?
000000
000000   %==============================================================================
000000   % 1.6.2   O P E N   F I L E   N U M B E R   T A B L E
000000
000000   SYMBOL FZERO=100               % FILE NUMBER FOR FIRST FILE
000000   SYMBOL FSPOO=120               % FILE NUMBER FOR FIRST SPOOLING FILE
000000   INTEGER DV100=?                % START OF OPEN FILE NUMBER TABLE
000000
000000   %==============================================================================
000000   % 1.6.3   O P E N   F I L E   T A B L E
000000
000000   % THERE IS ONE 64 WORD OPEN FILE TABLE ENTRY FOR EACH MASS STORAGE
000000   % FILE OPENED.
000000   % NOTE: COSMOS REMOTE OPENED FILES USE A DIFFERENT TABLE STRUCTURE.
000000
000000   DISP 4
000000
000000   INTEGER RWFIELD                % DATAFIELD ADDRESS FOR MONITOR CALLS
000000   INTEGER OFACC                  % OPENED ACCESS CODE
000000   INTEGER OFFTP                  % FILE TYPE
000000   INTEGER OFFLG                  % FLAG WORD
000000      SYMBOL PERMF=17             % PERMANENT OPENED FILE
000000      SYMBOL INFLG=16             % CHANGE INDEX BUFFER FLAG
000000                                  % 0 - FIRST BUFFER LAST CHANGED
000000                                  % 1 - LAST BUFFER LAST CHANGED
000000
000000      SYMBOL OFWRT=15             % FILE OPENED FOR WRITE
000000      SYMBOL INDB1=14             % WRITE BACK INDEX BUFFER ONE
000000      SYMBOL INDB2=13             % WRITE BACK INDEX BUFFER TWO
000000      SYMBOL OFSCR=12             % SCRATCH FILE
000000      SYMBOL 5REMPO=11            % REMOTE OPENED FILE
000000      SYMBOL 5RCFIL=10            % REMOTE OPENED COSMOS FILE
```

```
000000   INTEGER OFBUF                  % BUFFER POINTER
000000   INTEGER OFNBR=OFBUF            % OR OPEN FILE NUMBER (REMOTE FILE)
000000   INTEGER OFLIB                  % LIBRARY BUFFER POINTER
000000   INTEGER OFLOC=OFLIB            % LOCAL FILE NUMBER OF REMOTE OPENED FILE
000000   INTEGER OFPDIR=OFLIB           % DIRECTORY INDEX OF MAG.TAPE OR FLOPPY
000000                                  % OPENED AS PERIPHERAL
000000   INTEGER OFCB                   % CURRENT BUFFER FILLING INTO
000000   INTEGER OFNB                   % NUMBER OF BUFFERS IN BUFFER QUEUE
000000   INTEGER OFBLZ                  % LOGICAL BLOCK SIZE
000000   INTEGER OFDIR                  % DIRECTORY INDEX
000000   INTEGER OFOBJ                  % OBJECT INDEX
000000   INTEGER OFIP1,OFIP2
000000   DOUBLE OFIP=OFIP1              % BYTE POINTER (CURRENT)
000000   INTEGER OFOP1,OFOP2
000000   DOUBLE OFOP=OFOP1              % BYTE POINTER (MAX)
000000   INTEGER OFIND                  % INPUT DATAFIELD IF PERIFHERAL
000000   INTEGER OFOUD                  % OUTPUT DATAFIELD IF PERIFHERAL
000000   DOUBLE OFIOD=OFIND             % DATAFIELDS POINTERS
000000   DOUBLE OFPAG=OFIND             % NUMBER OF PAGES EXPANDED OR NUMBER
000000                                  % OF PAGES IN FILE IF CONTINIOUS FILE
000000   DOUBLE OFFP                    % FILE POINTER
000000   DOUBLE INDX1                   % CURRENT INDEX IN FIRST INDEX BUFFER
000000   DOUBLE ARRAY INDA1(10)         % FIRST INDEX BUFFER
000000   DOUBLE INDX2                   % CURRENT INDEX IN SECOND INDEX BUFFER
000000   DOUBLE ARRAY INDA2(10)         % SECOND INDEX BUFFER
000000
000000   PSID
000000
000000   %==========================================================================
000000   % 1.6.4    B U F F E R   P O O L   F O R   O P E N   F I L E   T A B L E S ,
000000   %         L I B R A R Y   T A B L E S   A N D   S E Q U E N T I A L
000000   %         B U F F E R S
000000
000000   % THERE MUST BE AT LEAST ONE 64 WORD BUFFER FOR EACH
000000   % OPEN FILE NUMBER.   EACH BUFFER IS PRECEEDED BY A LINK CELL
000000
000000   DISP -1
000000
000000   INTEGER BLINK                  % BUFFER LINK
000000   INTEGER ARRAY BDATA(100)       % BUFFER DATA
000000
000000   PSID
000000
000000
000000   %==========================================================================
000000   % 1.7   B I T   F I L E   B U F F E R
000000
000000   %      THERE IS ONE 16 WORD BUFFER FOR EACH DISK DIRECTORY ENTRY.
000000   %      ONE BUFFER WILL HOLD CURRENT PART OF THE BIT FILE.
000000
000000   DISP 0
000000
000000   INTEGER BDIRI                  % DIRECTORY INDEX
000000   INTEGER BPART                  % CURRENT 16 WORD BLOCK IN BUFFER
000000   INTEGER BLOCN                  % LOCK NUMBER FOR BIT FILE BUFFER
000000   INTEGER ARRAY BBUFF(20)        % 16 WORDS BIT TABLE BUFFER
000000
000000   PSID
000000
000000   SYMBOL BBLEN=23                % LENGTH OF BIT FILE BUFFER
```

```
000000
000000    INTEGER BFBUF=?
000000    INTEGER ENDBF=?
000000
000000    %=========================================================================
000000    %
000000    % 1.8   U S E R   F I L E   B U F F E R
000000
000000    %       THERE IS ONE 32 WORD USER FILE BUFFER. THE BUFFER IS PRECEDED BY A
000000    %       16 WORD BUFFER FOR 10 INDEXES POINTING TO THE CURRENT USER FILE.
000000
000000    DISP 0
000000
000000    INTEGER ULOCK                  % LOCK NUMBER FOR USER FILE BUFFER
000000    INTEGER UDIRI                  % DIRECTORY INDEX
000000    INTEGER UPART                  % CURRENT USER INDEX IN BUFFER
000000    DOUBLE  UINDP                  % FIRST INDEX NUMBER IN INDEX BUFFER
000000    DOUBLE  ARRAY UINDX(10)        % ARRAY FOR 10 INDEXES
000000    INTEGER UENTE                  % ENTER COUNT AND USED FLAG (START OF USER ENTRY).
000000       SYMBOL UUSED=17             % USED FLAG BIT
000000       SYMBOL UOFLG=10             % USER/OBJECT ENTRY FLAG (1 FOR USER ENTRY)
000000    INTEGER ARRAY UNAME(10)        % USER NAME (16 BYTES)
000000    INTEGER UPASS                  % PASSWORD
000000    DOUBLE  UDATE                  % DATE CREATED
000000    DOUBLE  UDENT                  % LAST DATE ENTERED
000000    DOUBLE  UPAVA                  % PAGES AVAILABLE FOR THIS USER
000000    INTEGER UPUS1,UPUS2
000000    DOUBLE  UPUSE=UPUS1            % PAGES USED BY THIS USER
000000    INTEGER UNDEX                  % USER INDEX OF THIS ENTRY
000000    INTEGER MAILF                  % MAIL FLAG
000000    INTEGER DFIAC                  % DEFAULT FILE ACCESS
000000       SYMBOL MESSF=0              % MESSAGE FLAG BIT
000000       SYMBOL BROAF=1              % BROADCAST FLAG BIT
000000    INTEGER USPRV                  % PREVIOUS USER ENTRY.
000000    INTEGER USNXT                  % NEXT USER ENTRY.
000000    INTEGER UFREE                  % FREE
000000    INTEGER ARRAY UFRIE(10)        % FRIEND TABLE FOR  8 FRIENDS
000000
000000    PSID
000000
000000    SYMBOL  NUSER=400              % MAX NUMBER OF USERS ON A DEVICE
000000
000000    %=========================================================================
000000    % 1.9   O B J E C T   F I L E   B U F F E R
000000
000000    %       THERE IS ONE 32 WORD OBJECT FILE BUFFER. THE BUFFER IS PRECEDED BY A
000000    %       16 WORD BUFFER FOR 10 INDEXES POINTING TO THE CURRENT OBJECT FILE.
000000
000000    DISP 0
000000
000000    INTEGER OLOCK                  % LOCK NUMBER FOR OBJECT FILE BUFFER
000000    INTEGER ODIRI                  % DIRECTORY INDEX
000000    INTEGER OPART                  % CURRENT OBJECT INDEX
000000    DOUBLE  OINDP                  % FIRST INDEX NUMBER IN INDEX BUFFER
000000    DOUBLE  ARRAY OINDX(10)        % ARRAY FOR 10 INDEXES
000000    INTEGER OFLAG                  % FLAGS (START OF OBJECT ENTRY)
000000       SYMBOL OUSED=17             % USED FLAG BIT
000000       SYMBOL OWRTE=16             % OPENED FOR WRITE FLAG BIT
000000       SYMBOL ORESE=15             % FILE RESERVED BIT
000000       SYMBOL OBACK=14             % SPOOLING FILE USED
```

```
000000      SYMBOL OCOMM=13          % FILE OPENED FOR COMMON ACCESS.
000000                               % BITS 0-12: TERMINAL NUMBER OF THE RESERVING USER; =0 : FILE RESERVED BY AN
000000   INTEGER ARRAY ONAME(10)     % OBJECT NAME (16 BYTES)
000000   INTEGER ARRAY OTYPE(2)      % TYPE (4 BYTES)
000000   INTEGER ONEXT               % NEXT VERSION
000000   INTEGER OPREV               % PREVIOUS VERSION
000000   INTEGER OACCE               % ACCES WORD
000000   INTEGER OFTYP               % FILE TYPE
000000      SYMBOL OTBIT=0           % TERMINAL FILE
000000      SYMBOL OPBIT=1           % PERIFHERAL FILE
000000      SYMBOL OSBIT=2           % SPOOLING FILE
000000      SYMBOL OIBIT=3           % INDEX-SEQUENTIAL FILE
000000      SYMBOL OCBIT=4           % CONTINIOUS FILE
000000      SYMBOL OABIT=5           % ALLOCATED FILE
000000      SYMBOL OMBIT=6           % MAG.TAPE FILE
000000      SYMBOL OLBIT=7           % LIBRARY FILE
000000      SYMBOL OTMBIT=10         % TEMPORARY FILE
000000   INTEGER ODEVN               % DEVICE NUMBER
000000   INTEGER OUSER               % USER NUMBER OF RESERVING USER
000000   INTEGER ONDEX               % OBJECT INDEX OF THIS OBJECT ENTRY
000000   INTEGER OCOUN               % OOPEN COUNT
000000   INTEGER OOPEN               % TOTAL NUMBER OF TIMES OPENED
000000   DOUBLE  ODATC               % DATE CREATED
000000   DOUBLE  ODATR               % LSAT DATE OPENED FOR READ
000000   DOUBLE  ODATW               % LAST DATE OPENED FOR WRITE
000000   DOUBLE  OPAGE               % PAGES IN FILE
000000   DOUBLE  OBYTE               % BYTES IN FILE
000000   DOUBLE  OPOIN               % FILE POINTER
000000      SYMBOL SUBIN=17          % SUBINDEX POINTER
000000      SYMBOL INDX=16           % INDEX POINTER
000000
000000   PSID
000000
000000   SYMBOL NOBJE=400            % MAX NUMBER OF OBJECTS PR. FILE
000000
```

```
000000   *
000000   %==========================================================================
000000   % 1.11   S P O O L I N G   D A T A
000000
000000   %%   S P O O L I N G   F O R   S I N T R A N   I I I
000000
000000   %% DESIGN GOALS:
000000   %% - SPOOL PERIPHERAL OUTPUT VIA MASS STORAGE FILES
000000   %% - AUTOMATIC PRINTING OF THE SPOOLING FILES
000000   %% - MAINTAIN A QUEUE OF SPOOLING FILES READY FOR PRINTING
000000   %% - USER FILES AND SPOOLING FILES MAY BE MIXED IN THE SPOOLING QUEUE
000000   %% - ANY NUMBER OF COPIES OF A FILE ALLOWED
000000   %% - ANY NUMBER OF PERIPHERAL UNITS OF THE SAME TYPE MAY PRINT FROM THE
000000   %%   SAME SPOOLING QUEUE
000000   %% - A LIMITED NUMBER OF SYSTEM PAGES ARE USED FOR SPOOLING FILES
000000   %% - NO PRINT IS LOST, EVEN WHEN THE SPOOLING PAGE AREA IS FULL
000000   %% - THE USER SHALL NOT SEE THE DIFFERENCE BETWEEN SPOOLING AND NO SPOOLING
000000   %%   EXCEPT AS DELAY IN THE PRINT-OUT, AND HEADER AND TRAILER ON THE PRINT.
000000
000000
000000   %% THE FOLLOWING COMMANDS ARE AVAILABLE:
000000
000000   % START-SPOOLING:          START-THE-SPOOLING-PROCESS-FOR-SPECIFIED-PERIPHERAL
000000
000000   % STOP-SPOOLING:           STOP-THE-SPOOLING-PROCESSES-FOR-SPECIFIED-PERIPHERAL-
000000   %                              AFTER-CURRENT-PRINT-OUT
000000
000000   % ABORT-SPOOLING-PRINT:    ABORT-THE-CURRENT-PRINT-OUT-ON-SPECIFIED-PERIPHERAL
000000
000000   % LIST-SPOOLING-QUEUE:     LIST-THE-FILES-IN-THE-SPOOLING-QUEUE-FOR-SPECIFIED-
000000   %                              PERIPHERAL
000000
000000   % APPEND-SPOOLING-FILE:    APPEND-SPECIFIED-FILE-TO-THE-SPOOLING-QUEUE-FOR-
000000   %                              SPECIFIED-PERIPHERAL
000000
000000   % DELETE-SPOOLING-FILE:    DELETE-SPECIFIED-FILE-FROM-THE-SPOOLING-
000000   %                              QUEUE-FOR-SPECIFIED-PERIPHERAL
000000
000000   % GIVE-SPOOLING-SPACE:     INCREMENT-NUMBER-OF-SYSTEM-PAGES-THAT-MAY-BE-USED-FOR-
000000   %                              SPOOLING
000000
000000   % TAKE-SPOOLING-SPACE:     DECREMENT-NUMBER-OF-SYSTEM-PAGES-THAT-MAY-BE-USED-FOR-
000000   %                              SPOOLING
000000
000000   % SPOOLING-PAGES-LEFT:     GIVE-NUMBER-OF-PAGES-LEFT-FOR-SPOOLING
000000
000000
000000   %% THE SPOOLING PROCESSES ARE IMPLEMENTED AS A SET OF RT-PROGRAMS.
000000   %% EACH RT-PROGRAM HAS THE FOLLOWING STRUCTURE:
000000
000000   % SPOOLING-PROGRAM:        DO
000000   %                                WHILE NO-STOP-SPOOLING-COMMAND
000000   %                              TAKE-FILE-FROM-SPOOLING-QUEUE
000000   %                              MAKE-SPECIFIED-NUMBER-OF-COPIES-OF-THE-FILE-
000000   %                              ON-THE-PERIPHERAL
000000   %                              OD
000000
000000   %% NOTE THAT MORE THAN ONE SPOOLING PROGRAM MAY TAKE SPOOLING FILES FROM
000000   %% THE SAME QUEUE
000000
```

```
000000
000000   %% SOME OF THE FILE SYSTEM ROUTINES ARE AFFECTED:
000000
000000   % FIND-FILE-TO-OPEN:      ......
000000   %                            IF      SPOOLING-PERIPHERAL
000000   %                              THEN  FIND-FREE-SPOOLING-FILE
000000   %                            FI
000000   %                            ......
000000
000000   % FILE-CLOSE:             ......
000000   %                            IF      SPOOLING-FILE
000000   %                              THEN  IF    OPENED-FOR-READ
000000   %                              THEN  SET-AS-FREE-SPOOLING-FILE
000000   %                              ELSE  INSERT-FILE-IN-SPOOLING-QUEUE
000000   %                            FI
000000   %                            FI
000000   %                            ......
000000
000000   % GET-PAGE-FOR-FILE:      ......
000000   %                            IF      SPOOLING-FILE-AND-NO-MORE-PAGES-FOR-SPOOLING
000000   %                              THEN  LET-PROGRAM-WAIT-FOR-PAGES
000000   %                            FI
000000   %                            ......
000000
000000   % SET-PERIPHERAL-FILE:    ......
000000   %                            IF      MORE-VERSIONS-OF-THE-FILE
000000   %                              THEN  SET-AS-SPOOLING-FILES-IF-NOT-ALREADY-
000000   %                            PERIPHERAL-FILES
000000   %                            FI
000000   %                            ......
000000
000000   % CREATE-NEW-VERSION:     ......
000000   %                            IF      OLD-VERSION-IS-A-PERIPHERAL-OR-SPOOLING-FILE
000000   %                              THEN  SET-THIS-VERSION-AS-A-SPOOLING-FILE
000000   %                            FI
000000
000000
000000
000000   %% THE SPOOLING FILES MUST HAVE PUBLIC WRITE ACCESS AND SYSTEM READ AND WRITE
000000   %% ACCESS, BUT NOT PUBLIC READ ACCESS
000000   %% ALL OTHER FILES LINKED TO THE SPOOLING QUEUE MUST HAVE PUBLIC READ ACCESS,
000000   %% OR THE OWNER MUST BE A FRIEND OF THE SYSTEM AND THE FILE MUST HAVE
000000   %% FRIEND READ ACCESS.
000000
000000
000000   %% THE SPOOLING SYSTEM MUST BE TESTED FOR THE FOLLOWING NON-TRIVIAL CASES:
000000   %% - TWO SPOOLING PROGRAMS TAKING FILES FROM THE SAME QUEUE
000000   %% - TWO OR MORE SPOOLING QUEUES
000000   %% - DELETING OF FILES THAT ARE ELEMENTS IN A SPOOLING QUEUE
000000   %% - LINKING FILES TO THE SPOOLING QUEUE THAT DOES NOT HAVE SUFFICIENT
000000   %%   ACCESS
000000   %% - SPOOLING SPACE FULL AND SPOOLING FILES IN THE QUEUES
000000   %% - SPOOLING SPACE FULL AND NO SPOOLING FILES IN THE QUEUES, BUT ONE PRINTING
000000   %% - SPOOLING SPACE FULL AND NO SPOOLING FILES IN THE QUEUES OR PRINTING, I.E.
000000   %%   INITIATE FORCED PRINTING
000000   %% - FORCED PRINTING INITIATED AND SPOOLING SPACE FULL FOR ANOTHER SPOOLING
000000   %%   FILE
```

```
000000
000000   %%   G L O B A L   S P O O L I N G   D A T A
000000
000000   %% THE SPOOLING SYSTEM USES SOME GLOBAL VARIABLES, ALL CORE RESIDENT.
000000
000000   %==========================================================================
000000   % 1.11.1   A U X I L I A R Y   D A T A
000000
000000   % SEMAPHORES:
000000   % - SPOOLING-SEMAPHORE        RESERVED WHEN THE FREE-SPOOLING-PAGES-COUNTER
000000   %                             AND PRINTING-OR-WAITING-FOR-PRINT-COUNTER ARE
000000   %                             USED
000000   % - WAIT-FOR-USED-PAGES-SEMAPHORE RESERVED AND RELEASED BY THE SPOOLING
000000   %                              PROGRAMS TO WAIT FOR MORE PAGES UNDER FORCED
000000   %                              PRINT, RELEASED AND RESERVED BY GET-PAGE-
000000   %                              FOR-FILE AND FILE-CLOSE TO RESTART
000000   %                              THE WAITING SPOOLING PROGRAM
000000   % - WAIT-FOR-FREE-PAGES-SEMAPHORE RESERVED AND RELEASED BY GET-PAGE-FOR-FILE
000000   %                              TO LET THE CALLING PROGRAM WAIT FOR SPOOLING
000000   %                              PAGES
000000   % - TWO SEMAPHORES FOR EACH QUEUE:
000000   %   QUEUE-SEMAPHORE            USED TO LOCK AND UNLOCK THE QUEUE
000000   %   QUEUE-I/O-SEMAPHORE        USED TO WAIT FOR SOMETHING TO BE INSERTED IN
000000   %                              THE QUEUE
000000
000000   INTEGER SLDN=?            % SPOOLING SEMAPHORE
000000   INTEGER WULDN=?           % WAIT FOR USED PAGES SEMAPHORE
000000   INTEGER WFLDN=?           % WAIT FOR FREE PAGES SEMAPHORE
000000   INTEGER FDFSE=?           % FORM DATAFIELDS SEMAPHORE
000000
000000   % GLOBAL VARIABLES:
000000   % - FREE-SPOOLING-PAGES-COUNTER CONTAINS NUMBER OF PAGES LEFT TO USE FOR
000000   %                            SPOOLING FILES
000000   % - SPOOLING-FILES-PRINTING-OR-WAITING-FOR-PRINT-COUNTER NUMBER OF
000000   %                            SPOOLING FILES IN THE QUEUES OR CURRENTLY
000000   %                            PROCESSED BY SPOOLING PROGRAMS
000000   % - PROGRAM-WAITING-FOR-FORCED-PRINT (0 IF NO PROGRAM WAITING)
000000
000000   INTEGER XSPFMESS=?            % SPOOLING FILE MESSAGE (ON SYSTEM SEGMENT)
000000   INTEGER FXSPFMESS=?
000000
000000   %==========================================================================
000000   % 1.11.2   S P O O L I N G   D A T A F I E L D
000000
000000   DISP -40
000000   INTEGER ARRAY ACCBUFF(40)     % GENERAL PURPOSE BUFFER USED BY SPOOLING PROGRAM
000000   PSID
000000   DISP 1
000000   INTEGER SPROG             % ADDRESS OF RT-DESCRIPTION
000000   INTEGER SPERI             % PERIPHERAL DEVICE NUMBER
000000   INTEGER SQUEU             % SPOOLING QUEUE NUMBER
000000   INTEGER SQSEM             % SPOOLING QUEUE SEMAPHORE
000000   INTEGER SQIOS             % SPOOLING QUEUE I/O SEMAPHORE
000000   INTEGER SSTOP             % STOP COMMAND FLAG
000000   INTEGER SABOR             % ABORT COMMAND FLAG
000000                                        % 1 - ABORT CURRENT PRINT
000000                                        % 2 - RESTART CURRENT PRINT
000000                                        % 3 - STOP PRINT
000000      SYMBOL BBASP=10                   %     BACKSPACE PRINT
000000      SYMBOL BFSSP=11                   %     FORWARD SPACE PRINT
```

```
000000      SYMBOL BNFIQ=12                          %     NEW FILE IS SET FIRST IN QUEUE
000000   INTEGER SPINX              % SPOOLING INDEX (1 - 25)
000000   INTEGER POINTER HEADER,TRAILER              % DEVICE DEPENDENT ROUTINES
000000   INTEGER POINTER PRINTBUFFER                 %            "
000000   INTEGER FILENUMBER                          % FILENUMBER
000000   INTEGER LASTPAGE,PAGENUMBER                 % LAST PAGENO IF ANY AND CURRENT PAGE
000000   INTEGER REMAINING,BYTENUMBER                % BYTENO ON LAST PAGE AND CURRENT BYTENO
000000   INTEGER BUFHAD                              % ADDRESS OF DEVICE BUFFER RESERVED FOR SPECIAL USE
000000   INTEGER PCBNK                               % DEVICE BUFFER ADDRESS, BANK NO
000000   INTEGER PCADR                               % DEVICE BUFFER ADDRESS WITHIN BANK
000000   DOUBLE PCORAD=PCBNK                         % PHYSICAL ADDRESS OF DEVICE BUFFER
000000   INTEGER SNPAGES                             % NUMBER OF PAGES TO BACKSPACE/ADVANCE
000000   INTEGER SNLINES                             % NUMBER OF LINES TO BACKSPACE/ADVANCE
000000   INTEGER SCONDITIONS            %  = 0 - DO NOT PRINT FILE NAME
000000                                  %  = 1 - PRINT FILE NAME BUT DO NOT STOP.
000000                                  %  = 2 - PRINT FILE NAME AND WAIT FOR START-PRINT.
000000   INTEGER NLPAGE                 % NUMBER OF LINES PER PAGE
000000   INTEGER SPMODE                 % SPOOLING MODE; 0: PRINT; 1: PLOT
000000   INTEGER SPAGENO                % SAVED PAGENUMBER
000000   INTEGER WSNLINE                % SAVED SNLINE
000000   INTEGER NOSINQ=WSNLINE         % NO OF ELEMENTS IN SPOOLING QUEUE
000000   INTEGER WSNPAGE                % SAVED SNPAGE
000000   INTEGER NUMINQ=WSNPAGE         % NO OF ELEMENTS IN SPOOLING QUEUE
000000   INTEGER LFCOUNT                % LINE-FEED COUNTER
000000   INTEGER ARRAY SPFNAM(16)       % PERIPHERAL FILE NAME FOR SPOOLING DEVICE
000000   INTEGER STORX                  % GENERAL SAVE LOCATION
000000   INTEGER BUFFAD                 % ADDRESS OF GENERAL DATA BUFFER
000000   INTEGER FPAR1LIST,FPAR2LIST    % GENERAL PARAMETER LIST
000000   INTEGER FPAR3LIST              %
000000   INTEGER NBYTS                  % NUMBER OF BYTES TO PRINT (FOR OUTSTRING)
000000   INTEGER POINTER LNKSP          % RETURN ADDRESS
000000   INTEGER SAVFN                  % SAVED FILE NUMBER
000000   INTEGER STORT                  % GENERAL SAVE LOCATION
000000   INTEGER DBHCACHE               % ADDRESS OF DEVICE BUFFER HEADER IN DISC CACHE
000000   INTEGER SOUR1,SOUR2            % ADDRESS OF DEVICE BUFFER IN CACHE
000000   DOUBLE  SOURX=SOUR1            % ADDRESS OF DEVICE BUFFER IN CACHE
000000   INTEGER RFIEL                  % ADDRESS OF SPOOLING DATAFIELD IN RESIDENT
000000   INTEGER ARRAY SBUFR(0)         % QUEUE ELEMENT BUFFER
000000   INTEGER NOCOPYS                          % FIRST WORD=NO OF COPIES
000000   INTEGER FSPMESS                          % FLAG FOR SPOOLING-FILE-MESSAGE (RIGHT BYTE)
000000                                            % AND BACKGROUND/RT (LEFT BYTE)
000000   INTEGER ARRAY FNAME(57)                  % POSITION OF SPOOLING FILENAME
000000   INTEGER NAME1=FNAME                      %
000000   INTEGER ARRAY SPJNAME(0)                 % PROJECT NAME BUFFER
000000   INTEGER SPJN1                                % 0 IF NO ACC., RT-DESCR IF RT-ACC
000000   INTEGER SPJN2                                % ACCTAB INDEX IF RT-ACC.
000000   INTEGER ARRAY SPJN3(6)
000000   INTEGER ARRAY SPMESSBU(117)              % USER MESSAGE BUFFER
000000                                  % END OF QUEUE ELEMENT BUFFER
000000   PSID
000000
000000   SYMBOL SPLEN=12                % LENGTH OF TABLE ENTRY IN RESIDENT
000000   INTEGER SPTAB=?                % START OF SPOOLING TABLE IN RESIDENT
000000   INTEGER ENDSP=?                % END OF SPOOLING TABLE IN RESIDENT
000000
000000   SYMBOL SSPLEN=340              % LENGTH OF TABLE ENTRY ON SEGMENT
000000   INTEGER SSPTAB=?               % START OF SPOOLING TABLE ON SEGMENT
000000   INTEGER SENDSP=?               % END OF SPOOLING TABLE ON SEGMENT
000000
000000   %==========================================================================
```

```
000000   % 1.11.3   Q U E U E   E L E M E N T
000000
000000   DISP 0
000000   INTEGER CXNOCOP                 % NUMBER OF COPIES
000000   INTEGER CXSPMESS                % FLAG FOR USER MESSAGE (RIGHT BYTE)
000000                                   % AND BACKGROUND/RT (LEFT BYTE)
000000
000000   INTEGER ARRAY CXFNAME(57)       % FILE NAME
000000   INTEGER CSPFNAM=CXFNAME
000000   INTEGER ARRAY CXSPJNAME(0)      % PROJECT NAME
000000   INTEGER CSPJ1                      % 0 IF NO ACCOUNTING, RT-DESCR IF RT-ACC.
000000   INTEGER CSPJ2                      % ACCTAB INDEX IF RT-ACCOUNTING
000000   INTEGER ARRAY CSPJ3(6)             % PROJECT NAME
000000   INTEGER ARRAY SPUMESS(117)      % USER MESSAGE IN QUEUE ELEMENT
000000   PSID
000000
000000   SYMBOL QBSEM=527                % SPOOLING QUEUE BUFFER LOCK
000000   SYMBOL SQELSIZE=210             % SPOOLING QUEUE ELEMENT SIZE
000000
```

```
000000
000000   @DEV 1
000000   @DEV (S-S-J)XMSG-VALUES
000000
000000   %==========================================================================
000000   %
000000   %       XMSG-VALUES:SYMB  Defines the values for symbolic
000000   %       ----------------  names for functions and error codes.
000000   %
000000   %==========================================================================
000000
000000   @DEC
000000   % =*=*=*= Warning: This file is now in DECIMAL to keep PLANC happy! =*=*=*=
000000
000000   %          F U N C T I O N   V A L U E S
000000
000000   SYMBOL XFDUM=0                 % Dummy function
000000   SYMBOL XFDCT=1                 % Disconnect from message system
000000   SYMBOL XFGET=2                 % Get message space
000000   SYMBOL XFREL=3                 % Release message space
000000   SYMBOL XFRHD=4                 % Read header from a message (6 bytes)
000000   SYMBOL XFWHD=5                 % Write header to a message (6 bytes)
000000   SYMBOL XFREA=6                 % Read from message to user buffer
000000   SYMBOL XFWRI=7                 % Write from user to message
000000   SYMBOL XFSCM=8                 % Set current message
000000   SYMBOL XFMST=9                 % Get message status
000000   SYMBOL XFOPN=10                % Open port
000000   SYMBOL XFCLS=11                % Close port
000000   SYMBOL XFSND=12                % Send message to a remote port
000000   SYMBOL XFRCV=13                % Receive a message on a given port
000000   SYMBOL XFPST=14                % Get local port status
000000   SYMBOL XFGST=15                % General status or wait
000000
000000          % SERVICE FUNCTIONS
000000
000000   SYMBOL XFSIN=16                % Service initialisation function
000000   SYMBOL XFSRL=17                % Service release function (obsolete)
000000   SYMBOL XFABR=18                % Absolute read block from POF area
000000   SYMBOL XFABW=19                % Absolute write block to POF area
000000   SYMBOL XFMLK=20                % Lock message system (obsolete)
000000   SYMBOL XFMUL=21                % Unlock message system (obsolete)
000000   SYMBOL XFM2P=22                % Magic number to port id.
000000   SYMBOL XFP2M=23                % Port to magic number
000000   SYMBOL XFRIN=24                % Routing initialise (obsolete)
000000   SYMBOL XFCRD=25                % Create driver with context
000000   SYMBOL XFSTD=26                % Start driver
000000
000000          % INDIRECT BUFFER HANDLING FUNCTIONS (OPTIONAL)
000000
000000   SYMBOL XFDIB=27                % Define indirect buffer
000000   SYMBOL XFRIB=28                % Read from indirect buffer
000000   SYMBOL XFWIB=29                % Write to indirect buffer
000000
000000          % FUNCTIONS ADDED AFTER THE FIRST RELEASE
000000
000000   SYMBOL XFPRV=30                % Request privilege
000000   SYMBOL XFRTN=31                % Write word 0 and return message
000000   SYMBOL XFRRH=32                % Receive message and read word 0
000000   SYMBOL XFDUB=33                % Define user buffer area for current message
000000   SYMBOL XFWDF=34                % Define wake-up context (drivers only!)
```

```
000000    SYMBOL XFDBK=35                % Define bank no (drivers only)
000000    SYMBOL XFSMC=36                % Start multi-function call
000000    SYMBOL XFDMM=37                % Define max memory for a task
000000    SYMBOL XFALM=38                % Allocate messages to a task
000000    SYMBOL XFFRM=39                % Free allocated messages
000000    SYMBOL X5FUN=40                % == END MARKER == LEAVE ME HERE PLEASE
000000
000000    % BIT VALUES IN FUNCTION CODE REGISTER (T-REG)
000000
000000    SYMBOL XFWTF=15  % If set then wait if operation not terminated
000000    SYMBOL XFWAK=14  % In RCV/PST/GST: Do RTENTRY on status change
000000    %       XFPRM=13  % In XFOPN: Permanent open requested (Obsolete)
000000    SYMBOL XFOPS=12  % In XFOPN: Specified port number required (not impl.)
000000    SYMBOL XFPON=13  % In XFSTD: Driver to run with paging on
000000    SYMBOL XFWOK=13  % In XFDIB: Allow write access to indirect buffer
000000    SYMBOL XFUSG=13  % In XFREA/WRI: Use the user segno defined by XFDUS
000000    SYMBOL XFHIP=13  % In XFSND: If not XFROU then high-priority message
000000    SYMBOL XFRRO=13  % In XFSND: If XFROU then non-local XROUT (mcno in A-reg)
000000    SYMBOL XFRES=12  % In XFWRI: Reset XMLEN to XMPTR (forget rest)
000000    SYMBOL XFBNC=12  % In XFSND: Bounce message
000000    SYMBOL XFFWD=11  % In XFSND: Forward message
000000    SYMBOL XFROU=10  % In XFSND: Message to be sent to XROUT (see XFRRO)
000000    SYMBOL XFSEC=9   % In XFSND: Secure message (Return if not deliv'd)
000000    % === warning: bits 8, 9 (decimal) are used for bank no in XFABR, XFABW ===
000000    SYMBOL XFSYS=7   % Bit set to indicate system mode call (system domain)
000000
000000    % Function codes when calling DRXMSG from drivers
000000
000000    SYMBOL XDINF=1                % Get message info
000000    SYMBOL XDSBP=2                % Set byte pointer
000000    SYMBOL XDPUS=3                % Put byte sequential
000000    SYMBOL XDPUR=4                % Put byte random
000000    SYMBOL XDGES=5                % Get byte sequential
000000    SYMBOL XDGER=6                % Get byte random
000000
000000    % MESSAGE TYPES: RETURNED AS SUCCESSFULL STATUS FROM XFRCV (NB: MAX 4 BITS)
000000
000000    SYMBOL XMTNO=1                % Normal message
000000    SYMBOL XMROU=2                % Routed message (Via XROUT)
000000    SYMBOL XMTHI=3                % High priority message
000000    SYMBOL XMTRE=4                % Return message (Abnormal condition)
000000    SYMBOL XMKIK=5                % XROUT has been kicked (no message)
000000    SYMBOL XMTPS=6                % Pseudo message (not used)
000000    %
000000    %       U S E R    E R R O R    S Y M B O L S   (Returned in T-reg)
000000    %
000000    SYMBOL XKXXX=16896            % Base for standard error system (41000B)
000000    SYMBOL XENOT= -1              % No more XT-blocks free
000000    SYMBOL XEIRM= -2              % Non-local remote port illegal here
000000    SYMBOL XETMM= -4              % Task is not allowed any more memory
000000    SYMBOL XENIM= -5              % Facility not yet implemented
000000    SYMBOL XEIBP= -6              % Illegal message buffer pointer
000000    SYMBOL XEBNY= -7              % Message buffer not yours
000000    SYMBOL XEIRT= -8              % Illegal function for RT-programs (only drivers)
000000    SYMBOL XENOP= -9              % No more ports available
000000    SYMBOL XEIDR= -10             % Function not available to drivers
000000    SYMBOL XENDM= -11             % No default message
000000    SYMBOL XEMCH= -12             % Message is already chained
000000    SYMBOL XEBFC= -13             % Message is in a queue.
000000    SYMBOL XEAIN= -14             % XMSG Kernel already initialised
```

```
000000   SYMBOL XEBNC= -15          % Return of a bounce message
000000   SYMBOL XEWNA= -16          % Write Not Allowed (Indirect buffer)
000000   SYMBOL XENVI= -17          % No Valid Indirect buffer defined
000000   SYMBOL XEILF= -18          % Illegal function code in monitor call
000000   SYMBOL XEIMA= -19          % Invalid magic number
000000   SYMBOL XEMFL= -20          % Message space full
000000   SYMBOL XEILM= -21          % Illegal message size or not enough space left
000000   SYMBOL XEIPN= -22          % Illegal port number
000000   SYMBOL XEPRV= -23          % Privileged function called without privilege.
000000   SYMBOL XEPVR= -24          % Privilege request refused
000000   SYMBOL XERNA= -25          % Remote system not available
000000   SYMBOL XEROV= -26          % Remote task space overflow
000000   SYMBOL XEXBF= -27          % Message already has XMSG buffer (XFDUB)
000000   SYMBOL XEDRI= -28          % Driver called XMSG before return from previous
000000   SYMBOL XENDP= -29          % No port open (so 'default port' param invalid)
000000   SYMBOL XEITL= -30          % Illegal transfer length for read/write
000000   SYMBOL XEIDP= -31          % Illegal displacement in read/write
000000   SYMBOL XEILR= -32          % Illegal use of reentrant segment in XFDIB
000000   SYMBOL XENOS= -33          % Indirect Buffer not on valid segment
000000   SYMBOL XENSE= -34          % Network sequencing error
000000   SYMBOL XERND= -35          % Remote system not defined
000000   SYMBOL XEPCL= -36          % Remote port closed while message queued
000000   SYMBOL XENRU= -37          % XMSG not running === do not move - used in S3 ===
000000   SYMBOL XENTO= -38          % Timeout detected by network layer
000000   SYMBOL XENUS= -39          % No user segment info defined (XFUSG option)
000000   SYMBOL XEREJ= -40          % Network remote reject (request retransmit)
000000   SYMBOL XEIXT= -41          % Driver called XMSG with illegal XT-block
000000   SYMBOL XECRA= -63          % XMSG crash (Info in Basefield)
```

```
000000
000000    %
000000    %       X R O U T   -   S E R V I C E   V A L U E S
000000    %              Values in byte 1 of message. Bit 6 is set => service request
000000    %
000000    SYMBOL XSNUL=64                % Null command returns 0 status to sender
000000    SYMBOL XSLET=65                % Send a letter
000000    SYMBOL XSNAM=66                % Give name to this port
000000    SYMBOL XSCNM=67                % Clear name of this port
000000    SYMBOL XSGNM=68                % Get name of port (param: MAGNO)
000000    SYMBOL XSGNI=69                % Get name (param: MC/PORTNO)
000000    SYMBOL XSGMG=71                % Get magic number from name (PRIV)
000000    SYMBOL XSDRN=73                % Define remote name
000000    SYMBOL XSDSY=74                % Define routing for system N (PRIV)
000000    SYMBOL XSDMC=XSDSY             % Define routing for system N (PRIV) - old
000000    SYMBOL XSGSY=75                % Get routing info for system N
000000    SYMBOL XSGMC=XSGSY             % Get routing info for system N - old symbol
000000    SYMBOL XSLKI=76                % Start up specified link
000000    SYMBOL XSTIN=77                % Initialise tracing (open file, ..)
000000    SYMBOL XSTCL=78                % Close tracing
000000    SYMBOL XSTDC=79                % Define tracing conditions
000000    SYMBOL XSCRS=80                % Create service (name, init no of SP's)
000000    SYMBOL XSNSP=81                % New service point(s) (increment in SP's)
000000    SYMBOL XSGIN=82                % Get information about name
000000    SYMBOL XSDLO=83                % Define local system
000000    SYMBOL XSLEK=84                % Send letter and kick if unavailable
000000    SYMBOL XSNET=85                % Start/stop gateway (network server) (PRIV)
000000    SYMBOL XSSCI=86                % Set crash information
000000    SYMBOL XSMAX=XSSCI             % Maximum legal service value
000000    %
000000    %       X R O U T   /   F I L E   S E R V E R   -   E R R O R S
000000    %
000000    %       Error values returned in byte 1 of return message (Bit 6 reset)
000000    %
000000    SYMBOL XRXXX=16960             % Base for XROUT errors: 41100B
000000    SYMBOL XRSOK=0                 % OK - not an error!
000000    SYMBOL XRISN=1                 % Illegal service number
000000    SYMBOL XRUNN=2                 % No open port has this name
000000    SYMBOL XRDDF=3                 % Another port already has this name
000000    SYMBOL XRNSP=4                 % No space left for names
000000    SYMBOL XRIPT=5                 % Illegal parameter type
000000    SYMBOL XRMMP=6                 % Missing mandatory parameter
000000    SYMBOL XRUNM=7                 % Unknown magic number
000000    SYMBOL XRMTL=8                 % Resulting message too long
000000    SYMBOL XRSMF=9                 % Standard message format not handled
000000    SYMBOL XRPRV=10                % Caller was not privileged
000000    SYMBOL XRISY=11                % Illegal system number parameter
000000    SYMBOL XRNRO=12                % No access to remote system
000000    SYMBOL XRIIV=13                % Illegal integer value
000000    SYMBOL XRNEI=14                % Cannot define route to a neighbour
000000    SYMBOL XRNXM=15                % Invalid service request - no multi-mc XMSG
000000    SYMBOL XRILN=16                % Illegal/Reserved Log. unit no. for link
000000    SYMBOL XRNXL=17                % No more XL-Blocks (Link Descriptors)
000000    SYMBOL XRNXD=18                % Not enough XD-Blocks for LKINI
000000    SYMBOL XRNTR=19                % No trace generated
000000    SYMBOL XRTRA=20                % Trace already active
000000    SYMBOL XRTRP=21                % Trace passive
000000    SYMBOL XRTFE=22                % Trace file open error (see param 1)
000000    SYMBOL XRTRT=23                % Trace RT-prog (XTRACE) not found
000000    SYMBOL XRTIS=24                % Illegal system number
```

```
000000    SYMBOL XRBLK=25              % Bad link - open unsuccessful
000000    SYMBOL XRSYD=26              % Attempt to redefine local system no
000000    SYMBOL XRNLS=27              % No local system number defined yet
000000    SYMBOL XRTRE=28              % Too many remote names to this system
000000    SYMBOL XRRNA=29              % Old service calls (<64) cannot go remote
000000    SYMBOL XRBUS=30              % Service points busy
000000    SYMBOL XRNSE=31              % This is not a service port
000000    SYMBOL XRRPN=32              % Remote port statically declared
000000    SYMBOL XRUKS=33              % Unknown remote system name
000000    SYMBOL XRMFL=34              % Remote system message table space full
000000    SYMBOL XRROV=35              % Remote task message space used up
000000    SYMBOL XRRFU=36              % Routing table full (too many systems)
000000    SYMBOL XRNRB=37              % No remote batch service available
000000    SYMBOL XRURT=38              % Unknown RT name
000000    SYMBOL XRSNR=39              % This server is not running
000000    SYMBOL XRRND=40              % Netserver: remote system is not defined
000000    SYMBOL XRNNA=41              % Netserver: network not available
000000    SYMBOL XRISE=42              % Netserver: internal server error
000000    SYMBOL XRIRQ=43              % Netserver: invalid request
000000    SYMBOL XRNGA=44              % XMSG not congigurated with gateway code
000000    SYMBOL XRRNL=45              % Remote system not on same LAN
000000    SYMBOL XRNCO=46              % No connection to this system (unknown status)
000000
000000    %   XMSG Crash Codes (on System Console and saved in Basefield)
000000
000000    SYMBOL XXEIE=1               % Illegal entry ptr to XCRMG
000000    SYMBOL XXIOW=2               % Illegal owner of buffer
000000    SYMBOL XXBIN=3               % Memory allocn. inconsistency
000000    SYMBOL XXMCE=4               % Message queue length inconsistency
000000    SYMBOL XXIEN=5               % ZRALL gave port not in XQTAB
000000    SYMBOL XXIFL=6               % INIT; ZFUNC Function >XFMX1
000000    SYMBOL XXIRT=7               % Illegal RT-Description add used.
000000    SYMBOL XXNBF=8               % INIT: No Buffer space available
000000    SYMBOL XXRIN=9               % Inconsistency in resource allocation
000000    SYMBOL XXNMM=10              % More memory released than owned
000000    SYMBOL XXNIM=11              % Not implemented (Cannot recover)
000000    SYMBOL XXCLS=12              % Inconsistency in port chain in CLOSE
000000    SYMBOL XXCHE=13              % Double chaining attempted
000000    SYMBOL XXNOR=14              % No XMSG-RESIDENT found by POF
000000    SYMBOL XXICM=15              % Inconsistency in XMPRT/XPCMS Pair
000000    SYMBOL XX100=16              % This can only be done on N100'S
000000    SYMBOL XXMON=17              % Inconsistency in level 5 monitor queues
000000    SYMBOL XXMMC=18              % Multisystem XMSG not implemented/generated
000000    SYMBOL XXFBI=19              % Frame buffer handling inconsistency
000000    SYMBOL XXPER=20              % Protocol error in communications system
000000    SYMBOL XXILN=21              % Illegal LOG NO for HDLC (bad LOGPH)
000000    SYMBOL XXROU=22              % No legal routing port defined
000000    SYMBOL XXHER=23              % Error in HDLC Driver or interface to it
000000    SYMBOL XXRO2=24              % XROUT fatal error - see XROUT Basefield
000000    SYMBOL XXTAS=25              % Task handling (wait,resume) error
000000    SYMBOL XXMER=26              % Message handling error (kernel-net interface)
000000    SYMBOL XXNER=27              % Network gateway error
000000    SYMBOL XXNGA=30              % Gateway code not generated
000000    SYMBOL XXRI2=31              % XROUT : Resource inconsistency
000000    SYMBOL XXINP=32              % XROUT : Releasing invalid name unit (ynrem)
000000    SYMBOL XXBER=33              % XROUT : Buffer handling error (xb... routines)
000000    SYMBOL XXRUT=34              % XROUT : Response with no request outstanding
000000    SYMBOL XXNXM=35              % XROUT : No multi-system xmsg - inconsistent!
000000    SYMBOL XXNLD=36              % XROUT : Not loaded correctly with RT loader
000000    SYMBOL XXUSR=37              % XROUT : Stopped by 'STOP-XMSG' in sintr-service
```

```
000000    SYMBOL XXN10=38              % XROUT : 8X100 library mark set, but this is ND-10
000000    SYMBOL XXNSG=39              % XROUT : No space available for fixing segment 35
000000    SYMBOL XXNI2=40              % XROUT : Not yet implemented
000000    SYMBOL XXMIN=41              % XROUT : System / routing table inconsistency
000000    SYMBOL XXTAI=42              % XROUT : Table allocation inconsistency
000000    SYMBOL XXWS3=43              % XROUT : Wrong SINTRAN III SYMBOL--LISTs
000000    SYMBOL XXN33=44              % XROUT : Not enough space for seg 33 (or too big)
000000    SYMBOL XXVSX=45              % XROUT : PIT3 version must run on S3/VSX
000000    SYMBOL XXNCX=46              % XROUT : CX version must run on CX
000000    SYMBOL XXWSY=47              % XROUT : Not generated for this system
000000    SYMBOL XXILL=50              % XROUT : Not allowed on this system (MON CPUST)
000000    @OCT
000000    @DEV 1
000000    SYMBOL SLAKK=0               % TEMPORARY SYMBOL FOR FREE SPACE
000000    SYMBOL BCSTA=110000          % START OF COMMAND SEGMENT
000000    SYMBOL OP2BG=110000          % START OF S-S-P AND MAIL SEGMENT
000000    SYMBOL ERRFL=110000          % START OF ERROR PROGRAM SEGMENT
000000    SYMBOL 5BFPAG=76             % LOGICAL PAGE NUMBER FOR BUFFER PAGE
000000    SYMBOL 5UBFPAGE=77           % LOGICAL PAGE NUMBER FOR USER WINDOW IN VSX
000000    SYMBOL L12LGP=74             % LOGICAL PAGE WINDOW FOR TERMINAL INPUT DRIVER
000000    SYMBOL L10LGP=77             % LOGICAL PAGE WINDOW FOR TERMINAL OUTPUT DRIVER
000000    SYMBOL L4LGP=75              % LOGICAL PAGE WINDOW FOR TERMINAL HANDLING ON LEVEL 4
000000    SYMBOL 3BFPAG=5BFPAG+300
000000    @DEV (S-S-J)SIN-VARIABLES
000000
000000
```

```
000000
000000   %=============================================================================
000000   % 1.4        V A R I A B L E S
000000
000000
000000   *0/
000000   *JMP *;JMP *                    % START IN CORE
000002
000002   INTEGER DSKTYP                  % MAIN SWAP-DEVICE TYPE
000003   INTEGER LOADI                   % FLAG=1 IF RT-LOADER IS INITIALIZED
000004   INTEGER BACKGR                  % BACKGROUND RT-PROGRAM
000005   INTEGER DEMAND                  % CURRENT PROG. HAS DEMAND SEGM.
000006   INTEGER MTOR:=1                 % EX. QUQUE SHOULD BE SCANNED
000007   INTEGER RTREF                   % CURRENT PROG. IN EXEC. QUEUE
000010   INTEGER CURPROG                 % CURRENT RT-PROG RUNNING
000011   INTEGER MQUEUE:=-1              % BEGINNING OF MONITOR QUEUE
000012   INTEGER BTIMQU:=-1              % BEGINNING OF TIME QUEUE
000013   INTEGER BEXQU:=STSIN            % BEGINNING OF EXECUTION QUEUE
000014   INTEGER BSEGLINK:=XSGRT         % BEGINNING OF SEGMENT LINK
000015   INTEGER SGMAX                   % HIGHEST SEGMENT NUMBER
000016   INTEGER USEGM                   % FIRST USER SEGMENT
000017   INTEGER ND500TEST               % USED BY ND-500 TEST PROGRAMS
000020
000020   % RESTART AFTER POWER FAIL:
000020   *20/; JMP I *1; RESTA
000022
000022   % START (LOAD FROM IMAGE):
000022   *22/; JMP I *1; SINTR
000024
000024   INTEGER RTSTART:=DUMMY          % START OF RT-DESCRIPTIONS
000025   INTEGER SEGSTART:=XSEGS         % START OF SEGMENT TABLE
000026   INTEGER CORMSTART:=XCORM        % START OF CORE MAP
000027   INTEGER CORAD                   % START OF SWAPPING AREA P.T. 0
000030   INTEGER ARRAY BLST:=(5BL01,5BLST,0,0,0,0,0,0)  % START OF SEGMENT FILES IN MASS.ADDR
000040   INTEGER SBLST=BLST
000040   DOUBLE DDBLST=BLST
000040   DOUBLE ARRAY DBLST=BLST
000040   INTEGER BGFPAGE                 % BACKGROUND FIRST PAGE
000041   INTEGER BGLPAGE:=77             % BACKGROUND LAST PAGE
000042   INTEGER RTFPAGE                 % RT FIRST PAGE
000043   INTEGER RTLPAGE:=77-CCNO        % RT LAST PAGE
000044   INTEGER CCFPAGE:=100-CCNO       % RT-COMMON FIRST PAGE
000045   INTEGER CCLPAGE:=77             % RT-COMMON LAST PAGE
000046
000046   %-----------------------------------------------------------------------
000046   %      C P U   A N D   S Y S T E M   I N F O R M A T I O N
000046
000046   INTEGER        SYSNO:=33CPN              % SYSTEM (CPU) NUMBER
000047   INTEGER ARRAY HWINFI(3)                  % HARDWARE INFORMATION
000052   INTEGER ARRAY SINVER:=(3SINV,0)          % SINTRAN VERSION
000054   INTEGER        REVLEV:=0                 % PATCH REVISION LEVEL
000055   INTEGER ARRAY GENDAT(5)                  % GENERATION DATE (MINUTES - YEAR)
000062   %-----------------------------------------------------------------------
000062
000062   INTEGER ARRAY IDNTS:=(ITB10,ITB11,ITB12,ITB13)  % IDENT TABLES
000066   INTEGER ARRAY EXTDS:=(ITE10,ITE11,ITE12,ITE13)  % IDENT EXTEND TABLES
000072   INTEGER ARRAY TABLES:=(TMRTAB,BACKTAB,BCHTAB,CCTAB)
000076   INTEGER USTART                  % USER START SUBROUTINE (IN START PROGRAM)
000077   INTEGER URESTART                % USER RESTART SUBROUTINE (AFTER POWER FAIL)
000100   INTEGER URPRO                   % USER RESTART RT-PROG.
```

```
000101   %%INTEGER SAFILNO:=(0,RESYS)   % SEG. FILE NO. FOR MEMORY SAVE
000101   INTEGER DASAX(2)              % START OF SAVE AREA
000103   DOUBLE DDASA=DASAX
000103   INTEGER HENTFLAG              % IF ><0 SEGMENTS WILL NOT BE INITIALIZED
000104   INTEGER UNAFLAG:=1            % SYSTEM UNAVAILABLE
000105   INTEGER FLPT3                 % FIRST FREE LOGICAL PAGE ON PAGE INDEX TABLE 3
000106   %%INTEGER DASAX                 % START OF SAVE AREA
000106   INTEGER FIXCLC0               % CLC IN CORE COMMON - RT-LOADER
000107   INTEGER CNTXLOC:=0            % MONITOR CONTEXT LOCK: 0 = NOT LOCKED ><0 LOCKED
000110   INTEGER CAMID                 % CAMAC IDENT
000111   %TIME VARIABLES:
000111   INTEGER 9TIM0:=310            % 20 MS BETWEEN CLOCK INTERRUPTS
000112   DOUBLE ARRAY BCLCN:=(0,1)     % BASIC UNIT SIZE
000114   DATA(0); INTEGER 9TIM1:=62    % SECOND
000116   DATA(0); INTEGER 9TIM2:=5670 % MINUTE
000120   INTEGER 9TIM3:=2,9TIM4:=137440  % HOUR
000122   INTEGER 9TIM5:=101,9TIM6;=165400% DAY
000124          % CALENDAR UPDATED ON MONITOR LEVEL (START TIME IS SYSTEM GENERATIN TIME)
000124   INTEGER 9CL00                 % BASIC TIME UNITS
000125   INTEGER 9CL01                 % SECOND
000126   INTEGER 9CL02:=77MIN          % MINUTES
000127   INTEGER 9CL03:=77HOU          % HOURES
000130   INTEGER 9CL04:=77DAY          % DAYES
000131   INTEGER 9CL05:=77MON          % MONTH
000132   INTEGER 9CL06:=77YEA          % YEAR
000133   INTEGER ARRAY ACL7(0)
000133   INTEGER ATIM1,ATIM2; DOUBLE ATIME=ATIM1  % COUNTED ON CLOCK LEVEL (13)
000135   INTEGER MTIM1,MTIM2; DOUBLE MTIME=MTIM1  % COUNTED ON MONITOR LEVEL
000137   INTEGER IMASK:=3736           % INTERNAL INTERRUPTS
000140   INTEGER NOPGS                 % NUMBER OF PAGES FOR SWAPPING
000141   INTEGER ENDCOR:=7ENDC         % END OF CORE RESIDENT PART
000142   INTEGER USEGADR:=SEGTX        % ADDRESS OF FIRST FREE ENTRY IN SEG.TABLE
000143   INTEGER ADRMASSNO:=MASSNO     % ADDRESS OF THE ARRAY MASSNO
000144   INTEGER UZEROFLAG             % IF SET, THE USER AREA IS ZEROED ON LOGOUT
000145   INTEGER BUFASTART             % START OF BLOCK DEVICE BUFFER AREA RELATIVE TO START OF BANK
000146   INTEGER XALTSYS               % FLAG EQUALS 1 IF GET-ALTERNATIVE-SYSTEM IS USED
000147   INTEGER XSGFNAM:=SGFTABLE     % POINTER TO THE SEGMENT FILE NAMES (IN RT-L)
000150   INTEGER USRTB                 % BATCH SUPERVISOR RT-PROGRAM
000151   INTEGER CACHLIM:=0\1          % UPPER AND LOWER LIMIT FOR NOCACHE AREA N-10
000152   INTEGER ECORMAP               % END OF CORMAP
000153   INTEGER XTMRTERM:=-1          % SAVED TMRTERM
000154   INTEGER ESGTABLE:=9ESGT       % END OF SEGMENT TABLE
000155   INTEGER MRTLADDRESS           % ADDRESS OF RT-LOADER IF "GET-ALT-SYS" IS USED
000156   INTEGER XERDEV:=1             % LOGICAL DEVICE NUMBER OF ERROR DEVICE
000157   INTEGER ENDPAGE:=777          % LAST PAGE NO. TO CHECK FOR EXISTING MEMORY
000160   INTEGER POINTER PPRTDR:=PRTDR  % ADDRESS OF ROUT. TO START RT.PROG FROM DT.
000161   INTEGER ARRAY XMSGU:=(0,0,XXRPT,MGOTA,0) % SPECIAL USE BY XMESSAGE
000166   INTEGER USLOGOUT              % ADDR OF USER DEFINED LOGOUT ROUTINE
000167   DOUBLE FREEC:=(0,764)         % FREE SPOOLING PAGES COUNTER
000171   INTEGER PRINC                 % NO. OF FILES PRINTING OR WAITING
000172   INTEGER FORCE                           % RT-PROG WAITING FOR FORCED PRINT
000173   INTEGER LCACHLIM:=-1          % LOWER CACHE INHIBIT LIMIT FOR N-100
000174   INTEGER UCACHLIM:=0           % UPPER CACHE INHIBIT LIMIT FOR N-100
000175   INTEGER MBSYMOD               % MEMORY BANK FOR SYNC.MODEM I/O BUFFERS.
000176   INTEGER CPSTA                 % SAVED CPU STATUS REGISTER
000177   INTEGER CORMBANK(2)           % ADDRESS OF CORMAP START (32 BITS)
000201   DOUBLE DCORMSTART=CORMBANK
000201   INTEGER BUFMAPSTA             % START OF COREMAP ELEMENTS FOR DEVICE BUFFERS
000202   INTEGER LRESP                 % LAST PAGE OF RESIDENT AREA
000203   INTEGER CSTART                % MONITOR CALL START ADDRESS
```

370

310

```
000204   INTEGER ALTMASK                  % ALT. PT. MASK FOR MONITOR CALLS
000205   INTEGER MONNO                    % LAST MONITOR CALL NUMBER
000206   INTEGER ARRAY 9ERRP(5)           % GLOBAL ERROR PARAMETERS (SET BY 9ERR/9ERRA)
000213   INTEGER ABPRO:=0                 % 0 IF ABORTED BY SYSTEM, ELSE ABORTED PROGR.RTDSC
000214   INTEGER RTTERM:=0                % USER DEFINED TERMINATION HANDLING RT-PROGRAM
000215   INTEGER FLRTTERM:=0              % TERMINATION HANDLING ENABLE/DISABLE FLAG
000216   INTEGER ACCFLAG:=0               % BACKGROUND ACCOUNTING ON/OFF FLAG
000217   INTEGER RTACCFLAG:=0             % RT-ACCOUNTING ON/OFF FLAG
000220   INTEGER NBRTP:=BRTN              % NUMBER OF USER RT-PROGRAMS
000221   INTEGER NBSRT:=NXRTP             % NUMBER OF SYSTEM-RT-PROGRAMS
000222   INTEGER ICCRT:=0                 % RT-DESCRIPTION OF ACCRT PROGRAM
000223   INTEGER PACTAB:=ACTAB            % ADDRESS OF ACTAB
000224   INTEGER PIOACTAB:=IOACTAB        % ADDRESS OF I/O ACTAB
000225   INTEGER POINTER PGNFLAG:=GNFLAG % POINTER TO ROUTINE IN FILSYS
000226                                     % CALLED FROM MEL-PROGRAM
000226   INTEGER CSGST:=0                 % FIX/UNFIX FLAG FOR NORD-NET SEGMENT
000227   INTEGER LGCOLDSTART:=1           % LOG.UNIT EXECUTING THE @COLD-START COMMAND
000230   INTEGER PN500D:=N500D            % POINTER TO N-500 DATAFIELD
000231   INTEGER PDDEBU:=DDEBU            % POINTER TO NSD (DEBUGGER) DATAFIELD
000232   INTEGER BUFBANK                  % MEMORY BANK (0<=BUFBANK<=3) IN WHICH DEVICE BUFFERS LIE
000233   INTEGER ECBKF:=ECBR7             % FLAG TO INDICATE IF BRK & ECHO TABLE 7 IS PRESENT
000234   INTEGER VDDFLAG:=0               % SYSTEM WITH NO DISK: LOADING OF SINTRAN IS DONE VIA VDD/VDP.
000235   INTEGER SPACCFLAG:=0             % SPOOLING ACCOUNTING FLAG
000236   INTEGER BYPINITC:=1              % FLAG TO BYPASS INITIAL COMMANDS (IF NECESSARY)
000237   INTEGER FPOFP                    % FIRST PAGE IN "PAGING OFF" AREA
000240   INTEGER LPOFP                    % LAST PAGE IN "PAGING OFF" AREA
000241   INTEGER 7EPOF:=9EMRE             % LAST ADDRESS USED IN "PAGING OFF" AREA
000242   INTEGER CURMAIL                  % CURRENT MAIL USER
000243   INTEGER TDFPAGE                  % FIRST PHYSICAL PAGE USED FOR DATAF. OUTSIDE RESIDENT
000244   INTEGER TDFBANK                  % MEMORY BANK FOR TERMINAL DATAFIELDS OUTSIDE RESIDENT
000245   INTEGER LAMBANK                  % MEMORY BANK FOR LAMU TABLES
000246   INTEGER LAMDT                    % ADDRESS OF LAMU DESCRIPTION TABLE
000247          DOUBLE DLAMDT=LAMBANK
000247   INTEGER LAMACT                   % ADDRESS OF ACTIVE LAMU TABLE
000250   INTEGER GNLAMU:=MXLAMU           % MAX NUMBER OF LAMUS IN THE SYSTEM
000251   INTEGER GNLPRT:=MXLPRT           % MAX NUMBER OF LAMUS PER PROGRAM (RT-DESCRIPTION)
000252   INTEGER ARRAY SIZF:=(0,0,0,0)       % SIZE OF SEGMENT FILES (NO. OF PAGES)
000256   INTEGER 7RTDLGADDR               % ADDR OF RT-DESCRIPTION EXTENTION TABLE IN POF
000257   INTEGER BUFERASE                 % BUFFER-ERASE OPTION ON/OFF
000260   INTEGER SREBBANK,SREBADDR        % ADDR OF PAGE-OWNER TABLE FOR MON SPLRE
000262          DOUBLE DSREBADDR=SREBBANK
000262   INTEGER XZRTT                    % ADDR OF TABLE FOR XMSG USED TO TELL MONITOR LEVEL
000263                                    % WHICH RT-PROGS SHOULD BE STARTED
000263   INTEGER EXSECUR:=7               % EXTRA SECURITY WANTED
000264          SYMBOL 7NCMLINE=0         % BIT  0: NO LISTING OF COMMAND LINE FOR OTHER USERS
000264          SYMBOL 7ZMEMORY=1         % BIT  1: ZEROING OF BACKGROUND SEGMENT WHEN LOGGING OUT OR LOGGING IN
000264          SYMBOL 7ZSCRATCH=2        % BIT  2: ZEROING OF SCRATCH FILE WHEN LOGGING OUT OR LOGGING IN
000264          SYMBOL 7ZFPAGE=3          % BIT  3: ZEROING OF FILE PAGES WHEN PAGES ARE RELEASED
000264          SYMBOL 7NPASS=4           % BIT  4: NOT ALLOWED TO LOG IN WITHOUT PASSWORD
000264
000264   INTEGER ARTFPAGE                 % FIRST LOGICAL PAGE IN RT-COMMON
000265   INTEGER ARTLPAGE                 % LAST LOGICAL PAGE IN RT-COMMON
000266   INTEGER RWPOF:=1IREA             % ADDRESS OF FIRST OF ROUTINES TO READ/WRITE IN POF.
000267
000267   % POINTERS TO VARIABLES ON SYSTEM SEGMENT USED BY REMOTE FILE ACCESS SEGMENTS:
000267   INTEGER RRPOOL:=RPOOL
000270   INTEGER SSPOOL:=SPOOL
000271   INTEGER ARRAY REMUS:=(RUSNA,0,21)
000274   INTEGER ARRAY LUSER:=(LOCUS,0,21)
000277   INTEGER ARRAY OPFTA:=(OPTAB,0,FMAX)
```

```
000302   INTEGER ARRAY PSYSN:=(RSYSN,0,17)
000305   INTEGER ARRAY PRJPA:=(RPRPA,0,17)
000310   INTEGER MAXOP:=DV100
000311   INTEGER INIFL:=INSFG
000312   INTEGER CRTRF:=CRTRE
000313
000313   % LOGICAL NUMBER TABLE PARTS
000313   @ICR;
000313   INTEGER ARRAY CNVRT:=(DV000,DV100,DV200,DV300,DV400,DV500,DV600,DV700,
000323          D1000,D1100,D1200,D1300,D1400,D1500,D1600,D1700,
000333          D2000,D2100,D2200,D2300,D2400);
000340   @CR;
000340   INTEGER FIXPAGES              % ACTUAL NUMBER OF PAGES FIXED
000341   INTEGER SWPFLAG:=1            % =0: SWAPPING AND DISC-RESERVING AS IN H-VERSION
000342   INTEGER CPULOOPTIME:=1750     % 1000 MIC.SECS. IN ONE IDLE LOOP
000343   INTEGER NMATP:=24             % NUMBER OF RFA ATTEMPTS BEFORE FORCED LOGOUT
000344
000344
```

```
000344
000344   %=========================================================================
000344   % 1.5        S E G M E N T   D A T A   S T R U C T U R E
000344
000344   %  SEGMENT TABLE ELEMENT:
000344
000344   DISP 0
000344   INTEGER SEGLINK     % LINK OF NON-FIXED SEGMENTS WITH PAGES IN MEMORY
000344   INTEGER BPAGLINK    % START OF PAGES-IN-MEMORY LINK FOR SEGMENT
000344   INTEGER LOGADR      % BIT 0-7 : FIRST LOGICAL PAGE IN SEGMENT
000344                       % BIT 8-15: LENGTH OF SEGMENT IN PAGES
000344   INTEGER MADR        % BIT 0-13: MS ADDR. OF SEGMENT WITHIN THE SEGMENT FILE
000344                       % BIT 14-15: SEGMENT FILE NUMBER
000344   INTEGER FLAG        % SEGMENT STATUS
000344   PSID
000344
000344   % BITS IN FLAG
000344   SYMBOL 5OK=0                  % SEGMENT IS READY
000344   SYMBOL 5DEMAND=1              % DEMAND SEGMENT
000344   SYMBOL 5FIX=2                 % SEGMENT CANNOT BE SWAPPED OUT (FIXED)
000344   SYMBOL 5INHB=3                % SEGMENT NOT BUILT
000344   SYMBOL 5SYSEGM=4              % SYSTEM-SEGMENT
000344   SYMBOL 5SPROT=5               % PROTECTED SEGMENT
000344   SYMBOL 5SREEP=6               % REENTRANT SUBSYSTEM SEGMENT
000344
000344   SYMBOL 5SEGSIZE=5             % SEGMENT TABLE ELEMENT SIZE
000344
000344
000344
000344   % CORE MAP ELEMENT FOR NORD 100 EXTENDED ADDRESS MODE
000344   DISP 0
000344   INTEGER PAGLINK               % LINK OF PAGES-IN-MEMORY FOR A SEGMENT
000344   INTEGER ARRAY POINTER ALOGNO        % LOGICAL PAGE NUMBER
000344   INTEGER PAGPROT               % BIT 9-15: SAME AS THE CONTENT OF BIT 9-15 IN
000344                                 %           FIRST WORD IN PAGE-TABLE ENTRY
000344                                 % BIT 0-9: SEGMENT NUMBER
000344   INTEGER EPAGPHYS              % PHYSICAL PAGE NUMBER
000344   PSID
000344
000344   % CORE MAP ELEMENT FOR N-10 AND N-100 IN NORMAL ADDRESS MODE
000344
000344   % PAGLINK                     % LINK OF PAGES-IN-MEMORY FOR A SEGMENT
000344   % ALOGNO                      % LOGICAL  PAGE NUMBER
000344   DISP 2
000344   INTEGER PAGPHYS               % BIT 9-15: SAME AS THE CONTENT OF BIT 9-15 IN
000344                                 %           PAGE-TABLE ENTRY
000344   PSID
000344   % BITS IN PAGPROT AND PAGE TABLE
000344   SYMBOL 5PGU=13                % PAGE IS USED
000344   SYMBOL 5WIP=14                % WRITTEN IN PAGE
000344   SYMBOL 5WPM=17                % WRITE PERMIT BIT
000344
000344   SYMBOL DPAGL=0,DALOG=10,DPGPR=20,DPAGP=30
000344
000344   SYMBOL 5CLOAD=501             % SEGMENT TRANSFER SEMAPHORE
000344
000344   %  VARIABLES:
000344
000344   INTEGER SEGMA,SEGMB,SEGMC     % CURRENT SEGMENTS
000347   TRIPLE FSABC=SEGMA
```

```
000347   INTEGER NSEGA,NSEGB,NSEGC     % NEW SEGMENTS
000352   TRIPLE FNABC=NSEGA
000352   INTEGER TSEGA,TSEGB,TSEGC     % SEGMENTS FOR SWAPPING RT-PROG.
000355   INTEGER ARRAY TSEGS(0)
000355   INTEGER SEGPROG               % PROGRAM USING SEGMA,B,C
000356   INTEGER PNUMB,CPNUMB,NUMBER,WIND1
000362   INTEGER SEGREF,SAVEFLAG,SRTREF,SGCNUMBER
000366   INTEGER MAXP:=200,FIXMAX            % FIXMAX=NUMBER-OF-PAGES-FOR-SWAP.-4
000370   INTEGER ARRAY MASSNO:=(5DSK,5DSK,5DSK,5DSK),MASSUNIT:=(5MAU1,5MAU1,5MAU1,5MAU1)
000400   INTEGER ARRAY ABLPAGE:=(5ABLP,5ABLP,5ABLP,5ABLP)
000404
000404   %=================================================================
000404   %       T A D  STACK VARIABLES
000404   %
000404   INTEGER TSBANK                % MEMORY BANK FOR TAD STACK-AREA
000405   INTEGER TSSIZE:=100           % SIZE IN WORDS FOR EACH TAD STACK
000406
000406   @DEV 1
000406   @DEV (S-S-J)SINA
000406
000406   %****************************  S I N A ****************************
000406
000406   SUBR STRTC
000406   RBUS
000406
000406   %=====================================================================
000406   %       FIXED ADDRESS SUBROUTINE ENTRY POINTS
000406   %
000406   %
000406   %msSUBR CLXMSG,PFXMSG,MFXMSG,DRXMSG
000406   %msCLXMSG: EXIT; 0/\0            % MULTI MACHINE XMSG TIMER
000406   %msPFXMSG: EXIT; 0/\0            % MULTI MACHINE XMSG POWERFAIL
000406   %msMFXMSG: EXIT; 0/\0            % MULTI MACHINE XMSG FATAL MEMORY ERROR
000406   %msDRXMSG: A:=-45; EXIT          % DRIVER CALLS TO XMSG
000406   %msRBUS
```

```
000406
000406   %==========================================================================
000406   %
000406   %  2.0          M O N I T O R   K E R N E L
000406   %
000406   %==========================================================================
000406   % 2.1        E R R F A T A L
000406
000406
000406   % SYSTEM ERROR:
000406   SUBR ERRFATAL
000406   ERRFATAL: *IOF; WAIT
000410          GO ERRFATAL
000411   RBUS
000412
000412   %==========================================================================
000412   % 2.2        M O N I T O R   E N T R Y
000412   %            M O N E N   S T U P R   R W A I T   R W   P M O N E N
000412   %            P R W A I T   P S T U P R   P R W
000412   %
000412   % THIS IS THE MAIN ENTRY AND EXIT POINT FOR THE MONITOR LEVEL
000412
000412   SUBR MONEN,STUPR,RWAIT,RW,PRW,PMONEN,PSTUPR,PRWAIT,IRWAIT
000412
000412   PMONEN:
000412   MONEN:*PION; IOF
000414          IF MQUEUE><-1 THEN     % SOMETHING IN MONITOR QUEUE
000420                X:="MQUEUE-5"
000421   LOOP:        X=:B
000422                IF MLINK.MLINK><T GO LOOP     %T=-1
000426                T=:MLINK; 0=:X.MLINK; X.MFUNC
000431                *ION
000432                A=:P              % EXECUTE ROUTINE; X=DATAFIELD
000433          FI
000433          IF MTOR=0 OR CNTXLOC><0 THEN                % LEAVE MONITOR LEVEL
000437                ALEV; *MST PID
000441                MLEV;*MCL PID; ION
000444                GO MONEN
000445          FI
000445          *ION
000446
000446   %MAY BE A DIFFERENT PROGRAM SHOULD BE STARTED
000446   %SEARCH EXEC. QUEUE
000446
000446   STUPR: IF CNTXLOC><0 GO MONEN                 % CONTEXT SWITCHING LOCKED?
000451          0=:MTOR                                % START ON BEG. OF EXEC QUEUE
000452          X:=BEXQUE=:RTREF; GO CRW
000455
000455   PRWAIT: *PON
000456   RWAIT: IF CNTXLOC=0 GO CRWAI; GO MONEN        % CONTEXT SWITCHING ALLOWED?
000461   RW:    IF CNTXLOC=0 GO CRW; GO MONEN          % CONTEXT SWITCHING ALLOWED?
000464   CRWAI: X:=RTREF.WLINK                         % START AFTER CURRENT PROG.
000466   IRWAIT: X=:RTREF
000467   CRW:   IF X.STATUS BIT 5WAIT GO CRWAI  % I/O-WAIT
000472          0=:DEMAND
000473          IF X.ACTSEG><0 THEN
000475                *POF
000476                CALL SEGADM
000477          FI
000477          IF X=CURPROG GO PMONEN
```

```
000502
000502   %THE PROGRAM IS FOUND
000502          *PIOF                    % TO PROTECT CONTEXT SWITCH
000503          X.ACTPRI/\100000 SHR 1 =:BACKGR
000507          X.ACTPRI/\3773; *TRR PCR
000512          X=:T
000513          IF X:=CURPROG><0 THEN
000515             X:=X.RTDLGADDR; *SRB ALEVB
000517          FI; X:=T=:CURPROG:=X.RTDLGADDR; *LRB ALEVB
000523          GO PMONEN
000524
000524   PSTUPR: *PON
000525          GO STUPR
000526   PRW:   *PON
000527          GO RW
000530   RBUS
000546
000546
000546   %=================================================================
000546   % 2.3        C X L O C    C X U L O C
000546   %
000546   % ROUTINES TO LOCK/UNLOC CONTEXT SWITHCING.
000546   SUBR CXLOC,CXULOC
000546   INTEGER CALLER                          % ADRESS OF UNLOC CALL IF ERROR
000547   CXLOC:  MIN CNTXLOC; 0/\0; EXIT         % LOCK CONTEXT SWITCHING
000552   CXULOC: CNTXLOC-1=:CNTXLOC              % UNLOC CONTEXT SWITCHING
000555           IF A<0 THEN
000556             A:=L=:CALLER; CALL ERRFATAL   % UNLOC CALLED WHEN NOT LOCKED
000561           FI
000561           EXIT
000562   RBUS
```

```
000564
000564   %==========================================================================
000564   %
000564   %   3.0            M O N I T O R   C A L L   D E C O D E R
000564   %
000564   %==========================================================================
000564   % 3.1          C A L L P R O C   W T   M R E T   S W A P P R
000564   %              B B C A L   M C A L   T Y P E T A B
000564   %              T M C T A B   M C T A B   C S T A R T   A L T M A S K
000564   %
000564
000564   INTEGER MOFTYPE=?
000564
000564   SUBR CALLPROC,WT,MRET,SWAPPR,BBCAL,MCAL,MONDEM,BBCLB,MFBBCL
000564   @ICR
000564   % ROUTINE ADDRESSES ON MONITOR LEVEL
000564   INTEGER ARRAY TYPETAB:=(
000564           ERR,ERR,XMCAL,BMCAL,M500,M500,ERR,XBPNT,                % 0-3
000574           JETTYPE,JETTYPE,RTEXT,BRTWT,RTWT,BBRTWT,DIREX,DIREX,     %4-7
000604           MCAL,BBCAL,MFIX,BMFIX,MOFIL,MOFI2,ERR,BBCAL,            % 10-13
000614           MFIX,BXCAL,ERR,MOFI2,XMCAL,BXCAL,ERR,MOCOM,             % 14-17
000624           WFRQI,WFRQI,RTEXT,YBRTWT,MFIX,MFIX,ERR,MFIX,            % 20-23
000634           MSIBA,MSIBA,RRSIO,MOFIA,MOFIL,MOFIB,ERR,MOFIB,          % 24-27
000644           ERR,MOFIA,MPIOM,MPIOM,MSIB,MSIB,MUDMA,MUDMA,            % 30-33
000654           XMLGIN,XMLGIN,0,0);                                     % 34-35
000660
000660   % INDICIES IN TYPETAB:
000660   @CR;
000660   *9TMCT=*
000660   @ICR
000660   INTEGER ARRAY TMCTAB:=(
000660           21\16,16\10,10\15,15\10,          % 0-7
000664           10\16,17\10,10\15,16\16,          % 10-17
000670           15\0,0\0,0\12,16\22,              % 20-27
000674           16\22,27\16,16\30,16\11,          % 30-37
000700           12\12,12\12,12\13,13\13,          % 40-47
000704           12\11,16\22,12\12,23\23,          % 50-57
000710           2\11,12\0,26\26,10\10,            % 60-67
000714           17\16,16\12,12\12,12\12,          % 70-77
000720           1\1,1\1,16\1,1\1,                 % 100-107
000724           1\1,1\22,27\11,11\10,             % 110-117
000730           10\1,16\16,1\1,1\1,               % 120-127
000734           1\7,1\1,5\6,1\1,                  % 130-137
000740           16\16,1\25,10\14,1\1,             % 140-147
000744           1\11,22\1,1\14,14\1,              % 150-157
000750           11\14,14\20,22\11,11\22,          % 160-167
000754           1\1,1\1,1\1,1\1,                  % 170-177
000760           0\22,0,3\3,22\22,                 % 200-207
000764           0,22\12,12\12,12\12,              % 210-217
000770           12\12,25\4,4\4,4\16,              % 220-227
000774           16\12,12\12,12\12,12\12,          % 230-237
001000           12\15,15\12,12\12,12\12,          % 240-247
001004           12\12,12\12,14\31,12\12,          % 250-257
001010           10\1,12\14,10\10,10\16,           % 260-267
001014           12\12,12\12,12\12,13\13,          % 270-277
001020           13\13,13\13,24\32,16\16,          % 300-307
001024           16\12,7\10,12\22,10\17,           % 310-317
001030           17\30,22\22,7\30,34\12,           % 320-327
001034           30\0,16\33,22\22,0);              % 330-337
001040   % ADDRESSES ON RT LEVEL:
```

```
001040   @CR;
001040   *9MCTA=*
001040   @ICR
001040   INTEGER ARRAY MCTAB:=(
001040   RTEXT,YFGET,YFPUT,ECHOM,BRKM,RDISK,WDISK,XRPAGE,          % 0-7
001050   XWPAGE,TIME,SETOLD,CIBUF,COBUF,SETUP,MGTTY,MSTTY,         % 10-17
001060   WCI,MBINB,MBOUT,BBINB,BBOUT,SETW,LSTC,RDSC,               % 20-27
001070   GTRT,EXIOX,MSG,MALTN,MALTF,IOUT,NOWAIT,AIRDW,             % 30-37
001100   SPCLO,MROBJ,OLDOP,CLOFI,MRUSE,BDBRK,BGBRK,BSBRK,          % 40-47
001110   OPFIL,GBRKD,MTERM,MRSEG,MDLFI,RSPQE,MPASE,MPAGE,          % 50-57
001120   N500M,FIXC5,RMAX,B4INW,ERMSG,QERMS,ISIZE,OSIZE,           % 60-67
001130   COMSB,MCDES,MCEES,SMAX,SETBY,REABT,SBSIZ,SETBC,           % 70-77
001140   RT,SET,ABSET,INTV,HOLD,ABORT,CONCT,DSCNT,                 % 100-107
001150   PRIOR,UPDAT,CLADJ,CLOCK,TUSED,MOFIX,MUNFIX,XRFILE,        % 110-117
001160   XWFILE,WAITF,RESRV,RELES,PRSRV,PRLS,DSET,DABST,           % 120-127
001170   DINTV,ABSTR,MCALL,MEXIT,RTEXT,RTWT,RTON,RTOFF,            % 130-137
001200   WHERE,IOSET,ERRMON,RSIO,MAGTP,ACM,IPRIV,CAMAC,            % 140-147
001210   GL,GRTDA,GRTNA,IOXN,ASSIG,PLOTT,TRACB,ENTSG,              % 150-157
001220   FIXC,3INSTR,3OUTST,WFRQI,WSEG,DIW,DOLW,REENT,             % 160-167
001230   US0,US1,US2,US3,US4,US5,US6,US7,                          % 170-177
001240   XMSGX,XTLX,JETM0,JETM2,BRPNT,DEBUG,EDTRM,RERRP,           % 200-207
001250   P0EARL,P1EARL,SREEN,MUIDI,GUSNA,DROBJ,DWOBJ,GUIOI,        % 210-217
001260   DOPEN,CRALF,GBGSZ,JETM3,JETM4,JETM5,JETM6,MSDAE,          % 220-227
001270   MGDAE,EXPFI,MRNFI,STEFI,SPEFI,SCROP,SPERM,SFACC,          % 230-237
001300   APSPF,SUSCN,RUSCN,FDINA,GDIEN,GNAEN,RESDI,RELDI,          % 240-247
001310   FDFDI,COPAG,BCLOS,CRALN,GERDV,PIOCM,DEABF,FOPFN,          % 250-257
001320   USCNT, SYCNT, CPUST, GDEVTY,500RF, 500WF, 500MTP,TMOUT,   % 260-267
001330   RDPAG, WDPAG, DELPG, MGFIL, FOBJN, SETTF, ELOFU, DLOFU,   % 270-277
001340   EUSEL,DUSEL,ELON,ELOFF,MAPSIB,MSIBB,GTMOD,TNOWAI,         % 300-307
001350   T8INB,WDIEN,MOINF,IBRSIZ,SDRUS,MLAMU,SLRMO,UECOMSB,       % 310-317
001360   UELOGIN,UEADM,GSGNO,SPLRE,MOCTBU,MBECHO,MLOGIN,WRBIX,     % 320-327
001370   TRTER,MSYSU,TREPP,UDMA,GETXM,EXABS,0,0);                  % 330-337
001400   @CR;
001400   *9MCMX=*-9MCTA
001400
001400   %AUXILIARY SUBROUTINE
001400   % ENTRY:      A=WORKING FIELD
001400   %  IOBT OR MON. LEVEL
001400   SWAPPR:
001400   SWAPREG: X:=A=:B+5REG; T:=ALTMASK=:D:=CSTART; *SRB ALEVB
001407           *IRW ALEVB DX; IRR ALEVB DA; IRW ALEVB DB
001412           IF A:=T=0 GO ERR; *IRW ALEVB DP
001415           T:=RTREF.ACTPRI=:OLDPAG
001420           174174\/D/\T+2=:X.ACTPRI/\3773; *TRR PCR
001427           1; *IRW ALEVB 0
001431           EXIT
001432   *)FILL
001437
001437   % MAIN ENTRYPOINT FOR MONITOR CALLS, ACTIVATED FROM LEVEL 14
001437   % ENTRY:      X=MON. CALL NO.
001437   % MONITOR LEVEL
001437
001437   CALLPROC: 0=:ALTMASK
001440           MCTAB(X)=:CSTART; X=:MONNO
001443           IF A><0 THEN
001444                 T:="TMCTAB"; *LBYT
001446           FI; A+A+BACKGROUND:=:X; T:=TYPETAB(X)=:P
001453
001453
001453   % ILLEGAL MONITOR CALL
```

```
001453   % IOBT OR MON. LEVEL
001453   ERR:    MONNO; CALL 9ERR(#D0); "BRTEXT"; *IRW MLEVB DP
001460           MLEV; *MST PID
001462           CALL BRTEXT
001463
001463   % EXECUTE MONITOR CALL DIRECTLY (ADDRESS FROM MCTAB) ON LEVEL 3
001463   DIREX: CSTART=:P
001465
001465   % SPECIAL PIOC MONITOR CALL
001465   MPIOM: T:=600
001466           A:="SEMPI"; GO TINMFX
001470
001470   % ENTRY FOR MON LOGIN
001470   XMLGIN: IF "MLIDFIELD"=0 GO ERR
001472            T:=600; GO TINMFX
001474
001474   % SPECIAL ENTRY POINT FOR NORD SYMBOLIC DEBUGGER MON CALL
001474   XBPNT: T:=0; PDDEBU; GO TINMFX
001477
001477   % SPECIAL ENTRY POINT FOR ND-500 MON CALL
001477   M500:    T:=600; PN500D
001501   TINMFX: T=:ALTMASK; GO INMFX
001503
001503   % BACKGROUND ENTRIES
001503   BMFIX: IF RTREF.ACTPRI NBIT 1 GO ERR; GO MFIX
001510   BMCAL: IF RTREF.ACTPRI NBIT 1 GO ERR
001514   BXCAL: 600=:ALTMASK
001516   BBCAL: "BBCLB"; *IRW BLEVB DP
001520           BLEV; *MST PID
001522           GO MONEN
001523
001523   % IOBT LEVEL:
001523   MFBBCL: A=:MOFTYPE; T=:"BGFIELD"."MRSTA"
001526            A:=D=:CSTART
001530   BBCLB: "BGFIELD"; CALL SWAPREG; *WAIT
001533
001533   % NORMAL ENTRIES FROM RT-PROGRAMS:
001533   RRSIO: A:=3; *IRW ALEVB DA
001535           GO MONEN
001536   MFIX:   600=:ALTMASK; GO MFX
001541   XMCAL:  600=:ALTMASK
001543   MCAL:   IF DEMAND><0 THEN
001545   MFX:            "DEMFIELD"
001546   INMFX:          A=:B; X:=RTREF; CALL BRESERVE; IF A<0 GO WT
001553                   B=:A; CALL SWAPREG; GO MONEN
001556           ELSE                        % NON-DEMAND
001557   MONDEM:         "NDEMFIELD"-5REG; CALL SWAPREG; MLEV; *MCL PIE
001564                   CALL ERRFATAL
001565           FI
001565
001565
001565   % SPEICAL SIBAS MONITOR CALLS
001565   MSIBA: K:="0"; GO INMSI
001567   MSIB:  K:=1
001570   INMSI: *IRR ALEVB DT
001571           IF A>>MXSIBAS OR T=0 THEN
001576              A:=174; *IRW ALEVB DA
001600              GO MONEN
001601           FI; X:=A
001602           IF K THEN SIBBDEVS(X) ELSE SIBAPDEVS(X) FI
```

```
001607          GO INMFX
001610   *)FILL
001644
001644   %- -RETURN TO MONITOR LEVEL AFTER MONITOR CALL PROCESSING
001644   % X=DATAFIELD, L=MON. ADDRESS
001644
001644   MRET:   X=:B+5REG; *LRB ALEVB
001647           CURPROG.ACTPRI/\40000\/OLDPAG=:X.ACTPRI/\3773; *TRR PCR
001656           EXIT                    %TO MONEN,STUPR,RWAIT,RTEXT
001657
001657   % SET IN WAITING QUEUE AND RESUBMIT CALL WHEN FREE:
001657   %B=DATAFIELD, X=RTPROG
001657   WT:     CALL FREXQU; CALL TOWQU; CALL ANTIJAMMER; *IRR ALEVB DP
001663           A-1; *IRW ALEVB DP
001665           GO STUPR
001666
001666   % MOINF MONITOR CALL
001666   MOINF: *IRR ALEVB DA
001667           IF A<<"9MCMX" THEN MCTAB(A) ELSE A:=0 FI; *IRW ALEVB DA
001677           IF A><0 THEN
001700                 *IRR ALEVB DP; AAA 1; IRW ALEVB DP
001703           FI; GO MONEN
001704   RBUS
001716
001716   %=============================================================================
001716   % 3.5          D N A L T O N   D A L T O N   B A L T O N   R A L T O N
001716   %             A L T O N   A L T O F F   S A L T O N
001716   %
001716   % SUBROUTINES TO SET AND RESET ALTERNATIVE PAGETABLE
001716
001716
001716   SUBR DNALTON,DALTON,BALTON,RALTON,ALTON,ALTOFF
001716   INTEGER XREG,LREG; REAL TADREG
001723
001723   %NORM. IN D-REG=:ALT
001723   DNALTON: *PIOF
001724           TAD=:TADREG; 3000/\D SHZ -2; GO INALT
001731
001731   % ALT. IN D-REG=:ALT
001731   DALTON: *PIOF
001732           TAD=:TADREG; 600/\D; GO INALT
001736
001736   % ALT IN A-REG=:D
001736   BALTON: *PIOF
001737           TAD=:TADREG; A/\600; GO INALT
001742
001742   % SET ALT=2; CALLED FROM RT-LOADER
001742   RALTON: *PIOF
001743           TAD=:TADREG; A:=400; GO INALT
001746
001746   % USERS ALT=:ALT. NB!     B-REG=WORKING AREA (MUST BE DEMFIELD OR NDEMFIELD)
001746   ALTON: *PIOF
001747           TAD=:TADREG; OLDPAG/\600
001752   INALT: X=:XREG; A=:T
001754           CURPROG.ACTPRI/\177177\/T; GO OUT
001761
001761   % D=:ALT
001761   ALTOFF: *PIOF
001762           TAD=:TADREG; X=:XREG
001764           CURPROG.ACTPRI/\177177
```

```
001767   OUT:    A=:X.ACTPRI/\3773; *TRR PCR; BSET ONE 0
001773           TADREG; X:=XREG; *PION
001776           EXIT
001777   RBUS
002005
002005
002005   % GET ALTERNATIVE PAGE TABLE
002005   % ENTRY:      X=MONITOR CALL WORKING FIELD
002005   % EXIT:       D=ALTERNATIVE PAGE TABLE, A IS DESTROYED
002005   SUBR GAPIT
002005   GAPIT: IF BACKGROUND=0 THEN
002007              IF X.MTFLG BIT MC5SPRT GO NRMPT
002012              GO ALTPT
002013           FI
002013           IF X.OLDPAGE/\3000><0 THEN
002016   ALTPT:     X.OLDPAG
002017           ELSE
002020   NRMPT:     X.OLDPAG/\177177=:D; X.OLDPAGE/\3000 SHZ -2\/D
002027           FI; A=:D; EXIT
002031   RBUS
002034
```

```
002034
002034   *CSSLO=*
002034   *RESI2=CORES+SLAKK
002034   *RESI2/
006355
006355   %==========================================================================
006355   % 3.3          R T C H E C K   X R T C H E C K
006355   %
006355   %CHECK FOR GOOD RT-PROGRAM POINTER
006355   % ENTRY:       A=RT-PROG. FOR XRTCHECK; D0=RT-PROG. FOR RTCHECK
006355   % RETURN:      X=RT-PROG.
006355   % RT LEVEL
006355
006355   SUBR RTCHECK,XRTCHECK
006355   XRTCHECK: IF =0 THEN RTREF FI; GO XRT
006360   RTCHECK: IF D0=0 THEN RTREF=:D0 FI
006364   XRT:    A=:X; IF >=SEGSTART GO ERR
006370           IF A-RTSTART<0 GO ERR
006372           A=:D:=0; T:=5RTSIZE; *RDIV ST
006376           IF D><0 GO ERR
006400           EXIT
006401   ERR:    CALL 9ERRA(#01); CALL RETXIT    %ILLEGAL RT-PROG.
006404   RBUS
006411
006411   %==========================================================================
006411   % 3.4          R E T   R E T S T U P R   R E T R W A I T   R E T X I T
006411   %              R E T R T W A I T   A B R E T X I T   M 6 1 R E T
006411   % RETURN FROM MONITOR CALL
006411   % ENTRY:       B=WORKING AREA
006411   % RT-LEVEL
006411
006411   SUBR RET,RETSTUPR,RETRWAIT,RETXIT,RETRTWAIT,ABRETXIT,M61RET
006411   RETRTWAIT: "RTWT"; GO XRET
006413   RETXIT: "BRTEXT"; GO XRET
006415   ABRETXIT: "RTEXT"; GO XRET
006417   RETSTUPR: "STUPR"; GO XRET
006421   RETRWAIT: "RWAIT"; GO XRET
006423   M61RET: "STUPR"; *PIOF; IRW MLEVB DL
006426            GO L1
006427   RET:    "MONEN"
006430   XRET:   *PIOF; IRW MLEVB DL
006432           IF B="DEMFIELD" THEN X:=CURPROG; CALL BRELEASE FI
006437   L1:     B=:A; *IRW MLEVB DX
006441           "MRET";*IRW MLEVB DP
006443           MLEV; *MST PID; MST PIE; PION
006447           CALL ERRFATAL          %RETURN IN IOF
006450   RBUS
006463
006463   %==========================================================================
006463   %
006463   %  - - - 4.0    K E R N E L    M O N I T O R   C A L L S
006463   %
006463   %==========================================================================
006463   % 4.1          R T
006463   %
006463   SUBR RT,PRIOR,ABORT
006463
006463   %CALL RT(PROG)
006463   RT:     CALL GET1; CALL RTCHECK
006465           CALL RTENTRY; GO RETSTUPR
```

```
006467
006467   %=========================================================================
006467   % 4.2          P R I O R
006467
006467   % A=OLD PRIORITY= PRIOR(PROG,PRI)
006467   INTEGER PMASK:=177400,QMASK:=377,BREG
006472   PRIOR: CALL GET2; CALL RTCHECK; IF D1/\PMASK><0 GO ERR
006477          X.STATUS/\QMASK=:ZAREG; X.STATUS/\PMASK\/D1=:X.STATUS
006506          CALL FRWQU; IF ><0 THEN A:=:B=:BREG; CALL TOWQU; BREG=:B FI
006515          GO RETSTUPR
006516
006516   ERR:   CALL 9ERRA(#02); CALL RETXIT
006521
006521   %=========================================================================
006521   % 4.3          A B O R T
006521
006521   %CALL ABORT(PROG)
006521   ABORT: CALL GET1; CALL RTCHECK; X=:D
006524          IF "CLFIE".RTRES=D THEN                  % PROGRAM HAS SWAPPING-SEMAPHORE RESERVED
006530             X:="BRESLINK"+D
006532             DO WHILE X:=X.RESLINK><D              % AREA ANY SWAPPING-DEVICES RESERVED?
006535                IF X.TYPRING/\3=3 THEN             % SWAPPING-DEVICES HAVE TYPRING=3
006542                   IF RTREF=D GO NORMAL            % ABORT CURRENT PROGRAM; IMPOSSIBLE
006545                   D.ACTPRI/\100170+2=:X.ACTPRI    % SET PT=APT=0, RING=2
006552                   X:=X.RTDLGADR; "XABORT"; *PIOF
006555                   A=:X.DPREG; GO RET              % DELAY EXECUTION OF MON ABORT
006557                FI
006557             OD
006560          FI
006560   NORMAL: X:=D; CALL FRWQU; CALL FTIMQU
006563          X.STATUS/\1777=:X.STATUS
006566          0=:X.RSEGM
006567          X =: RTREF
006570          A:=X; CALL CLLAM                 % CLEAR ENTRY IN ACTIVE LAMU TABLE
006572          GO ABRETXIT
006573
006573   % DELAYED EXECUTION OF MON ABORT BECAUSE THE PROGRAM HAD THE SWAPPING-SEMAPHORE
006573   % RESERVED WHEN THE MON ABORT WAS EXECUTED.
006573   %
006573   INTEGER NULL; INTEGER ARRAY PLIST:=(NULL)
006575   XABORT: A:="PLIST"; *MON 2ABOR
006577           CALL ERRFATAL
006600   RBUS
006626
006626   %=========================================================================
006626   % 4.4          P R S R V
006626   %
006626
006626   % MONITOR CALLS:
006626   SUBR RESRV,RELES,PRSRV,PRLS,WHERE
006626   INTEGER BREG
006627
006627   % INTERNAL SUBROUTINE TO CHECK LOGICAL NUMBER; RETURN:A=DATAFIELD
006627   DISP 0; INTEGER POINTER LREG=D3; PSID
006627   IOCHECK: A:=L=:"LREG"; IF D0/\177700=100 GO ERR; D0; CALL LOGPH
006640           IF =0 OR T:=D1><0 THEN D=:A FI; IF =0 GO ERR; GO LREG
006647
006647   %CALL PRSRV(DEV,OUTP,PROG)
006647   PRSRV: CALL GET3; D2; CALL XRTCHECK; X=:D2; GO RES
006654
```

```
006654   %=========================================================================
006654   % 4.5          R E S R V
006654   %
006654   % CALL RESRV(DEV,OUTP,FLAG)
006654   RESRV: CALL GET3; X:=RTREF
006656   RES:   CALL IOCHECK; A:=:B=:BREG
006661          T:=3; IF TYPRING/\T=T THEN A:=2 FI
006667          IF A>X.STATUS SH 6 SHZ -16 GO ERRB
006674          CALL BRESERVE
006675          X:=BREG
006676          IF A>=0 OR T:=X.D2><0 THEN A=:X.ZAREG; X=:B; GO RET FI
006705          X:=RTREF; CALL FREXQU; CALL TOWQU; CALL ANTIJAMMER
006711          BREG=:B; 0=:ZAREG;GO RETSTUPR
006715
006715   %=========================================================================
006715   % 4.6          P R L S
006715   %
006715   %CALL PRLS(DEV,OUTP)
006715   PRLS:  CALL GET2; CALL IOCHECK; X:=0; A:=:B=:BREG; GO REL
006723
006723   %=========================================================================
006723   % 4.7          R E L E S
006723   %
006723   %CALL RELES(DEV,OUTP)
006723   RELES: CALL GET2; CALL IOCHECK
006725          X:=RTREF; A:=:B=:BREG; IF RESLINK><0 AND RTRES><X GO ERRB
006735   REL:   IF TYPRING BIT 5RFILE GO ERRB; CALL BRELEASE
006741          BREG=:B; 0=:ZAREG; GO RET
006745
006745   %=========================================================================
006745   % 4.8          W H E R E
006745   %
006745   %CALL WHERE(DEV,OUTP)
006745   WHERE: CALL GET2; CALL IOCHECK; A.RTRES=:ZAREG; GO RET
006753
006753   ERRB: BREG=:B
006755   ERR:   -1=:ZAREG; GO RET              %ILL. PAR. FOR RESRV OR RELES
006760   RBUS
006775
006775   %=========================================================================
006775   % 4.9          R T W T  R T E X T  B R T E X T  P S B R T E X T
006775   %              X B R T W T   P B R T E X T   P R T E X T
006775
006775   %MONITOR CALLS TO TERMINATE PROGRAMS
006775   %MONITOR LEVEL
006775   SUBR RTWT,RTEXT,BRTEXT,XBRTWT,YBRTWT,PBRTEXT,PRTEXT,PSBRTEXT
006775
006775   INTEGER SAVPR
006776   RTWT:  RTREF.STATUS BONE 5RWAIT=:X.STATUS; GO FORB2
007003   PBRTEXT:
007003   BRTEXT:  A:=RTREF
007004   PSBRTEXT: A=:X; *PON                          % ENTRY FROM XCSEGS (POF)
007006          CALL IFTRM; 0=:ABPRO                  % INDICATE ABORTED BY SYSTEM
007010          IF X.ACTPRI BIT 5BACKGR GO YBRTWT
007013          T:="BRTEXT"; CALL RL5PDESC            % TERMINATE ND-500 PROC AND RELEASE ND-500 PROC-DESC.
007015          X.STATUS BZERO 5INT BZERO 5REP=:X.STATUS
007021          GO FORB1
007022   RTEXT: X:=RTREF
007023          T:="RTEXT"; CALL RL5PDESC             % TERMINATE ND-500 PROC AND RELEASE ND-500 PROC-DESC.
007025          CALL IFTRM; X=:ABPRO; X.STATUS        % TERMINATED WITH ABORT/RTEXT
```

```
007030   FORB1: A BZERO 5RWAIT=:X.STATUS
007032          DO
007032             IF X.BRESLINK=X GO OUT
007035             A=:B; CALL BRELEASE
007037          OD
007040   OUT:   T:=1; *MON 2XMSG
007042          T:=201; *MON 2XMSG       % DISCONNECT FROM XMSG (XFDCT)
007044   %
007044   %  IF RT-TERMINATION ENABLED AND USER RT-PROGRAM THEN ......
007044          CALL IFTRM; GO OK; GO FORB2
007047   OK:            X=:SAVPR
007050                  X:="TERMP"; CALL RTENTRY  % START TERMIN. CLEANUP-PROGRAM ON PT 0
007052                  X:=SAVPR
007053   FORB2: CALL FREXQU
007054          IF X.STATUS BIT 5REP THEN CALL RTENTRY FI
007060          GO STUPR
007061   PRTEXT: *PON
007062          GO RTEXT
007063
007063   %
007063   %  LOCAL SUBROUTINE TO CHECK FOR TERMINATION ENABLED ETC.
007063   %  IF TERMINATION ENABLED AND USER RT-PROGRAM EXCEPT USER TERMINATION
007063   %  HANDLING RT-PROGRAM THEN EXITA, ELSE EXIT
007063
007063   IFTRM: IF FLRTTERM><0 AND X>="RTBES" AND X<"XSEGS" AND X><RTTERM THEN
007076                  EXIT       % IF TERMINATION
007077          ELSE
007100                  EXITA      % IF NOT TERMINATION
007101          FI
007101   RBUS
007124
007124
007124   %==========================================================================
007124   % 4.10        M A L T N   M A L T F
007124
007124   % MONITOR CALLS:
007124   % CALL ALTON(PAGE TABLE NO); PT=0,1,2,3; ONLY 3 FOR BACKGROUND
007124   % CALL ALTOF
007124
007124   SUBR MALTN,MALTF
007124   MALTN: CALL GET1
007125          IF D0 >=0 AND <=3 THEN
007132             A SH 7=:D
007134             IF BACKGROUND><0 THEN
007136                CURPROG.ACTSEG/\377*5SEGSIZE+SEGSTART=:X
007144                IF X.S2 SHZ -10><200 GO RET
007151                X.S2/\377+100 SH 1=:D
007156                BACTPRI/\177177\/D=:BACTPRI
007162             FI
007162             OLDPAGE/\177177\/D=:OLDPAGE
007166          FI
007166          GO RET
007167
007167   MALTF: CALL GET0; OLDPAG
007171          *BLDA 110 DA; BSTA 70 DA; BLDA 120 DA; BSTA 100 DA
007175          A=:OLDPAG; IF T:=BACKGROUND><0 THEN A/\177770=:BACTPRI FI
007203          GO RET
007204   RBUS
007220
007220   %==========================================================================
```

```
007220   % 4.11        R D S C
007220
007220   % MONITOR CALL: ICNCT=RTDSC(RTPROG,IARR)
007220   SUBR RDSC
007220   DISP 0; INTEGER ADR=D2,COUNT=D3
007220           INTEGER ARRAY POINTER DVXXX=D4
007220   PSID
007220   CMOVB:  *BSET ONE
007221           T=:D BONE 16; *MOVB; JMP *; EXIT
007226
007226   RDSC:   T:="P1"; CALL GET2; T=:A=:ADR
007232           T:=27; X:=RTREF=:D; CALL CHLIM; GO ERR; CALL RTCHECK
007240           IF DEMAND><0 THEN
007242              MLEV; *MST PIE
007244           FI
007244           X:=ADR; D0; T:=20; CALL ALTON; CALL CMOVB
007251           *BSET ZRO
007252           ADR+10=:L; X:=D0.RTDLGADDR; T:=0; L:=:B; *BSET ONE
007262           *LDDTX 00; STD 0,B; LDDTX 20; STD 2,B
007266           *LDDTX 40; STD 4,B; LDDTX 60; STD 6,B; BSET ZRO
007273           B+10=:X; B:=L; D0+10; T:=16; CALL CMOVB; *BSET ZRO
007303           0=:COUNT=:ZAREG; D0=:D
007307           MLEV; *MCL PIE; POF
007312           FOR COUNT TO 24 DO
007316                 IF X:=COUNT><1 THEN CNVRT(X)=:"DVXXX"; DVXXX(0)+A=:L
007330                   FOR T:=1 STEP 2 TO L DO X:=DVXXX(T)
007335                     IF X><0 AND X.TYPRING BIT 5CONCT AND X.DCNRT=D THEN MIN ZAREG FI
007345                   OD
007347                 FI
007347           OD
007353           GO RET
007354   ERR:    -1=:ZAREG; GO RET
007357   RBUS
007367
007367   %==========================================================================
007367   % 5.0         L O G P H
007367
007367   %SUBROUTINE TO CONVERT LOGICAL NUMBER
007367   % ENTRY:      A=LOGICAL NUMBER
007367   % RETURN:     AD=DATAFIELDS, 0 IF ERROR
007367   % ANY LEVEL
007367
007367   SUBR LOGPH
007367   DISP 0; INTEGER LOGMAX; DOUBLE IOFIELDS; PSID
007367   INTEGER K37777:=37777,K140000:=140000,SAVX,SAVA,COUT=?
007373   INTEGER ARRAY PIOARR(0); *PIOF;PON;ION;PION
007377   LOGPH: A=:SAVA; AD SHZ -6                % A=LOG.NO/100
007401          IF >RDLOO THEN                   % REMOTE DEVICE OR
007404          IF <RDHII THEN A-RDLOO ELSE 0=:A=:D; EXIT FI; FI    %  NO SUCH GROUP
007414          X=:T:=A; *TRA STS; PIOF          % TEST STATUS & TURN OFF INTERRUPT & PAGING
007420          A/\K140000; D SHZ -12\/A         % D=LOGNO MOD.100 & PON/ION BITS
007423          IF X=A:=1 THEN
007426             *PION                         % ACCESS FILES IN PION
007427          FI
007427          X:=CNVRT(X)                      % GET LOGNO GROUP
007430          IF A:=D/\K37777-X.LOGMAX>>=0 THEN
007435             A:=0=:X                       % NON-EXCISTING
007437          ELSE
007440             A:=D/\K37777; X+A+A           % GET INDEX
007444             X.S1; X:=X.S2                 % A & X CONTAINS INPUT/OUTPUT PART
```

```
007446          FI
007446          *PIOF                                   % PROTECT SAVX
007447          X=:SAVX                                 % SAVE OUTPUT PART
007450          D SHZ -16; X:=PIOARR(D)=:COUT           % ORIGINAL STATE OF PAG. & INTERR.
007454          X:=SAVX=:D                              % D CONTAINS OUTPUT PART
007456  INTEGER COUT                                    % PION-ION-PON-PIOF
007457          T=:X; EXIT                              % RESTORE X
007461
007461  RBUS
007463
007463  %=================================================================
007463  % 5.1          R T E N T R Y
007463
007463  % SUBROUTINE TO START AN RT-PROGRAM
007463  %X=RT-PROG; IT IS PUT INTO EX.QUEUE, AND POSSIBLY INITIALISED
007463
007463  SUBR RTENTRY
007463  DISP 0; REAL FDTIME=DTIME; PSID
007463  INTEGER POINTER LREG
007464  RTENTRY: A:=L=:"LREG"; IF X=0 THEN CALL ERRFATAL FI
007471          IF X.ACTPRI BIT 5RTOFF GO TFINT
007474          IF X.WLINK=0 THEN
007476                X.STATUS BZERO 5REP=:X.STATUS
007501                IF A NBIT 5RWAIT THEN
007503                  X.STADR
007504                  IF T:=CURPROG=X THEN
007507                     *IRW ALEVB DP
007510                  ELSE
007511                     X=:D:=X.RTDLGADDR; *POF
007514                     A=:X.DPREG; X:=D; *PON
007517                  FI; X.SEGM=:X.ACTSEGM
007521                  X.ACTPRI/\154177; T:=14000/\A SH -2; A\/T\/ALEVB
007530                  T SH -2\/A; X.STATUS SHZ -10/\3\/T=:X.ACTPRI
007537                ELSE
007540                  X.STATUS BZERO 5RWAIT=:X.STATUS
007543                  IF A BIT 5TMOUT THEN                % TMOUT IN PROGRESS
007545                     IF X.TLINK><0 THEN CALL FTIMQ; A:=1 ELSE A:=0 FI
007553                     X=:D:=X.RTDLGADDR; *POF
007556                     A=:X.DAREG; X:=D; *PON
007561                     X.STATUS BZERO 5TMOUT=:X.STATUS
007564                  FI
007564                FI
007564                CALL TOEXQU; 1=:MTOR
007567                IF X.TLINK=0 THEN MTIME=:X.DTIME FI
007573          ELSE
007574                X.STATUS BONE 5REP=:X.STATUS
007577          FI
007577  TFINT:  IF X.STATUS BIT 5INT THEN
007602                IF X.TLINK><0 THEN CALL FTIMQU FI
007605                AD:=MTIME
007606                T:=X.DTIN2; D+T; A:=A+C+X.DTIN1
007612                AD=:X.DTIME; CALL TTIMQU
007614          FI
007614          GO LREG
007615  RBUS
007630
007630
```

```
007630
007630   %=========================================================================
007630   %
007630   %     - - -6.0     M O N I T O R    Q U E U E   O P E R A T I O N S
007630   %
007630   %=========================================================================
007630   % 6.1         R E S E R V E   R E L E A S E
007630
007630   % MONITOR LEVEL ROUTINES TO RESERVE AND RELEASE
007630   % A=LOG. UNIT, X=RT-PROGRAM
007630
007630   SUBR RESERVE,RELEASE
007630   INTEGER BREG; INTEGER POINTER LREG
007632   RESERVE: T:=L=:"LREG"; CALL LOGPH; IF =0 THEN CALL ERRFATAL FI
007637           A:=:B=:BREG; CALL BRESERVE; GO OUT
007643
007643   RELEASE: T:=L=:"LREG"; CALL LOGPH; IF =0 THEN CALL ERRFATAL FI
007650           A:=:B=:BREG; CALL BRELEASE
007653   OUT:    T:=BREG=:B; GO LREG
007656   RBUS
007662
007662   %=========================================================================
007662   % 6.2         X L O C K   X U N L O C K
007662   %
007662   % SUBROUTINES TO RESERVE AND RELEASE SYSTEM RESOURCES
007662   % ENTRY:      A=LOGICAL NUMBER; IF =0, RETURN
007662   % IF NON-EXISTENT, STOP IN ERRFATAL
007662   % IF ALREADY OCCUPIED, WAIT
007662   % RETURN: A=0
007662   % RT LEVEL
007662
007662   SUBR XLOCK,XUNLOCK
007662   XLOCK: IF =0 THEN EXIT FI; *IOF; IRW MLEVB DA
007666          "MLOCK"; GO XLO
007670   XUNLOCK: IF =0 THEN EXIT FI; *IOF; IRW MLEVB DA
007674          "MUNLOCK"
007675   XLO:   *IRW MLEVB DP
007676          MLEV; *MST PID
007700          A:=0; *ION; EXIT
007703
007703   % MONITOR LEVEL, LOCK:
007703   MLOCK: CALL LOGPH; IF =0 THEN CALL ERRFATAL FI; A=:B; X:=RTREF; CALL BRESERVE
007711          IF <0 THEN CALL FREXQU; CALL TOWQU; CALL ANTIJAMMER; GO STUPR FI
007716          GO MONEN
007717   % MONITOR LEVEL, UNLOCK:
007717   MUNLOCK: CALL LOGPH; IF =0 THEN CALL ERRFATAL FI; A=:B
007723          IF X:=RTREF=RTRES THEN CALL BRELEASE FI
007730          GO MONEN
007731
007731   RBUS
007745
007745   %=========================================================================
007745   % 6.3         B R E S E R V E
007745
007745   % BASIC ROUTINE TO RESERVE RESOURCES
007745   % ENTRY:      X=RT-PROG, B=DATAFIELD
007745
007745   SUBR BRESERVE
007745   BRESERVE: IF RESLINK=0 THEN   %RESOURCE FREE
007747                X.BRESLINK=:RESLINK; A:=B=:X.BRESLINK   %INSERT
```

```
007753                  X=:RTRES; A:=0
007755          ELSE IF X=RTRES THEN A:=0 ELSE -1 FI
007764          FI
007764          EXIT
007765   RBUS
007765
007765   %=========================================================================
007765   % 6.4          B R E L E A S E
007765
007765   % BASIC ROUTINE TO RELEASE RESOURCES
007765   % ENTRY:      X=RT-PROG,IF=0 ANY PROGRAM; B=DATAFIELD
007765
007765   SUBR BRELEASE
007765   TRIPLE SVTAD
007770   INTEGER POINTER LREG; INTEGER XREG
007772   BRELEASE: X=:XREG; TAD=:SVTAD; A:=L=:"LREG"
007776          IF B>=RTSTART AND B<SEGSTART THEN CALL ERRFATAL FI
010005          IF RESLINK=0 GO OUT; IF RTRES=0 THEN CALL ERRFATAL FI
010012          IF X><A AND X><0 THEN CALL ERRFATAL FI
010016          X:=A+5BRESLINK; CALL RESAJ
010021   LOOP:  IF X.RESLINK><B THEN A=:X; GO LOOP FI      % FIND PREVIOUS
010026          RESLINK=:X.RESLINK; 0=:RESLINK=:RTRES      % REMOVE
010032          IF TYPRING BIT 5IOBT THEN 0=:ISTAT FI
010036          IF SWPFLAG><0 THEN CALL SWPRELEASE FI      % RESET SWAPPING-QUEUE
010041   % ACTIVATE ANY WAITING EXCEPT PRIORITY 0 PROGRAMS(TO PREVENT DEADLOCK)
010041          IF X:=BWLINK><B THEN
010044             IF X.STATUS/\377=0 AND X><"DUMMY" AND X><"DUMM2" AND X><"STSIN" GO OUT
010060             X.WLINK=:BWLINK; 0=:X.WLINK             % REMOVE FROM WAIT QUEU
010063             CALL BRESERVE; CALL TOEXQU
010065             1=:MTOR
010067          FI
010067   OUT:   X:=XREG; TAD:=SVTAD; GO LREG
010072   RBUS
010107
010107   %=========================================================================
010107   % 6.5          T O E X Q   T O W Q U
010107
010107   % SUBROUTINES TO PUT RT-PROGRAMS INTO WAIT QUEUE
010107
010107   SUBR TOEXQ,TOWQU
010107   INTEGER XREG,BREG; INTEGER POINTER LREG
010112
010112   % ENTRY:      X=RT-PROG:
010112   TOEXQ: "BEXQU-5BWLINK":=:B=:BREG:=L=:"LREG"
010117          IF X.WLINK><0 THEN CALL FRWQU FI
010122          CALL TOWQU
010123          BREG=:B; GO LREG
010126
010126   % ENTRY:      X=RT-PROG, B=DATAFIELD
010126   TOWQU: X=:XREG
010127          IF X.WLINK><0 THEN CALL ERRFATAL FI
010132          X.STATUS/\377=:T          %PRIOR
010135          X:=B+"5BWLINK-5WLINK"
010137   LOOP:  X=:D:=X.WLINK
010141          IF X><B AND X.STATUS/\377>=T GO LOOP
010147          T:=D.WLINK; XREG=:X.WLINK; T=:A.WLINK   %INSERT
010155          EXIT
010156   RBUS
010163
010163   %=========================================================================
```

```
010163   % 6.6          F R E X Q U    F R W Q U
010163
010163   %SUBROUTINES TO REMOVE RT-PROGRAMS FROM EWAIT QUEUE
010163   % ENTRY:       X=RT-PROG.
010163   % RETURN:      FRWQU: A CONTAINS DATAFIELD ON RETURN
010163   SUBR FREXQU,FRWQU
010163   INTEGER BREG,XREG,AREG
010166
010166   FREXQU: X=:XREG; "BEXQU-5BWLINK":=:B=:BREG; GO BFR
010173   FRWQU: X=:XREG; A;=B=:BREG
010176          IF X.WLINK=0 THEN EXIT FI
010201          T:=RTSTART; SEGSTART
010203   LOOP1: X:=X.WLINK; IF X>=T AND X<A GO LOOP1
010210          X=:B=:AREG:=XREG
010213   %COMMON PART:
010213   BFR:   X=:T:=B+"5BWLINK-5WLINK"
010216   LOOP2: IF X.WLINK=B GO OUT        %NO SUCH PROGRAM
010221          IF ><T THEN A=:X; GO LOOP2 FI
010225          T=:B; WLINK=:X.WLINK; 0=:WLINK
010231   OUT:   X:=BREG=:B:=XREG; AREG; EXIT
010236   RBUS
010241
010241   %==========================================================================
010241   % 6.7          D M S T R
010241
010241   %DUMMY RT-PROGRAM, ALWAYS ACTIVE
010241   SUBR DMSTR
010241   DMSTR: *WAIT
010242          GO DMSTR
010243   RBUS
010244
010244
```

```
010244
010244   %==========================================================================
010244   %
010244   %  7.0            T I M E   H A N D L I N G
010244   %
010244   %==========================================================================
010244   % 7.1          I C  L K
010244
010244   %ENTRY FROM LEVEL 13, CLOCK
010244   %CALLS CALENDAR ROUTINE AND ACTIVATES PROGRAMS WHOSE TIMES ARE OUT
010244   % MONITOR LEVEL
010244
010244   SUBR ICLK
010244   REAL POINTER FATIME:=ATIM1
010245   ICLK:  CALL KALDR; *PIOF
010247          CALL 500HIST;
010250          CALL HISTI; CALL CLXMSG; *PION            % CLXMSG CAN RETURN WITH ION
010253          DO
010253                IF X:=BTIMQU=-1 GO OUT
010257                FATIME
010260                A-X.DTIM2:=X.DTIM1; *RADD ADC CM1 SA DT
010263                IF T BIT 17 GO OUT
010265                IF X.STATUS NBIT 5INT THEN
010270                   X.TLINK=:BTIMQU; 0=:X.TLINK   %OUT OF TIME QUEUE
010273                FI
010273             CALL RTENTRY
010274          OD
010275   OUT:   GO MONEN
010276   RBUS
010305
010305   %==========================================================================
010305   % 7.2          T T I M Q
010305
010305   % SUBROUTINE TO PUT AN RT-PROGRAM INTO TIME QUEUE
010305   % ENTRY:      X=RT-PROG
010305
010305   SUBR TTIMQ
010305   INTEGER BREG,XREG; INTEGER POINTER LREG
010310   TTIMQ: A:=L=:"LREG":=B=:BREG; X=:XREG
010315          IF X=0 THEN CALL ERRFATAL FI  %NO RT-PROGRAM
010320          IF X.TLINK><0 THEN CALL FTIMQ FI
010323          X.DTIME; L:=D-; *COPY CM1 ADC SA DT
010327          X:="BTIMQU"
010330   LOOP:  X=:B:=TLINK; IF X=A:=-1 GO OUT
010335          X.DTIME; D+L; A+T+C; IF <0 GO LOOP
010341   OUT:   XREG=:TLINK; A:=:X=:X.TLINK
010345          BREG=:B; GO LREG
010350   RBUS
010353
010353   %==========================================================================
010353   % 7.3          F T I M Q U
010353
010353   % SUBROUTINE TO REMOVE AN RT-PROGRAM FROM TIME QUEUE .
010353   % ENTRY:      X=RT-PROGRAM
010353
010353   SUBR FTIMQU
010353   INTEGER XREG
010354   FTIMQU: X=:XREG=:T
010356          X:="BTIMQU"; D:=-1
010360          DO
```

```
010360                  IF X=D GO OUTX
010362                  IF X.TLINK=T GO OUT
010365                  A=:X
010366          OD
010367   OUT:   A:=:X; T:=X.TLINK; D=:X.TLINK; A:=:X; T=:X.TLINK
010374   OUTX:  X:=XREG; EXIT
010376   RBUS
010377
010377   %=========================================================================
010377   % 7.4          K A L D R   K A L N X   S T R T
010377
010377
010377   %SUBROUTINE TO UPDATE CALENDAR, CALLED FOR EACH CLOCK INTERRUPT
010377   % -KALDR- CALLED FROM ICLK
010377   % -KALNX- CALLED FROM CLADJ WITH A=TIME AND X=UNIT
010377
010377   INTEGER ARRAY STRT:=(-62,-74,-74,-30);INTEGER MND:=(-37,-15)
010405                                  -50 60 60 24                31, 13
010405   SUBR KALDR,KALNX
010405   SYMBOL 5DAY=4,5MONTH
010405   INTEGER ARRAY 9CLOC=9CL00,X9CLOC=9CL01
010405   INTEGER POINTER LREG
010406   KALNX: T:=L=:"LREG"; A+9CLOC(X)=:9CLOC(X); GO KALN
010413   KALDR: A:=L=:"LREG"
010415   LOOP:  IF ATIM2-MTIM2<=0 GO LREG    %RETURN POINT FOR KALDR
010421          AD:=MTIME; D+1; A:=A+C; AD=:MTIME
010425          X:=0; MIN 9CLOC(X)          %BASIC UNITS
010427   KALN:  DO
010427                  IF 9CLOC(X)+STRT(X)<0 GO LOOP
010432   LOOPC:         MIN X9CLOC(X)
010433                  A+STRT(X); IF >=0 GO LOOPC
010435                  A-STRT(X)=:9CLOC(X)
010437                  IF X=5MONTH THEN MIN 9CLOC(X); GO LOOP FI
010444                  IF X=5DAY THEN MIN 9CLOC(X); CALL MONTH; T=:MND FI
010452                  X+1
010453          OD
010454   RBUS
010464
010464   %=========================================================================
010464   % 7.5          M O N T H   M N T H 1
010464
010464   % SUBROUTINE TO DETERMINE DAYS IN A MONTH (NEGATIVE)
010464   % A=MONTH, D=YEAR, RETURN WITH RESULT IN T
010464
010464   SUBR MONTH,MNTH1
010464   DOUBLE POINTER D9CL5:=9CL05
010465   MONTH: AD:=D9CL5
010466   MNTH1: IF A-14<=0 THEN A+14 FI
010472          IF A-2=0 THEN   %FEBRUARY
010474                  T:=-35
010475                  IF 3/\D=0 THEN T-1 FI
010501                  EXIT
010502          FI
010502          IF A-6<0 THEN ;*BSET BCM 0 DA
010505          FI
010505          T:=-37; IF A NBIT 0 THEN T-1 FI
010511          EXIT
010512   RBUS
010512
010512
```

```
010512
010512   %==========================================================================
010512   %
010512   %                    T I M E   M O N I T O R   C A L L S
010512   %
010512   %==========================================================================
010512   % 7.6          I N T V   D I N T V   S E T   D S E T
010512
010512   % RT LEVEL
010512   SUBR INTV,DINTV,SET,DSET
010512   DISP 0; DOUBLE DD1=D1; PSID
010512
010512   % CALL INTV(PROG,TIME,UNIT)
010512   INTV:  CALL GET3; CALL RTCHECK
010514          T:=D1; A:=D2
010516          CALL CLCON; X:=D0
010520          GO DINT1
010521
010521   %CALL DINTV(PROG,DOUBLE TIME)
010521   DINTV: CALL GETS2; CALL RTCHECK
010523          DD1
010524
010524   DINT1: IF A<0 GO RET; IF A=0 AND D=0 GO RET; AD=:X.DTINT
010532          X.STATUS BONE 5INT=:X.STATUS
010535          GO RET
010536
010536   %CALL SET(PROG,TIME,UNIT)
010536   SET:   CALL GET3; CALL RTCHECK
010540          T:=D1; D2; CALL CLCON; X:=D0
010544          GO DSET1
010545
010545   %CALL DSET(PROG,DOUBLE TIME)
010545   DSET:  CALL GETS2; CALL RTCHECK
010547          DD1
010550
010550   DSET1: A=:T; D=:L
010552          ATIME; D+L; A+T+C; AD=:X.DTIME
010556          X.STATUS BZERO 5ABS=:X.STATUS; CALL TTIMQ
010562          GO RET
010563   RBUS
010572
010572   %==========================================================================
010572   % 7.7          A B S E T   D A B S T
010572
010572   % MONITOR CALLS: ABSET, DABST; RT LEVEL
010572   DOUBLE ABS6=9TIM5              %BASIC UNITS PER DAY
010572   SUBR ABSET,DABST
010572   DISP 0; DOUBLE DD1=D1; PSID
010572   REAL POINTER FSEC:=9CL01       %SEC,MIN,HOUR
010573   DOUBLE ABS2
010575
010575   % CALL ABSET(PROG,SEC,MIN,HOUR)
010575   ABSET: CALL GET4; CALL RTCHECK
010577          TAD:=FSEC; A:=:D; CALL ABS1
010602          CALL CCNN
010603          T:=9CL00; D+T; A:=A+C; AD=:ABS2
010607          IF D2=:D<0 GO ERR      %MIN
010612          IF D3<0 GO ERR         %HOUR
010614          IF T:=D1<0 GO ERR      %SEC
010617          IF A>X:=27 GO ERR
```

```
010622          IF D>X:=73 GO ERR
010625          IF T>X GO ERR
010627          CALL ABS1; CALL CCNN
010631          AD:=ABS2; L-D; *RADD ADC CM1 SA DX
010634          IF X<0 THEN ABS6; L+D; X+A+C FI
010641          MTIME; D+L; A+X+C
010644          X:=D0; GO DABS3
010646
010646   ABS1:  A*74+D*17                %(HOUR*60+MIN)*15
010651          D:=0; AD SHR 22         %*4
010653          D+T; A:=A+C; T:=9TIM1     %+SEC
010656          EXIT
010657
010657   %CALL DABST(PROG,DOUBLE TIME)
010657   DABST: CALL GETS2; CALL RTCHECK
010661          DD1
010662
010662   DABS3: AD=:X.DTIME; X.STATUS BONE 5ABS=:X.STATUS
010666          CALL TTIMQ
010667          GO RET
010670
010670   ERR:   CALL 9ERRA(#10); GO RETXIT      %PAR. ERROR IN ABSET
010673   RBUS
010713
010713   %==========================================================================
010713   % 7.8          H O L D
010713
010713   % MONITOR CALL: CALL HOLD(TIME,UNIT)
010713   % APPL. LEVEL
010713
010713   SUBR HOLD
010713   HOLD:  CALL GET2;A:=D1
010715          IF T:=D0><0 THEN
010720             CALL CLCON
010721             ATIME; D+L; A+X+C; AD=:RTREF.DTIME; CALL FTIMQU
010727             X.STATUS BZERO 5ABS=:X.STATUS
010732             IF A NBIT 5REP THEN CALL TTIMQ FI; GO RETRTW
010736          ELSE
010737             IF RTREF.TLINK><0 THEN CALL FTIMQU FI
010743             X.STATUS BZERO 5ABS BZERO 5INT BZERO 5REP=:X.STATUS
010750          FI; GO RET
010751   RBUS
010761
010761   %==========================================================================
010761   % 7.9          T M O U T
010761
010761   % MONITOR CALL: ITM = TMOUT(NUNIT,TUNIT)
010761   SUBR TMOUT
010761   TMOUT: CALL GET2; T:=D0; A:=D1; CALL CLCON
010765          ATIME; D+L; A+X+C; AD=:RTREF.DTIME; CALL FTIMQ
010773          X.STATUS BZERO 5ABS BONE 5TMOUT=:X.STATUS
010777          IF A BIT 5REP THEN -1=:ZAREG      % REPEAT BIT WAS SET
011003          ELSE CALL TTIMQ; FI               % ELSE INSERT IN TIME QUEUE
011005          GO RETRTW
011006   RBUS
011015
011015
011015   %==========================================================================
011015   % 7.10         T I M E
011015
```

```
011015   % MONITOR CALL: T=TIME(0)
011015   SUBR TIME
011015   DISP 0; DOUBLE ADREG=ZAREG; PSID
011015   TIME:  CALL GET0; ATIME=:ADREG; GO RET
011021   RBUS
011024
011024   %=============================================================================
011024   % 7.11        C L O C K
011024
011024   % MONITOR CALL: CALL CLOCK(ARRAY)
011024   SUBR CLOCK
011024   DISP 0; INTEGER ADRCLOCK=D1; PSID
011024   CLOCK: T:="P0"; CALL GET0; T=:A=:ADRCLOCK
011030          T:=7; X:=RTREF=:D; CALL CHLIM; GO ERR
011035          FOR X:=-7 DO ACL7(X)=:ECL7(X) OD
011041          IF DEMAND ><0 THEN MLEV; *MST PIE
011045          FI
011045          X:=ADRCLOCK; A:="CL7"+B; T:=16=:D BONE 16; CALL ALTON
011054          *MOVB; JMP *; BSET ZRO
011057   %%       FOR X:=-7 DO ECL7(X); X:=:T; CALL ALTON; A=:X.S0; CALL ALTOFF; X:=:T
011057   %%              T+1; OD
011057          IF BACKGROUND><0 THEN CALL XBMRET FI; GO RET
011063   ERR:   CALL 9ERRA(#09); GO RETXIT
011066   RBUS
011102
011102   %=============================================================================
011102   % 7.12        C L A D J
011102
011102   % MONITOR CALL: CALL CLADJ(TIME,UNIT)
011102
011102   SUBR CLADJ
011102   INTEGER BREG,STQUE; DOUBLE TMP; REAL FTMP=TMP
011106   CLADJ: CALL GET2; T:=D0; D1
011111          IF T>=0 THEN
011113                CALL CLCON; D-; *COPY CM1 ADC SA DA
011116                AD=:TMP
011117                X:=D1-1; D0; CALL KALNX     %UPDATE CALENDAR
011123          ELSE
011124                T-; CALL CLCON
011126                AD=:TMP:=MTIME; D+L; A+X+C; AD=:MTIME
011133          FI
011133          FTMP; A=:L:=B=:BREG
011137          X:="BTIMQU"; 0=:STQUE
011141          DO
011141                X=:B:=TLINK; IF X=A:=-1 GO OUT
011146                IF X.STATUS BIT 5ABS THEN
011151                  X.DTIME; D+L; A+T+C; AD=:X.DTIME
011155                  X.TLINK=:TLINK       %REMOVE
011157                  STQUE=:X.TLINK; X=:STQUE; B=:X     %TO AUXILIARY QUEUE
011163                FI
011163          OD
011164   OUT:
011164   % RE-INSERT IN TIME QUEUE
011164          DO
011164                IF X:=STQUE=0 GO OUT2
011166                X.TLINK=:STQUE; 0=:X.TLINK
011171                CALL TTIMQU
011172          OD
011173   OUT2:  CALL SETCLOCK; BREG=:B; GO RET
011177   RBUS
```

```
011207
011207   %=========================================================================
011207   % 7.13        U P D A T
011207
011207   % MONITOR CALL: CALL UPDAT(MIN,HOUR,DAY,MONTH,YEAR)
011207
011207   SUBR UPDAT
011207   DISP 0; DOUBLE DD3=D3; PSID
011207   @DEC
011207   DATA(0,0,1,1,1982); INTEGER ARRAY LLIMU(0)
011214   DATA(60,24,31,12,2000); INTEGER ARRAY ULIMU(0)
011221   @OCT
011221   UPDAT: CALL GET5
011222          FOR X:=-5 DO
011223                IF CL7(X)<LLIMU(X) OR >ULIMU(X) GO ERR
011232                IF X=-4 AND A=30 THEN
011240                   X:=-5; IF CL7(X)><0 GO ERR        % MINUTES PAST 24 IS WRONG
011243                   X:=-4
011244                FI
011244          OD
011245          DD3; CALL MNTH1; IF D2-<T GO ERR
011253          T=:MND
011254          FOR X:=-5 DO CL7(X)=:ACL7(X) OD; 0=:9CL00=:9CL01
011262          CALL SETCLOCK; GO RET
011264
011264   ERR:   CALL 9ERRA(#11); GO RETXIT      %PAR. ERROR IN UPDAT
011267   RBUS
011303
011303   %=========================================================================
011303   % 7.14        C L C O N   C C N N
011303
011303   % SUBROUTINE TO CONVERT TIME UNIT TO BASIC UNITS
011303   % A=UNIT,T=TIME; RESULT IN AD AND ALSO IN X,L
011303
011303   SUBR CLCON,CCNN
011303   INTEGER POINTER LREG
011304   CLCON: IF A-1<0 OR A-3>0 GO ERR
011311          A+3
011312          A SH 1; BCLCN(A)
011315   CCNN:  X:=L=:"LREG";0=:X=:L
011321          DO
011321                IF T=0 GO OUT
011323                IF T BIT 0 THEN L+D; X+A+C FI
011327                IF A BIT 16 THEN A:=-1=:L; X:=77777; GO OUT FI          % OVERFLOW
011335                T SHZ -1; AD SH 1
011337          OD
011340   OUT:   L=:D; X=:A; GO LREG
011343
011343   ERR:   CALL 9ERRA(#12); GO RETXIT      %ILLEGAL TIME PARAMETER
011346   RBUS
011352
011352
011352   %=========================================================================
011352   % 7.15        T U S E D
011352
011352   %  MONITOR CALL TO GET TIME USED, RETURNED IN AD
011352
011352   SUBR TUSED
011352   TUSED: RTREF,DTINT; X:=ZXREG; EXIT
011356   RBUS
```

```
011357
011357
011357   %=========================================================================
011357   % 7.16        S T E R M                    (TERMP)
011357   %
011357   %  PROGRAM TO START USER-DEFINED CLEAN-UP RT-PROGRAM WHEN
011357   %  RT-PROGRAMS ARE ABORTED (IF ENABLED)
011357
011357   SUBR STERM
011357
011357   INTEGER POINTER USSUB
011360   INTEGER COUNT,PRTTERM:=RTTERM,SAVRT
011363
011363   STERM: GO L1                    % BYPASS CLOSE RT-OPENED FILES
011364          100=:COUNT
011366          FOR COUNT TO 177 DO              % CLOSE ALL FILES OPEND
011372                COUNT; CALL LOGPH          % BY ABORTED PROGRAM
011374                IF A><0 THEN
011375                   *POF; COPY SA DX; LDT ,X RTRES; PON
011401                   IF ABPRO=0 AND 9ERRP(4)=T THEN   % IF ABORTED BY SYSTEM
011407                      T:=COUNT; *MON 2CLOS; MON 2ERMS
011412                   ELSE IF ABPRO=T THEN   % TERMINATED WITH MON ABORT/RTEXT
011416                         T:=COUNT; *MON 2CLOS; MON 2ERMS
011421                      FI
011421                   FI
011421                FI
011421          OD
011425   L1:    IF "USSUB"><0 THEN CALL USSUB FI          % CALL USER SUBR ON PT 0
011430          IF ABPRO=0 THEN        % ABORTED BY SYSTEM
011432
011432   %% DO SOMETHING SPECIAL??
011432                RTREF=:SAVRT
011434          ELSE                   % TERMINATED WITH MON ABORT/RTEXT (A=RT-PROG)
011435
011435   %% DO SOMETHING ELSE???
011435                RTREF BONE 17=:SAVRT
011440          FI
011440          IF RTTERM><0 THEN "PRTTERM"; *MON 2RT % START USER RT-PROGRAM
011444          FI; *MON 2RTEX
011445   RBUS
011453
011453
011453   @DEV 1
011453   @DEV (S-S-J)SINI
```

```
011453
011453   %%%%%%%%%%%%%%%%%%%%%%%%%%%% S I N I %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
011453
011453
011453   %==========================================================================
011453   %
011453   %  8.0            I N T E R R U P T   L E V E L   R O U T I N E S
011453   %
011453   %==========================================================================
011453
011453   %==========================================================================
011453   % 8.1          S L V 1 0   S L V 1 1   S L V 1 2   S L V 1 3
011453   %
011453   % ROUTINES TO TURN OFF PAGING BEFORE STARTING THE WANTED ROUTINE
011453   %
011453   % ENTRY: T=ADDRESS OF ROUTINE
011453   %
011453   SUBR SLV10,SLV11,SLV12,SLV13
011453   SLV10:
011453   SLV11:
011453   SLV12:
011453   SLV13: *POF; COPY ST DP
011455   RBUS
011455
011455
011455   % SUBROUTINES CALLED FROM DRIVERS
011455   %==========================================================================
011455   % 8.2          I D 1 0   W T 1 0   I D 1 1       W T 1 1
011455   %             I D 1 2   W T 1 2   I D 1 3       W T 1 3
011455   %             E R R 2 2
011455
011455
011455   SUBR ID10,ID11,ID12,ID13,ERR22,WT10,WT11,WT12,WT13
011455
011455   % CHECK IF INTERRUPT FROM TERMINAL 65-128
011455   % ENTRY: D=MAX ENTRIES IN TERMINAL-IDENT-CODE TABLE
011455   %        X=ADDRESS TO TERMINAL-IDENT-CODE TABLE
011455   MBIDTERM:
011455           IF A<<="HIDTERM" AND A>>="LIDTERM" AND A-T<<D THEN
011466              X+A; X.S0=:B; IF A><0 GO DRIVER
011473           FI; EXIT
011474   *)FILL
011476
011476   INTEGER C10IDCODE
011477   ID10:   A:=L=:"DRIVER"
011501   WT10:   *PON; WAIT; COPY SA DA          % UNTIL INTERRUPT COMES
011504           *TRA STS
011505           IF A NBIT 16 THEN
011507                   *TRA PID; COPY SA DD; TRA PIE
011512                   A/\D; CALL TWT05
011514           FI
011514           A:=0; *IDENT PL10; POF
011517           IF A-1<<"MAX10" THEN T:=ITB10(A)=:B; IF T><0 GO DRIVER FI
011530           X:="MXX10"=:D:="ID10T"; A=:C10IDCODE; CALL MBIDTERM; A:=C10IDCODE
011536           X:="ITE10"; GO FAR EXTEND
011540
011540   ID11:   A:=L=:"DRIVER"
011542   WT11:   *PON; WAIT; COPY SA DA
011545           *TRA STS
011546           IF A NBIT 16 THEN
```

```
011550                  *TRA PID; COPY SA DD; TRA PIE
011553                  A/\D; CALL TWT10
011555          FI
011555          A:=0; *IDENT PL11; POF
011560          IF A-1<<"MAX11" THEN T:=ITB11(A)=:B; IF T><0 GO DRIVER FI
011571          X:="ITE11"; GO EXTEND
011573
011573  INTEGER C12IDCODE
011574  ID12:   A:=L=:"DRIVER"
011576  WT12:   *PON; WAIT; COPY SA DA
011601          *TRA STS
011602          IF A NBIT 16 THEN
011604                  *TRA PID; COPY SA DD; TRA PIE
011607                  A/\D; CALL TWT11
011611          FI
011611          A:=0; *IDENT PL12; POF
011614          IF A-1<<"MAX12" THEN T:=ITB12(A)=:B; IF T><0 GO DRIVER FI
011625          X:="MXX12"=:D:="ID12T"; A=:C12IDCODE; CALL MBIDTERM; A:=C12IDCODE
011633          X:="ITE12"; GO EXTEND
011635
011635  ID13:   A:=L=:"DRIVER"
011637  WT13:   *PON; WAIT; COPY SA DA
011642          *TRA STS
011643          IF A NBIT 16 THEN
011645                  *TRA PVL
011646                  A=:D SH 11 SHZ -14=:X; *EXR SD
011653                  CALL XFIPV
011654          FI
011654          A:=0; *IDENT PL13; POF
011657          IF A-1<<"MAX13" THEN T:=ITB13(A)=:B; IF T><0 GO DRIVER FI
011670          IF A>=377 AND A<"MAXCA"-1 THEN CALL CARTTE FI
011700          X:="ITE13"; GO EXTEND
011702  *)FILL
011732
011732  %FALSE INTERRUPT, A=IDENT NO.:
011732  ERR22: T:=A; *TRA STS
011734          A SH 4 SHZ -14=:X; CALL 9ERR(#22)
011741  ERRY:   X-12 GOSW WT10,WT11,WT12,WT13
011747
011747  %TEST FOR IDENT TABLE EXTENSION
011747  EXTEND: A+1; IF =0 GO ERR22
011752          IF A>=400 AND <"MAXCA" THEN CALL CATEST FI
011761          DO
011761             IF =X.S0 THEN
011764                 X.S1=:B                  % B => DATAFIELD
011766                 T-1:=:A                  % TO BE COMPATIBLE
011770                 GO DRIVER
011771             FI
011771             IF T+1=0 GO ERR22; X+2
011775          OD
011776
011776  RBUS
012007
012007  %==================================================================
012007  % 8.3            T W T 0 4   T W T 0 5   T W T 1 0 T W T 1 1   T W T 1 2
012007  %                X F I P V
012007  %
012007  %       FIND PREVIOUS LOWER LEVEL INTERRUPTED WITH PAGING OFF
012007
012007  INTEGER ARRAY PVLAD=?
```

```
012007   SUBR TWT04,TWT05,TWT10,TWT11,TWT12,XFIPV
012007
012007
012007   TWT12: *BSKP ZRO 140 DA
012010          GO NLV12
012011   TWT11: *BSKP ZRO 130 DA
012012          GO NLV11
012013   TWT10: *BSKP ZRO 120 DA
012014          GO NLV10
012015
012015   TWT05: *BSKP ZRO 50 DA
012016          GO NLV05
012017   TWT04: *BSKP ZRO 40 DA
012020          GO NLV04
012021   TWT03: *BSKP ZRO 30 DA
012022          GO NLV03
012023   TWT02: *BSKP ZRO 20 DA
012024          GO NLV02
012025   TWT01: *BSKP ZRO 10 DA
012026          GO NLV01
012027   TWT00: *BSKP ZRO 00 DA
012030          GO NLV00; CALL ERRFATAL
012032
012032   NLV12: X:=14; GO FIPVL
012034   NLV11: X:=13; GO FIPVL
012036   NLV10: X:=12; GO FIPVL
012040
012040   NLV05: X:=5; GO FIPVL
012042   NLV04: X:=4; GO FIPVL
012044   NLV03: X:=3; GO FIPVL
012046   NLV02: X:=2; GO FIPVL
012050   NLV01: X:=1; GO FIPVL
012052   NLV00: X:=0; GO FIPVL
012054   *)FILL
012055
012055   FIPVL: A:=X SHZ 3\/153602=:D; *EXR SD  % READ PREVIOUS LEVEL P-REG
012062   XFIPV: X:=PVLAD(X)=:T
012064          IF A<T OR A>T+2 THEN A=:X.S2 FI % SAVE P-REGISTER
012072          A:=D/\153572=:D; A:=X; *EXR SD  % MOVE P-REG ON PREVIOUS LEVEL
012077          EXIT
012100   RBUS
012103
012103
012103   @ICR;
012103   INTEGER ARRAY PVLAD:=(P0PVL,P1PVL,P2PVL,P3PVL,P4PVL,P5PVL,P6PVL,
012112   P7PVL,P8PVL,P9PVL,P10PVL,P11PVL,P12PVL,P13PVL);
012121   @CR;
012121
012121   SUBR P0PVL,P1PVL,P2PVL,P3PVL,P4PVL,P5PVL,P6PVL,P7PVL,P8PVL,P9PVL,P10PVL,P11PVL,P12PVL,P13PVL
012121   P0PVL: *POF; JMP I *+1; 0
012124   P1PVL: *POF; JMP I *+1; 0
012127   P2PVL: *POF; JMP I *+1; 0
012132   P3PVL: *POF; JMP I *+1; 0
012135   P4PVL: *POF; JMP I *+1; 0
012140   P5PVL: *POF; JMP I *+1; 0
012143   P6PVL: CALL ERRFATAL; *0;0
012146   P7PVL: CALL ERRFATAL; *0;0
012151   P8PVL: CALL ERRFATAL; *0;0
012154   P9PVL: CALL ERRFATAL; *0;0
012157   P10PVL: *POF; JMP I *+1; 012060
```

12161

```
012162   P11PVL: *POF; JMP I *+1; 0
012165   P12PVL: *POF; JMP I *+1; 0
012170   P13PVL: *POF; JMP I *+1; 0
012173
012173   RBUS
012174
012174
012174
012174   %==========================================================================
012174   % 8.4          E N T 1 4   E R R 1 4
012174   %            I N T E R N A L   I N T E R R U P T S
012174
012174   %ENTRY FOR INTERNAL INTERRUPTS, LEVEL 14
012174
012174   SUBR ENT14,ERR14,IOB14,OUT14,POFMONC,POFNMON,RET14,MRET14,FINSTR,BEG14
012174   DISP -1; INTEGER SN1; PSID
012174   BASE B14
012174          INTEGER PERR,ACTLV,IBITNO,PESERR,PEAERR,SADINSTR(0); *SAD
012202          DOUBLE MMAP:=(1041,2527)
012204   ESAB
012204   ENT14: "B14"=:B; GO BEG14
012207   IOB14:
012207   IOB:   *IRW BLEVB DP
012210          BLEV; *MST PID
012212   RET14:
012212   MRET14: *PON
012213   YWAIT: T:=1000=:D; *WAIT; COPY SA DA
012217   BEG14: *TRA IIC
012220          IF T=D GO NOMONCALL
012222   MONCALL: *TRA PGS; TRA STS; POF
012225          GO POFMONCALL
012226   NOMONCALL: A=:IBITNO; *TRA STS
012230          IF A NBIT 16 THEN
012232                  *TRA PVL
012233                  A=:D SH 11 SHZ -14=:X; *EXR SD
012240                  CALL XFIPV
012241          FI
012241          *POF
012242          GO POFNMONC
012243
012243   FINSTR: *PON                    % FETCH INSTRUCTION FROM USER-AREA, X=ADDR OF INSTRUCTION+1
012244          X.SN1=:D; *POF
012247          EXIT
012250
012250   ERR14: "B14"=:B; IBITNO; T:=PERR; CALL 9ERR(#40); "B14"=:B; GO RET14
012261   RBUS
012271
012271   %==========================================================================
012271   % 8.5        I D L E   L O O P:  E N T 0   P E N T 0
012271   %
012271
012271   %BACKGROUND LOOP ON LEVEL 0 TO DISPLAY A LOCATION IN T
012271   SUBR ENT0,PENT0
012271   PENT0: *PION; TRA PEA
012273   ENT0:  DO
012273              *COPY ST DX
012274              *JNC *
012275              *RINC DD; COPY SA ADC DA
012277          OD
012300   RBUS
```

```
012300
012300
012300
012300   %=============================================================================
012300   %              A U X I L I A R Y   R O U T I N E S
012300   %=============================================================================
012300   % 8.6            R D A T A   W D A T A
012300   %
012300
012300   % SUBROUTINES TO SET AND RESET WAITING STATE
012300   % X=RT-PROG, B=DATAFIELD
012300
012300   SUBR WDATA,RDATA
012300   WDATA: X.STATUS BONE 5WAIT=:X.STATUS; 1=:ISTATE
012305          EXIT
012306
012306   RDATA: IF ISTATE>0 THEN
012311                  0=:ISTATE
012312                  IF X><0 THEN X.STATUS BZERO 5WAIT=:X.STATUS FI
012316          FI
012316          EXIT
012317   RBUS
012317
012317   %=============================================================================
012317   % 8.7            R T A C T
012317   %SUBROUTINE TO PUT A DATAFIELD INTO THE MONITOR QUEUE
012317   %CALLED FROM INTERRUPT LEVELS
012317
012317   SUBR RTACT
012317   RTACT: *IOF
012320          IF MLINK=0 THEN MQUEUE=:MLINK; A:=B=:MQUEUE; MLEV; *MST PID
012330          FI
012330          *ION
012331          EXIT
012332   RBUS
012333
012333
012333   %=============================================================================
012333   % 8.8            X R T A C T
012333   %
012333   % SUBROUTINE TO PUT A DATAFIELD INTO MONITOR QUEUE
012333   % MUST BE CALLED WITH INTERRUPT OFF!
012333   % B=DATAFIELD
012333
012333   SUBR XRTACT
012333   XRTACT: IF MLINK=0 THEN MQUEUE=:MLINK; A:=B=:MQUEUE; MLEV; *MST PID
012343          FI; EXIT
012344   RBUS
012345
012345
012345   %=============================================================================
012345   % 8.9          X S E T U S E R   M X S E T U S E R
012345   %
012345   % SUBROUTINE TO SET USER PAGETABLES AND RING  (CALLED FROM OPCOM AND SYS SEG)
012345
012345   SUBR XSETUSER,MXSETUSER,XMXSETUSER
012345
012345   INTEGER POINTER PCMDFFIELD:=CMDFFIELD
012346
012346   XMXSETUSER: K:="0"; GO FELLS
```

```
012350   XSETUSER:
012350           BACTPRI
012351   MXSETUSER: K:=1
012352   FELLS: X=:D; A=:CURPROG.ACTPRI/\3773; *TRR PCR
012357           IF K THEN X.STATUS BZERO 11=:X.STATUS FI; D=:X
012365           0=:PCMDFFIELD; EXIT
012367
012367   RBUS
012372
012372
012372   %=============================================================================
012372   %                T I M E R    R T - P R O G R A M
012372   %=============================================================================
012372   % 9.0           T I M E R
012372
012372   % RT-PROGRAM FOR DEVICE TIME-OUT
012372
012372   @DEC; INTEGER TCNTI:=-3600; @OCT        % ONE HOUR
012373   SUBR TIMER
012373
012373   SYMBOL 5ESC=33
012373   INTEGER ARRAY POINTER BATAB:=BCHTAB
012374   INTEGER COBA,COBAMAX,BXSA,CADDR
012400   @DEC; INTEGER CNT:=-3600; @OCT          % ONE HOUR
012401   INTEGER POINTER IPITO:=177000+5BFPAGE+5BFPAGE
012402   DOUBLE POINTER DPITO=IPITO
012402
012402   TIMER: X:="TMRTAB"=:CADDR
012404           DO
012404              *PION; PIOF
012406              CADDR.SO
012410           WHILE A=:B><-1
012414   @LIB CXCPU-,
012414   @LIB CXCPU
012414              IF A><0 THEN
012415                 IF TYPRING BIT 5TERM THEN
012420                    T:=TDFBANK; X:=TDFLGADDR+5TMR; *LDATX
012424                    IF A><0 THEN
012425                       A+1; *STATX
012427                       IF A=0 THEN
012430                          CALL SPTOWINDOW; *PON
012432                          CALL TMSUB; *POF
012434                          0=:IPITO
012435                       FI
012435                    FI
012435                 ELSE
012436                    IF TMR><0 THEN MIN TMR; GO OUT2; CALL TMSUB; FI
012443                 FI
012443              FI
012443   @ELIB
012443   OUT2:      MIN CADDR
012444           OD
012445           *PION
012446           CALL READCLOCK
012447           BCHTAB=:COBAMAX; 1=:COBA
012453           FOR COBA TO COBAMAX DO
012457              COBA SHZ 2-1=:X; A:=BATAB(X); CALL LOGPH; A=:B; X-2
012467              IF BATAB(X)=0 GO PASS; X=:BXSA+1
012473              IF A<0 THEN X:=BATAB(X); GO ABJ FI
012476              IF MXTIM <0 GO PASS; T:=5670; *RMPY ST DA
```

```
012502              *LDX I BATAB,X; IOF
012504              A:=:D-X.DTIN2; *RDCR ADC DD
012507              A:=:D-X.DTIN1
012511              IF A<0 THEN
012512   ABJ:           *IOF
012513                  IF BSTATE=2 OR FLAGB BIT 5ESCON THEN
012522                    FLAGB BONE 5ABJOB=:FLAGB; 5ESC=:LAST; CALL ESCAPE; 0/\0
012531                    77777=:MXTIM; 1=:BATAB(BXSA)
012536                  FI
012536              ELSE
012537                  CALL CH5MXTIME; GO ABJ     % CHECK N-500 JOB TIMEOUT
012541              FI; *ION
012542   PASS:  OD
012546   % ENABLING ALL MEMORY ERRORS EACH HOUR
012546          IF TCNTI><0 THEN
012550              MIN CNT; GO OUT; TCNTI=:CNT; 4; *TRR ECCR
012556          FI
012556   OUT:   *IOF
012557          X:=RTREF; CALL FREXQU; 1=:MTOR
012563          "MONEN"; *IRW MLEVB DP
012565          MLEV; *MST PID; ION; JMP *                % INSTEAD OF MON RTEXT
012571   RBUS
012610
012610
```

```
012610
012610   %==========================================================================
012610   %
012610   %  11.0            B L O C K    I / O
012610   %
012610   %==========================================================================
012610   %      U S E R   M O N I T O R   C A L L S
012610
012610
012610
012610   % MAGTP IS FOUND AMONG THE OPTIONS AT THE END OF THE SYSTEM.
012610   % THE OTHERS ARE IN THE FILE SYSTEM.
012610   % THIS PORTION CONTAINS CALL PREPARATION
012610
012610   %==========================================================================
012610   % 11.1          M R F I   M R F I O
012610   %
012610   % ENTRIES FOR FGET-FPUT(FILE INBT/OUTBT) CALLED FROM FILE SYSTEM
012610   % INBT/OUTBT LEVEL
012610   SUBR MRFI,MRFO
012610   MRFI:  1; GO MRFIO
012612   MRFO:  2
012613   MRFIO: *IRW MLEVB DX
012614          "CALLPROC"; *IRW MLEVB DP
012616          MLEV; *MST PID; WAIT
012621   RBUS
012622
012622
012622   %==========================================================================
012622   % 11.2          M R F I L   M O F I L   M O F I 2 X R F I L E
012622   %               X W F I L E   M A G T P   X R P A G E   X W P A G E
012622   %               X F G E T   X F P U T
012622   %
012622
012622   INTEGER MOLDN=?     % DEVICE INDEPENDENT CALLS FROM RT
012622
012622   @ICR
012622   SUBR MRFIL,MOFIL,MOFI2,XRFILE,XWFILE,MAGTP,XRPAGE,XWPAGE,YFGET,YFPUT,XMRW,
012622   500RF,500WF,500MT,MOFIA,MOFIB;
012622   @CR;
012622
012622   % ENTRYPOINT FOR DEVICE INDEPENDENT CALLS; OPEN, CLOSE...
012622   % PROGRAM RWRT2 AND DATAFIELD USED FOR RT-PROGRAMS
012622   % MONITOR LEVEL:
012622   INTEGER SCSTART
012623   % FOR RT-PROGRAM:
012623   MOFIL: CSTART=:SCSTART
012625          IF DEMAND=0 THEN "NDAMOFIL"=:CSTART; GO MONDEM FI
012632          "AMOFIL"=:CSTART
012634          "DEMFIELD"=:B; X:=RTREF; CALL BRESERVE; IF A<0 GO WT
012642          A:=B; CALL SWAPPREG; SCSTART; *IRW ALEVB DT
012646          GO MONEN
012647   % FOR BACKGROUND PROGRAMS
012647   MOFIA: A:=1; GO MOFF
012651   MOFIB: A:=2; GO MOFF
012653   MOFI2: A:=0
012654   MOFF:  X:=CSTART; *IOF
012656          *IRW BLEVB DA
012657          "BMOFF"; *IRW BLEVB DD
012661          "MFBBCL"; *IRW BLEVB DP
```

```
012663          A:=X; *IRW BLEVB DT
012665          BLEV; *MST PID; ION
012670          GO MONEN
012671
012671   % RT-LEVEL ENTRIES:
012671   % T=MONITOR CALL ROUTINE ADDRESS (CSTART)
012671
012671   % RT-PROGRAMS:
012671   NDAMOFIL: T:=SCSTART
012672   AMOFIL: CALL GETO; T=:"MRSTA"; GO CMO
012675
012675   % BACKGROUND PROGRAMS:
012675   BMOFF: CALL GETO
012676   CMO:    1=:WFLAG; 0=:MTFLG
012701          IF BACKGROUND><0 THEN "DFS2"
012704          ELSE MOLDN; CALL LOGPH; IF =0 THEN CALL ERRFATAL FI
012711          FI; GO FAR DFRES
012712
012712   % ENTRYPOINTS FOR MAGTP,RFILE,WFILE
012712   % CALL MAGTP(FUNC,CORAD,LOGNO,MAX. WORDS,READ WORDS)
012712   %CCALTON: IF X.MTFLG=0 GO SALTON
012712   %          EXIT
012712   500MT: 2=:X.MTFLG; GO MAGT1
012715   500RF: 2=:X.MTFLG; A:=0; GO RWFF
012721   500WF: 2=:X.MTFLG; A:=1; GO RWFF
012725   MAGTP: 0=:X.MTFLG
012726   MAGT1: CALL GAPIT; CALL DALTON; "P1"; T:=P3; *BSET ZRO
012733          A=:X.ICORAD; T=:X.IMAXW; *BSET ONE
012736          "P4"; *BSET ZRO
012740          A=:X."IRETW"; *BSET ONE
012742          T:=P2; CALL GET1; DO=:IFUNC; T=:IOLOG
012747          1=:WFLAG; MTFLG BONE MCMAGTP=:MTFLG; -1=:IBLOADR; GO MRW
012757   *)FILL
013006
013006   % CALL RFILE/WFILE(LOG.NO,FLAG,ADDRESS,BLOCK NO,NO.WORDS)
013006   XRFILE: "0"; GO RWF
013010   XWFILE: 1
013011   RWF:    0=:X.MTFLG
013012   RWFF:   A=:X.IFUNC; CALL GAPIT; CALL DALTON; P3; T:=P4; *BSET ZRO
013020          A=:X.IBLOADR; T=:X.IMAXW; *BSET ONE
013023          T:="P2"; CALL GET2
013025          T=:ICORAD
013026          IF WFLAG><0 AND BACKGROUND=0 THEN "0" ELSE 1 FI; A=:WFLAG
013036
013036   XMRW:
013036   MRW:
013036          IF IOLOG>=100 AND <=177 THEN     % FILE OR DEVICE
013045             IF IFUNC=0 THEN "RFILE" ELSE IF A><1 GO ERFUN; "WFILE" FI; A=:"MRSTA"
013056             IOLOG; GO FCM
013060          FI % DEVICE
013060   DVCE:  T:="MC144"=:"MRSTA"; A:=:X
013063          IF IFUNC/\176777-77>0 THEN GO ERFUN FI; A:=X
013072          IF T:=BACKGROUND><0 THEN 2=:MOFTYPE; "DFS1"
013100          ELSE            % RT
013101             CALL LOGPH; IF =0 THEN D=:A FI; IF =0 THEN 132; GO ERR FI
013107             IF A.OFFLG BIT 5RCFIL AND X.TYPRING BIT 5RFILE THEN A:="DFRRT"; GO DFRES; FI
013120             IF X.TYPRING NBIT M144B THEN 174; GO ERR FI
013125             IF A BIT 5IBDV OR A BIT 5HDMA THEN
013131                IF IFUNC=1 OR =55 THEN X:=D FI
013141             FI
```

Level: 9

```
013141              IF X.TYPRING BIT 5COM THEN
013144                 IF X.DBPROG><0 THEN 174; GO ERR FI        % BACKGROUND CH.
013150                 IF IFUNC=0 THEN                           % READ
013152                    X.DFDCR; GO DFRES
013154                 ELSE                                      % WRITE
013155                    X=:T; X.DFOPP=:X; X.DFDCW; T=:X; GO DFRES
013163                 FI
013163              FI
013163              X.DFDEV; CALL LOGPH; IF A=0 THEN 174; GO ERR FI
013170           FI; GO DFRES
013171   ERFUN: 201
013172   ERR:   A=:ZAREG; GO RET
013174   *)FILL
013212
013212   % ENTRYPOINTS FOR RPAGE,WPAGE,FGET,FPUT:
013212   XRPAGE: T:="RPAGE"; GO RP
013214   XWPAGE: T:="WPAGE"; GO RP
013216   YFGET:  T:="FGET"; GO RP
013220   YFPUT:  T:="FPUT"
013221   RP:     CALL GETO; T=:"MRSTA"; 1=:WFLAG; ZTREG; 0=:MTFLG
013227
013227   FCM:    IF T:=BACKGROUND><0 THEN "DFS1"; 0=:MOFTYPE
013234           ELSE                                          % RT
013235              CALL LOGPH; IF A=0 THEN D=:A FI IF =0 THEN 132; GO ERR FI
013243              IF A.OFFLG BIT 5RCFIL AND X.TYPRING BIT 5RFILE THEN A:="DFRRT"; GO DFRES; FI
013254              IF X.TYPRING NBIT 5RFILE THEN 133; GO ERR FI; X.RWFIELD
013262           FI
013262
013262   % COMMON PART; A=DF-DATAFIELD
013262   DOUBLE DOLDFI=?
013262   INTEGER ARRAY POINTER OLDFI=?,NEWFI=?   % DEFINED BELOW
013262   DFRES: A=:"NEWFI":=:B=:"OLDFI"
013265           IF BACKGROUND=0 THEN                           % RT
013267                   X:=RTREF; CALL BRESERVE
013271                   IF <0 THEN CALL FREXQU; CALL TOWQU; CALL ANTIJAMMER
013275                     "OLDFI"=:B; ZPREG-1=:ZPREG; GO RETSTUPR
013303                   FI
013303                   IF SSREF=0 THEN   % NO TRANSFER GOING ON
013305   @LIB CXCPU
013305                     "5MRSTA-5REG+1"=:L; X:=5REG; AD:=DOLDFI; D+X; T:=A+X; *MOVNN
013315   @ELIB
013315   @LIB CXCPU-,
013315                     X:=RTREF=:SSREF; X.ACTSEG=:STRSEG
013321                     IF WFLAG><0 THEN CALL WDATA ELSE 0=:"OLDFI".ZAREG FI
013327                     X:=DRT; IF X="RTRFA" THEN X.ACTPRI BZERO 5RTOFF=:X.ACTPRI; FI
013336                     CALL RTENTRY; "OLDFI"=:B
013341                   ELSE                          % OLD TRANSFER NOT FINISHED, WAIT
013342                     X:=RTREF; CALL WDATA; 0=:WFLAG; "OLDFI"=:B; ZPREG-1=:ZPREG
013352                   FI; GO RETSTUPR
013353           ELSE                              % BACKGROUND
013354   @LIB CXCPU
013354                   "5MRSTA-5REG+1"=:L; X:=5REG; AD:=DOLDFI; D+X; T:=A+X; *MOVNN
013364   @ELIB
013364   @LIB CXCPU-,
013364                   CALL COMENTRY             % CONTINUE ON SYSTEM SEGMENT
013365           FI
013365
013365   %SUBROUTINE TO COPY DATAFIELD CONTENTS:
013365   DOUBLE DOLDFI(0)
013365   INTEGER ARRAY POINTER NEWFI,OLDFI
```

50 200

```
013367   @LIB CXCPU-,
013367   RBUS
013413
013413   %=============================================================================
013413   % 11.3          C O M M O N
013413   %
013413
013413   % RT-PROGRAM FOR BLOCK I/O TRANSFER
013413   % THIS IS THE COMMON PART OF THE TRANSFER RT-PROGRAMS
013413   % THE ENTRY-POINTS ARE FOUND AT THE DF-DATAFIELDS
013413   % ENTRY: A POINTS TO DF-DATAFIELD
013413
013413   SUBR COMMON
013413   DISP 0; DOUBLE DADREG=DAREG; PSID
013413   INTEGER POINTER CRTR:=CRTREF            % ON SYSTEM SEGMENT
013414   INTEGER POINTER PFWFLAG:=FWFLAG         % ON SYSTEM SEGMENT
013415   INTEGER POINTER SDFL:=SDFLAG
013416   INTEGER HPAR:=("1","1")                 % HOLD 1 BASIC TIME UNIT
013420   INTEGER XRGX=?,TRGT=?,ARGA=?,DRGD=?
013420   TRIPLE TADR=?,TXTA=?
013420   INTEGER CURSEG
013421   *)FILL
013422
013422   COMMON: A=:B
013423           A:=SSREF; X:=RTREF; CALL COLAM             % COPY ORIGINATORS ACT. LAMUS
013426   NYTRY: "OFLDN"; CALL XLOCK
013430           IF "MRSTA"="MC144" THEN CALL REEIN; T:=SSREF.ACTSEG; CALL MMEXY; GO L1 FI
013441           IF A="OPFIL" OR ="OLDOP" OR ="CLOFI" THEN T:="5ERRSEG\5FILSEG"; X:="FRSG1"
013454           ELSE IF A="ERMSG" OR A="QERMSG" THEN T:=SSREF.ACTSEG; X:=0
013466           ELSE T:=5FILSEG; X:="FRSG1"
013471           FI; FI
013471           MLEV; *MCL PIE; POF
013474           T=:CURSEG; CALL XCSEGS; CALL SETRT; T:=CURSEG
013500           MLEV; *IOF; MST PIE; PON
013504           A:=X; X:=SSREF; CALL RTSVCALLSEGS
013507   L1:     *BSET ONE
013510           SSREF=:CRTR; 0=:SDFL; WFLAG=:PFWFLAG
013515           ZTADREG; X:=ZXREG; CALL MRSTA; GO ERET; GO OK1
013522           K:=1
013523   FELL1: M:=1; *PIOF
013525           TAD=:TADR; X=:XRGX
013527   FELL2: IF X:=SSREF=RTRES AND WFLAG><0 THEN
013535              X:=X.RTDLGADDR; X.DPREG
013537              IF K THEN A+1 FI; IF M THEN A+1 FI; A=:X.DPREG
013546              TAD:=TXTA=:X.TDXREG; DRGD=:X.DDREG
013552           FI; *PION
013553           IF "MRSTA"="MC144" THEN
013557              IF RTREF.RSEGM><0 THEN
013562                 MLEV; *MCL PIE; POF
013565                 CALL XCSEGS; 0=:X.RSEGM=:X.ACTSEG
013570                 MLEV; *PON; MST PIE
013573                 CALL REEUT
013574              FI
013574           ELSE
013575              X:=SSREF; CALL RTRSCALLSEGS
013577           FI; *IOF
013600           T:=RTRES; X:=SSREF:=:T; GO RETE; *)FILL
013626   RETE:   CALL RDATA; IF X=T THEN CALL BRELEASE FI
013632           A:=RTREF; CALL CLLAM; 0=:SSREF
013635           IF X:=RTREF="RTRFA" THEN
```

```
013641              X.ACTPRI BONE 5RTOFF=:X.ACTPRI
013644              "OFLDN"; CALL XUNLOCK
013646              *ION; MON 135; JMP I (RRT1
013651           FI
013651           *ION; MON 2RTEX
013653   INTEGER XRGX,TRGT,ARGA,DRGD
013657   TRIPLE TADR=TRGT,TXTA=XRGX
013657   OK1:    K:="0"; GO FELL1
013661
013661
013661
013661   ERET:   *PIOF
013662           TAD=:TADR; X=:XRGX:=SSREF
013665           IF A+1=0 THEN T:=X; GO RETE FI
013671           IF A+1=0 THEN
013673              *PION
013674              CALL RTRSCALLSEGS; "OFLDN"; CALL XUNLOCK
013677              "HPAR"; *MON 2HOLD
013701              GO FAR NYTRY
013702           FI; M:="0"; K:="0"; GO FELL2
013705
013705   RBUS
013720
013720   %=========================================================================
013720   % 11.4          W A I T F
013720
013720   % MONITOR CALL: T=WAITF(FILE NO,CONTINUE FLAG)
013720   SUBR WAITF
013720   WAITF: CALL GET2; D0; CALL LOGPH
013723           IF =0 THEN D=:A FI; IF =0 THEN 132; GO ERR FI
013730           IF A.TYPRING BIT 5RFILE THEN X:=X.RWFIELD
013735           ELSE IF BIT M144B THEN X:=X.DFDEV ELSE 174; GO ERR FI
013744           FI; X=:D
013745           IF X:=X.RTRES=RTREF THEN
013751                   IF D1><0 THEN -1; GO ERR FI %NOT FINISHED
013755                   B:=:D; CALL WDATA; D=:B     %WAIT
013760           FI; A:=0
013761   ERR:    A=:ZAREG; GO RETSTUPR
013763   RBUS
013770
013770   SUBR LNK1SWAP
013770   RBUS
013770
013770
013770   %=========================================================================
013770   % 11.5          A B S T R
013770
013770   % MONITOR CALL: CALL ABSTR(READ-WRITE,CORAD,MASS.ADDR,NO. HARDW.BLOCKS)
013770   % T=DEV.NO.; CALLED FROM CORE ONLY
013770   %  MONITOR LEVEL
013770
013770   SUBR ABSTR
013770   INTEGER BRG
013771   ABSTR: IF RTREF.ACTPRI NBIT 1 THEN 131; CALL 9ERR(#00); GO BRTEXT FI
014001           *IRR ALEVB DT
014002           A=:X BZERO 17; CALL LOGPH; IF X<0 THEN D=:A FI
014010           IF =0 THEN CALL ERRFATAL FI; A=:B=:BRG
014014           X:=RTREF; CALL BRESERVE; IF A<0 GO WT
014020           *IRR ALEVB DA
014021           CALL MTRANS
```

```
014022
014022   %  RETURN WHEN TRANSFER IS FINISHED
014022          CALL BRELEASE; A:=B=:BRG
014025   L2:    IF HSTAT BIT 4 THEN A BONE 17 ELSE A BZERO 17 FI
014033          IF X=CURPROG THEN
014036             *IRW ALEVB DA
014037          ELSE
014040             X:=X.RTDLGADDR; *POF
014042             A=:X.DAREG
014043          FI; GO PSTUPR
014044   RBUS
014056
014056   %==============================================================================
014056   % 11.6          M T R N S   R E T R A N S
014056   %
014056
014056   % MONITOR LEVEL SUBROUTINE TO GIVE CONTROL TO LEVEL 11, RETURN WHEN FINISHED
014056   % A=PAR.LIST, B=DATAFIELD
014056   % RETURN: X=RTRES
014056
014056   SUBR MTRNS,RETRANS
014056   MTRNS: *IOF; IRW LV11B DX
014060          A:=L=:"TRLREG"; B=:A; *IRW LV11B DB
014064          "STDRIV"; *IRW LV11B DT
014066          "SLV11"; *IRW LV11B DP
014070          LV11; *MST PID
014072          X:=RTRES; CALL WDATA
014074          *PION
014075          IF X=RTREF GO RWAIT; GO MONEN
014101   % - -RETURN, X=BFIELD
014101   RETRANS: *TRR CCLR                          % CLEAR CHACHE
014102          X=:B:=RTRES; IF X=0 GO MONEN
014106          CALL RDATA                          % RESER WAITING STATE
014107          GO TRLREG
014110   RBUS
014117
```

```
014117
014117
014117   %=========================================================================
014117   % 11.7          M T M R S U B   -   F D T M S
014117   %
014117   % FDTMS       : FOR FLOPPY DISCS
014117   % MTMRSUB     : FOR DISCS AND OTHER MASS STORAGE DEVICES
014117   %
014117   SUBR MTMRSUB,FDTMS
014117
014117   %TIME-OUT SUBROUTINE, CALLED BY TIMER RT-PROGRAM
014117
014117   FDTMS: "BUFDISC"; GO L1
014121   MTMRSUB:
014121           "DRIVER"
014122   L1:     *IRW LV11B DT
014123           B=:A; *IRW LV11B DB
014125           "SLV11"; *IRW LV11B DP
014127           LV11; *MST PID
014131           EXIT
014132   RBUS
014135
014135
```

```
014135
014135   %==========================================================================
014135   %
014135   %                    C H A R A C T E R   I / O
014135   %
014135   %==========================================================================
014135   %
014135   %              I / O - M O N I T O R   C A L L S
014135   %
014135   %==========================================================================
014135   %
014135   %
014135   %==========================================================================
014135   % 11.8          I N B T   O U T B T
014135   %               I O R E S T A R T   F I O R E S   W D X
014135   %
014135   % MONITOR CALL INBT OUTBT; T=DEV, A=CHAR,SKIP RETURN IF OK
014135   % INBT/OUTBT LEVEL
014135
014135   SUBR INBT,FINBT,OUTBT,FOUTBT,IORESTART,FIORES,WDX,TERWDX,M8RET
014135
014135
014135   INTEGER POINTER PTTNO=?,PBCHFLAG=?,PCPNT=?,PNCOMPL=?
014135   INTEGER POINTER PLACTPRI=?,TTIF=?,PBMECHO=?,PM4LRG=?,PCI4DFELT=?
014135   INTEGER SACTPRI=?,PCSTRING=?
014135
014135   OUTBT: *IRR ALEVB DT
014136          IF X:=BACKGROUND><0 THEN
014140             0=:PM4LRG
014141             IF A=1 OR =PTTNO THEN A:=PTTNO
014150                IF T:=PBCHFLAG><0 THEN
014153                   CALL LOGPH; D=:B; A=:PCI4DFELT
014156                   X:=B; T:="BCHOSTS"; CALL XGTDFADDR; IF A=0 GO FAR DUM
014163                   X:=PCI4DFELT; CALL CHBMECHO; GO OUTB1; T:=B=:PCI4DFELT
014170                   IF A BIT BOTTERM AND A NBIT BXOTTERM THEN
014174                      "OUTBO"=:PM4LRG; GO OUTB2
014177   OUTBO:             PCI4DFELT=:B
014201                   FI
014201   OUTB1:          X:=B; T:="ROFIL"; CALL XGTDFADDR
014204                FI
014204          FI FI
014204          IF A=0 GO FAR DUM
014206          CALL LOGPH; IF D=0 GO FAR 123ERR
014211          D=:B; A=:D
014213   OUTB2: IF TYPRING NBIT 5IOBT GO FAR 123ERR
014216          IF A NBIT 5RFILE GO FAR IOBTX
014220          IF OFFTP NBIT OPBIT AND NBIT OTBIT GO FOUTB; OFOUD=:B
014227          GO FAR IOBTX; *)FILL
014243
014243   INTEGER POINTER PTTNO:=TTNO,PBCHFLAG:=BCHFLAG,PCPNT:=CPNT,PNCOMPL:=NCOMPL
014247   INTEGER POINTER PLACTPRI:=LACTPRI,TTIF:=TTIFIELD,PBMECHO:=BMECHO
014252   INTEGER POINTER PM4LRG:=M4LRG,PCI4DFELT:=CI4DFELT
014254   INTEGER SACTPRI,PCSTRING:=COMSTRING
014256
014256   INBT: *IRR ALEVB DT
014257          IF X:=BACKGROUND><0 THEN 0=:PM4LRG FI
014262          IF A=0 THEN
014263             IF X=0 GO FAR DUM                          % BACKGROUND
014265             IF X:=PNCOMPL=0 AND X:=PCPNT><0 THEN
014272                MIN PCPNT; T:=PCSTRING; *LBYT
```

```
014275                IF A=15 THEN 0=:PCPNT FI; CALL SETPARITY; *IRW ALEVB DA
014303                GO FAR DUM
014304             FI
014304             IF X:=PBCHFLAG><0 GO BTINBT        % BATCH OR MODE
014307             "EDINB"=:CSTART; "MOFIB"; GO FAR MONACT
014313          FI
014313          IF X:=BACKGROUND><0 THEN
014315             IF A=1 OR =PTTNO THEN
014323   BTINBT:      A:=PTTNO
014324                IF T:=PBCHFLAG><0 THEN
014327                   CALL LOGPH; A=:B=:X; *IRR ALEVB DP
014333                   T:="DBADR"; CALL XSTDFADDR; CALL CHBMECHO; GO INBT1
014337                   IF A BIT BITERM THEN "INBT4"=:PM4LRG; GO INBT3 FI
014344   INBT1:          "BAPROG"; CALL SSPALEV
014346                   X:=B; T:="RIFIL"; CALL XGTDFADDR; *IRW ALEVB DT
014352                FI
014352          FI FI
014352          CALL LOGPH; IF A=0 GO FAR 124ERR; A=:B
014356   INBT3: IF TYPRING NBIT 5IOBT GO FAR 124ERR
014361          IF A NBIT 5RFILE GO IOBTX
014363          IF OFFTP NBIT OPBIT AND NBIT OTBIT GO FINBT; OFIND=:B
014372          GO IOBTX
014373
014373   SSPALEV: *IRW ALEVB DP
014374             RTREF.ACTPRI=:SACTPRI
014377             "100002+ALEVB"=:X.ACTPRI/\3773; *TRR PCR
014403             SACTPRI=:PLACTPRI; EXIT                    % ON SYSTEM SEGMENT
014406
014406   % IF MODE THEN SKIP-RETURN WITH T=BSTATE, A=BMECHO ELSE RETURN
014406   INTEGER LRG
014407   CHBMECHO: IF PBCHFLAG><0 THEN
014411                A:=L=:LRG; T:="BSTATE"; CALL XGTDFADDR; A=:T:=LRG=:L
014420                IF 5BUSER=T OR PBMECHO BIT BXOTTERM THEN L+1; PBMECHO FI
014430             FI; EXIT
014431   INBT4: "BXBAPROG"; CALL SSPALEV
014433   FELL1:
014433   @LIB CXCPU
014433          *PIOF; SAA 42; TRR PCR
014436   @ELIB
014436          GO LPION
014437   *)FILL
014467
014467
014467   %  B=DATAFIELD:
014467   @LIB CXCPU
014467   INTEGER CDFADDR=?
014467   INTEGER POINTER PCDFADDR:=CDFADDR
014470   @ELIB
014470   IOBTX: *PIOF
014471   @LIB CXCPU
014471          IF TYPRING BIT 5TERM THEN
014474             A:=B=:PCDFADDR; CALL SETBFPAGE; *PON
014500             IF PCDFADDR.RTRES><RTREF THEN
014505                IF X.TYPRING NBIT 5NORESERV GO ERR5
014510                IF CFREE=0 THEN
014512                   IF DFOPP><0 THEN
014514                      X:=1777; X/\A; A:=B/\176000; X+A
014521                      IF X.DFLAG BIT 50XON GO DUM
014524                   FI; GO TERWDX
014525                FI
```

```
014525              FI; IF DERROR><0 GO DERRERR
014527              *IRR ALEVB DA
014530              CALL IOTRANS; GO TERWDX
014532              GO FELLS; *)FILL
014540           FI
014540   @ELIB
014540           IF X:=RTREF><RTRES THEN
014544              IF TYPRING NBIT 5NORESRV GO ERR5
014547              IF A NBIT 5BAD THEN
014551                 IF CFREE=0 THEN
014553                    IF DFOPP><0 AND A.DFLAG BIT 5OXON THEN
014561                       GO DUM ELSE GO WDX
014563           FI FI FI FI
014563           IF DERROR><0 THEN
014565   DERRERR:  0=:DERROR
014566   ERR:      *IRW ALEVB DA
014567             GO FELL1
014570           FI; *IRR ALEVB DA
014571           CALL IOTRANS; GO WDX
014573   FELLS: *IRW ALEVB DA
014574           CALL STDEV
014575   DUM:
014575   @LIB CXCPU
014575           *PIOF; SAA 42; TRR PCR
014600   @ELIB
014600           *PION
014601           IF BACKGROUND><0 THEN
014603              *LDA I (M4LRG; STZ I (M4LRG
014605              IF A><0 THEN A=:P FI
014607           FI
014607   RETU:   *IRR ALEVB DP
014610           A+1; *IRW ALEVB DP      %SKIP RETURN
014612   LPION: *PION; WAIT
014614           CALL ERRFATAL
014615
014615   MBRET: *IOF
014616           CALL STDEV
014617   @LIB CXCPU
014617           *PIOF
014620           A:=42; *TRR PCR
014622   @ELIB
014622           GO RETU
014623
014623   124ERR: A:=124; GO ERR
014625   123ERR: A:=123; GO ERR
014627   ERR5:   A:=5; GO ERR
014631   ERR3:   A:=3; GO ERR
014633   *)FILL
014640
014640   %NOT READY:
014640
014640   WDX:    IF DERROR><0 GO DERRERR                    % ERROR
014642           IF ISTATE<0 THEN CALL STDEV; GO ERR3 FI
014646           IF X:=RTREF=RTRES THEN CALL WDATA FI
014653           IF X=RTRES OR TYPRING NBIT 5NORESERV OR CFREE><0 THEN CALL STDEV FI
014664   INWDX: *PIOF; IRR ALEVB DP
014666           A-1; *IRW ALEVB DP
014670           "RWAIT"
014671   MONACT: *IRW MLEVB DP
014672           MLEV; *MST PID
```

```
014674   @LIB CXCPU
014674           A:=42; *TRR PCR
014676   @ELIB
014676           *PION; WAIT
014700           CALL ERRFATAL
014701   *,FILL
014705   @LIB CXCPU
014705   INTEGER CDFADDR
014706   TERWDX: IF DERROR><0 GO DERRERR
014710           IF TDRADDR.ISTATE<0 THEN CALL STDEV; GO ERR3 FI
014715           IF X.RTRES=RTREF THEN
014721              1=:X.ISTATE; T.STATUS BONE 5WAIT=:X.STATUS
014727              CALL STDEV
014730           ELSE
014731              IF X.TYPRING NBIT 5NORESERV OR CFREE><0 THEN CALL STDEV FI
014737           FI; *PIOF
014740           GO INWDX
014741   @ELIB
014741
014741
014741   % RESTART AFTER TRANSFER, X=DATAFIELD; ENTRY FROM DRIVER LEVEL
014741   IORESTART: X=:B:=RTRES
014743   FIORES:IF ISTATE>=0 THEN CALL RDATA ELSE IF X><0 THEN CALL RTENTRY FI FI
014751           GO STUPR
014752   RBUS
014756   @LIB CXCPU
014756
014756   %=========================================================================
014756   %              C X R T A C T   C X X R T A C T
014756   %
014756   % RTACT FOR TERMINALS
014756   %
014756   % ENTRY:      B=ADDR. OF TERMINAL DATAFIELD OUTSIDE RESIDENT
014756   %             A=MONITOR ROUTINE ADDR.
014756   % CXXRTACT:   MUST BE CALLED IN IOF
014756   %
014756   SUBR CXRTACT,CXXRTACT
014756   INTEGER CJMP:=124001,CION(0); *ION
014760   CXXRTACT: X:=CJMP; GO FELL
014762   CXRTACT: *IOF
014763           X:=CION
014764   FELL:   X=:D:=TDRADDR
014766           IF T:=X.MLINK=0 THEN
014771              A=:X.MFUNC; MQUEUE=:X.MLINK; A:=X=:MQUEUE
014776              MLEV; *MST PID
015000           FI; *EXR SD; EXIT
015002   RBUS
015003
015003   @ELIB
015003   @LIB CXCPU-,
015003   %=========================================================================
015003   % 11.9          I O S E T
015003
015003   % MONITOR CALLS TO SET CONTROL INFORMATION FOR A DEVICE
015003   % APPL. LEVEL
015003   %CALL IOSET(DEV,OUTP,PROG,VALUE)
015003   SUBR IOSET
015003   IOSET: CALL GET4; D2; CALL XRTCHECK; X=:D2; IF D0/\177700=100 GO ERR
015014           D0; CALL LOGPH; IF =0 OR T:=D1><0 THEN D=:A FI
015023           IF A><0 AND A.TYPRING BIT 5ISET AND D2=X.RTRES AND BACKG=0 OR RTREF=T THEN
```

```
015041   @LIB CXCPU
015041              IF X.TYPRING BIT 5TERM THEN
015044                   *IOF
015045                   T:="KSETDV"; CALL XGTDFADDR; A=:L; L:=:P; GO L1
015052              FI
015052   @ELIB
015052                   D3; X:=:B; *IOF
015055                   D:=0; CALL SETDV; X:=:B
015060   L1:             *ION
015061                   A BZERO 17=:ZAREG
015063           ELSE
015064   ERR:            -1=:ZAREG
015066           FI; GO RET
015067   RBUS
015100
015100   %==========================================================================
015100   % 11.10          C I B U F   C O B U F
015100
015100   %BACKGROUND MONITOR CALLS TO CLEAR BUFFERS
015100   %APPL. LEVEL; T=DEV.NO, SKIP IF OK
015100   SUBR CIBUF,COBUF
015100   CIBUF: CALL GETO
015101          CALL GZTREG; GO ERR
015103          A=:X; GO CIO
015105   COBUF: CALL GETO
015106          CALL GZTREG; GO ERR
015110          D=:X
015111   CIO:   IF X=0 OR X.TYPRING NBIT 5ISET THEN
015115             2=:ZAREG
015117          ELSE
015120             IF X.RTRES><RTREF THEN
015124                5=:ZAREG
015126             ELSE
015127                IF X.TYPRING NBIT 5BAD AND A BIT M144B OR BIT 5COM THEN T:=-2 ELSE T:=-1 FI
015141   @LIB CXCPU
015141                IF A BIT 5TERM THEN
015143                   T:="KSETDV"; CALL XGTDFADDR; A=:L; *IOF
015147                   L:=:P; GO L1
015151                FI
015151   @ELIB
015151               X:=:B; *IOF
015153               A:=T; CALL SETDV
015155               X:=:B
015156   L1:         *ION
015157               MIN ZPREG; 0/\0
015161             FI
015161          FI; GO RET
015162   ERR:   2=:ZAREG; GO RET
015165   RBUS
015173
015173   %==========================================================================
015173   % 11.11          I S I Z E   O S I Z E
015173
015173   %MONITOR CALLS TO READ BUFFER CONTENTS
015173   %APPL.LEVEL; T=DEV.NO, SKIP IF OK
015173   SUBR ISIZE,OSIZE,PISIZ,BOSIZ,T1P01,T1P02
015173   ISIZE: CALL GETO
015174          CALL GZTREG; GO ERR
015176          A=:X
015177          IF X><0 THEN
```

```
015200              IF X.TYPRING BIT 5IOBT AND NBIT 5COM AND NBIT 5BAD THEN
015207                  T:="BHOLD"
015210   INISIZE:       CALL XGTDFADDR; GO OK
015212              ELSE
015213                 IF X.TYPRING BIT 5COM THEN              % NORD-NET CHANNEL
015216                    X.BYTS-; GO OK
015221                 FI
015221                 IF X.TYPRING BIT 5BAD THEN              % BAD
015224   T1P01:           GO PISIZ; 0/\0
015226                 FI
015226                 GO ERR
015227              FI
015227           FI; GO ERR
015230   OSIZE: CALL GETO
015231           CALL GZTREG; GO ERR
015233           D=:X
015234           IF X><0 THEN
015235              IF X.TYPRING BIT 5IOBT AND NBIT 5COM AND NBIT 5BAD THEN
015244                 T:="CFREE"; GO INISIZE
015246              ELSE
015247                 IF X.TYPRING BIT 5COM THEN           % NORD-NET CHANNEL
015252                      MSTOR-BYTS; GO OK
015255                 FI
015255                 IF X.TYPRING BIT 5BAD THEN           % BAD
015260   T1P02:             GO BOSIZ; 0/\0
015262                 FI
015262              FI
015262           FI
015262   ERR:    240=:ZAREG; GO RET    % ILLEGAL DEVICE TYPE
015265   OK:     A=:ZAREG; MIN ZPREG; 0/\0; GO RET
015271   RBUS
015303
015303   %=========================================================================
015303   %       G Z T R E G
015303   %
015303   % GET LOGICAL DEVICE NUMBER
015303   %
015303   % ENTRY:      LOG.DEVICE PARAMETER IN ZTREG
015303   %
015303   % EXIT:       ERROR
015303   % EXIT+1:     A=POINTER TO INPUT DATAFIELD
015303   %              D=POINTER TO OUTPUT DATAFIELD
015303   %              T IS DESTROYED
015303   SUBR GZTREG
015303   INTEGER POINTER PTTNO:=TTNO,PBCHFLAG:=BCHFLAG
015305   GZTREG: IF ZTREG/\177700><100 THEN L+1 FI
015313           IF BACKGROUND><0 AND PBCHFLAG=0 AND ZTREG=1 THEN PTTNO ELSE ZTREG FI
015326           GO LOGPH
015327   RBUS
015332
015332
015332   %=========================================================================
015332   % 11.12        B R K M   E C H O M   F D A T A F I E L D
015332
015332   %MONITOR CALLS FOR ECHO AND BREAK SETTING
015332   %A=STRATEGY NUMBER; T=LOG. NO. IF RT-PROGRAM
015332   INTEGER ARRAY BRKTB=?,ECHTB=?
015332   SUBR BRKM,ECHOM,FDATAFIELD,T1P03
015332   DISP 0; INTEGER POINTER LREG=D0; PSID
015332   DISP 0; INTEGER CBCHFLAG=D0; PSID
```

```
015332   INTEGER POINTER TTIF:=TTIFIELD,PBCHFLAG:=BCHFLAG
015334
015334   %LOCAL SUBROUTINE TO FIND DATAFIELD, RETURNED IN X-REGISTER
015334   FDATAFIELD: L=:T; CALL GETO; T=:"LREG"
015337           IF BACKGROUND><0 THEN
015341              X:=TTIF; T:=PBCHFLAG
015343           ELSE
015344              IF ZTREG/\177700=100 GO RET
015351              ZTREG; CALL LOGPH; IF A=0 GO RET
015355   @LIB CXCPU
015355              IF A.TYPRING BIT 5TERM THEN
015361                 CALL XSETBFPAGE
015362              ELSE
015363                 IF A NBIT 5COM AND A NBIT 5BAD GO RET
015367              FI
015367   @ELIB
015367   @LIB CXCPU-,
015367              T:=0
015370           FI; GO LREG
015371   *)FILL
015377
015377   DISP 0; INTEGER DREG=D2; INTEGER POINTER LRG=D1; PSID
015377   FELLS: A+X=:D0:=L=:"LRG":=D=:DREG
015405           RTREF=:D; ZXREG; T:=10; CALL CHLIM; GO RET
015413           IF DEMAND><0 THEN
015415                   MLEV; *MST PIE
015417           FI; X:=ZXREG; OLDPAG=:D; CALL DALTON; *BSET ZRO
015424   @LIB CXCPU
015424           A:=10=:L:=ZXREG=:D
015430           X:=TDFBANK; T:=D0; *BSET ONE; MOVAN
015434   @ELIB
015434   @LIB CXCPU-,
015434           CALL ALTOFF
015435           A:=DREG=:D:=D0; GO LRG; *)FILL
015450   BRKM:   CALL FDATAFIELD; T=:CBCHFLAG
015452           IF ZAREG<0 THEN
015454                   A:=0=:X.BRKMAX
015456           ELSE
015457                   IF A>11 GO RET
015462                   IF A-2>0 THEN A:=ZDREG ELSE A:=0 FI
015470                   A=:X.BRKMAX; X=:D
015472                   IF CBCHFLAG=0 THEN 0=:X.NCBRK FI
015475                   IF ZAREG=11 THEN D:=1; GO TCOM FI % DO NOT CHANGE TABLE
015503                   IF A=7 THEN
015506                      IF ECBKF=0 OR CBCHFLAG=1 GO RET
015514                      "PBRK7"; CALL FELLS            % COPY BREAK TABLE FROM USER
015516                   ELSE
015517                      IF A=10 THEN
015522                         IF ECBKF=0 OR CBCHFLAG=1 GO RET
015530                         "PBRK7"+X                   % USE EXISTING USER BREAK TABLE
015532                      ELSE
015533                         BRKTB(A)                    % STANDARD TABLE 0-6
015535                      FI
015535                   FI; X:=D
015536           FI; A=:X.BRKTAB; D:=1; GO TCOM
015541   ECHOM:  CALL FDATAFIELD; T=:CBCHFLAG
015543           IF ZAREG<0 THEN
015545                   A:=0
015546           ELSE
015547                   IF A>7 GO RET
```

```
015552                  X=:D
015553                  IF A=7 THEN
015556                     IF ECBKF=0 OR CBCHFLAG=1 GO RET
015564                     "PECH7"; CALL FELLS
015566                  ELSE
015567                     ECHTB(A)
015571                  FI; X:=D
015572          FI; A=:X.ECHOTAB; 2=:D
015575   TCOM:  IF X.TYPRING BIT 5COM THEN ZAREG/\377=:T; CALL BRECC FI
015604          IF X.TYPRING BIT 5BAD THEN
015607   T1P03:    CALL BBREC; 0/\0
015611          FI
015611          GO RET
015612
015612   RBUS
015624
015624
015624   %=================================================================
015624   % 11.13         M G T T Y    M S T T Y
015624   %
015624   % MONITOR CALLS TO SET OR GET THE TERMINAL TYPE
015624   %
015624   % MGTTY:        ENTRY: T=LOGICAL DEV. NO
015624   %               EXIT:  A=TERMINAL TYPE
015624   %
015624   % MSTTY:        ENTRY; T=LOGICAL DEV. NO
015624   %                      A=TERMINAL TYPE
015624   %               FROM BACKGROUND MEANS T=0 YOUR OWN TERMINAL
015624   %
015624   %       ERROR EXIT: NORMAL RETURN WITH A:=ERRORCODE
015624   %
015624   SUBR MGTTY,MSTTY,T1P04
015624   INTEGER POINTER PTTNO:=TTNO,PPASS:=PASSTYPE
015626   MSTTY: K:=1; GO MSGTTY
015630   MGTTY: K:="0";
015631   MSGTTY: CALL GETO; A:=ZTREG
015633          IF T:=BACKGROUND><0 THEN
015636             IF A=0 OR A=1 OR A=PTTNO THEN
015645                PTTNO
015646             ELSE
015647                IF K AND T:=PPASS=0 GO E25
015654             FI
015654          FI; IF A=:D/\177700=100 GO ERRFI
015661          A:=D; CALL LOGPH; IF A=0 GO ERRNO
015664          IF A.TYPRING NBIT 5TERM AND A NBIT 5COM AND A NBIT 5BAD GO ERRDT
015674          T:="CTTYP"
015675          IF K THEN ZAREG; CALL XSTDFADDR ELSE CALL XGTDFADDR; A=:ZAREG FI
015704          IF X.TYPRING BIT 5BAD THEN   % BAD
015707   T1P04:    CALL BSTTY; 0/\0
015711   OUT:   FI; MIN ZPREG; 0/\0; GO RET
015714   ERRDT: 240; GO ERRF                    % ILLEGAL DEVICE TYPE
015716   ERRNO:  33; GO ERRF                    % NO SUCH LOGICAL UNIT
015720   ERRFI:   2; GO ERRF                    % BAD FILE NUMBER
015722   E25:    25
015723   ERRF:  A=:ZAREG; GO RET
015725
015725   RBUS
015737
015737
015737   %=====================================================================
```

```
015737   % 11.14          N O W A I T    T N O W A I
015737
015737   % MONITOR CALL: CALL NOWAIT(LOG.NO.,I/O-FLAG,NOWAITFLAG)
015737   % MONITOR CALL: CALL TNOWAI(LOG.NO.,I/O-FLAG,NOWAITFLAG)
015737   SUBR NOWAIT,TNOWAI,T1P05
015737   DISP 0; INTEGER WTYPE=D3; PSID
015737   NOWAIT: CALL GET3; A:=-1; GO FELLS
015742   TNOWAI: CALL GET3; A:=-2
015744   FELLS: A=:WTYPE
015745          IF D0/\177700=100 GO ERR
015752          D0; CALL LOGPH; IF =0 OR T:=D1><0 THEN D=:A FI
015761          IF ><0 AND A.TYPRING BIT 5ISET AND X.RTRES=RTREF THEN
015772             IF D2><0 THEN
015774                IF X.TYPRING BIT 5TERM THEN WTYPE ELSE -1 FI
016002             FI
016002             IF A><X.ISTATE THEN
016005                A=:X.ISTATE
016006                IF X.TYPRING BIT 5BAD AND D1=0 THEN
016013                   A:=X.ISTATE; *PIOF
016015   T1P05:          CALL NWSTA; 0/\0; *PION
016020                FI
016020             FI; A:=0
016021          ELSE
016022   ERR:      -1
016023          FI; A=:ZAREG; GO RET
016025   RBUS
016033
016033   %=========================================================================
016033   % 11.15          E X I O X
016033   %
016033
016033   % MONITOR CALL: I=EXIOX(VALUE,DEV.NO.)
016033   SUBR EXIOX
016033   DISP 0; INTEGER ADR=D2; PSID
016033   EXIOX: T:="P0"; CALL GET2; T=:ADR
016036          FOR X:="IOXTAB" TO "EIOXTAB" DO
016042             IF -1=X.S0 GO ERR
016046             IF D1=T THEN
016051                D0; *IOXT
016053                A=:T; IF DEMAND><0 THEN MLEV; *MST PIE
016060                FI; X:=ADR; CALL ALTON; T=:X.S0; CALL ALTOFF
016064                IF BACKGROUND><0 THEN CALL XBMRET FI
016067                A:=0; GO OKR
016071             FI
016071          OD
016073   ERR:   A:=-1
016074   OKR:   A=:ZAREG; GO RET
016076   RBUS
016107
016107   %=========================================================================
016107   % 11.16          L S T C
016107
016107
016107   % MONITOR CALL: I=LASTC(LOG. NO)
016107   %      RETURN WITH -1 MEANS WRONG UNIT
016107   SUBR LSTC
016107   INTEGER POINTER PPASST:=PASSTYPE
016110   LSTC: CALL GET1
016111          IF BACKGROUND><0 AND PPASST><2 GO ERR
016117          D0; CALL LOGPH
```

```
016121          IF A=0 OR A.TYPRING NBIT 5IOBT OR A BIT 5RFILE GO ERR
016130          IF A BIT 5TERM OR A BIT 5COM OR A BIT 5BAD THEN
016136             T:="LAST"; CALL XGTDFADDR; GO OUT
016141          FI
016141          IF A BIT 5IBDV THEN A:=X.CHARI; GO OUT FI
016145   ERR:   A:=-1
016146   OUT:   A=:ZAREG; GO RET
016150   RBUS
016156
016156   %==========================================================================
016156   % 11.17          S E T P A R I T Y
016156
016156   % SUBROUTINE TO SET PARITY
016156   SUBR SETPARITY
016156   SETPARITY: *BSET ZRO 70 DA; BSKP ZRO 60 DA; BSET BCM 70 DA
016161                *BSKP ZRO 50 DA; BSET BCM 70 DA; BSKP ZRO 40 DA; BSET BCM 70 DA
016165                *BSKP ZRO 30 DA; BSET BCM 70 DA; BSKP ZRO 20 DA; BSET BCM 70 DA
016171                *BSKP ZRO 10 DA; BSET BCM 70 DA; BSKP ZRO 0 DA; BSET BCM 70 DA
016175                EXIT
016176   RBUS
016176
```

```
016176
016176   %==========================================================================
016176   %              I N T E R N A L   D E V I C E S
016176
016176
016176   %==========================================================================
016176   % 11.18       I P T C H   I G T C H   I S T D V O S T D V
016176   %             P U T W   G E T W   I N I O S E T
016176
016176   SUBR IPTCH,IGTCH,ISTDV,OSTDV,PUTW,GETW,INIOSET,RBGET,RWGET
016176   INTEGER POINTER LREG
016177   IPTCH: T:=L=:"LREG":=DFOPP=:B; IF T:=CFREE><0 THEN
016206                 T:=377/\A=:CHARI; CALL RBPUT; MIN "LREG"
016213          FI; GO OUT
016214
016214   PUTW:  T:=L=:"LREG":=DFOPP=:B; IF T:=CFREE><0 THEN
016223                 A=:CHARI; CALL RWPUT; MIN "LREG"
016226          FI
016226   OUT:   T:=DFOPP=:B; GO LREG
016231
016231   IGTCH: IF BHOLD><0 THEN L+1; GO RBGET FI; EXIT
016236   GETW:  IF BHOLD><0 THEN L+1; GO RWGET FI; EXIT
016243
016243   ISTDV: A:=L=:"LREG"
016245          IF BHOLD=0 THEN DFOPP=:B; CALL XRTACT; DFOPP=:B FI
016254          GO LREG
016255
016255   OSTDV: A:=L=:"LREG"; DFOPP=:B
016261          IF BHOLD>=IMAXBHOLD OR CHARI=TERM THEN
016271                IF ISTATE = -1 OR = -2 THEN CALL NW5ST; GO NST; GO OSRET FI
016303   NST:         CALL XRTACT
016304          FI
016304   OSRET: DFOPP=:B; GO LREG
016307
016307   % SETDV-ROUTINE FOR OUTPUT:
016307   INIOSET: A:=L=:"LREG"; DFOPP=:B; CALL XRTACT; DFOPP=:B; A:=0; GO LREG
016320   RBUS
016327
016327
016327   @LIB CXCPU-,
016327   INTEGER ARRAY BRKTB:=(BRK0,BRK1,BRK2,BRK3,BRK4,BRK5,BRK6,-1)
016337   @LIB CXCPU-,
016337   INTEGER ARRAY ECHTB:=(ECH0,ECH1,ECH2,ECH3,ECH4,ECH5,ECH6,-1)
016347
016347
016347   %==========================================================================
016347   % 11.20        T T I M R
016347
016347   % TIMEOUT SUBROUTINE,CALLED FROM TIMER RT-PROG.
016347   % USED FOR KEEPING THE TERMINAL ACTIVATED.
016347   % THE ROUTINE ALSO CHECK IF THE TERMINAL LINE IS OK AND TAKES
016347   % APPROPRIATE ACTION.
016347
016347   SUBR TTIMR,TETTO
016347
016347   TTIMR: IF TYPRING BIT 5TERM THEN                    % TERMINAL
016352             IF DFLAG BIT 5LBRK THEN                   % LINE DEAD FOR DURATION OF TIMEOUT
016355                T:=HDEV+DST; *IOXT                     % CHECK IF LINE HAS RECOVERED
016360                IF A NBIT 13 THEN                      % IS CARRIER MISSING?
016362                   FLAGB BZERO 5LSTA=:FLAGB            % LINE HAS RECOVERED, SET STATUS OK
```

12 10X WDATA

```
016365                DFLAG BZERO 5LBRK=:DFLAG                          %
016370              ELSE                                                % LINE IS DOWN, WE NOW CONSIDDER IT LOST
016371                T:=FLAGB BONE 5LSTA=:FLAGB                        % SET TERMINAL LINE STATUS AS DEAD
016374                IF BSTATE><0 THEN                                 % BACKGROUND
016376                   IF T BIT 5LBLOG THEN                           % LINE TERMINATION ENABLED
016400                     IF T NBIT 5LOGOUT THEN                       % NOT ALREADY IN LOGOUT
016402                       T BONE 5LOGOUT=:FLAGB                      % SET LOGOUT BIT
016404                       IF T NBIT 5TLREP THEN                      % TERMINAL LINE REPPORT NOT ENABLED
016406                         IF BSTATE=5BCOM THEN
016412                           IF FLAGB NBIT 5ESCON THEN
016415                             A BONE 5ESC2SET=:FLAGB; GO TETTO
016420                           FI
016420                           A BZERO 5ESC2SET=:FLAGB  .
016422                         ELSE
016423                           IF A=5BUSER THEN MIN BSTATE FI
016427                         FI
016427                         "MESCAPE"; GO RSTPR
016431                       ELSE
016432                         IF IN5MSG><0 THEN "IBMOVE" ELSE "IORESTART" FI; GO RSTPR
016440                       FI
016440                     FI
016440                   ELSE
016441                      IF IN5MSG><0 THEN "IBMOVE" ELSE "IORESTART" FI; GO RSTPR
016447                   FI
016447                ELSE
016450                  IF RTRES=0 THEN FLAGB BZERO 5TLREP=:FLAGB FI
016455              FI; FI
016455            ELSE
016456              FLAGB BZERO 5LSTA=:FLAGB
016461            FI
016461          FI
016461   TETTO: TTMR=:TMR; CNTREG; T:=HDEV+"DCONT"; *IOXT
016467          IF DFLAG BIT 5FIMO THEN
016472             T:=HDEV+4; *IOXT
016475          FI; EXIT
016476
016476   @LIB CXCPU
016476   INTEGER CLR
016477   RSTPR: A:=:L=:CLR:=L; CALL CXXRTACT; CLR=:L
016505   @ELIB
016505   @LIB CXCPU-,
016505   GO TETTO
016506   RBUS
016514
```

```
016514
016514   %==========================================================================
016514   %              T E R M I N A L   O U T P U T
016514   %==========================================================================
016514   % 11.22       T T P U T   T R T P U T   D M O U T P U T   T T O M R   I O N I O F
016514   %
016514
016514   SUBR CXRBPUT
016514   RBUS
016514
016514   SUBR TTPUT,TRTPUT,DMOUTPUT,TTOMR,PDMOUTPUT,DIOUT,IONIOF,MLTTOMR
016514
016514   %IOTRANS SUBROUTINE, CALLED FROM OUTBT
016514   TTPUT: IF T:=CFREE=0 THEN EXIT FI
016520          L+1; GO RBPUT
016522
016522   % TERMINAL IOTRANS ROUTINE
016522   TRTPUT: IF T:=CFREE=0 THEN EXIT FI          % BUFFER FULL
016526           IF T:=MINBHOLD<0 THEN EXIT FI       % BUFFER LOCKED
016532           L+1; GO CXRBPUT
016534
016534   @LIB CXCPU-,
016534   @LIB CXCPU
016534   INTEGER POINTER PXRTACT:=CXXRTACT
016535   @ELIB
016535   %STDEV SUBROUTINE, CALLED FROM OUTBT, TO ACTIVATE LEVEL 10
016535   INTEGER SVT,SVL
016537   DIOUT: T=:SVT; GO IDIOUT
016541   TTOMR: T=:SVT; GO ITTOMR
016543   DMOUTPUT: T=:SVT
016544           T:=HDEV+DST; *IOXT
016547           IF NBIT 0 THEN
016551   % TERMINAL OUTPUT TIME-OUT SUBROUTINE
016551   ITTOMR:
016551   @LIB CXCPU
016551              IF DFOPP><0 AND X:=TYPRING BIT 5TERM THEN
016556                 X:=1777; X/\A; A:=B/\176000; X+A; IF X.DFLAG NBIT 5LBRK GO L1
016566   @ELIB
016566   @LIB CXCPU-,
016566                 IF CFREE=MAX THEN          % OUTPUT BUFFER EMPTY
016572                    MINBHOLD BZERO 5BLOC=:MINBHOLD
016575                 FI
016575                 IF X.DBPROG><0 THEN
016577                    X.DFLAG BZERO 5LBRK=:X.DFLAG; T:=HDEV+DST-4; *IOXT
016606                    IF A BIT 13 THEN           % MISSING CARRIER
016610                       A:=L=:SVL; CALL CLBUF; "IORESTART"; CALL PXRTACT; SVL=:L
016617                    FI
016617              FI FI
016617   @LIB CXCPU
016617   L1:        IF TYPRING BIT 5TERM THEN TDRADDR; GO L2 FI
016624   @ELIB
016624
016624   %  DIGITAL I/O TIMEOUT ENTRY POINT  (DR11C)
016624   IDIOUT:    B=:A
016625   L2:        *IRW LV10B DB
016626              "STDRIV"; *IRW LV10B DT
016630              "SLV10"; *IRW LV10B DP
016632              LV10; *MST PID
016634           FI
016634           T:=SVT; TTMR=:TMR
```

```
016637          EXIT
016640
016640
016640   INTEGER LREG
016641   PDMOUTPUT: *PON; IOF
016643          A:=L=:LREG; CALL DMOUTPUT; LREG=:L; *ION; POF; EXIT
016653
016653   %
016653   %  SUBROUTINE TO DO ION; IOF WITH PAGING ON
016653   %
016653   %  CALLED FROM (ROUTINES IN POF): MBINTERM,........
016653
016653   IONIOF: *PION; PIOF; EXIT
016656
016656   % SPECIAL TIME-OUT SUBROUTINE FOR TERMINALS TAKING OUTPUT
016656   % FROM THE MAIL SYSTEM
016656
016656   MLTTOMR: IF CURMAIL=X:="F1205".RTRES THEN
016663                L=:D; CALL CLBUFF; X.STATUS BZERO 5WAIT=:X.STATUS; L:=D
016671            ELSE
016672                "TTOMR"=:"TMSUB"; -10=:TTMR
016676            FI; EXIT
016677
016677   RBUS
016713
016713
016713   %==========================================================================
016713   % 11.23          C L B U F   M C L R   C E X I T   T E X I T T E X I A
016713
016713   % SUBROUTINES TO CLEAR DEVICE, AND DUMMY SUBROUTINES
016713   SUBR CLBUF,MCLR,CEXIT,TEXIT,TEXIA
016713   MCLR:  T:=HDEV+DCONT; 20; *EXR ST
016717   CLBUF: 0=:BHOLD=:HENTE=:FYLLE; MAX=:CFREE
016724   CEXIT: A:=0
016725   TEXIT: EXIT
016726   TEXIA: EXITA
016727   RBUS
016727
016727   %==========================================================================
016727   % 11.23B      C T R T I N   C T R T O U
016727
016727   % SETDV SUBROUTINES FOR TERMINAL INPUT OUTPUT
016727   SUBR CTRTIN,CTRTOU
016727   @LIB CXCPU
016727   INTEGER SVX
016730   CTRTIN: X=:SVX                                          % X=RESIDENT DATAFIELD
016731          X:=X.TDFLGADDR=:D; T:=TDFBANK
016734          X:="DFLAG"+D; *LDATX
016737          A BZERO 5ECHO; *STATX
016741          X:="RSISTE"+D; A:=-1; *STATX
016745          GO L1
016746   @ELIB
016746   @LIB CXCPU-,
016746   @LIB CXCPU
016746   CTRTOU: X=:SVX:=X.TDFLGADDR=:D; T:=TDFBANK
016752          X:="MINBHOLD"+D; *LDATX
016755          A BZERO 5BLOC; *STATX
016757   L1:    X:="BHOLD"+D; *STZTX
016762          X:="HENTE"+D; *STZTX
016765          X:="FYLLE"+D; *STZTX
```

```
016770          X:="DERROR"+D; *STZTX
016773          X:="MAX"+D; *LDATX
016776          X:="CFREE"+D; *STATX
017001          X:=SVX; A:=0; EXIT
017004  @ELIB
017004  RBUS
017016
017016  %==========================================================================
017016  %              T A P E    R E A D E R
017016  %==========================================================================
017016  % 11.24         T R G E T   R S T D E V   D T A P T I M   B S T D E V
017016  %
017016
017016  SUBR TRGET,RSTDEV,DTAPTIM,BSTDEV
017016
017016  % IOTRANS SUBROUTINE TO READ A CHAR., CALLED FROM INBT
017016  TRGET: IF BHOLD=0 THEN EXIT FI
017021         L+1; GO RBGET
017023
017023  % STDEV SUBROUTINE TO ACTIVATE LEVEL 12
017023  RSTDEV: IF BHOLD<MINBHOLD THEN
017027  BSTDEV:         B=:A; *IRW LV12B DB
017031                  "STDRIV"; *IRW LV12B DT
017033                  "SLV12"; *IRW LV12B DP
017035                  LV12; *MST PID
017037         FI
017037         EXIT
017040
017040  %TIMEOUT SUBROUTINE
017040  DTAPTIM: IF ISTATE><0 THEN
017042                  IF BHOLD=0 THEN 12=:DERROR ELSE TTMR=:TMR FI
017051         FI
017051         20; T:=HDEV+DCONT; *EXR ST        %DEVICE CLEAR
017055         IF X:=RTRES><0 GO XRTACT
017060         EXIT
017061  RBUS
017065
017065
```

```
017065
017065   %==========================================================================
017065   %              C O N N E C T - D E V I C E S
017065   %==========================================================================
017065   % 11.25          C O N C T
017065   %
017065
017065   % MONITOR CALL: CALL CONCT(PROG,DEV.NO.)
017065   %APPL. LEVEL
017065
017065   SUBR CONCT
017065   CONCT: CALL GET2; CALL RTCHECK
017067          IF D1/\177700=100 GO ERR
017074          D1; CALL LOGPH; IF =0 GO ERR
017077          A:=:X; IF T:=X.TYPRING NBIT 5CONCT GO ERR
017103          A=:X.DCNRT
017104          GO RET
017105   ERR:   CALL 9ERRA(#27); GO RETXIT
017110   RBUS
017117
017117   %==========================================================================
017117   % 11.26          D S C N T
017117
017117   %MONITOR CALL: CALL DSCNT(PROG)
017117   SUBR DSCNT
017117   DISP 0; INTEGER COUNT=D2; INTEGER RTSAV=D3; PSID
017117   DSCNT: CALL GET1; CALL RTCHECK
017121          X.STATUS BZERO 5INT BZERO 5REP=:X.STATUS
017125          CALL FTIMQU; X=:RTSAV; 0=:COUNT
017130          FOR COUNT DO
017130                  IF X:=COUNT>24 GO OUT
017134                  IF X><1 THEN X:=CNVRT(X)=:D        % NOT FILE
017141                  *PIOF
017142                  X.S0+A=:L                          % # ENTRIES IN TABLE *2->L
017145                  *PION
017146                  FOR T:=1 STEP 2 TO L DO
017151                     *PIOF
017152                     X:=D+T; X:=X.S0                 % PICK UP DATAFIELD ADDRESS
017155                     *PION
017156                     IF X><0 AND X.TYPRING BIT 5CONCT AND X.DCNRT-RTSAV=0 THEN
017165                                 0=:X.DCNRT
017166                     FI
017166                  OD
017170                  FI
017170          OD
017172   OUT:   GO RET
017173   RBUS
017200
017200   %==========================================================================
017200   % 11.27          D D R I V E R
017200
017200   %DRIVER ON LEVEL 12
017200   SUBR DDRIVER
017200   DDRIVER: *TRA STS
017201          A SH 4 SHZ -14; A-12; A SH 1=:X
017206          CALL RTACT; P+X
017210          CALL ID10; GO DDRIVER
017212          CALL ID11; GO DDRIVER
017214          CALL ID12; GO DDRIVER
017216          CALL ID13; GO DDRIVER
```

```
017220   RBUS
017226
017226
017226   %==========================================================================
017226   % 11.28        D M O N I T O R
017226
017226   %MONITOR LEVEL, START RT-PROG.
017226   SUBR DMONITOR
017226   DMONITOR: IF X:=X.DCNRT><0 THEN CALL RTENTRY FI
017231           GO STUPR
017232   RBUS
017234
017234
017234   %==========================================================================
017234   % 11.29        P I C K F P A R   P I C K L P A R
017234   %
017234   % SUBROUTINE TO FETCH PARAMETERS FROM ABSTR PARAMETER LIST
017234   % CALLED FROM HIGH INTERRUPT LEVEL
017234   %
017234   SUBR PICKFPAR,PICKLPAR,PICKXLPAR,PICKYLPAR
017234   DISP 0; DOUBLE POINTER DP1=P1,DP2=P2,DP3=P3; PSID
017234   PICKFPAR: *PONN
017235           T:=P0; AD:=DP1; *POF
017240           EXIT
017241   PICKLPAR: *PONN
017242           T:=P3; "P3"=:D; P2; *POF
017247           EXIT
017250   PICKXLPAR: *PONN
017251           T := P2; AD := DP3; *POF
017254           EXIT
017255   PICKYLPAR: *PONN
017256           T:=P3; AD:=DP2; *POF
017261           EXIT
017262   RBUS
017262
017262   %=========================================================================
017262   % 11.29.1    P S T M R E T    P S T D R E T
017262   %
017262   %       SUBROUTINE TO RETURN PARAMETERS FROM HIGH LEVELS
017262   %       TO ABSTRANS PARAMETER LIST
017262
017262   SUBR PSTMRET, PSTDRET
017262
017262   DISP 0
017262           DOUBLE POINTER ZMRET=MRETURN
017262   PSID
017262
017262   PSTMRET:
017262           *PONN
017263           A=:MRETURN; *POF
017265           EXIT
017266
017266   PSTDRET:
017266           *PONN
017267           AD=:ZMRET; *POF
017271           EXIT
017272
017272   RBUS
017272
017272   %=========================================================================
```

```
017272   % 11.30      M T E R M O D E
017272   %
017272   %   MONITOR CALL TERMINAL-MODE:  CALL MTERM(IDEV,ICODE)
017272   %
017272   %   ENTRY:   IDEV        = LOGICAL DEVICE NUMBER
017272   %              ICODE BIT 0 = CAPITAL LETTERS
017272   %                    BIT 1 = DELAY AFTER CR
017272   %                    BIT 2 = STOP ON FULL PAGE
017272   %                    BIT 3 = LOGOUT ON MISSING CARRIER
017272   %
017272   %   RETURN:  IF OK A:=0 ELSE A:=ERRORCODE.
017272   %
017272   %
017272
017272   SUBR MTERMODE,T1P06
017272   INTEGER POINTER PTTNO:=TTNO, PBCHFLAG:=BCHFLAG
017274   INTEGER BSAV,LPAR1
017276
017276   MTERMODE: CALL GET2; IF D0/\177700=100 GO FAR ERRFI
017304           A:=D1=:LPAR1; IF <0 OR >17 THEN GO ERRF FI   % LEGAL FUNCTION CODE?
017313           IF BACKGROUND><0 AND PBCHFLAG=0 AND D0=1 THEN PTTNO ELSE D0 FI
017326           CALL LOGPH; IF A=0 OR D=0 THEN GO ERRP FI
017333           A=:X:=B=:BSAV; D=:B                        % B=OUTPUT DATAFIELD, X=INPUT DATAFIELD
017337           IF X.TYPRING BIT 5COM THEN BSAV=:B; GO ERRR FI  % IF REMOTE, USE LOCAL MODE
017345
017345   %  CAPITAL LETTERS?
017345           T:="DFLAG"; CALL XGTDFADDR
017347           A BZERO 5CAPITAL; IF T:=LPAR1 BIT 0 THEN A BONE 5CAPITAL FI
017354           T:="DFLAG"; CALL XSTDFADDR
017356           GO L1; *)FILL
017367
017367   %  CR DELAY?
017367   L1:     X:=:B; T:="TYPRING"; CALL XGTDFADDR; A BZERO 5CRDLY
017373           IF T:=LPAR1 BIT 1 THEN A BONE 5CRDLY FI
017377           A=:X.TYPRING; T:="TYPRING"; CALL XSTDFADDR
017402
017402   %  STOP ON FULL PAGE?
017402           A:=0; IF T:=LPAR1 BIT 2 THEN A+1 FI
017407           T:="SCREEN"; CALL XSTDFADDR; X:=:B
017412
017412   %  LOGOUT ON MISSING CARRIER?
017412           T:="FLAGB"; CALL XGTDFADDR; A BZERO 5LBLOG
017415           IF T:=LPAR1 BIT 3 THEN A BONE 5LBLOG FI
017421           T:="FLAGB"; CALL XSTDFADDR
017423           IF X.TYPRING BIT 5BAD THEN
017426              A:=LPAR1
017427   T1P06:     CALL BTMOD; 0/\0
017431           FI
017431           BSAV=:B; 0=:ZAREG; GO RET
017435
017435   %  ERROR RETURN
017435   ERRR:  A:=-1; GO ERET                    % USE LOCAL MODE
017437   ERRF:  A:=201; GO ERET                   % ILLEGAL FUNCTION CODE
017441   ERRFI: A:=2; GO ERET                     % BAD FILE NUMBER
017443   ERRP:  A:=174                            % ILLEGAL PARAMETER
017444   ERET:  A=:ZAREG; GO RET
017446   RBUS
017456
017456   %==================================================================
017456   % 11.30.2     G T M O D
```

```
017456   %
017456   % MONITOR CALL TO GET TERMINAL MODE
017456   %
017456   % ENTRY:        T = LOG. NO IF RT-PROGRAM
017456   % SKIP-RETURN:  A = TERMINAL-MODE
017456   % RETURN:       A = ERROR-CODE
017456   SUBR GTMOD
017456   INTEGER FTMOD,BSAV
017460   GTMOD: CALL GETO; CALL GZTREG; 0/\0; X:=B=:BSAV; D=:B
017466          IF A=0 THEN 33; GO ERR FI
017471          IF A.TYPRING NBIT 5TERM AND A NBIT 5BAD THEN
017477             A:=240; GO ERR                                  % ILLEGAL DEVICE TYPE
017501          FI
017501          0=:FTMOD
017502          T:="DFLAG"; CALL XGTDFADDR
017504          IF A BIT 5CAPITAL THEN FTMOD BONE "0"=:FTMOD FI    % CAPITAL LETTERS?
017511          T:="FLAGB"; CALL XGTDFADDR
017513          IF A BIT 5LBLOG THEN FTMOD BONE 3=:FTMOD FI        % LOGOUT ON MISSING CARRIER?
017520          B=:X; T:="SCREEN"; CALL XGTDFADDR
017523          IF A><0 THEN FTMOD BONE 2=:FTMOD FI                % STOP ON FULL PAGE?
017527          T:="TYPRING"; CALL XGTDFADDR
017531          IF A BIT 5CRDLY THEN FTMOD BONE 1=:FTMOD FI        % CR DELAY?
017536          BSAV=:B; FTMOD=:ZAREG; MIN ZPREG; 0/\0; GO RET
017545
017545   ERR:   X:=BSAV=:B; A=:ZAREG; GO RET
017551   RBUS
017562
017562   %==========================================================================
017562   % 11.31        M C D E S C F U    M C E E S C F U
017562   %
017562   %      MONITORCALL DISABLE-/ENABLE- ESCAPE FUNCTION
017562   %
017562   %      ENTRY: T = LOGICAL DEVICE NUMBER IF RT-PROGRAM
017562   %
017562
017562   SUBR MCDESCFU,MCEESCFU,T1P07
017562
017562   MCDESCFU: K:=1; GO FELLS
017564   MCEESCFU: K:="0"
017565   FELLS: CALL FDATAFIELD
017566          IF K THEN
017570                X.DFLAG BONE 5IESC
017572          ELSE
017573                X.DFLAG BZERO 5IESC
017575          FI; A=:X.DFLAG
017576          IF X.TYPRING BIT 6BAD THEN
017601   T1P07:    CALL BCESC; 0/\0
017603          FI
017603          GO RET
017604   RBUS
017607
017607   %==========================================================================
017607   %  11.32        M S D A E      M G D A E
017607   %
017607   %      MONITORCALL TO SET-/GET DISCONNECT CHARACTER AND ESCAPE CHARACTER
017607   %
017607   %      ENTRY: T = LOGICAL DEVICE NUMBER IF RT-PROGRAM
017607   %
017607
017607   SUBR MSDAE,MGDAE,T1P08
```

```
017607
017607   MSDAE: K:=1; GO FELLS
017611   MGDAE: K:="0"
017612   FELLS: CALL FDATAFIELD
017613          IF K THEN            % SET
017615             ZAREG=:X.CESCP
017617             IF X.TYPRING BIT 5BAD THEN
017622   T1P08:       CALL BSDAE; 0/\0
017624             FI
017624          ELSE
017625             X.CESCP=:ZAREG
017627          FI; GO RET
017630   RBUS
017633
017633   %=============================================================================
017633   % 11.33        U S C N T    S Y C N T
017633
017633   % USCNT: USER-CONTROLL   MONITOR CALL (MON 260)
017633   % SYCNT: SYSTEM-CONTROLL MONITOR CALL (MON 261)
017633   %        ZTREG=DEV, ZAREG=CONTROLL NUMBER
017633   %        SKIP IF OK
017633   SUBR USCNT,SYCNT
017633   INTEGER POINTER PTTNO:=TTNO,PBCHFLAG:=BCHFLAG
017635   USCNT: K:=1; GO FELLS
017637   SYCNT: K:="0"
017640   FELLS: CALL GETO
017641          CALL GZTREG; GO E240
017643          D=:X; IF X=0 OR X.TYPRING NBIT 5BAD GO E240
017650          IF A NBIT 5NORESRV THEN
017652             IF X.RTRES><RTREF THEN 5; GO ERR FI
017660          FI
017660          ZAREG=:D; IF K THEN 24 ELSE 23; MIN ZPREG; 0/\0 FI; X:=:B; *IOF
017673          CALL SETDV; *ION
017675          X:=:B; GO RET
017677   E240:  240                  % ILLEGAL DEVICE TYPE
017700   ERR:   A=:ZAREG; GO RET
017702   RBUS
```

```
017707
017707   %=================================================================
017707   %  11.34      G D E V T Y
017707   %
017707   SUBR GDEVTY,GDEV3TY
017707
017707   GDEVTY: CALL GETO; MLEV; *MST PIE
017712           CALL SPT3PIT; GO GDEV3TY
017714   RBUS
017717
017717   %=============================================================================
017717   %
017717   %  11.36      T R E P P
017717   %
017717   %=============================================================================
017717
017717   % RESIDENT PART OF TREPP MONITOR CALL
017717
017717   SUBR TREPP,T3REPP
017717   TREPP: CALL GETO; CALL SPT3PIT; GO T3REPP
017722   RBUS
017725
017725   %=================================================================================
017725   %              G E T X M
017725   %
017725   % RESIDENT PART OF MONCALL GET-ERROR-TEXT (MON 334)
017725   %
017725   SUBR GETXM,3GETXT
017725   GETXM: CALL GETO; CALL SPT3PIT; GO 3GETXT
017730   RBUS
017733
017733   %=============================================================================
017733   %              E X A B S
017733   %
017733   % RESIDENT PART OF MONCALL EXECUTE-ABSTR (MON 335)
017733   %
017733   SUBR EXABS,3EXABS
017733   EXABS: CALL SPT3PIT; GO 3EXABS
017735   RBUS
```

```
017737
017737   %=============================================================================
017737   %
017737   %  12.0      E R R O R   H A N D L I N G
017737   %
017737   %=============================================================================
017737
017737
017737   %=============================================================================
017737
017737   % 12.2          9 E R R   9 E R R A
017737
017737   % ERROR RECORDING ROUTINE, ANY LEVEL
017737   % 9ERRA CALLED FROM MONITOR CALLS, B POINTS TO WORKING AREA
017737
017737   INTEGER 9EXREG,9ETREG,9EAREG,9EDREG; TRIPLE 9ESSAVTAD=9ETREG
017743
017743   SUBR 9ERR,9ERRA
017743
017743   INTEGER COUT=?
017743   INTEGER LREG
017744
017744   9ERRA: T:=ZPREG; *IOF
017746          TAD=:9ESSAVTAD; A=:D:=T; GO ERRX
017752   9ERR:  *IOF
017753          TAD=:9ESSAVTAD; A=:D; *IRR ALEVB DP
017756   ERRX:  X=:9EXREG:=L=:LREG:=X.S0; A=:L; *TRA STS; POF
017765          IF A BIT 16 THEN "PON" ELSE "POF" FI; A=:COUT
017773          A:=L; CALL NW9ERR                                  % IN POF-AREA
017775          A:=LREG=:L; X:=9EXREG; TAD:=9ESSAVTAD
020001   INTEGER COUT; *ION
020003          EXITA
020004   RBUS
020011
020011
020011   @DEV 1
020011   @DEV (S-S-J)SINB-1
```

```
020011
020011   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% S I N B - 1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
020011
020011   %==========================================================================
020011   %
020011   %    13.0        B A C K G R O U N D   P R O C E S S O R
020011   %
020011   %                M E M O R Y   R E S I D E N T   P A R T
020011   %
020011   %==========================================================================
020011
020011   INTEGER ARRAY ACTAB=?          % RT ACCOUNTING TABLE
020011   INTEGER ARRAY BACKTAB=?        % BACKGROUND TABLE
020011   INTEGER ARRAY TSLSTATUS=?      % TIME SLICE STATUS TABLE
020011   INTEGER BNUMB=?                % BACKGROUND TABLE SIZE
020011   INTEGER TDVN=?                 % OUTPUT DEVICE (IN FILE SYSTEM)
020011
020011   SUBR XS5ESCF
020011   RBUS
020011
020011   %==========================================================================
020011   %             E S C A P E   H A N D L I N G
020011   % 13.3       E S C A P E   R E S C A P
020011   %
020011
020011   % SUBROUTINE CALLED FROM INPUT TT DRIVER, TAD-DRIVER AND STOP-TERM
020011   % TO DETECT LOCAL-CHARACTER, ESCAPE-CHARACTER OR -1 LOGGOUT
020011   SUBR ESCAPE,RESCAP,STESCAPE,ACTESC
020011   @LIB CXCPU
020011   % CALLED FROM XSTOPTERM ON OPCOM SEGMENT
020011   INTEGER POINTER CLL; DOUBLE DCPIT
020014   INTEGER POINTER IPTO:=177000+5UBFPAGE+5UBFPAGE
020015   DOUBLE POINTER DPTO=IPTO
020015   STESCAPE: *POF
020016           A:=L=:"CLL"
020020           AD:=DPTO=:DCPIT
020022           A:=TDFPHPAGE=:D:=162000; AD=:DPTO
020026           TDFLGADDR/\1777+"5UBFPAGE*2000"=:B; *PON
020033           CALL ESCAPE; 0/\0; *POF
020036           DCPIT=:DPTO; *PON
020041           GO CLL
020042   *)FILL
020046   @ELIB
020046
020046   % LOCAL ROUTINE TO ACTIVATE ESCAPE MONITOR FUNCTIONS
020046   % FUNCTION IN A-REGISTER
020046   ACTESC:
020046   @LIB CXCPU
020046           IF T:=TYPRING BIT 5TERM GO CXRTACT
020051   @ELIB
020051           A=:MFUNC; GO RTACT
020053           EXIT
020054
020054   ESCAPE:
020054   @LIB CXCPU
020054   % B=DATAFIELD (OUTSIDE RESIDENT IF TERMINAL)
020054           IF TYPRING BIT 5TERM THEN TDRADDR.RTRES ELSE RTRES FI
020063           IF A><0 AND A><DBPROG THEN EXITA FI
020070   @ELIB
020070   @LIB CXCPU-,
```

```
020070   % ----  ENTRY FROM NORD-NET I
020070   RESCAP: IF DFLAG BIT 5IESC THEN EXITA FI
020074          IF LAST<0 THEN
020076             FLAGB BONE 5LOGOUT BZERO 5USESC BZERO 5LCHAR=:FLAGB
020103             GO ESCA
020104          FI
020104
020104          IF CESCP SHZ-10=LAST THEN        % L O C A L - C H A R A C T E R
020111             IF FLAGB NBIT 5LCHAR GO CHRUB          % LOCAL FUNCTION OFF
020114             IF A BIT 5ESCLOF THEN A BONE 5WLOC=:FLAGB; EXIT FI
020121             IF BSTATE=5BCOMM THEN
020125                IF FLAGB NBIT 5ESCON THEN T:=K5LOC2SET; GO XS5ESCF FI
020132                A BZERO 5ESCON=:FLAGB
020134             FI
020134             "LOCACT"; GO ACTESC                    % ACTIVATE MONITOR FUNCTION
020136          FI
020136          IF CESCP/\377=LAST THEN          % E S C A P E - C H A R A C T E R
020143             IF FLAGB BIT 5ESCLOC THEN A BONE 5WESC=:FLAGB; EXIT FI
020151   ESCA:     IF BSTATE=5BCOMM THEN
020155                IF FLAGB NBIT 5ESCON THEN T:=K5ESC2SET; GO XS5ESCF FI
020162                A BZERO 5ESCON=:FLAGB
020164             FI
020164             IF BSTATE=5BUSER THEN MIN BSTATE FI
020171             IF FLAGB BIT 5USESC THEN
020174                5BUSER=:BSTATE                  % USER ESCAPE
020176                "USESCAP"; GO ACTESC            % ACTIVATE MONITOR FUNCTION
020200             ELSE                               % NORMAL ESCAPE
020201   BACT:        "MESCAPE"; GO ACTESC            % ACTIVATE MONITOR FUNCTION
020203             FI
020203          FI
020203
020203   %-----------------------------------------------------------------
020203   %       C H E C K   I F   R U B O U T   I N   N O R D - N E T
020203
020203   CHRUB: IF LAST=177 AND BSTATE=5REMOT THEN MIN BSTATE; GO BACT FI
020215          IF BSTATE><0 THEN EXITA FI
020220          EXIT
020221   RBUS
020232
020232   %=================================================================
020232   % 13.4        M E S C A P E   L O C A C T   U S E S C A P
020232   %             B B R T W T   B R T W T   X B R T W T   U S R T W T
020232   %
020232   % ROUTINES ON MONITOR LEVEL TO PERFORM ESCAPE AND EXIT MONITOR CALLS
020232   % X=DATAFIELD
020232   SUBR MESCAPE,USESCAP,LOCACT,BBRTWT,BRTWT,XBRTWT,USRTWT,YBRTWT,T1P12
020232   INTEGER SYSUSF=?,ESCADR=?,CDBPROG=?
020232   INTEGER POINTER PESCADR:=ESCADR,PSYSUF:=SYSUSF ,PCDBPROG:=CDBPROG
020235
020235   % ACTIVATED FROM "ESCAPE"
020235   LOCACT:  T:="LUSADD"; CALL XGTDFADDR; A=:PESCADR          % LOCAL-FUNCTION
020240            1=:PSYSUSF; X=:B; GO FELLS
020244   USESCAP: T:="EUSADD"; CALL XGTDFADDR; A=:PESCADR          % USER ESCAPE
020247            1=:PSYSUSF; 0=:PCDBPROG; GO ERESP
020253   MESCAPE: "ESCOPCOM"=:ESCADR; 0=:SYSUSF=:CDBPROG           % NORMAL ESCAPE
020257            CALL MBABPROC; GO ERESP                          % ALLOCATE BACKGROUND PROCESS
020261            X=:CDBPROG:=B
020263
020263   ERESP: X=:B
020264   T1P12: CALL CBERSP; 0/\0                          % SEND ESCAPE RESPONSE IF TAD
```

ok set Flagb bit 4 and Escape priority → EXIT

72456

```
020266          IF CDBPROG><0 THEN X:=A; GO LABL1 FI       % DYNAMIC ALLOCATED BACKGR. PROG.
020272   FELLS: T:="DBPROG"; CALL XGTDFADDR
020274          IF A=0 GO MONEN; A=:CDBPROG
020277          IF A=CURPROG THEN
020302             CALL ESC500; IF SYSUSF=0 THEN CALL ESCDBUG FI; X:=B; GO STCUR
020310          FI; IF A.ACTPRI BIT 5RTOF GO MONEN
020314          X+5BRESLINK
020315          DO WHILE X:=X.RESLINK<RTSTART OR X>=SEGSTART
020324                IF X.ISTATE>0 AND X.TYPRING BIT 5IOBT OR BIT M144B THEN
020334                  IF X.TYPRING BIT 5COM THEN X.DFLAG BONE 5RQI=:X.DFLAG FI
020342                  0=:X.ISTAT; CDBPROG.STATUS BZERO 5WAIT=:X.STATUS; GO OUT
020350                FI
020350          OD
020351   OUT:   X:=CDBPROG; CALL MRLCLFIE            % RELEASE SWAPPING-SEMAPHORE
020353          CALL ESC500; IF SYSUSF=0 THEN CALL ESCDBUG FI
020357          X:=CDBPROG.RTDLGADDR; *POF
020362          X.DPREG; X:=B; T:="DBADR"; CALL XSTDFADDR
020366          IF SYSUSF=0 THEN ESCADR ELSE ESCADR=:CDBPROG.RTDLGADDR.DXREG; "GACTPR" FI
020377          A=:CDBPROG.RTDLGADDR.DPREG; X:=CDBPROG; *PON
020404          IF RTRES=0 THEN X.STATUS BZERO 5WAIT=:X.STATUS FI
020411   LABL1: "100002+ALEVB"=:X.ACTPRI; CALL RTENTRY; CALL FRWQU; CALL TOEXQU
020416          CALL S5ESCF; X:=CDBPROG
020420          GO ACTI; *)FILL
020447
020447   INTEGER SYSUSF,ESCADR,CDBPROG
020452
020452   % MON 135, RTWT:
020452   BBRTWT: IF CURPROG.STATUS BIT 11 GO RTWT
020456   % MON 134, RTEXT:
020456   BRTWT: T:="BRTWT"; CALL 1RL5PDEC        % TERMINATE ND-500 PROC AND RELEASE ND-500 PROC-DESC.
020460          0=:SYSUSF; "XJABORT"; GO XBR
020463   %  ERROR OR MON 0
020463   YBRTWT: T:="YBRTWT"; CALL 1RL5PDEC      % TERMINATE ND-500 PROC AND RELEASE ND-500 PROC-DESC.
020465           0=:SYSUSF; "XEROPCOM"; GO XBR
020470   % USER ESCAPE OR LOCAL FUNCTION:
020470   USRTWT:T:=1=:SYSUSF; GO XBR
020473   % ESCAPE:
020473   XBRTWT: "ESCOPCOM"; 0=:SYSUSF
020475   XBR:    A=:ESCADR; CALL GETDATAFIELD; X=:B
020500   STCUR: *IRR ALEVB DP
020501          T:="DBADR"; CALL XSTDFADDR
020503          IF SYSUSF=0 THEN
020505             ESCADR; *IRW ALEVB DP
020507          ELSE
020510             "GACTPR"; *IRW ALEVB DP
020512              ESCADR; *IRW ALEVB DX
020514          FI
020514          "100002+ALEVB"=:CURPROG.ACTPRI/\3773; *TRR PCR
020521   ACTI:  IF SYSUSF=0 THEN
020523             0=:ISTATE
020524          ELSE
020525             IF ISTATE>0 THEN 0=:ISTATE FI
020531          FI
020531          X.SEGM; T:=X.STATUS BONE 11
020534   ACTU:  A=:X.ACTSEG; T=:X.STATUS
020536          GO STUPR
020537
020537   INTEGER POINTER PSTPNT:=STPNT           % POINTER TO SUBROUTINE STACK POINTER
020540   GACTPR: CALL XBMRET; "STBEG"=:PSTPNT; -1=:L
020545          T:=BACTPRI; "ACTEFUN"; *IRW MLEVB DP
```

```
020550          MLEV; *MST PID; ION
020553          CALL ERRFATAL
020554
020554   ACTEFUN: *IRR ALEVB DT
020555          A=:CURPROG.ACTPRI/\3773; *TRR PCR
020561          *IRR ALEVB DX; IRW ALEVB DP
020563          X.SEGM SHZ -10+1=:T; X.SEGM/\177400\/T; T:=X.STATUS BZERO 11
020574          GO ACTU
020575   RBUS
020622
020622   %==========================================================================
020622   % 13.5        M O C O M
020622   %
020622
020622   % MONITOR LEVEL ROUTINE FOR MONITOR CALL CMND
020622   SUBR MOCOM
020622   @LIB CXCPU
020622   INTEGER ARRAY CC5BREGBLOCK(11)
020633   INTEGER CDFADDR
020634   @ELIB
020634   MOCOM: CALL GETDATAFIELD
020635   @LIB CXCPU-,
020635   @LIB CXCPU
020635          X=:CDFADDR; CSTART; T:="DBACTPRI"; CALL XSTDFADDR
020641          "CC5BREGBLOCK-5REG"; CALL SWAPPREG
020643          T:="5BREGBLOCK"; X:=CDFADDR; CALL FXTADDR
020646          A:=11=:L:="CC5BREGBLOCK"=:D; *MOVNP
020653   @ELIB
020653          "MENTRY"; *IRW ALEVB DP
020655          GO MONEN
020656   RBUS
020670
020670   %==========================================================================
020670   % 13.6        B G E R R
020670
020670   % SUBROUTINE CALLED FROM 9ERR ROUTINE, ANY LEVEL
020670   SUBR BGERR
020670   INTEGER CDFADDR
020671   INTEGER POINTER LREG
020672   DISP 0; DOUBLE DD0,DD2; PSID
020672   BGERR: X=:B:=L=:"LREG"; CALL GETDATAFIELD
020676   @LIB CXCPU-,
020676   @LIB CXCPU
020676          X=:CDFADDR; T:="DERO"; CALL FXTADDR
020701          A:=4=:L:=0; D:=B; *MOVPP
020706          T:="BSTATE"; 5ERROR; X:=CDFADDR; CALL XSTDFADDR
020712   @ELIB
020712          "YBRTWT"; *IRW MLEVB DP
020714          MLEV; *MST PID; MST PIE
020717          GO LREG
020720   RBUS
020726
020726   %==========================================================================
020726   % 13.7        E S C O N   E S C O F F
020726
020726   %ESCAPE ON AND OFF SUBROUTINES; ALLOW ESCAPE IN COMMAND MODE
020726   SUBR ESCON,ESCOFF,T1P13
020726   INTEGER BREG,XREG,LREG; REAL TADREG
020734   INTEGER CCFLAGB
020735   ESCON: *IOF
```

```
020736          TAD=:TADREG; A:=B=:BREG; X=:XREG:=L=:LREG; CALL GETDATAFIELD
020745          T:="FLAGB"; CALL XGTDFADDR
020747          IF A=:CCFLAGB BIT 5LCHAR AND A BIT 5LOC2SET THEN
020754             T:="LUSADD"; CALL XGTDFADDR; *IRW MLEVB DA
020757             "USRTWT"; *IRW MLEVB DP
020761             MLEV; *MST PID
020763             GO ESON
020764          FI
020764          IF A BIT 5ESC2SET THEN
020766   T1P13:    CALL CXRESP; O/\O
020770             CCFLAGB
020771             CALL ESC5ON; GO OUT
020773             IF CCFLAGB BIT 5USESC THEN
020776                T:="EUSADD"; CALL XGTDFADDR; *IRW MLEVB DA
021001                "USRTWT"
021002             ELSE
021003                "XBRTWT"
021004             FI
021004             *IRW MLEVB DP
021005             MLEV; *MST PID
021007          FI
021007   ESON:  CCFLAGB BONE 5ESCON; GO OUT
021012
021012   ESCOFF: *IOF
021013          TAD=:TADREG; A:=B=:BREG; X=:XREG:=L=:LREG; CALL GETDATAFIELD
021022          T:="FLAGB"; CALL XGTDFADDR
021024          A BZERO 5ESCON; GO OFOUT
021026   OUT:   A BZERO 5ESC2SET BZERO 5LOC2SET
021030   OFOUT:
021030          T:="FLAGB"; CALL XSTDFADDR
021032          X:=XREG; LREG=:L; BREG=:B; TADREG; *ION; EXIT
021042   RBUS
021054
021054   %=========================================================================
021054   % 13.8        G G L O C   P P L O C
021054
021054   %ROUTINES TO GET/PUT WORDS IN SEGMENTS,CALLED FROM GETLOC/PUTLOC
021054   %X=ADR, A=VALUE, D=PAGE TABLE NO., SKIP RETURN IF OK
021054   SUBR GGLOC,PPLOC
021054   INTEGER XREG,VALUE
021056   GGLOC: *PIOF
021057          X=:XREG; A:=D SH 7\/RTREF.ACTPRI=:X.ACTPRI/\3773; *TRR PCR; BSET ONE 0
021070          X:=XREG; *PION
021072          X.SO; GO GPRET
021074
021074   PPLOC: *PIOF
021075          X=:XREG; A=:VALUE:=D SH 7\/RTREF.ACTPRI=:X.ACTPRI/\3773; *TRR PCR;BSET ONE 0
021107          X:=XREG; VALUE; *PION
021112          A=:X.SO
021113   GPRET: *PIOF
021114          A=:D; X=:XREG
021116          T=:RTREF.ACTSEG; X.ACTPRI BZERO 7 BZERO 10=:X.ACTPRI/\3773; *TRR PCR
021126          X=:XREG; "STUPR"; *IRW MLEVB DP
021131          MLEV; *MST PID; PION
021134          D=:A; EXIT
021136   RBUS
021141
021141   %=========================================================================
021141   % 13.9        S T P I O F   S T D P I O F   L D P I O F   L D D P I O F
021141
```

```
021141   SUBR STPIOF,LDPIOF,GETPTABLE,STDPIOF,LDDPIOF
021141   STPIOF: *PIOF
021142           A=:X.S0; *PION
021144           EXIT
021145   STDPIOF: *PIOF
021146           AD=:X.DS0; *PION
021150           EXIT
021151   GETPTABLE:
021151   LDPIOF: *PIOF
021152           A:=X.S0; *PION
021154           EXIT
021155   LDDPIOF: *PIOF
021156           AD:=X.DS0; *PION
021160           EXIT
021161   RBUS
021161
021161
021161   SUBR RCOM
021161   RBUS
021161
021161
021161   %==========================================================================
021161   %       2 B D U M P  -  2 B R E C O V E R  -  2 B S R E C O V E R
021161   %       2 B D B R E C O V E R
021161   %
021161   % SUBROUTINES USED BY THE RECOVER,PLACE-PROGRAM AND DUMP COMMANDS
021161   %
021161   % ENTRY:      T=FILE NUMBER
021161   %             X=ADDRESS OF "PROG-FILE" DESCRIPTOR BLOCK
021161   %
021161   INTEGER 2BSEG=?,2BXREX=?,2BMCALL=?,2BNWRD=?,2BLREG=?,2BBLCKNO=?,2BALTON=?
021161   INTEGER 2BFILNO=?,2BMEMAD=?,2B64K=?,2BSSTART=?,2BOLDSEG=?,BSTART=?,BCSEGM=?
021161   INTEGER ARRAY 2BRWPAR=?
021161
021161   SUBR 2BDUMP,2BRECOVER,2BSRECOVER,2BDBRECOVER
021161
021161   DISP 0
021161   INTEGER 2BSTART                % START ADDR
021161   INTEGER 2BRSTART               % RESTART (CONTINUE) ADDR.
021161   INTEGER 2BLPAD                 % LOWER PROGRAM BANK ADDR.
021161   INTEGER 2BUPAD                 % UPPER PROGRAM BANK ADDR.
021161   INTEGER 2BLDAD                 % LOWER DATA BANK ADDR.
021161   INTEGER 2BUDAD                 % UPPER DATA BANK ADDR.
021161   INTEGER D2BXAD                 % LAST WORD IN PROG-FILE HEADER.
021161   DOUBLE DD2PMEMAD=2BLPAD,DD2DMEMAD=2BLDAD
021161   PSID
021161
021161   DOMCALL:
021161           T=:2BBLCKNO; A=:2BMEMAD
021163           IF A=0 AND D=-1 THEN 100000=:2B64K ELSE D-A+1=:A; 0=:2B64K FI
021176           A=:2BNWRD; 2BALTON=:D; A:=L; CALL DALTON; A=:L; *BSET ZRO
021205           A:=2BMCALL=:D:="2BRWPAR"; T:=2BFILNO; *EXR SD
021212           IF A><0 THEN EXIT FI
021214           IF 2B64K><0 THEN
021216              2BBLCKNO+200=:2BBLCKNO; 100000=:2BMEMAD
021223              2BALTON=:D; A:=L; CALL DALTON; A=:L; *BSET ZRO
021231              A:=2BMCALL=:D:="2BRWPAR"; *EXR SD
021235              IF A><0 THEN EXIT FI
021237           FI; EXITA
021240    *)FILL
```

```
021253
021253   INTEGER MON117(0); *MON 117
021254   INTEGER MON120(0); *MON 120
021255
021255   2BDUMP: 3=:2BSSTART; MON120=:2BMCALL; GO FELLS
021262   2BDBRECOVER: L=:D; CALL SPTOPIT; L:=D; A:=2; GO FLREC
021267   2BSRECOVER: A:=0; GO FLREC
021271   2BRECOVER: A:=1
021272   FLREC: A=:2BSSTART:=MON117
021274   FELLS: A=:2BMCALL:=L=:2BLREG; T=:2BFILNO; X=:2BXREG
021301          T:=BCSEGM=:T; CALL MMEXY; T=:2BOLDSEG
021305          BCSEGM/\377*5SEGSIZE+SEGSTART=:2BSEG
021312          A.LOGADR/\300 SH 1=:2BALTON
021317          T:=1; 2BXREG.DD2PMEMAD; IF A>>D GO E198
021324          CALL DOMCALL; GO ERR
021326          IF 2BSSTART><3 THEN KBACTPRI=:BACTPRI FI; 2BXREG.DD2DMEMAD
021336          IF A<<=D AND T:=X.D2BXADR=0 THEN
021343             IF 2BSEG.LOGADR SHZ -10><200 THEN
021351                ER184
021352   GWARN:       *MON 64
021353                GO MBSTART
021354             FI; X.LOGADR/\300+100 SH 1=:2BALTON
021361             T:=401; 2BXREG.DD2DMEMAD; CALL FAR DOMCALL; GO MBERR
021366             IF 2BSSTART><3 THEN KABACTPRI=:BACTPRI FI
021374          FI
021374   MBSTART: CALL ALTOFF
021375          IF 2BSSTART><2 THEN
021401             T:=2BFILNO; *MON 2CLOS; JMP ERR
021404          FI; MIN 2BLREG
021405          IF A=0 THEN BSTART; GO RCOM FI
021410   RETU:  IF A><2 THEN
021413             T:=2BOLDSEG; CALL MMEXY; A:=D
021416          FI; T:=2BLREG
021417          IF 2BSSTART-2=0 THEN CALL S3NOALTPIT FI
021423          T=:P
021424
021424   E198:  ER198
021425   MBERR: CALL ALTOFF
021426          IF A=22 THEN ER185; GO GWARN FI
021433   ERR:   CALL ALTOFF; A=:D; GO RETU
021436
021436   RBUS
021472
021472
021472   %=========================================================================
021472   % 16.2        R E T B A C K   X R E T B A C K
021472
021472   INTEGER LACTPRI=?
021472
021472   % RETURN FROM MONITOR CALL PROCESSING, APPL. LEVEL
021472
021472   SUBR RETBACK,XRETBACK
021472
021472   INTEGER POINTER PTTIFIELD:=TTIFIELD
021473   XRETBACK: CALL XBMRET
021474             A:=D=:LACTPRI
021476   RETBACK: CALL XBMRET
021477             X:=PTTIFIELD; CALL CHDFPAGE; X.BSTATE; *IRW MLEVB DA
021503             "MRETBACK"; *IRW MLEVB DP
021505             MLEV; *MST PID; ION
```

```
021510
021510  %MONITOR LEVEL
021510  MRETBACK: IF A=5BUSER THEN
021513                LACTPRI; CALL MXSETUSER; BACTPRI=:LACTPRI
021517            ELSE
021520                LACTPRI; CALL XMXSETUSER
021522            FI; X:="MONBLOCK"; *LRB ALEVB
021524            GO MONEN
021525  RBUS
021536
021536
021536  %=========================================================================
021536  %              T O R T L O A D E R
021536  % SUBROUTINE TO CALL THE RT-LOADER
021536  % ENTRY: D=ENTRY POINT IN RT-LOADER
021536  %
021536  SUBR TORTLOADER
021536  INTEGER SVT,SVA
021540  INTEGER POINTER CSUBR(0)
021540  INTEGER SVD,SVX
021542  TRIPLE SVTAD=SVT
021542  INTEGER POINTER LREG
021543  TORTLOADER: TAD=:SVTAD; A:=L=:"LREG":=BCSEG/\177400+5RT2SG=:T; CALL MMEXY
021553              5RTSG; CALL MMREENT; 200=:D; CALL DALTON
021560              TAD:=SVTAD; CALL CSUBR; TAD=:SVTAD; X=:SVX
021564              CALL ALTOFF; D=:RTREF.RSEGM; T:="377\50PSEG"; CALL MMEXY
021571              TAD:=SVTAD; X:=SVX; GO LREG
021574  RBUS
021605
021605
021605  %=========================================================================
021605  % 16.9        M M E X I T   O M E X I T
021605
021605  SUBR MMEXIT,OMEXIT
021605  OMEXIT: *IOF
021606          RTREF.STATUS BZERO 10 BONE 11=:X.STATUS; *ION
021614  MMEXIT: *MON 2MEXI; JMP *
021616  RBUS
021617
021617  %=========================================================================
021617  % 16.5        S T U S E R
021617
021617  % START USER,CALLED FROM RECOVER, LOAD-B, ETC.,APPL. LEVEL
021617  SUBR STUSER
021617  INTEGER POINTER PTTIFIELD:=TTIFIELD
021620  STUSER: "0"; *MON 2BRKM; MON 2ECHO
021623          IF RTREF.ACTSEG><BCSEG THEN CALL MMEXY FI
021631          5BUSER=:PTTIFIELD.BSTATE; CALL XBMRET
021635          "MSTUSER"; *IRW MLEVB DP
021637          MLEV; *MST PID; ION
021642
021642  %MONITOR LEVEL:
021642  MSTUSER: CALL XSETUSER; X:="ESCBLOCK"; *LRB ALEVB
021645          GO MONEN
021646  RBUS
021656
021656
021656  %=========================================================================
021656  INTEGER ARRAY EGURSEGM(16)              % SEGMENT INFO FOR RTERR
021674  INTEGER ARRAY SVTXT:=(SVER0,SVER1,SVER2,SVER3,SVER4,SVER5) % SINTRAN VERSION
```

```
021702
021702   %=========================================================================
021702   % 15.14        1 I R E A D   1 D R E A D   1 R R E A D
021702   %               1 I W R I T E   1 D W R I T E   1 R W R I T E
021702   %
021702   %  ROUTINES TO READ/WRITE LOCATIONS IN POF-AREA
021702
021702   SUBR 1IREAD,1DREAD,1RREAD,1IWRITE,1DWRITE,1RWRITE
021702
021702   1IREAD:        *PIOF; LDA ,X; JMP L1
021705   1DREAD:        *PIOF; LDD ,X; JMP L1
021710   1RREAD:        *PIOF; LDF ,X; JMP L1
021713   1IWRITE:       *PIOF; STA ,X; JMP L1
021716   1DWRITE:       *PIOF; STD ,X; JMP L1
021721   1RWRITE:       *PIOF; STF ,X
021723   L1:            *PION; EXIT
021725
021725   RBUS
021725
021725   @LIB CXCPU-,
021725
021725   %=========================================================================
021725   % 16.24        B A B O R T
021725
021725   % ABORT CURRENT BATCH PROCESS
021725   SUBR BABORT
021725   BABORT:"YBFIELD"=:B; CALL 5BABORT; T:="377\50PSEG"; CALL OMEXIT; CALL BOBORT
021733   RBUS
021740
021740
021740
021740
021740   %=========================================================================
021740   %       PARAMETER LIST FOR RESERVE AND RELEASE OF ACCOUNTING SEMAPHORE
021740   %       PARAMETER LIST FOR ABSTR CALL TO SET
021740   %                        ALPHANUMERIC/PLOT MODE IN SPOOLING PROGRAM
021740   INTEGER ACCSEMRE:=("530","0","0")
021740   INTEGER SPLMOD:=("31","0","0","0")          % SET PLOT MODUS
021743   INTEGER SALMOD:=("30","0","0","0")          % SET ALPHANUMERIC MODE
021747   *)FILL
021753
021757   *CSSL=*
021757   @LIB CXCPU-,
021757
021757   *CSSL1=*
021757
```

```
021757
021757   *BGSYS=CSSL1+1777@-12@12
021757   *BGSYS/
022000   %==========================================================================
022000   %
022000   %   16.0        S Y S T E M   S E G M E N T
022000   %
022000   %==========================================================================
022000   % 16.1        S Y S T E M   S E G M E N T   D A T A   D E F I N I T I O N S
022000   %             D A T A   P A R T
022000
022000   BASE BFIELD
022000
022000   REAL FV0,FV3,FV6                  % WORKING SPACE FOR SUBROUTINES
022000   REAL FV11; INTEGER V12            % WORKING SPACE FOR GLPAR
022011   REAL SAVTAD; INTEGER INDEX        % USED BY ENTER AND LEAVE ROUTINES
022015   INTEGER V0=FV0,V3=FV3,V6=FV6
022021   INTEGER TTNO:=1                   % LOGICAL NUMBER FOR TERMINAL
022021   INTEGER TTIFIELD:=DT01R           % DATAFIELD FOR TERMINAL
022022   INTEGER CPNT                      % BYTE POINTER IN COMMAND STRING
022023   INTEGER OPNT                      % BYTE POINTER IN OLD STRING
022024   INTEGER CSTRING                   % START OF COMMAND STRING
022025   INTEGER OSTRING                   % START OF OLD STRING
022026   INTEGER PASSTYPE                  % 2=SYSTEM, 1=RT, 0=OTHERS
022027   INTEGER SPASTYPE                  % SAVED PASSTYPE
022030   INTEGER ARRAY ACTPAR(6)           % PARAMETERS FOR MONITOR CALL COMMANDS
022031   INTEGER ARRAY CPLIST:=(ACTPAR,ACTPAR+1,ACTPAR+2,ACTPAR+3,ACTPAR+4,ACTPAR+5)
022037   INTEGER TEXTADR                   % POINTS TO THE LOCATION AFTER LAST OUTP.STRING
022045   INTEGER GPSEGM                    % SEGMENT USED BY LOOK-AT
022046   INTEGER BCSEGM:=BSGM              % SYSTEM AND USER SEGMENTS
022047   INTEGER STPNT                     % SUBROUTINE STACK POINTER
022050   INTEGER BSTART                    % SUBSYSTEM START
022051   INTEGER BRESTART:=1               % SUBSYSTEM RESTART
022052   INTEGER LDADR                     % SUBSYSTEM FIRST LOCATION
022053   INTEGER HDADR                     % SUBSYSTEM LAST LOCATION
022054   INTEGER LDDAD                     % LOWER DATA BANK ADDR.
022055   INTEGER HDDAD                     % UPPER DATA BANK ADDR.
022056   INTEGER 2BXAD                     % WORD 6 IN A PROG.FILE
022057   INTEGER W789PRG(0);*0;0;0         % WORD 7,8 AND 9 IN PROG.FILE HEADER SHOULD BE ZERO
022060   INTEGER BCHFLAG:=0                % 0=TERM,1=BATCH,2=MODE,-1=IDLE BATCH
022063   INTEGER BINDV                     % INT. DEV. FOR BATCH QUEUE
022064   INTEGER FOBUF                     % POINTER FOR OPENING BATCH OUTPUT
022065   INTEGER FILNO                     % OPEN FILE NO. FOR RECOVER ETC.
022066
022067   INTEGER NULL
022070   INTEGER ARRAY P3RETU(0)           % PARAMETER FOR MCALL
022070   INTEGER S3RETU                    % RETURN ADDRESS ON SEGMENT 3
022071   INTEGER OPSEG:=377\5OPSEG         % COMMAND SEGMENT
022072   INTEGER RTSEG:=377\5RTSG          % RT-LOADER SEGMENT
022073   INTEGER FILSEG:=377\5FILSEG       % FILE SYSTEM SEGMENT
022074   INTEGER OP2SEG:=377\5OP2SEG       % SERVICE-PROG. AND MAIL SEGMENT
022075   INTEGER COMTAB:=3COMTAB           % COMMAND TABLE
022076   INTEGER RERNUM,RERPREG,RN1,RN2,RRTPROG  % ERROR INFORMATION
022103   INTEGER CURUSER                   % CURRENT USER INDEX
022104   INTEGER ARRAY POINTER BATAB:=BCHTAB   % BATCH TABLE
022105   DOUBLE SARDFILP                   % BATCH INPUT FILE POINTER BETWEEN JOBS
022107   DOUBLE TIMON(0); INTEGER 1TIMON,2TIMON   % LOG ON TIME
022111   DOUBLE TIOOF(0); INTEGER 1TIOOF,2TIOOF   % TERMINAL TIME
022113   INTEGER PRJN                      % PROJECT NUMBER; PROJECT NAME IS FOUND IN PRNAM (16.27)
022114   INTEGER REMLIN
022115   INTEGER CHNR
```

```
022116   INTEGER BATINT
022117   INTEGER KTRVF
022120   INTEGER MSTPN
022121   DOUBLE ARRAY POINTER MOSTK:=MODST
022122   INTEGER FLMAIL                  % FLAG FOR MAIL
022123   INTEGER SCRSIZE:=100            % DEFAULT NO. OF PAGES IN SCRATCH FILE AFTER CLOSE
022124   INTEGER ENTCT:=0                % NUMBER OF ATTEMTS TO ENTER
022125   INTEGER SSPAS=ENTCT             % USER ENTC TO SAVE THE PASSWORD TYPE FOR THE CHANGE USER CONTEXT MON CALL.
022125   INTEGER MAXCT:=12               % MAX. NUMBER OF ATTEMPTS (DEFAULT 10)
022126   INTEGER NAROUTSITCH             % USED IN "NAMSR" ROUTINES
022127   INTEGER POINTER FILERR:=3FILERR,ENTER:=3ENTER,LEAVE:=3LEAV,OUTTEXT:=3OUTTEXT
022133   INTEGER REESS                   % SAVED REENTRANT SEGMENT FOR NAMSR/INAMS
022134   INTEGER XXSAV                   % USED BY NAMSR/INAMS
022135   INTEGER ARRAY NAMSARR(0)        % SAVED REENT.BITMAP FOR NAMSR/INAMS
022135   INTEGER EMOD                    % EDITING MODE IN EDIT
022136   INTEGER POINTER EDSVL           % SAVED L-REG IN EDINBT
022137   INTEGER EDSVB                   % SAVED B-REG IN EDINBR
022140   INTEGER EDOSG                   % SAVED SEGMENTS IN EDINBT
022141   INTEGER NCOMPL                  % RETURN ADDRESS IN EDINBT WHEN NOWAIT RETURN
022142   INTEGER 5COBSTATE=XXSAV         % USED IN STSUPER TO MARK ND-500 USER BREAK
022142   INTEGER M4LRG                   % USED IN INBT/OUTBT FOR MODE/BATCH
022143   INTEGER CI4DFELT                % CURRENT DATAFIELD ADDR USED IN INBT/OUTBT
022144   INTEGER EDIRSEGM                % SAVED REENTRANT SEGMENT FOR INBT WITH T-REG=0 (MON EDIT)
022145   INTEGER XCSTRING=XXSAV          % CURRENT STRING USED IN 3CWRITE,3OWRITE ETC.
022145   *NAMSA+10/
022145   ESAB
022145
022145   *YBFIE=BFIEL
022145
022145   % SUBROUTINES
022145   @ICR
022145   SUBR CREAD,OREAD,OWRITE,
022145          CWRITE,BACK,SCAB,
022145          TEXTN,OCTU,DECU,
022145          CRLF,TCO2,TCO,TCI,ETCI,SRCHINT,
022145          LEAV2,L3EAVE,LEAV3;
022145   RBUS
022145   @CR;
022145
022145   DISP 0; INTEGER XTTNO=TTNO,XTTIFIELD=TTIFIELD,XBCHFLAG=BCHFLAG; PSID
022145
022145   INTEGER BMECHO                  % INPUT/OUTPUT CONTROL IN BATCH/MODE
022146   INTEGER COBSTATE                % SAVED OBSTATE
022147   INTEGER CMDFFIELD               % ADDRESS OF REGISTER BLOCK IN MON. CALL
022150   INTEGER PANAMSR                 % ADDRESS OF PARAMETER FOR NAMSR, INAMS
022151   INTEGER ARRAY ESCBLOCK(10)      % SAVED REGISTER BLOCK ON ESCAPE
022161   INTEGER ARRAY BGFIELD(5REG)     % WORKING AREA FOR MONITOR CALLS
022170   INTEGER ARRAY MONBLOCK(30)      % ITS REGISTER BLOCK
022220   INTEGER ARRAY DFS1(44)          % FOR FILE SYSTEM MONITOR CALLS
022264   INTEGER ARRAY DFS2(44)          % FOR FILE SYSTEM MONITOR CALLS
022330   INTEGER ARRAY DCOM(21)          % FOR COMND MON. CALL
022351   INTEGER ARRAY STBEG(12*STDELTA),STEND(0)  % SUBROUTINE STACK
022527          DATA(6415); *  *-1/
022527   INTEGER ARRAY COMSTRING(5WCBUFSIZE+1)     % COMMAND BUFFER
022614   INTEGER ARRAY FIBUF(5WCBUFSIZE+1)         % FILE NAME BUFFER
022701   INTEGER ARRAY WORKA(5WCBUFSIZE+1)         % WORKING AREA
022766   DOUBLE ARRAY DWORKA=WORKA
022766   INTEGER ARRAY TYPS:='SYMB',TYPD:='DATA'  % FOR OPEN-FILE
022774   INTEGER ARRAY SC100:='(SCRATCH)SCRATCH01',XSC100:='0' % SCRATCH FILE
023007   INTEGER BAUSER:='SYSTEM'
```

603507

769D

```
023013   INTEGER ARRAY STACK=?; INTEGER CSTCK=?            % FILE SYSTEM STACK
023013   INTEGER CUSER=?,USNO=?,USDI=?                     % IN FILE SYSTEM
023013   INTEGER ARRAY MODST(2*12); INTEGER MSTEN:=MSTEN-MODST      % MODE STACK
023040   INTEGER ARRAY USPAR(5)          % USER-PARAMETERS
023045   INTEGER FLBGTERM:=0             % FLAG TO INDICATE ENABLED/DISABLED
023046                                   % TERMINATION HANDLING
023046                                   % BIT 0=USER-BREAK, BIT 1=FATAL-ERROR
023046   INTEGER FLQERM:=0               % FLAG TO INDICATE MON 65 TERMINATION
023047   INTEGER FLLIPCOM:=0             % FLAG TO INDICATE ENABLED/DISABLED
023050                                   % AUTOMATIC START OF DEFAULT SUBSYSTEM
023050   INTEGER 6PASSW:=0               % PASSWORD FOR TERMINAL
023051   INTEGER MINUS:=-1               % FOR IOOPAR
023052   INTEGER LACTPRI:=KBACTPRI       % SAVED ACTPRI WHEN INBT FROM BATCH/MODE
023053   INTEGER MOFTYPE                 % "FILESYSTEM MONITOR CALL TYPE"
023054   INTEGER INLOGGED:=0             % FLAG INDICATING USER LOGGED IN OR NOT
023055   INTEGER UEFLG                   % USER ENVIRONMENT FLAGS (SEE BELOW)
023056   SYMBOL 5UEMO=0                  % MON 0 TRIGGERS UE TERMINATION HANDLING
023056   SYMBOL 5UETE=1                  % UE TERMINATION HANDLING ENABLED
023056   SYMBOL 5UEIE=2                  % INHIBIT ERROR MESSAGES FROM S III ROUTINES
023056   SYMBOL 5UETLOG=3                % TERMINATION HANDLINKG ENABLED FOR LOGOUT
023056   SYMBOL 5UERUN=4                 % USER ENVIRONMENT IS RUNNING
023056   SYMBOL 5UEPRUN=5                % USER ENVIRONMENT WILL SOON BE RUNNING
023056   SYMBOL 5UECM=6                  % THE MON.CALL CMD OR UELOGIN IS EXECUTING
023056   SYMBOL 5UEST=7                  % STOP-TERMINAL HAS BEEN EXECUTED
023056   SYMBOL 5UEES=10                 % ESCAPE HAS BEEN PRESSED
023056   SYMBOL 5UETM=11                 % BATCH JOB CPU TIMEOUT
023056   SYMBOL 5UELOGD=12               % USER WANTS TO LOG OUT
023056   SYMBOL 5UEAL=13                 % AUTOMATIC LOGIN AND START OF SUBSYSTEM
023056   INTEGER ARRAY UEDAT(30)         % SAVED INFORMATION FOR UE
023106   SYMBOL UEMAX=27                 % LAST WORD IN UEDAT
023106   INTEGER UEXREG;REAL UECMRET     % RETURN INFORMATION FOR NCMD AND UELOGIN
023112   INTEGER BACTPRI
023113   INTEGER TXAPPS(0); *#''
023114
023114   % VARIABLES USED IN THE RECOVER AND PLACE COMMANDS
023114
023114   INTEGER 2BFILNO                 % FILE NUMBER
023115   INTEGER 2B64K                   % 128KB PROGRAM OR DATA AREA
023116   INTEGER 2BBLCKNO                % BLOCK NUMBER IN 2BFILNO
023117   INTEGER 2BMCALL                 % MON RFILE OR MON WFILE
023120   INTEGER 2BLREG                  % L-REG
023121   INTEGER 2BXREG                  % X-REG.
023122   INTEGER 2BOLDSEG                % ORIGINAL SEGMENTS
023123   INTEGER 2BSEG                   % ADDR OF SEGMENT TABLE ENTRY
023124   INTEGER 2BSSTART                % "START-USER" INDICATOR
023125   INTEGER 2BNWRD                  % NUMBER OF WORDS TO TRANSFER
023126   INTEGER ARRAY 2BRWPAR:=(2BFILNO,NULL)
023130   INTEGER 2BMEMAD                 % MEMORY ADDRESS IN PARAMETER LISTE 2BRWPAR
023131   INTEGER ARRAY 22BRWPAR:=(2BBLCKNO,2BNWRD)         % REST OF 2BRWPAR
023133   INTEGER 2BALTON                 % CURRENT ALTERNATIVE PAGE TABLE FOR RECOVER ETC
023134   INTEGER 1XUSTATUS               % EXTRA "USER STATUS"
023135          SYMBOL 1ILLCONTINUE=0            % @CONTINUE AND @GOTO ILLEGAL
023135
023135   INTEGER ARRAY 3BDUREE(14)       % USED BY @DUMP-REENTRANT ETC....
023151                                   % FOR HOLDING NAME OF REENTRAN SUBSYSTEMS
023151
023151   SUBR OPCOM; RBUS       %MAKE GLOBAL ENTRYPOINT
023151
023151   %=========================================================
023151   %
```

```
023151   %                    M I S C .  R O U T I N E S
023151   %
023151   %=====================================================
023151
023151   %%%%%%%% ROUTINES WHICH MUST BE ON THE SAME PAGE AS THE BASE FIELD %%%%%%%%%%%%
023151   %=============================================================================
023151   % 16.3        X B M R E T
023151
023151   % SUBROUTINE TO GET THIS PAGE, CALLED FROM MON. CALL ENTRY, GETO
023151   SUBR XBMRET
023151   XBMRET: *IOF; EXIT
023153   RBUS
023153
023153   %=============================================================================
023153   % 16.4        C O M E N T R Y
023153
023153   % FILE SYSTEM MONITOR CALLS FROM BACKGROUND
023153   % ENTRY: B="DFS1 DFS2" DATAFIELDS; MONITOR LEVEL DISABLED
023153
023153   SUBR COMENTRY
023153   DISP 0; DOUBLE ZADREG=ZAREG; PSID
023153   INTEGER POINTER TTIF:=TTIFIELD,CRTR:=CRTREF
023155   DISP 3; INTEGER STRRSEGM,OBSTATE,OFLGB; PSID
023155   %RT LEVEL:
023155   COMENTRY: X:=RTREF=:SSREF; X.ACTSEG=:STRSEG; X.RSEGM=:STRRSEGM
023163           "100002+ALEVB"=:X.ACTPRI
023165           X.STATUS BONE 11=:X.STATUS
023170           X:=TTIF; CALL CHDFPAGE
023172           X.BSTATE=:OBSTATE=:COBSTATE
023175           5BCOMM=:X.BSTATE
023177           X.FLAGB=:OFLGB
023201           IF COBSTATE=5BUSER THEN
023205                 X:=B+5REG; *LRB BLEVB
023210                 X:="ESCBLOCK"; *SRB BLEVB
023212                 A:=B=:CMDFFIELD
023214           FI; MLEV; *MST PIE
023216           CALL ESCOFF; X:=RTREF=:CRTR
023221           IF MOFTYPE=1 THEN
023225                 T:="377\5OPSEG"; CALL MMEXY; GO OK; *)FILL
023242           ELSE IF A=0 THEN
023244                 IF STRSEG/\377=5FILSEG OR A=ECODSEG THEN
023254                    T:="377\5FILSEG"; "FRSG1"; CALL FSVCALLSEGS
023257                 ELSE
023260                    T:="377\5FILSEG"; "FRSG1"; CALL SVCALLSEGS
023263                 FI
023263           FI    FI
023263   OK:     X:=ZXREG; ZTADREG; CALL MRSTA; GO ERET; GO OK1; MIN ZPREG; 0/\0
023272   OK1:    MIN ZPREG; 0/\0
023274   ERET:   TAD=:ZTADREG; X=:ZXREG
023276           IF RTREF.ACTSEG><STRSEG OR X.RSEGM><STRRSEGM THEN
023307              IF X.RSEGM="FRSG1" OR A="FRSG2" THEN
023316                 IF STRSEG/\377=5FILSEG OR A=ECODSEG THEN CALL FRSCALLSEGS ELSE CALL RSCALLSEGS FI
023331              ELSE
023332                 T:=STRSEG; CALL MMEXY
023334              FI
023334           FI
023334           IF OBSTATE=5BUSER THEN CALL ESCON; CALL ESCOF
023342           ELSE IF OFLGB BIT 5ESCON THEN CALL ESCON FI
023347           FI
023347           OBSTATE=:TTIF.BSTATE; "MORET"; *IOF; IRW MLEVB DP
```

464 03
100110 11
115/03

241

```
023355          MLEV; *MST PID; ION
023360  %MONITOR LEVEL:
023360  MORET: *IRR ALEVB DB
023361          IF A.OLDPAG NBIT 1 THEN CALL MXSETUSER ELSE CALL XMXSETUSER FI
023370          X+7; *LRB ALEVB
023372          0=:COBSTATE=:CMDFFIELD
023374          GO MONEN
023375  RBUS
023415
023415  %==========================================================================
023415  % 16.7        M E N T R Y
023415  %
023415  % MONITOR CALL TO EXECUTE COMMAND STRING
023415  % A=POINTER TO COMMAND STRING
023415
023415  SUBR MENTRY
023415  INTEGER OPS:=(0,377\50PSEG),OBSTATE
023420  INTEGER POINTER TTIF:=TTIFIELD
023421  %RT LEVEL:
023421  MENTRY: X+5REG=:B:=TTIF; *IOF
023425          CALL CHDFPAGE; X:="DCOM":=:B; *LRB BLEVB
023431          X:=B+5REG; *SRB BLEVB
023434          IF TTIF.BSTATE=5BUSER THEN                  %%      SEQUENCES      %%
023441                X+"5BREGBLOCK-5REG"; X.OLDPAG=:OLDPAG  %%%%%%%%%%%%%%
023444                X:="ESCBLOCK"; *SRB BLEVB              %%%%%%%%%%%%%%%%%%%%%%%%%
023446                A:=B=:CMDFFIELD                        %%%%%%%%%%%%%%%%%%%%%%%%%
023450          FI; *ION                                     %%%%%%%%%%%%%%%%%%%%%%%%%
023451          "100002+ALEVB"=:RTREF.ACTPRI; X.STATUS BONE 11=:X.STATUS
023457          TTIF.BSTATE=:OBSTATE=:COBSTATE; 5BCOMM=:X.BSTATE; X.DBACTPRI=:OPS
023467          ZAREG; X:="BFIELD":=:B; T:="STBEG"=:STPNT:="OPS"; *MON 2MCAL
023476          CALL MMEXY; X=:B; OBSTATE=:TTIF.BSTATE; CALL ESCON; CALL ESCOFF
023505          "MCMRET"; *IOF; IRW MLEVB DP                 %%%%%%%%%%%%%%%%%%%%%%%%%
023510          MLEV; *MST PID; ION                          %%%%%%%%%%%%%%%%%%%%%%%%%
023513  %MONITOR LEVEL:
023513  MCMRET: X:=TTIF
023514          CALL CHDFPAGE
023515          IF X.BSTATE=5BUSER THEN
023521             "DCOM".OLDPAG; CALL MXSETUSER
023524          FI
023524          0=:COBSTATE=:CMDFFIELD
023526          X:="DCOM+5REG"; *LRB ALEVB
023530          GO MONEN
023531  RBUS
023552
023552  %==========================================================================
023552  % 16.8        T Q C O P Y   X Q C O P Y   R T D C O P Y
023552  % SUBROUTINES TO COPY TO THE WORK AREA WORKA (IN THE BASEFIELD)
023552  % CALLED FROM LIST COMMANDS
023552
023552  SUBR TQCOPY,XQCOPY,RTDCOPY
023552  INTEGER ARRAY POINTER RTADR
023553  TQCOPY: MLEV; *MCL PIE
023555          "BTIMQU"
023556          FOR D:=0 TO "5WORKSIZE-1" DO A.TLINK=:WORKA(D) WHILE ><-1; OD
023573          GO OUT
023574
023574  % COPY EXEC. OR WAITING QUEUE; X=DATAFIELD
023574  XQCOPY: MLEV; *MCL PIE
023576          X=:"RTADR"+"5BWLINK-5WLINK"=:A
023601          FOR D:=0 TO "5WORKSIZE-1" DO A.WLINK=:WORKA(D)
```

F B+11
X B+10

```
023611                  IF ="RTADR" THEN -1=:WORKA(X); GO OUT FI
023617          OD; GO OUT
023622
023622    % X=RT-DESCRIPTION
023622    RTDCOPY: MLEV; *MCL PIE
023624          X=:"RTADR"
023625          FOR X:=0 TO "5RTSIZE-1" DO RTADR(X)=:WORKA(X) OD
023635          "WORKA+5RTSIZE"=:B; X:="RTADR".RTDLGADDR; T:=0
023642          *LDDTX 00; STD 0,B; LDDTX 20; STD 2,B
023646          *LDDTX 40; STD 4,B; LDDTX 60; STD 6,B
023652          "BFIELD"=:B; 5RTSIZE+10=:D; "RTADR"+5BRESLINK   % GET RESOURCES
023661          FOR D TO "5WORKSIZE-1" DO A.RESLINK=:WORKA(D)
023670                  IF ="RTADR" THEN -1=:WORKA(X); GO OUT2 FI
023676          OD
023700    OUT2:  IF "WORKA".WLINK><0 THEN
023703                  DO WHILE X.WLINK>=RTSTART AND <SEGSTART; X:=X.WLINK OD
023714          FI; A=:X
023715    OUT:  MLEV; *MST PIE; EXIT
023720    RBUS
023726                                                                    %%%
023726    %%%
023726    %%%%%%% THE ROUTINES UP TO THIS POINT MUST BE ON THE SAME PAGE %%%%%%%%%%%%%%%%
023726                                                                    %%%
023726    %%%
```

```
023726
023726   %=====================================================================
023726   %
023726   %              E S C A P E   H A N D L I N G
023726   %
023726   % 16.10        E S C Q E R M   E S C O P C O M   E N T O P C O M
023726   %              X J A B O R T
023726   %
023726
023726   % PREPARATIONS BEFORE ENTERING THE COMMAND PROCESSOR, STSUPER
023726
023726   SUBR ESCQERM,ESCOPCOM,ENTOPCOM,XJABORT,XEROPCOM
023726
023726   INTEGER XREG,AREG,TREG,OPC:=(STSUPER,377\5OPSEG)
023733   INTEGER POINTER TTIF:=TTIFIELD,PTTNO:=TTNO,PNCOMPL:=NCOMPL
023736
023736   % ENTRY FROM MONITOR CALL QERM
023736   ESCQERM: X:="DFS2+5REG"; *LRB BLEVB
023740          X:="ESCBLOCK"; *SRB BLEVB
023742          5BUSER=:TTIF.BSTATE; 0=:X.ISTAT; 1=:FLQERM; GO ENTF1
023751
023751   % ENTRY FROM ESCAPE, ERROR OR MON 0:
023751   XEROPCOM:  0=:COBSTATE; GO FELLS
023753   ESCOPCOM: UEFLG BONE 5UEES=:UEFLG
023756   FELLS: X=:XREG; A=:AREG; T=:TREG
023761          IF COBSTATE=0 THEN
023763             % CLEAR ERROR INFO IF MON 0
023763             *STZ I (RERNU; STZ I (RERPR; STZ I (FLQER
023766          FI
023766          IF X:=CMDFFIELD><0 THEN 0=:CMDFFIELD; GO ENTF FI
023772          IF COBSTATE=0 AND TTIF.BSTATE=5BCOMM THEN GO ENTF FI
024002          X:="ESCBLOCK"; *SRB ALEVB
024004          AREG=:X.RAREG; TREG=:X.RTREG
024010          XREG=:X.RXREG; TTIF.DBADR=:"ESCBLOCK".RPREG
024016          GO ENTF
024017
024017   %INITIAL ENTRY:
024017   ENTOPCOM: KBACTPRI=:BACTPRI
024021   ENTF1:  0=:COBSTATE=:CMDFFIELD
024023   ENTF:   0=:PNCOMPL
024024          TTIF.FLAGB=:T BZERO 5FFGPAR=:X.FLAGB
024031          IF T BIT 5FFGPAR GO TORSCALLSEGS        % ESCAPE WHILE WAITING
024033                                                  % FOR INPUT CALLED FROM GLPAR
024033          IF RTREF.RSEGM><0 THEN
024036             IF A="FRSG1" OR A="FRSG2" THEN
024044   TORSCALLSEGS: CALL RSCALLSEGS
024045             ELSE IF A="5RTSG" THEN 0=:X.RSEGM
024052          FI; FI FI;
024052          T:="OPC"; *MON 2MCAL
024054
024054   % ABORT JOB IF BATCH(MON 134,135):
024054   XJABORT: X=:XREG:="ESCBLOCK"; *SRB ALEVB
024057          XREG=:X.RXREG; TTIF.DBADR=:"ESCBLOCK".RPREG; CALL JABORT
024066
024066   RBUS
024104
```

54130

152640

023736

023723

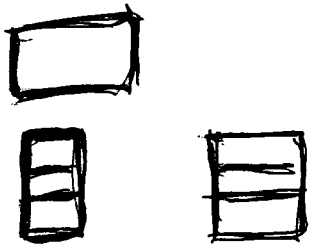
```
024104
024104   %==========================================================================
024104   %
024104   %              M O N I T O R   C A L L S
024104   %
024104   %==========================================================================
024104   % 16.11       B D B R K   B G B R K   B S B R K
024104
024104   %BREAKPOINT MONITOR CALLS:
024104   SUBR BDBRK,BGBRK,BSBRK
024104   INTEGER BPBLOCK,BPADR; INTEGER ARRAY POINTER BPNT,MACPNT
024110
024110   %MONITOR CALL TO DEFINE BREAKPOINT
024110   %T=ADDR. REGISTER BLOCK; X=ROUTINE ADDR.
024110   BDBRK: X=:B; ZTREG=:BPBLOCK; CALL CHMEM; A+7; CALL CHMEM
024116          ZXREG=:BPADR; CALL CHMEM; MIN ZPREG; 0/\0; AD:=DOLDPAGE; GO XRETBACK
024125
024125   %MONITOR CALL TO GET BREAKPOINT WHEN BREAKPOINT IS REACHED
024125   BGBRK: X=:B; BPBLOC; CALL CHMEM; ZPREG-1=:ZPREG; AD:=DOLDPAGE
024134          "5REG+10"+B=:"BPNT"; BPBLOC+10=:"MACPNT"
024142          FOR X:=-10 DO BPNT(X); CALL DNALTON; A=:MACPNT(X); CALL ALTOFF; OD
024150          BPADR=:ZPREG; GO XRETBACK
024153
024153   %MONITOR CALL TO SET BREAKPOINT ON USER RESTART
024153   %T=ADDR. REGISTER BLOCK
024153   BSBRK: X=:B; ZTREG; CALL CHMEM; A+7; CALL CHMEM
024160          "5REG+10"+B=:"BPNT"; ZTREG+10=:"MACPNT"; AD:=DOLDPAGE
024167          FOR X:=-10 DO CALL DNALTON; MACPNT(X); CALL ALTOFF; A=:BPNT(X); OD
024175          GO XRETBACK
024176   RBUS
024202
024202   %==========================================================================
024202   % 16.13       M S G
024202
024202   % MONITOR CALLS TO PRINT USER MESSAGE
024202   % X POINTS TO STRING, TERMINATED BY ',AND $ MEANS CRLF
024202   SUBR MSG
024202   DISP 0; INTEGER POINTER LREG=D0; PSID
024202   TCX:    T:=1; *MON 2OUTB
024204           GO FILERR; EXIT
024206
024206   MSG:    A:=L=:"LREG"
024210           AD:=DOLDPAG
024211           FOR X:=0 TO 1000 DO
024215                 T:=ZXREG; CALL DALTON; *LBYT
024220                 CALL ALTOFF WHILE><##'
024224                 IF A=##$ THEN 15; CALL TCX; 12 FI; CALL TCX
024233
024233           OD; A:=X SHZ -1+1+ZXREG=:X; TAD:=ZTADREG; GO LREG
024244   RBUS
024247
024247
024247
```

```
024247
024247   %=========================================================================
024247   %
024247   %            M I S C.   R O U T I N E S
024247   %
024247   %=========================================================================
024247   % 16.15       R E C O X   D U M P X   R E E C
024247
024247   %SUBROUTINES CALLED FROM RECOVER AND DUMP
024247
024247   SUBR RECOX,DUMPX,REEC,SREEC,RCOM,TOOPSEG
024247   INTEGER XPLIST:=(FILNO,NULL),XLDADR:=(0,BLADR,NWORDS)
024254   INTEGER NWORDS,BLADR,SSFLG=BLADR
024256   INTEGER RLIST:=SEGN,SEGN
024260
024260   % FOR REENTRANT SUBSYSTEMS, CALLED FROM RECOVER; A=SEG.NO.
024260   SREEC: K:=1; GO FELLS
024262   REEC:  K:="0"
024263   FELLS: L=:D; CALL ENTER;
024265          T:=BCSEG; CALL MMEXY; A=:SEGN; "RLIST"; *MON 2REEN
024272          IF K THEN
024274             A:=SEGN*5SEGSIZE+SEGSTART; A.LOGADR/\377; T:=X.LOGADR SHZ -10
024304             IF A+T>>200 THEN KABACTPRI ELSE KBACTPRI FI; A=:BACTPRI
024314             BSTART
024315   RCOM:     A=:"ESCBLOCK".RPREG
024317             IF BACTPRI=KABACTPRI THEN X.RSREG BONE "0" ELSE X.RSREG BZERO "0" FI
024330             A=:X.RSREG
024331             UEFLG; *BLDA 50 DA; BSTA 40 DA
024334             A BZERO 5UEPRUN=:UEFLG; GO STUSER
024337          FI
024337   TOOPSEG:
024337   OUT:   CALL ALTOFF; T:=OPSEG; CALL MMEXY; GO LEAVE
024343
024343   % READ SUBSYSTEM FROM FILE, CALLED FROM RECOVER
024343   % X=FIRST ADDRESS, A=NO. OF WORDS, T=BLOCK ADDR.
024343   RECOX: K:="0"; GO REDU
024345
024345   % SUBROUTINE TO WRITE PROG-FILE, CALLED FROM DUMP
024345   % X=FIRST ADDRESS, A=NO. OF WORDS, T=BLOCK ADDR.
024345   DUMPX: K:=1
024346   REDU:  L=:D; CALL ENTER; T=:BLADR; A=:NWORDS; X=:XLDADR; A:=153117
024354          IF K THEN A+1; FI; A=:D
024360          T:=BCSEG; CALL MMEXY
024362          T:=FILNO; A:=BACTPRI; CALL BALTON; A:="XPLIST"; *EXR SD
024367          GO OUT
024370   RBUS
024410
024410   %=========================================================================
024410   % 16.16       L O D X
024410
024410   % PART OF LOAD- AND PLACE-BINARY COMMANDS
024410   % ENTER WITH X=START ADDRESS, A=NO. OF CHARACTERS
024410   % RETURN WITH A=CHECKSUM
024410   SUBR LODX
024410   DISP -200; INTEGER COUNT,CHSUM,CHAR; PSID
024410   LODX:  L=:D; CALL ENTER; A-=:COUNT; 0=:CHSUM; T:=BCSEG; CALL MMEXY
024417          BACTPRI; CALL BALTON; *BSET ZRO
024422          FOR COUNT DO T:=FILNO; *MON 2INBT
024424                GO FILERR; A SH 10=:CHAR; *MON 2INBT
024430                GO FILERR; A+CHAR; *BSET ONE
```

```
024433                  A=:X.SO; *BSET ZRO
024435                  A+CHSUM=:CHSUM; X+1
024440          OD; GO TOOPSEG              % T:=OPSEG; CALL MMEXY; GO LEAVE
024443   RBUS
024447
024447   %==========================================================================
024447   % 16.17        X M A C D   X G B R K D  S B R K D
024447
024447   %COMMAND TO ACTIVATE MACD
024447   SUBR XMACD,XGBRKD,SBRKD
024447   INTEGER YMACD:=(REFBP,SBRKD,PUT1L,GET1L,USEGM,SGMAX)
024455   INTEGER RESPAR:=("5DMSEM",NULL,"1")
024460   INTEGER BREG2,MDPROG
024462           INTEGER NULPA:=(NULL)
024463   XMACD: L=:D; CALL ENTER; "RESPAR"; *MON 2RESR
024467           IF <0 GO ERR; T:="377\5MACDSEG"; CALL MMEXY; CALL ESCON
024473           RTREF.ACTSEG=:MDSEG; X=:MDPROG; A:=B=:BREG2
024501           "NULPA"; *MON 2REEN
024503           3/\PASSTYPE; T:=OPSEG; X:="YMACD"; CALL MACD
024510           BREG2=:B; IF X=0 GO RETU; T:=OPSEG; CALL MMEXY
024515           "DMACNOTLO"; CALL OUTTEXT
024517   RETU:   "RESPAR"; *MON 2RELE
024521           GO LEAVE
024522   ERR:    "TALRES"; CALL OUTTEXT; GO LEAVE
024525   *)FILL
024542
024542   % DMAC BREAKPOINT, CALLED BY GBRKD IN CORE
024542   % T=SEGMENTS  OF CALLING PROGRAM
024542
024542
024542   DISP 0; REAL F0,F3,FZP=ZPREG,FZA=ZAREG; DOUBLE DB6,DZS=ZSREG; PSID
024542   INTEGER MDBLOCK:=REFBP,SAVSEG,BPFLAG,MDSTATUS,TESTPROG
024547   INTEGER MDADR:=MACD
024550   XGBRKD: T=:SAVSEG; 1=:BPFLAG
024553           X:=MDBLOCK; FZP; T-1; TAD=:X.F0; FZA=:X.F3; DZS=:X.DB6
024563           TTNO; CALL 9ERR(-1)      %MESSAGE TO TERMINAL
024566           *IOF
024567           X:=RTREF=:TESTPROG
024571           IF X=MDPROG THEN
024574                   X.STATUS=:MDSTATUS BONE 11=:X.STATUS; MDADR=:ZPREG
024602                   3/\PASSTYPE=:ZAREG; "YMACD"=:ZXREG
024607           ELSE X.STATUS BONE 5WAIT=:X.STATUS; SAVSEG=:X.ACTSEG
024615           FI
024615           GO RETSTUPR
024616
024616
024616   % SUBROUTINE TO RESTART USER, CALLED FROM MACD
024616
024616
024616   INTEGER XREG,BREG
024620   INTEGER POINTER LREG
024621   SBRKD: X=:XREG:=B=:BREG:=L+1=:"LREG"
024626           IF BPFLAG=0 GO OUT; 0=:BPFLAG; *IOF
024632           IF X:=RTREF=TESTPROG THEN
024636                   MDSTATUS; *BLDA 110 DA
024640                   X.STATUS; *BSTA 110 DA
024642                   A=:X.STATUS
024643                   SAVSEG=:X.ACTSEG; "MSB"; *IRW MLEVB DP
024647                   MLEV; *MST PID
024651           ELSE
```

```
024652   @LIB CXCPU-,
024652   @LIB CXCPU
024652                  10=:L; MDBLOCK=:D; T:=TESTPROG.RTDLGADDR; *MOVNN
024661   @ELIB
024661                  TESTPROG.STATUS BZERO 5WAIT=:X.STATUS
024665           FI; *ION
024666   OUT:    BREG=:B; X:=XREG; GO LREG
024672   %MON. LEVEL:
024672   MSB:    X:=MDBLOCK; *LRB ALEVB
024674           GO STUPR
024675   RBUS
024703
024703   %=================================================================
024703   % 16.17B      A B A D M I
024703
024703   % SUBROUTINE TO ACTIVATE TADADM CALLED FROM TADADM COMMANDS
024703   %
024703   % ENTRY:      X = 1  -  INITIALISE AND START TADADM
024703   %             X = 2  -  STOP TADADM
024703   %             X = 3  -  TADADM SUPERVISOR MODE
024703   %
024703   % RETURN:     X = 0     NORMAL EXIT
024703   %             X ><0     ERROR, TADADM NOT LOADED
024703
024703   SUBR ABADMI
024703   INTEGER POINTER LREG
024704   INTEGER BREG2
024705   INTEGER NAMII:='SYSTEM'
024711   @ICR
024711   INTEGER SINADD:=(0,SINVER,TADAD,LOGPH,SLV10,
024716                    INIBDR,INISND,STOTAD,BDDSCN,0,0,0,0);
024726   @CR;
024726
024726   ABADMI: A:=L=:"LREG"; A:=B=:BREG2; IF X><1 GO BYPA
024735           A:="WORKA"; X=:T:="NAMII"; *MON 44; JMP ERR
024742           WORKA(11)=:SINADD; T=:X
024746   BYPA:   T:="377\5BADM"; CALL MMEXY
024750           T:="SINADD"; CALL BADM
024752           BREG2=:B; T:=OP2SEG; CALL MMEXY; GO RETU
024757   ERR:    *MON 2ERMS
024760   RETU:   BREG2=:B; 0=:SINADD; GO LREG
024764   RBUS
024772
024772   %=================================================================
024772   % 16.18       G E T 1 L   P U T 1 L
024772
024772   %SUBROUTINES TO GET AND PUT LOCATIONS
024772   %T=SEGMENT,X=ADDRESS,A=VALUE; SKIP RETURN IF OK
024772   %
024772   SUBR GET1L,PUT1L
024772
024772   INTEGER VALUE,DREG,PTNO,SFLB
024776   DOUBLE DVALUE=VALUE
024776   INTEGER GPDAD,SEGD      %PARAMETERS FOR MCALL
025000   INTEGER BREG,ADR,ALTFL,CCRSEGM
025004   INTEGER POINTER LREG
025005
025005   GET1L: T=:SEGD:="GGLOC"; GO GPUTL
025010   PUT1L: T=:SEGD:="PPLOC"
025012   GPUTL: T=:GPDAD; X=:ADR; AD=:DVALUE; A:=L=:"LREG":="BFIELD":=:B=:BREG
```

```
025022          IF TTIFIELD.FLAGB=:SFLB BIT 5ESCON THEN CALL ESCOFF FI
025030          IF SEGD BIT 17 THEN A BZERO 17 =:SEGD; T:=100=:ALTFL ELSE 0=:ALTFL FI
025041          IF A>>SGMAX GO ERR
025044          IF A=0 THEN                                   % RESIDENT
025045             X:=ADR
025046             IF GPDAD="PPLOC" THEN VALUE; CALL STPIOF; ELSE CALL LDPIOF FI
025056             GO OK
025057          FI
025057          IF A=1 THEN                                   % RT-COMMONON
025062             X:=ADR; IF GPDAD="PPLOC" THEN T:=1 ELSE T:=0 FI
025072             A:=VALUE; CALL RTCGPLOC; GO ERR; GO OK
025076          FI                                            % SEGMENT
025076          A*5SEGSIZE+SEGSTART=:X; IF X.FLAG BIT 5INHB GO ERR
025104          A:=X.LOGADR/\377=:D:=X.LOGADR SHZ -10+D=:X
025113          T:=ADR SHZ -12; A:=D/\300+T+ALTFL
025121          IF A<D OR A>=X GO ERR; A SHZ -6=:D
025127          0=:CCRSEGM
025130          IF SEGD><RTREF.SEGM SHZ -10+1 THEN X.RSEGM=:CCRSEGM; 0=:X.RSEGM FI
025142          VALUE; X:=ADR; T:="GPDAD"; *MON 2MCAL
025146          A=:VALUE
025147          IF CCRSEGM><0 THEN A=:RTREF.RSEGM; T:=X.ACTSEG; CALL MMEXY FI
025155          A:=VALUE
025156   OK:    MIN "LREG"
025157   ERR:   IF T:=SFLB BIT 5ESCON THEN CALL ESCON FI
025163          X:=DREG=:D; X:=ADR; T:=BREG=:B:=SEGD; GO LREG
025172   RBUS
025212
025212   %==========================================================================
025212   % 16.19       C H M E M
025212
025212
025212   %SUBROUTINE TO CHECK FOR LEGAL ADDRESS, IN A-REG.
025212   SUBR CHMEM,OPCOR
025212   CHMEM: EXIT
025213   RBUS
025213
025213   %==========================================================================
025213   % 16.20       U S E T
025213
025213   % SUBROUTINE TO SET PART OF USER AREA
025213   % X=LOW ADDRESS, D=HIGH ADDRESS, A=CONTENTS, T=SEGMENTS
025213   SUBR USET
025213   INTEGER POINTER LREG
025214   USET:   A:=:L=:"LREG":=L; CALL MMEXY; *BSET ONE
025221           DO A=:X.S0 WHILE X><D; X+1 OD; *BSET ZRO
025227           CALL MMEXY; GO LREG
025231   RBUS
025232
025232   %==========================================================================
025232   % 16.21       3 F I L E R R
025232
025232   %I/O ERROR MESSAGE ROUTINE
025232   %A=ERROR NUMBER
025232   SUBR 3FILERR,OPCOR
025232   INTEGER LREG
025233   3FILERR: A=:X:=L=:LREG;  "BFIELD"=:B; T:=OPSEG; CALL MMEXY
025242            A:=X BZERO 17; *MON 2ERMS
025245           GO OPCOR
025246   RBUS
025251
```



```
025251   %==========================================================================
025251   % 16.22       X A L T O N
025251
025251   % SUBROUTINE TO SET ALT.PAGE TABLE, CALLED FROM FILESYSTEM(STRING.MON)
025251   SUBR XALTON
025251   DISP 5; INTEGER BREG; PSID  % AS IN FILESYST.
025251   INTEGER XREG; DOUBLE ADREG; INTEGER POINTER LREG
025255   XALTON: AD=:ADREG; X=:XREG:=L=:"LREG"
025261           AD:=BREG.DOLDPAGE; CALL DALTON
025264           X:=XREG; ADREG; GO LREG
025267   RBUS
025270
025270   SUBR LOGOUT
025270   RBUS
025270
025270   %==========================================================================
025270   %
025270   %              B A T C H   I / O
025270   %
025270   % 16.23       B A P R O G   O B A E R R   J A B O R T   I E R   T O U S
025270   %
025270   % USER LEVEL, ENTERED FROM BINBT
025270
025270   SUBR BAPROG,OBAERR,JABORT,STSUPER,IER,TOUS,BXBAPROG
025270
025270   INTEGER XREG=?,EOJ=?,MONBP=?,TPEK=?
025270   DISP 0; REAL SF0,SF3; DOUBLE SF6; PSID
025270   SYMBOL EEOF=3
025270
025270   BAPR1: T:=XREG=:X.RXREG; T:="BFIELD"=:B; T:=TTIFIELD.DBADR+1=:MONBP.RPREG
025301          EXIT
025302
025302          A:=-2
025303   BAPROG:A BONE 17; X=:XREG:=MONBP; *SRB ALEVB
025307          T:=1=:X.RTREG; CALL BAPR1
025312   NCON:  IF A=33 THEN                    % ESCAPE
025315             IF EOJ><0 GO NWCAL; MIN EOJ
025320   TOUS:     A:=3=:TTIFIELD.BSTATE=:X.DFOPP.BCHOSTS; CALL 5BABORT
025326             T:=OPSEG; CALL OMEXIT
025330             IF INLOGGED<0 GO LOGOUT                % MODE FILE AFTER COLD-START
025333             GO STSUPER
025334          FI; IF A=27 GO ENDFI; IF A<0 GO IBAERR
025340          0=:EOJ; IF A=12 OR =0 GO NWCAL; A=:D
025347          IF TTIFIELD.BSTATE=5BUSER AND BMECHO BIT BECHOFF GO NOECH
025357          IF BCHFLAG>0 AND X.RIFIL><TTNO OR X.DFOPP.ROFIL><T THEN
025372             T:=1; A:=D; *MON 2OUTB; JMP I (OBAER
025376   NOECH: FI; A:=D
025377          IF X:=TTIFIELD.BSTATE=5BUSER THEN
025404             X:=TPEK; IF A BZERO 7=15 THEN T:=-1=:TPEK FI; T:="COMSTRING"; *SBYT
025415             TPEK+1/\77=:TPEK
025421          FI; IF BCHFLAG=2 AND TTIFIELD.CESCP/\377=X.LAST BZERO 7 THEN GO TOUS FI
025435   BRETB: X:=MONBP; "MONBLOC"=:B; X.SF0=:SF0; X.SF3=:SF3; X.SF6=:SF6
025446          GO RETBACK
025447
025447   INTEGER EOJ,MONBP:=BATBLOC,TPEK:=0
025452   INTEGER XREG,SASEG; *)FILL
025472
025472   NWCAL: MONBP.RPREG-2=:X.RPREG; GO BRETB
025477
025477   IBAERR:IF A=-2 GO NWCAL; A BZERO 17
```

27352/ 124002 142005

24

```
025503          IF A=EEOF OR =12 THEN
025511   ENDFI:     IF MSTPN=0 THEN                          % ALSO ABORT IF BATCH. BEFORE H-VERSION: IF BCHFLAG=2
025513                 IF BCHFLAG>0 GO TOUS
025516                 IF TTIFIELD.DFOPP.BCHOSTS=0 GO ATOUS
025522              FI; RTREF.STATUS=:XREG BZERO 10 BONE 11=:X.STATUS; X.ACTSEG=:SASEG
025532              T:=OPSEG; CALL MMEXY; CALL NWBPAR; T=:D:=SASEG; CALL MMEXY
025540              IF A<0 THEN
025541                 IF D=0 GO IER
025543                 IF BCHFLAG>=0 THEN
025545                    A BZERO 17; CALL 9ERR(#07); T:=OPSEG; CALL OMEXIT; GO ATOUS
025553                 FI
025553              FI; XREG=:RTREF.STATUS; GO NWCAL
025557          FI
025557   IER:   A=:XREG; T:=OPSEG; CALL OMEXIT; "BINPE"; CALL OUTTEXT
025564          A:=XREG BZERO 17; *MON 2ERMS
025567          CALL 9ERR(#06); "JOBAB"; CALL OUTTEXT
025573   ATOUS: 0=:TTIFIELD.BCHISTS; GO TOUS
025576
025576   OBAERR:0=:TTIFIELD.DFOPP.ROFIL; A BZERO 17; CALL 9ERR(#07); GO ATOUS
025605   *)FILL
025616
025616   % ABORT CURRENT JOB
025616   JABORT:"BFIELD"=:B; IF BCHFLAG=0 GO TOUS;
025623           CALL 5BABORT; T:=OPSEG; CALL OMEXIT; CALL JAB2
025627
025627   BXBAPROG: X=:XREG:=MONBP; *SRB ALEVB
025632             CALL FAR BAPR1; A=:D
025634             IF A/\177=15 OR TTIFIELD.DFOPP.ROFIL><TTNO THEN
025646                T:=BMECHO BONE BXOTTERM=:BMECHO; A:=D; T:=1; *MON 2OUTB; JMP OBAER
025655                BMECHO BZERO BXOTTERM=:BMECHO
025660             FI; GO FAR BRETB
025661   RBUS
025672
025672
025672
025672   %=================================================================================
025672   % 16.25       B A T B L O C   B E R N R   B E R T Y P   B L R E G
025672   %
025672   INTEGER ARRAY BATBLOC(10)
025702   INTEGER BERNR,BERTYP,BLREG
025705
025705
025705   %=================================================================================
025705   % 16.26       C M 1 4 4
025705
025705   % SUBROUTINE FOR THE COMMAND: DEVICE-FUNCTION
025705
025705   SUBR CM144
025705   DISP -200
025705   INTEGER TFLOG,TFFUN,TFNOT,TFNWO,TFSTAT
025705   INTEGER POINTER CLINK
025705   PSID
025705
025705   INTEGER ARRAY PM144:=(BFIELD+TFFUN,0,BFIELD+TFLOG,BFIELD+TFNWO,BFIELD+TFNWO)
025712
025712   CM144: A=:PM144(1):=L=:"CLINK"; T:=BCSEG; CALL MMEXY
025720          BACTPRI; CALL BALTON; "PM144"; *MON 2MAGT
025724          CALL ALTOFF; T:=OP2SEG; CALL MMEXY; GO CLINK
025730   RBUS
  5735
```

```
025735  %==========================================================================
025735  %      S M S G C O M M A N D    N D N C O M M A N D   S S T O P C O M
025735  %
025735  % SMSGCOMMAND:
025735  % CALLS COMMAND ROUTINES ON THE SERVICE PROGRAM/MAIL SEGMENT
025735  % NDNCOMMAND:
025735  % CALLS COMMAND ROUTINES ON THE ND-NET SEGMENT
025735  %
025735  % NDNTOCOM:
025735  % CALLS ROUTINES ON THE COMMAND SEGMENT FROM THE ND-NET SEGMENT.
025735  %
025735  % ENTRY:      X=COMMAND ROUTINE ADDRESS
025735  %
025735  SUBR SMSGCOMMAND,NDNCOMMAND,NDNTOPCOM
025735  DISP -200; INTEGER POINTER ROUTADDR; PSID
025735  INTEGER C5NNET:=377\5NNET
025736  NDNTOPCOM:    T:=OPSEG; GO L1
025740  NDNCOMMAND:   T:=C5NNET; GO FELLS
025742  SMSGCOMMAND: T:=OP2SEG
025743  FELLS: L=:D; CALL ENTER
025745  L1:     X=:"ROUTADDR"; CALL MMEXY; CALL ROUTADDR
025750          T:=OPSEG; CALL MMEXY
025752          GO LEAVE
025753  RBUS
025754
025754
025754  %==========================================================================
025754  % 16.27      P R N A M
025754  %
025754  INTEGER ARRAY PRNAM(10)       % ARRAY FOR PROJECT NAME (PADDED WITH BLANKS). WILL CONTAIN PROJECT NAME ONLY
025754                                % WHEN ACCOUNTING WAS RUNNING WHEN THE USER LOGGED IN; I.E. WHEN PRJN >< 0.
025764
025764
025764  %==========================================================================
025764  % 16.28      N D - 5 0 0   C O N N E C T   T I M E   A N D   C P U   T I M E.
025764  %
025764  DOUBLE CT500                  % SUM OF THE ND-500 CONNECT TIME DURING TIME USER IS LOGGED IN ON THE ND-100.
025766  DOUBLE CPUT5                  % SUM OF THE ND-500 CPU TIME DURING TIME USER IS LOGGED IN ON THE ND-100.
025770
025770
025770  *SYEND=*
025770
025770  @DEV 1
025770  @DEV (S-S-J)FILSYS-SYS
025770  * BGSYS+4000/
026000
026000  %==========================================================================
026000
026000  % 3   S Y S T E M   S E G M E N T   P A R T   O F   F I L E   S Y S T E M
026000
026000  %==========================================================================
026000  % 3.1   A U X I L I A R Y
026000
026000  INTEGER TDVN:=1               % LIST DEVICE NUMBER
026001  INTEGER CUSER:=-1             % CURRENT USER ENTERED
026002                                % (BIT 15-8: MAIN DIR. INDEX)
026002                                % (BIT 7-0: MAIN USER INDEX)
026002  INTEGER USDI                  % USERS DEFAULT DIRECTORY
026003  INTEGER USNO                  % USER INDEX OF DEF. DIRECTORY
026004  INTEGER SCUSE:=-1             % SAVED CUSER
026005  INTEGER SUSDN:=-1             % SAVE USDI AND USNO
```

```
026006
026006   INTEGER CRTREF                    % RTREF OF CALLING PROGRAM
026007
026007   INTEGER OFLCK:=OFLDN              % LOCK NUMBER FOR OPEN FILE TABLE. ( =0 FOR BACKGROUND PROGRAMS).
026010
026010   INTEGER ARRAY RUSNAM(11)          % REMOTE USER NAME  (RECORD USED BY FILE USER)
026021   INTEGER ARRAY LOCUS(11)           % LOCAL USER NAME
026032   INTEGER ARRAY RSYSNA(10)          % REMOTE SYSTEM NAME
026042   INTEGER ARRAY RPRPAS(10)          % REMOTE PROJECT PASSWORD
026052   INTEGER FACFLAG                   % FILE ACCESS FLAG, ><0 GO ALWAYS REMOTE
026053   INTEGER INSFG:=0                  % DATA SEGMENT INIT FLAG (REMOTE FILE ACCESS)
026054   INTEGER WPECT:=0                  % ERROR COUNT
026055   INTEGER DRFSG:=0                  % DATA SEGMENT NUMBER.   (REMOTE FILE ACCESS)
026056
026056   INTEGER UWLOGADR                  % LOGICAL PAGE NUMBER OF "USER WINDOW"
026057   INTEGER FILULREG                  % SAVED RETURN ADDRESS FOR CALL TO FILUSER
026060   INTEGER FENTLREG                  % RETURN ADDR USED IN TOFENTRY
026061   INTEGER FWFLAG                    % COPY OF WFLAG (SET BY COMMON)
026062   INTEGER SCRWRITTEN:=0             % WRITTEN IN SCRATCH FILE FLAG
026063
026063   %==========================================================================
026063   %            X S P F M E S S   F X S P F M E S S
026063   INTEGER FXSPFMESS:=0
026064   INTEGER XSPFMESS:=#''
026065   INTEGER XXSPM(60)
026145   %==========================================================================
026145   % 3.2   F I L E   S Y S T E M   S T A C K
026145
026145   %==========================================================================
026145   % 3.2.1   S T A C K
026145
026145   INTEGER ARRAY STACK(700)
027045   INTEGER ARRAY ESTCK(7)
027054   INTEGER AASTCK(0)
027054   DOUBLE ASTCK                      % A&D-REG SAVED BY ENTER
027056   INTEGER CSTCK:=STACK
027057   INTEGER XSTCX
027060
027060   %==========================================================================
027060   % 3.2.2   S T A C K   P U S H   R O U T I N E
027060
027060   % ( S P U S H )
027060
027060   % PUSH ROUTINE FOR  MACRO ENTER
027060
027060   SUBR SPUSH
027060
027060   DISP 0
027060   INTEGER XREG,TREG,AREG,DREG,LREG,BREG
027060   PSID
027060
027060   SPUSH: *STX I (XSTCX; LDX I (CSTCK; STD LREG,X
027063          *LDA I (XSTCX; STA ,X XREG
027065          *RADD SX DB; LDA (ESTCK; SKP IF DA GRE SB; JMP *
027071          *SWAP SX DB; STX I (CSTCK; LDD I (ASTCK; STF TREG,B; LDX XREG,B; EXIT
027077
027077   RBUS
027103
027103   %==========================================================================
027103   % 3.2.3   S T A C K   P O P   R O U T I N E
```

```
027103
027103    % (S P O P )
027103
027103    % POP ROUTINE FOR LEAVE
027103
027103    SUBR SPOP
027103
027103    DISP 0
027103    INTEGER XREG,TREG,AREG,DREG,LREG,BREG
027103    PSID
027103
027103    SPOP:   *ADD I (CSTCK; STA I (CSTCK; COPY SB DX; LDD LREG,B; COPY SA DL
027110            *COPY SD DB; LDF TREG,X; LDX XREG,X; EXIT
027114
027114    RBUS
```

122074/ 122070
227
1
3417

107540/ 112670 / 122070 / 122064
64 75 47
166003 166003 1
730 1756 3416
107554 105170 122070

150000-151777 172000-173777

```
027115   *
027115   %==========================================================================
027115   % 3.4   M O N I T O R   C A L L S
027115
027115   INTEGER BAPROG=?
027115
027115   %==========================================================================
027115   % 3.4.1   I N P U T   B Y T E
027115
027115   % ( F I N B T , S B I N B T )
027115
027115   % T-REG: FILE NUMBER
027115
027115   % RETURN - A-REG: ERROR CODE
027115
027115   % SKIPRETURN - A-REG: BYTE
027115
027115   SUBR FINBT,SBINBT
027115
027115   % ROUTINE ON INBT/OUTBT LEVEL
027115
027115   SBINBT: *IRR ALEVB DP; AAA 1; IRW ALEVB DP
027120           GO INBT
027121
027121   % ENTERED FROM SINTRAN IF FILE INBT:
027121
027121   FINBT: AD:=OFOP; IF A=D AND A=-1 GO ENOF                    % EMPTY FILE
027127          IF A<OFIP1 OR A=T AND D<<=OFIP2 THEN                 % END OF FILE
027137   ENOF:          A:=ER71; GO ERET; FI;
027141          IF OFIP2/\177=177 THEN
027146
027146   % CURRENT BUFFER EMPTY. TRY NEXT IN FILE SYSTEM RT-PROG.
027146                  CALL MRFI       % IN SINTRAN
027147          FI
027147
027147   % CURRENT BUFFER NOT EMPTY. GET BYTE
027147
027147          MIN OFIP2; GO SKIP1; MIN OFIP1; GO SKIP1
027153   SKIP1: T:=OFCB; A+1=:X; *LBYT
027157          *IRW ALEVB DA
027160   DUM:   *IRR ALEVB DP
027161          A+1; *IRW ALEVB DP                          % SKIP RETURN
027163          *WAIT
027164
027164   ERET:  *IRW ALEVB DA
027165          *WAIT
027166
027166   RBUS
027171
027171   %==========================================================================
027171   % 3.4.2   O U T P U T   B Y T E
027171
027171   % ( F O U T B T , S B O U T B T )
027171
027171   % T-REG: FILE NUMBER
027171
027171   % A-REG: BYTE
027171
027171   % RETURN - A-REG: ERROR CODE
027171
```

```
027171   % SIPRETURN - OK
027171
027171   SUBR FOUTBT,SBOUTBT
027171
027171   % MONITOR LEVEL ROUTINE FOR BACKGROUND PROGRAMS
027171
027171   SBOUTBT: *IRR ALEVB DP; AAA 1; IRW ALEVB DP
027174          GO OUTBT
027175
027175   % ENTERED FROM SINTRAN IF FILE OUTBT
027175
027175   FOUTBT:IF OFIP2/\177=177 THEN
027202
027202   % CURRENT BUFFER FULL. TRY NEXT (START FILE SYSTEM RT-PROG)
027202                CALL MRFO      % IN SINTRAN
027203          FI
027203
027203   % CURRENT BUFFER NOT FULL. PUT BYTE IN BUFFER
027203
027203          A+1=:X
027205          MIN OFIP2; GO SKIP1; MIN OFIP1; GO SKIP1
027211   SKIP1: AD:=OFOP; IF A=D AND A=-1 GO UPPOI          % EMPTY FILE
027217          OFIP2-D; A:=OFIP1; A:=A+C-1-OFOP1
027225          IF A>=0 THEN
027226   UPPOI:   OFIP=:OFOP
027230          FI
027230          T:=OFCB;  *IRR ALEVB DA
027232          *SBYT
027233   DUM:   *IRR ALEVB DP                                % SKIP RETURN
027234          A+1; *IRW ALEVB DP
027236          *WAIT
027237
027237   ERET:  *IRW ALEVB DA
027240          *WAIT
027241
027241   RBUS
027244
027244   %==========================================================================
027244   % 3.5   R O U T I N E S   T O   C A L L   O T H E R   S E G M E N T S
027244
027244   %==========================================================================
027244   % 3.5.1   O P C O M   C A L L   R O U T I N E
027244
027244   % ( O P C A L , E O P C A L , F I L C A L , S F I L C A L)
027244
027244   %       SUBROUTINES TO CALL ROUTINES ON OPERATOR COMM. SEGMENT, ERROR
027244   %       SEGMENT, THE FILE SYSTEM SEGMENT (FROM SPOOL SEGM.), AND THE SPOOLING
027244   %       SEGMENT FROM THE REENTRANT FILESYSTEM SEGMENT.
027244   %       THE ROUTINE IS ON THE SYSTEM SEGMENTS, AND ALL PARAMETERS
027244   %       MUST BE THERE OR CORE RESIDENT.
027244
027244   SUBR OPCAL, EOPCAL, FILCAL, SFILCAL
027244
027244   DISP 0
027244   INTEGER CCSEGM,CCRSEG
027244   INTEGER FEXRG,FEBRG,FETRG,FEARG,FEDRG,FELRG,ADDR
027244   INTEGER ARRAY FEURSEGM(16)
027244   TRIPLE FETADR=FETRG
027244   PSID
027244
```

```
027244   INTEGER POINTER SSLREG
027245
027245   FELL1: X:=:B; X=:FEBRG; TAD=:FETADR; A:=L=:"SSLREG"
027252          FELRG.SO=:ADDR
027255          IF BACKGROUND><0 THEN 177400 ELSE "ERRSEG\0" FI
027262          T:=CCSEGM \/A; CCRSEG; X:="FEURSEGM"+B; CALL SAVUSEGM
027270          TAD:=FETADR; X:="BFIELD"=:B; GO SSLREG
027274
027274   FELL2: X=:B; TAD=:FETADR; A:=L=:"SSLREG"
027300          X:="FEURSEGM"+B; CALL USAVUSEGM
027303          TAD:=FETADR; MIN FELRG; X:=FEBRG=:B; GO SSLREG
027310
027310   INTEGER ARRAY FUCNTX(0)
027310   INTEGER FXSEGS(0); *FSEG
027311   INTEGER FXRSG                                  % FILESYSTEM REENTRANT SEGMENT
027312   INTEGER FXREG,FBREG,FTREG,FAREG,FDREG
027317   INTEGER POINTER FLREG,ADDRF
027321   INTEGER ARRAY FIARR(16)
027337
027337   SFILCA: X=:FXREG:="FRSG2"; GO CCAL                 % CALL ROUTINE ON SPOOLING REENTRANT SEGMENT.
027342   FILCAL: X=:FXREG:="FRSG1"                          % CALL ROUTINE ON FILESYSTEM REENTRANT SEGMENT.
027344   CCAL:   X=:FXRSG:=L=:"FLREG":="FUCNTX"; CALL FELL1
027351           X:=FXREG; CALL ADDRF; GO FBY1; GO FBY2
027355           MIN "FLREG"
027356   FBY2:   MIN "FLREG"
027357   FBY1:   X=:FXREG:="FUCNTX"; CALL FELL2
027362           X:=FXREG; GO FLREG
027364
027364   *)FILL
027374
027374   INTEGER ARRAY OUCNTX(0)
027374   INTEGER OXSEGS(0); *OSEG; 0
027376   INTEGER OXREG,OBREG,OTREG,OAREG,ODREG
027403   INTEGER POINTER OLREG,ADDRO
027405   INTEGER ARRAY OIARR(16)
027423
027423   OPCAL: X=:OXREG:=L=:"OLREG":="OUCNTX"; CALL FAR FELL1
027430          X:=OXREG; CALL ADDRO; GO OBY1; GO OBY2
027434          MIN "OLREG"
027435   OBY2:  MIN "OLREG"
027436   OBY1:  X=:OXREG:="OUCNTX"; CALL FAR FELL2
027441          X:=OXREG; GO OLREG
027443
027443   *)FILL
027446
027446   INTEGER ARRAY EUCNTX(0)
027446   INTEGER ARRAY EXSEGS(0); *ECODS; 0
027450   INTEGER EXREG,EBREG,ETREG,EAREG,EDREG
027455   INTEGER POINTER ELREG,ADDRE
027457   INTEGER ARRAY EURSEGM(16)
027475
027475   EOPCAL: X=:EXREG:=L=:"ELREG":="EUCNTX"; CALL FAR FELL1
027502           X:=EXREG; CALL ADDRE; GO EBY1; GO EBY2
027506           MIN "ELREG"
027507   EBY2:   MIN "ELREG"
027510   EBY1:   X=:EXREG:="EUCNTX"; CALL FAR FELL2
027513           X:=EXREG; GO ELREG
027515
027515
027515   RBUS
```

```
027520
027520
027520
027520   INTEGER ARRAY GURSEGM(16)                % CALLER'S SEGMENT INFO.
027536
027536   %=====================================================================
027536   % 3.5.2    F I L S Y S   O P S Y S   O P 2 S Y   E R S Y S   S F I L S Y S
027536
027536   % SUBROUTINE TO CALL ROUTINES ON THE FILE SEGMENT
027536   % POSSIBLE CORE PARAMETERS MUST BE ON THE SYSTEM SEGMENT
027536
027536   SUBR FILSYS,ERSYS,OPSYS,OP2SY,SFILSYS
027536
027536   INTEGER POINTER SLREG,ADR
027540   INTEGER SXREG,STREG,SAREG,SDREG
027544   TRIPLE TADR=STREG
027544
027544   OPSYS:    TAD=:TADR; T:="377\OSEG"; GO FELL1
027547   OP2SY:    TAD=:TADR; T:="377\O2SEG"; GO FELL1
027552   ERSYS:    TAD=:TADR; T:="377\ECODSEG"
027554   FELL1:    A:=0; GO FELLS
027556   SFILSYS:  TAD=:TADR; A:="FRSG2"; GO FELL2
027561   FILSYS:   TAD=:TADR; A:="FRSG1"
027563   FELL2:    T:="377\FSEG"
027564   FELLS:    X=:SXREG:=L=:"SLREG":=X.S0=:"ADR"
027571             X:="GURSEGM"; CALL SAVUSEGM
027573             TAD:=TADR; X:=SXREG; CALL ADR; GO BYP1; GO BYP2
027600             MIN "SLREG"
027601   BYP2:     MIN "SLREG"
027602   BYP1:     TAD=:TADR; X=:SXREG:="GURSEGM"; CALL USAVUSEGM
027606             TAD:=TADR; X:=SXREG; MIN "SLREG"; GO SLREG
027612
027612   RBUS
027621
027621
027621   % 3.5.3      S V C A L L S E G S   R S C A L L S E G S   R T R S C A L L S E G S
027621   %            M M R E E N T
027621
027621   %      ROUTINES TO SAVE AND UNSAVE CALLER'S SEGMENTS
027621
027621   SUBR SVCALLSEGS,RSCALLSEGS,RTRSCALLSEGS,MMREENT,RTSVCALLSEGS,FSVCALLSEGS,FRSCALLSEGS
027621
027621   INTEGER ARRAY FURSEGM(16)
027637   INTEGER STREG,SAREG,SDREG,SXREG
027643   INTEGER POINTER SLREG
027644   TRIPLE TADR=STREG
027644
027644
027644   % SVCALLSEGS
027644   %      SAVE CURRENT SEGMENTS AND TAKE IN NEW SEGMENTS
027644   %
027644   %      ENTRY: X=CALLING PROGRAM
027644   %             T=NEW SEGMENTS
027644   %             A=NEW REENTRANT SEGMENT
027644   %
027644   SVCALLSEGS: TAD=:TADR; X=:SXREG:=L=:"SLREG"
027650               X:="GURSEGM"
027651   L1:         CALL SAVUSEGM
027652   OUT:        TAD:=TADR; X:=SXREG; GO SLREG
027655
```

```
027655   FSVCALLSEGS: TAD=:TADR; X=:SXREG:=L=:"SLREG"
027661          X:="FURSEGM"; GO L1
027663
027663   RTSVCALLSEGS: TAD=:TADR; X=:SXREG=:D:=L=:"SLREG"
027670          X:="GURSEGM"; CALL RTSVUSEGM
027672          GO OUT
027673
027673   % RSCALLSEGS
027673   %      RESTORE CALLER'S SEGMENT(S)
027673   %
027673   %      ENTRY: NONE
027673   %
027673   RSCALLSEGS: TAD=:TADR; A:=L=:"SLREG"; X=:SXREG
027677          X:="GURSEGM"; CALL USAVUSEGM
027701          GO OUT
027702
027702   FRSCALLSEGS: TAD=:TADR; A:=L=:"SLREG"; X=:SXREG
027706          X:="FURSEGM"; CALL USAVUSEGM; GO OUT
027711
027711
027711   % RTRSCALLSEGS
027711   %      CALLER'S BITMAP\/SAVED BITMAP=:CALLER'S BITMAP (OLD REEUT)
027711   %
027711   %      ENTRY: X=CALLER'S RT-DESCRIPTION
027711   %
027711   RTRSCALLSEGS: TAD=:TADR; A:=L=:"SLREG"; X=:SXREG=:D
027716          X:="GURSEGM"; CALL RTUSUSEGM
027720          GO OUT
027721
027721   % ROUTINE TO GET A REENTRANT SEGMENT WITHOUT MONITOR CALL
027721   % ENTRY: A=REENTRAN SEGMENT NUMBER
027721
027721   INTEGER SSXRG,CLREG
027723   MMREENT: *IOF
027724   @LIB CXCPU-,
027724   @LIB CXCPU
027724          X=:SSXRG:=L=:CLREG
027727          T:=RTREF.RSEGM; A=:X.RSEGM
027732          IF T=0 THEN
027734             A*5SEGSIZE+SEGSTART=:X
027737             IF X.FLAG BIT 50K THEN
027742                X=:SEGMC=:NSEGC; CALL CSTSEG; GO RETU1
027746             FI
027746          FI
027746          "STUPR"; *IRW MLEVB DP
027750          MLEV; *MST PID
027752   RETU:  X:=RTREF; CALL CLRBMAP
027754   RETU1: X:=SSXRG
027755          CLREG=:L; *ION; EXIT
027761   @ELIB
027761   RBUS
027777
027777   %=================================================================
027777   % 3.5.4      E F I L S Y S   E O P S Y S
027777   %
027777   % SUBROUTINES TO CALL FILESYSTEM AND OPCOM FROM ERROR PROGRAM
027777   %
027777   SUBR EFILSYS,EOPSYS
027777
027777   INTEGER POINTER ADDR
```

```
030000    INTEGER SXREG,STREG,SAREG,SDREG
030004    INTEGER POINTER SLREG
030005    TRIPLE TADR=STREG
030005
030005    EFILSYS: TAD=:TADR; T:="377\5FILSEG"; A:="FRSG1"; GO FELLS
030011    EOPSYS:  TAD=:TADR; T:="377\5OPSEG"; A:=0
030014    FELLS: X=:SXREG:=L=:"SLREG":=X.S0=:"ADDR"
030021           X:="EGURSEGM"; CALL SAVUSEGM
030023           TAD:=TADR; X:=SXREG; CALL ADDR; GO BYPASS; MIN "SLREG"
030030    BYPASS: TAD=:TADR; X=:SXREG:="EGURSEGM"; CALL USAVUSEGM
030034           TAD:=TADR; X:=SXREG; MIN "SLREG"; GO SLREG
030040    RBUS
030045
030045
030045
030045    % 3.5.5      S A V U S E G M   U S A V U S E G M
030045    %
030045    % ROUTINES TO SAVE AND UNSAVE THE CALLER'S SEGMENTS AND REENTRANT INFO,
030045    % AND TAKE IN NEW SEGMENTS
030045    %
030045    %
030045    SUBR SAVUSEGM,USAVUSEGM,RTSVUSEGM,RTUSUSEGM
030045
030045    % SAVUSEGM
030045    %       ENTRY: X=ADDRESS OF WORKING FIELD (10 LOCATIONS)
030045    %              A=NEW REENTRAN SEGMENT
030045    %              T=NEW SEGMENT(S)
030045    %
030045    SAVUSEGM: X:=:B; X=:BXBRG:=L=:BXLRG; A=:BXARG
030052           RTREF.RSEGM=:XURSEGM; 0=:X.RSEGM
030056           CALL MMEXY; T=:XUSEGM
030060    L1:    CALL BMFRTD
030061           BXARG; CALL MMREENT
030063    OUT:   X:=BXBRG; BXLRG=:L; X:=:B; EXIT
030070
030070    RTSVUSEGM: X:=:B; X=:BXBRG:=L=:BXLRG; A=:BXARG
030075           D.RSEGM=:XURSEGM; X.ACTSEG=:XUSEGM; CALL MMEXY; GO L1
030104
030104    % USAVUSEGM
030104    %       ENTRY: X=ADDRESS OF WORKING FIELD
030104    %
030104    USAVUSEGM: X:=:B; X=:BXBRG:=L=:BXLRG
030110           XURSEGM=:RTREF.RSEGM; CALL BMTRTD
030114           T:=XUSEGM; CALL MMEXY; GO OUT
030117
030117    RTUSUSEGM: X:=:B; X=:BXBRG:=L=:BXLRG
030123           *IOF
030124           X:=D; CALL BMOR
030126           *ION
030127           GO OUT
030130    RBUS
030136
030136    % MONITOR CALL ERMSG AND QERMS  (MON 64 - MON 65)
030136
030136    SUBR ERMSG,QERMS
030136
030136    INTEGER SBREG,SAREG,CFLAG,SOLDSEGS
030142    INTEGER POINTER SLREG
030143    ERMSG: T:=0; GO FELLS
030145    QERMS: T:=1
```

```
030146   FELLS: T=:CFLAG:="BFIELD":=:B=:SBREG; A=:SAREG:=L=:"SLREG"
030155          IF BACKGROUND><0 THEN "177400" ELSE "ERRSEG\0" FI
030162          T:=ECODSEG\/A; CALL MMEXY; T=:SOLDSEGS
030166          SAREG; CALL ERLIST; T:=SOLDSEGS; CALL MMEXY
030172          IF CFLAG><0 THEN
030174             IF BACKGROUND><0 GO ESCQERM
030177             IF SBREG.RTRES=X.SSREF THEN
030204                T.ACTPRI/\174774 BONE 1=:X.ACTPRI; "MMLE"; X:=X.RTDLGADDR; T:=0; *STATX     %DPREG
030215             FI
030215          FI; T:=SBREG=:B; TAD:=ZTADREG; X:=ZXREG; GO SLREG
030222   RBUS
030233
030233
030233
030233
030233   % 3.6   B U F F E R   P O O L   F O R   O P E N   F I L E S
030233
030233   % ( B P O O L )
030233
030233   % THERE MUST BE AT LEAST ONE 64 WORD BUFFER FOR EACH
030233   % OPEN FILE NUMBER.   EACH BUFFER IS PRECEEDED BY A LINK CELL
030233
030233   INTEGER SPOOL                  % START OF FREE LIST
030234   INTEGER RPOOL                  % START OF RESERVED LIST
030235   INTEGER NPOOL:=5NPOO
030236
030236   INTEGER ARRAY BPOOL=?
030236
030236   INTEGER SDFLAG
030237
030237   % BPOOL IS INITIALLY USED FOR SWAPP DRIVER AND IOBUTAB
030237   % BPOOL MUST THEREFOR BE AVAILABLE WHEN THE SYSTEM IS GENERATED
030237
030237
030237   % 3.3   O P E N   F I L E   T A B L E
030237
030237   INTEGER DV100; **-1/FMAX
030240   INTEGER ARRAY OPTAB(2*20)
030300   INTEGER ARRAY OPSPO(2*2)
030304   INTEGER ENDOP=?
030304   *CONTX=OPTAB+40+40+3434                % DEFAULT LENGTH OF OPTAB AND BPOOL
030304   *OPTAB<CONTX-1; )ZERO
```

```
030304   *
030304   @DEV 1
030304   @DEV (S-S-J)SINB-2
030304
030304   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%% S I N B - 2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
030304
030304   *BCSTA/;*<*+3;)ZERO;*+3/      % FREE LOCATIONS FOR LIMIT INDICATORS
110003   %=========================================================================
110003   %
110003   %   17.0      C O M M A N D   S E G M E N T
110003   %
110003   %=========================================================================
110003
110003   %=========================================================================
110003   % 17.1        U S T X   R E S I P A R   R E S O P A R
110003   %
110003   INTEGER SVERALL:='$ SINTRAN III - '
110014   INTEGER SVLETTER:=' J '
110016   INTEGER SVER0:='VS',SVER1:='VSE',SVER2:='VSE/500'    %%%%%,SVER3:='RTP'
110026   INTEGER SVER4:='VSX',SVER5:='VSX/500'
110034   INTEGER USTX5:='@'
110035   INTEGER LUSTX:=USTX5
110036   INTEGER USTX4:='R@'
110040   INTEGER ARRAY UDD4(20)
110060   INTEGER USTXT:=##'
110061   INTEGER ARRAY UDD(70)
110151   INTEGER RESIPAR:=(TTNO,NULL,NULL),RESOPAR:=(TTNO,"1",NULL)
110157   INTEGER UNEXTRA=?              % FOR SET-UNAVAILABLE
110157   INTEGER EXUNAFLAG:=1
110160   INTEGER LSBUF                  % LOGICAL ADDRESS OF DEVICE BUFFER
110161   INTEGER DBPCOLDSTART           % SAVED DBPROG FOR THE TERMINAL USED AFTER @COLD-START
110162
110162   %=========================================================================
110162   %             C O M M A N D   P R O C E S S I N G
110162   %
110162   %=========================================================================
110162   %             M I S C .   R O U T I N E S
110162   %
110162   % SEQUENCES WHERE PAGE FAULTS ARE NOT ALLOWED
110162
110162   %=========================================================================
110162   % 17.2        S T O P S Y S
110162   %
110162   % COMMAND: STOP-SYSTEM - SIMULATING POWER FAIL
110162   SUBR STOPSYS
110162   STOPSYS: L=:D; CALL ENTER; "PPWFAIL"; *IOF; IRW 160 DP
110167           40000; *MST PID; ION
110172           GO LEAVE
110173   RBUS
110176
110176   %=====================================================================
110176   % 17.15.1     R E S E S C
110176
110176   % SUBROUTINE TO RESET ALL USER DEFINED ESCAPE/LOCAL HANDLING TO
110176   % NORMAL SINTRAN MODE WHEN COMMAND-MODE IS ENTERED. THE ROUTINE
110176   % IS ALSO CALLED AT LOGOUT.
110176   % ZERO IN ESCMASK: 5LCHAR,5USESC,5ESCLOFF,5WESC,5WLOC,5LOC2SET
110176   SUBR RESESC
110176   RESESC: TTIFIELD.FLAGB/\ESCMASK=:X.FLAGB; EXIT
110203   RBUS
```

```
110204
110204
110204   %=============================================================================
110204   % 17.3        X S T O P T E R M
110204   %
110204   % LAST PART OF THE COMMAND STOP-TERMINAL
110204   % D=DATAFIELD, A=BACKGROUND PROGRAM
110204   SUBR XSTOPTERM
110204   INTEGER BRG,LRG
110206   XSTOPTERM: *IOF
110207           IF A><RTREF AND A=D.RTRES THEN
110216             T:=L=:LRG
110220   @LIB CXCPU
110220             A:=B=:BRG
110222             IF D.TYPRING BIT 5TERM THEN
110226               T:="FLAGB"; CALL XGTDFADDR; A/\ESCMASK; T:="FLAGB"; CALL XSTDFADDR
110233               T:="LAST"; -1; CALL XSTDFADDR
110236               T:="DFLAG"; CALL XGTDFADDR; A BZERO 5IESC; T:="DFLAG"; CALL XSTDFADDR
110243               X=:B; CALL STESCAPE
110245             ELSE
110246               X=:B; -1=:LAST; DFLAG BZERO 5IESC=:DFLAG
110254               FLAGB /\ESCMASK=:FLAGB; CALL ESCAPE; 0/\0
110261             FI; BRG=:B
110263   @ELIB
110263   @LIB CXCPU-,
110263             LRG=:L
110265           FI; *ION
110266           EXIT
110267   RBUS
110300
110300
110300   %=============================================================================
110300   % 17.4        X B L O G O U T
110300   %
110300   % LAST PART OF BATCH LOGOUT ROUTINE
110300   SUBR XBLOGOUT
110300   INTEGER RTBPAR:=USRTB
110301   DISP -200; INTEGER BI,BO,SAVLINK; PSID
110301   XBLOGOUT:  A:=TTNO; CALL LOGPH; A=:BI:=D=:BO; MLEV; *MCL PIE
110310           A:=RTREF.BRESLINK
110312           DO WHILE A><X
110314             A:=:B; T:=RESLINK; A:=:B; T=:SAVLINK
110320             IF A><BI AND ><BO THEN A=:B; CALL BRELEASE; T:="BFIELD"=:B FI
110332             A:=SAVLINK
110333           OD MLEV; *MST PIE
110336           IF RTBPAR.SO><0 THEN "RTBPAR"; *MON 2RT
110343           FI; T:=201; *MON 2XMSG; SAT 1; MON 2XMSG  % DISCONNECT ALL XMSG CONNECTIONS
110347           GO STSUPER
110350   RBUS
110357
110357
110357   @LIB SSCOM
110357
110357   %=============================================================================
110357   % 17.7        N S T A R T
110357   %
110357   % SUBROUTINE TO RESTART BATCH PROCESS
110357   % ENTRY: X = INDEX IN BATCH TABLE OF PROCESS TO START
110357   %        L = ADDRESS OF START ADDRESS
110357   SUBR NSTART
```

```
110357   DISP -200; INTEGER POINTER ENTPOINT,LREG; PSID
110357
110357   % USER LEVEL
110357   NSTART:A:=L=:"ENTPOINT"+1=:"LREG"; X+2; A:=BATAB(X); CALL LOGPH; X-1; T:=BATAB(X)
110370          IF T=X:=RTREF THEN ENTPOINT=:"ENTPOINT"; GO ENTPOINT FI; *IOF
110377          *IRW MLEVB DB
110400          T=:A; *IRW MLEVB DX
110402          A:=ENTPOINT; *IRW MLEVB DA
110404          "MNSTAR"; *IRW MLEVB DP
110406          MLEV; *MST PID; ION
110411          GO LREG
110412
110412   % MONITOR LEVEL
110412   % B = DATAFIELD, X = RT-DESCRIPTION, A = START ADDRESS
110412   MNSTAR:
110412          X=:D:=X.RTDLGADDR; T:=0; *STATX
110416          D.STATUS BONE 11 BZERO 5WAIT=:X.STATUS; "100002+ALEVB"=:X.ACTPRI
110425          CALL FRWQU; CALL TOEXQU
110427          CALL S5ESCF; GO STUPR
110431   RBUS
110441
110441
110441
110441   @LIB SSCOM
110441
110441
110441   %==========================================================================
110441   % 17.12       X S R T O F   X R R T O F
110441   %
110441   % SET RTON/RTOFF ON AN RT-PROGRAM
110441   %
110441   % ENTRY: X=RTDESCRIPTION ADDRESS
110441   %
110441   SUBR XSRTOF,XRRTOF
110441   XSRTOF: *IOF
110442           IF X.WLINK=0 THEN X.ACTPRI BONE 5RTOF=:X.ACTPRI; L+1 FI
110450           *ION; EXIT
110452   XRRTOF: *IOF
110453           X.ACTPRI BZERO 5RTOF=:X.ACTPRI
110456           *ION
110457           EXIT
110460   RBUS
110460
110460   INTEGER ARRAY RPAR=?,WPAR=?
110460   INTEGER FBLCK=?,TBLCK=?,SBANK=?,SBUF=?,NBLCK=?,XFUNC=?,SBANK=?
110460   DOUBLE DTBLCK=?,DFBLCK=?
110460
110460   %==========================================================================
110460   %       C C B R E L E A S E
110460   %
110460   % SUBROUTINE TO CALL BRELEASE, CALLED FROM STSUPER
110460   % ENTRY:      A=ADDRESS OF DATAFIELD
110460   %
110460   SUBR CCBRELEASE
110460   INTEGER POINTER LLRG; INTEGER XXRG
110462   CCBRELEASE: A=:T:=MLEV; *MCL PIE
110465           A:=L=:"LLRG"; X=:XXRG
110470           IF T><0 AND T.RTRES=RTREF THEN
110477              X=:B:=RTREF; CALL BRELEASE; "BFIELD"=:B
110504           FI; "LLRG"=:L; X:=XXRG; MLEV; *MST PIE
```

```
110511          EXIT
110512  RBUS
110515
110515
110515  %=========================================================================
110515  %             C C B R S R V E
110515  %
110515  %  SUBROUTINE TO CALL BRESERVE
110515  % ENTRY:      A=ADDRESS OF DATAFIELD TO RESERVE
110515  %             T=0; WAIT FOR RESOURCE TO BE FREE
110515  %             T><0: DO NOT WAIT FOR RESOURCE TO BE FREE (RETURN)
110515  % EXIT:       RESOURCE OCCUPIED
110515  % EXIT+1:     THE RESOURCE IS RESERVED BY THE CALLER
110515  %
110515  SUBR CCBRSRVE
110515  INTEGER BRG,LRG,XXRG,CFLG
110521  CCBRSRVE: A=:D; *IOF
110523          D:=:B; T=:CFLG:=D=:BRG:=L=:LRG; X=:XXRG:=RTREF; CALL BRESERVE
110534          IF A<0 THEN
110535             IF CFLG><0 GO OUT
110537             CALL FREXQU; CALL TOWQU
110541             "STUPR"; *IRW MLEVB DP
110543              MLEV; *MST PID
110545          FI; MIN LRG
110546  OUT:    X:=XXRG; BRG=:B; LRG=:L
110553          *ION
110554          EXIT
110555  RBUS
110562
110562
110562  %=========================================================================
110562  %       S T S L U T T
110562  %
110562  % ROUTINE TO SET PRIORITY TO ZERO FOR THE STSIN PROGRAM
110562  % AND TO TERMINATE THE STSIN PROGRAM
110562  % CALLED FROM OLDSTART
110562  %
110562  SUBR STSLUTT
110562
110562  STSLUTT: MLEV; *MCL PIE
110564          DO WHILE RTREF.BRESLINK><X; A=:B; CALL BRELEASE OD
110573          CALL FREXQU; 1000=:X.STATUS
110576          "STUPR"; *IOF; IRW MLEVB DP
110601          MLEV; *MST PIE; MST PID; ION; JMP *
110606  RBUS
110613
110613
110613
110613  %=========================================================================
110613  % 17.14       N W B P A R   N W 2 P A R
110613  %
110613  % OPEN BATCH INPUT AND OUTPUT FILES
110613  % USER LEVEL
110613  % NWBPAR: FETCH NEW PARAMETERS FROM INTERNAL DEVICE
110613  % NW2PAR: REOPEN OLD FILES
110613  INTEGER RINPA:=(BINDV,NULL,NULL)
110616
110616  SUBR NWBPAR,NW2PAR
110616
110616  INTEGER MONBP=?
```

```
110616
110616   IDLTEST:*MON 2ISIZ; JMP *
110620          X=:D
110621          IF A=0 THEN 0=:TTIFIELD.IDLE; 1=:X.ISTATE; T:=X.RIFIL; *MON 2CLOS; 0
110631            T:=X.DFOPP.ROFIL; *MON 2CLOS; 0
110635          ELSE
110636            X:=TTIFIELD; 1=:X.IDLE; 0=:X.ISTATE
110642          FI
110642          T:=BINDV; D=:X
110644          EXIT
110645
110645   DISP 0; REAL SF0,SF3; DOUBLE SF6; PSID
110645   INTEGER XREG
110646
110646   NWBPAR:A:=L=:BLREG; 0=:BMECHO
110651          X:=MONBP; "WORKA"=:B; X.SF0=:SF0; X.SF3=:SF3; X.SF6=:SF6; "BFIELD"=:B
110664          IF MSTPN>0 THEN
110667            A-2=:MSTPN=:XREG
110672            DO WHILE X:=XREG>=0
110674              AD:=MOSTK(X)
110675              IF A=TTIFIELD.RIFIL OR D=T THEN 0=:X.RIFIL FI
110704              IF D=X.DFOPP.ROFIL OR A=T THEN 0=:X.ROFIL FI
110713              XREG-2=:XREG
110716            OD
110717            T:=TTIFIELD.RIFIL; *MON 2CLOS; JMP *1
110723            T:=X.DFOPP.ROFIL; *MON 2CLOS; JMP *1
110727            AD:=MOSTK(MSTPN); A=:TTIFIELD.RIFIL; A:=D=:X.DFOPP.ROFIL; 0=:BERNR; GO UTNW
110740          FI
110740          "RINPA"; *MON 2RESR
110742          X:=0; DO T:=BINDV; CALL IDLTEST; 0/\0; *MON 2INBT; JMP *
110750          T:="FIBUF"; *SBYT
110752          X+1; IF A-##'=0 GO OUT OD
110756   OUT:   IF X BIT "0" THEN X+1 FI A:=X SHZ "-1"+T=:FOBUF; X:=0
110766          DO T:=BINDV; CALL IDLTEST; *MON 2INBT; JMP *1
110772          T:=FOBUF; *SBYT
110774          IF A=##' GO OUT2; X+1 OD
111001   OUT2:  MIN TTIFIELD.IDLE; 0=:X.ISTATE; "RINPA"; *MON 2RELE
111006          GO NW2; *)FILL
111016
111016   INTEGER MONBP:=BATBLOC
111017
111017   NW2PAR:A:=L=:BLREG; 0=:D; 0=:BMECHO; GO NW22
111024   NW2:   1=:D
111026   NW22:  0=:BERNR; X:=TTIFIELD; CALL XSTBACK
111031          T:=TTIFIELD.RIFIL; 0=:X.RIFIL; *MON 2CLOS; JMP *1
111036          A:="TYPS"; X:="FIBUF"; T:=1=:BERTYP; *MON 2DOPE; JMP IOER
111044          IF A=0 THEN 127; GO IOER FI
111047          A=:TTIFIELD.RIFIL; MIN X.BCHISTS; CALL FILSYS(SETPO); *JMP *
111055          IF D=0 THEN
111057            T:=TTIFIELD.DFOPP.ROFIL; *MON 2CLOS; JMP *1
111064            A:="TYPS"; X:=FOBUF; T:=5; *MON 2DOPE; JMP OOER
111071            IF T:=BCHFLAG<0 AND A>=100 THEN
111077              T:=A; *MON 2CLOS; JMP *1; JMP UTNW
111103            FI
111103            A=:TTIFIELD.DFOPP.ROFIL; CALL FILSYS(SETPO); *JMP *
111111            MIN X.BCHOSTS
111112          FI
111112   UTNW:  X:=MONBP; "WORKA"=:B; SF0=:X.SF0; SF3=:X.SF3; SF6=:X.SF6; "BFIELD"=:B
111125          A:=BERNR; T:=BERTYP; X:=BLREG=:P
111131
```

```
111131   IOER:  A BONE 17=:BERNR; 0=:BERTYP; GO UTNW
111135   OOER:  A BONE 17=:BERNR; GO UTNW
111140   RBUS
111152
111152   %=============================================================================
111152   %              M L G R S T A R T
111152   %
111152   SUBR MLGRSTART
111152   MLGRSTART: L=:D; CALL ENTER
111154          T:=TTNO=:D:=A; *IOF
111160          IF "MLIDFIELD".RTRES><0 AND X.DO=D THEN
111166             T=:X.MLIMSTATUS; X.RTRES.STATUS BZERO 5WAIT=:X.STATUS
111173          FI; *ION
111174          TTIFIELD.FLAGB BZERO 5SPSIO=:X.FLAGB
111200          GO LEAVE
111201   RBUS
111202
111202
111202   %=============================================================================
111202   % 25.2        O I S Y S E G
111202   %
111202   % INITIALIZE-SYSTEM-SEGMENT <SEGMENT>
111202   %
111202   % CALLED FROM SINTRAN-SERVICE-PROGRAM ON SEGMENT 23 (OP2SEG)
111202
111202   SUBR OISYSEG
111202
111202   DISP -200
111202   INTEGER CINDX,CADDR
111202   PSID
111202
111202   INTEGER PWSEG:=CSEG
111203   INTEGER ERT1:='$ILLEGAL SEGMENT NUMBER'
111217   INTEGER ERT2:='$NOT SYSTEM SEGMENT'
111231   INTEGER TPACTIVE:='$BACKGROUND PROGRAM IS ACTIVE$'
111251   INTEGER TERRO:='$NO BACKGROUND PROGRAM WILL USE THIS SEGMENT'
111300   INTEGER 6DBPROG,CSEG
111302
111302   OISYSEG: L=:D; CALL ENTER
111304          "GGSGM"; CALL SGPAR; GO ERM64; A=:CSEG     % GET SEGMENT NUMBER
111310          IF A=0 OR A>>SGMAX GO ERR1                 % ILLEGAL SEGMENT NUMBER
111314          A*5SEGSIZE+SEGSTART
111316          IF A.FLAG NBIT 5SYSEGM GO ERR2             % NOT SYSTEM SEGMENT
111322          0=:6DBPROG
111323          FOR X:="BAK01" STEP 5RTSIZE TO "9LBPR-5RTSIZE" DO
111327             IF X.SEGM SHZ -10=CSEG THEN             % CHECK IF PROG IS ACTIVE
111334                CALL XSRTOF; GO ERR3                 % SET RTOFF IF PROG IS INACTIVE
111336                X=:6DBPROG; GO L1
111340             FI
111340          OD; GO EROUT
111343   L1:    IF "BPTMP"=0 THEN                  % NOT BACKGROUN PROC. ALLOCATION SYSTEM
111345             0=:CINDX; "BACKT"=:CADDR
111350             DO WHILE X:=CADDR.S0><-1       % SEARCH FOR ENTRY IN BACKGROUND TABLE
111355                IF X><0 THEN                % ENTRY IS USED
111356                   MIN CINDX; T:="DBPROG"; CALL XGTDFADDR
111361                   IF A.SEGM SHZ -10=CSEG GO ISYS1  % ENTRY FOUND
111367                FI; MIN CADDR
111370             OD; GO EROUT                   % ENTRY NOT IN BACKGROUND TABLE
111372          ELSE
111373             IF 6DBPROG=0 GO EROUT
```

```
111375              A-"BAK01"=:D:=0; T:=5RTSIZE; *RDIV ST
111402              A=:CINDX
111403              IF X:="BACKT"+A<<"BBCHT" THEN X:=0 FI; X=:CADDR
111412           FI
111412
111412   ISYS1: T:=MASSNO(0); 2000; CALL G3BUF; CALL ERRFATAL
111417          T:=A=:SBANK; X:=RTREF.ACTPRI:=:D=:SBUF; CALL DBTRANS; X=:LSBUF
111427          T:=2; X:=CADDR; CINDX; CALL MOVSYSEG
111433          T:=MASSNO(0); CALL R3BUF; CALL ERRFATAL
111437   OUT:   IF X:=6DBPROG><0 THEN CALL XRRTOF FI
111442          "PWSEG"; *MON 2WSBC
111444          GO LEAVE
111445
111445   EROUT: "TERRO"
111446   EFEL:  T:=1=:TDVN; CALL OUTTEXT; GO LEAVE
111452   ERR1:  "ERT1"; GO EFEL
111454   ERR2:  "ERT2"; GO EFEL
111456   ERR3:  "TPRACTIVE"; GO EFEL
111460   ERM64: *MON 64
111461          GO LEAVE
111462   RBUS
111521
111521   INTEGER EDIRTL:='$ERROR DETECTED BY RT-LOADER IN BUILDING SEGMENT'
111552   INTEGER TSGNN:='GGSEGMENT: ',TSGNAM:='GSSEGMENT NAME: '
111571
111571   %==========================================================================
111571   % 18.20      L I R E E N T
111571   %
111571   % COMMAND: LIST-REENTRANT (<NAME>)
111571
111571   SUBR LIREENT
111571   DISP -200; INTEGER HEAFLG; PSID
111571   INTEGER STRCOM:='S NAME: '
111576   INTEGER STHEA:='  START RESTART SEGMENT NAME$'
111615   INTEGER T5STD:='ND-500 STANDARD DOMAIN: '
111632
111632   LIREENT: L=:D; CALL ENTER
111634          0=:HEAFLG; "STRCOM"; CALL KGPAR; A:="TXAPPS"; A=:CSTRING
111641          X:="REECOMTAB"
111642          DO WHILE X.S0><-1
111646             IF X.S1><0 THEN
111650                0=:CPNT; X.S0; CALL ABL1
111653                IF A><2 THEN
111656                  IF HEAFLG=0 THEN "STHEA"; CALL OUTTEXT; 1=:HEAFLG FI
111664                  IF X.S1=-1 THEN
111670                     "T5STD"; CALL OUTTEXT
111672                  ELSE
111673                     X.S2; CALL OCTU; X.S3; CALL OCTU; X.S1; CALL OCTU
111701                     40; CALL TCO; CALL TCO; CALL TCO
111705                  FI; X.S0; CALL OUTTEXT; CALL CRLF
111710                FI
111710             FI; X+4
111711          OD; GO LEAVE
111713   RBUS
111725
111725
111725   %==========================================================================
111725   %            L R E S E G N
111725   % COMMAND:   LOAD-REENTRANT-SEGMENT <FILE NAME> <SEGMENT NAME>
111725   %
```

```
111725   SUBR LRESEGN
111725   DISP -200; INTEGER CFNO; PSID
111725   LRESEGM: L=:D; CALL ENTER
111727           "STRFILE"; CALL GPAR; CALL PROPEN; GO CFIERR; A=:CFNO
111734           "TSGNAM"; CALL GPAR; T:=CFNO; CALL PZRTLOADER
111740           IF A=0 THEN "EDIRTL"; CALL OUTTEXT FI
111743           T:=CFNO; *MON 2CLOS; JMP * 1
111746           CALL CRLRTL
111747           GO LEAVE
111750   CFIERR: *MON 64
111751           GO LEAVE
111752   RBUS
111761
111761
```

```
111761
111761   * *-1@-12+1@12/
112000   INTEGER REEPNT:=REECOMTAB,REECSPNT:=REECTEXT                          %%%%%
112002   *-1;*-1/                                                              %%%%%
112002   INTEGER ARRAY REECOMTAB(4*113)  % REENTRANT COMMAND TABLE              %%%%%
112456   INTEGER REECEND:=-1; INTEGER ARRAY REECTEXT(1024),REETEND(0)          %%%%%
113503                                                                         %%%%%
113503                                                                         %%%%%
113503   %=============================================================================
113503   %             F R E E S G N                                            %%%%%
113503   %                                                                      %%%%%
113503   % FIND REENTRANT SEGMENT IN REECOMTAB                                 %%%%%
113503   %                                                                      %%%%%
113503   % ENTRY:      A=SEGMENT NUMBER                                        %%%%%
113503   % EXIT:       SEGMENT NOT FOUND                                       %%%%%
113503   % EXIT+1:     X=ADDR TO ENTRY IN REECOMTAB                            %%%%%
113503   %                                                                      %%%%%
113503   SUBR FREESGN                                                          %%%%%
113503   FREESGN: A=:D; MLEV; *MCL PIE                                         %%%%%
113506           X:="REECOMTAB"                                                %%%%%
113507           DO WHILE X.S0><-1                                             %%%%%
113513              IF X.S1=D THEN L+1; GO OUT FI                              %%%%%
113520              X+4                                                        %%%%%
113521           OD                                                            %%%%%
113522   OUT:    MLEV; *MST PIE                                                %%%%%
113524           EXIT                                                          %%%%%
113525   RBUS                                                                  %%%%%
113526                                                                         %%%%%
113526                                                                         %%%%%
113526   %=============================================================%%%%%======
113526   %             5 D L R E E N T                                          %%%%%
113526   %                                                                      %%%%%
113526   % DELETE A ND-500 STANDARD DOMAIN FROM THE "REENTRAN SUBSYSTEM" TABLE %%%
113526   %                                                                      %%%%%
113526   % ENTRY:      A= ND-500 STANDARD DOMAIN INDEX                         %%%%%
113526   %                                                                      %%%%%
113526   SUBR 55DLREENT                                                        %%%%%
113526
113526   DISP -200; INTEGER CDOMINDX; PSID
113526
113526   55DLREENT: L=:D; CALL ENTER                                           %%%%%
113530           A=:CDOMINDX; "REESM"; T:=0; CALL CCBRSRV; 0/\0                %%%%%
113535           X:="REECOMTAB"; CDOMINDX=:D; MLEV; *MCL PIE                   %%%%%
113542           DO WHILE X.S0><-1                                             %%%%%
113546              IF X.S1=-1 AND X.S2=D THEN                                 %%%%%
113555                 MLEV; *MST PIE                                          %%%%%
113557                 CALL DLRETAB; GO OUT                                    %%%%%
113561              FI; X+4                                                    %%%%%
113562           OD; MLEV; *MST PIE                                            %%%%%
113565   OUT:    "REESM"; CALL CCBRELEASE                                      %%%%%
113567           GO LEAVE                                                      %%%%%
113570   RBUS                                                                  %%%%%
113575   %=============================================================================
113576   %             D L R E T A B                                            %%%%%
113575   % CLEAR AN ENTRY IN THE "REENTRAN-SUBSYSTEM-TABLE" (REECOMTAB)         %%%%%
113575   % ENTRY:      X=ADDRESS IN REECOMTAB OF ENTRY TO DELETE               %%%%%
113575   %                                                                      %%%%%
113575   SUBR DLRETAB                                                          %%%%%
113575                                                                         %%%%%
113575   INTEGER DUMS:=''                                                      %%%%%
```

```
113576   INTEGER C1ADR,C2ADR                                                  %%%%%
113600   INTEGER PWSBC:="5OPSEG"
113601                                                                        %%%%%
113601   DLRETAB: L=:D; CALL ENTER                                            %%%%%
113603            X=:C1ADR; X.S0=:C2ADR                                       %%%%%
113606            T:="DUMS"; -1=:D                                  %%%%%%%%%%%%%%%%%%%%%%%%
113611            DO                                                %%%%%%%%%%%%%%%%%%%%%%%%
113611               MLEV; *MCL PIE                                 %%%%%%%%%%%%%%%%%%%%%%%%
113613               X.S4=:X.S0; X.S5=:X.S1; X.S6=:X.S2; X.S7=:X.S3 %%%%%%%%%%%%%%%
113623               IF X.S4><-1 THEN T=:X.S4 FI                    %%%%%%%%%%%%%%%%%%%%%%%%
113630               MLEV; *MST PIE                                 %%%  MUST NOT CROSS  %%%
113632            WHILE X.S0><D                                     %%%  PAGE LIMIT      %%%
113635               X+4                                            %%%%%%%%%%%%%%%%%%%%%%%%
113636            OD; X=:REEPNT                                     %%%%%%%%%%%%%%%%%%%%%%%%
113640            DO WHILE C1ADR.S0><-1                             %%%%%%%%%%%%%%%%%%%%%%%%
113645               A=:D; X:=0                                     %%%%%%%%%%%%%%%%%%%%%%%%
113647               MLEV; *MCL PIE                                 %%%%%%%%%%%%%%%%%%%%%%%%
113651               DO                                             %%%%%%%%%%%%%%%%%%%%%%%%
113651                  T:=D; *LBYT                                 %%%%%%%%%%%%%%%%%%%%%%%%
113653                  T:=C2ADR; *SBYT                             %%%%%%%%%%%%%%%%%%%%%%%%
113655               WHILE A><##'                                   %%%%%%%%%%%%%%%%%%%%%%%%
113660                  X+1                                         %%%%%%%%%%%%%%%%%%%%%%%%
113661               OD; A:=X SHZ -1+C2ADR+1; T:=C2ADR=:C1ADR.S0; A=:C2ADR %%%%%%%
113672               X+4=:C1ADR                                     %%%%%%%%%%%%%%%%%%%%%%%%
113674               MLEV; *MST PIE                                 %%%%%%%%%%%%%%%%%%%%%%%%
113676            OD; C2ADR=:REECSPNT
113701            "PWSBC"; *MON 2WSEG
113703            GO LEAVE
113704   RBUS
113711
113711
113711   %==========================================================================
113711   %            5 D F R E E N T   -   5 D L R E E N T
113711   %
113711   %  ENTRIES FROM ND-500 MONITOR
113711   %
113711   SUBR 5DFREENT,5DLREENT
113711   INTEGER POINTER 5LRG
113712   INTEGER 5BRG
113713   5DFREENT: K:="0"; GO FELLS
113715   5DLREENT: K:=1
113716   FELLS:    X:=L=:"5LRG":="BFIELD":=:B=:5BRG
113723             IF K THEN CALL 55DLREENT ELSE CALL 55DFREENT FI
113730             X:=5BRG=:B; GO 5LRG
113733   RBUS
113736
113736
113736   %==========================================================================
113736   % 18.19       R E E D U M P   R E E C O M T A B
113736   %            D P R E E N T   D F P R E E N T
113736   %            L R E S E G
113736
113736   % COMMAND: DUMP-REENTRANT NAME,START,RESTART,BIN.FILE,SEGMENT-NAME
113736   %         : DUMP-PROGRAM-REENTRANT NAME,PROG-FILE,SEGMENT NAME
113736   %         : DEFINE-REENTRANT-PROGRAM  NAME,START,RESTART,SEGMENT NAME
113736
113736
113736
113736   SUBR REEDUMP,DPREENT,DFPREENT,55DFREENT
113736   INTEGER STRCOM:='S NAME: '
```

```
113743   INTEGER SEGF:='NO SUCH REENTRANT SEGMENT'
113760   INTEGER TABF:='$REENTRANT SUBSYSTEM TABLE IS FULL'
114002   INTEGER NAMAL:='NAME ALREADY USED ON REENTRANT SUBSYSTEM'
114027   INTEGER TXLONGNAME:='$TOO LONG SUBSYSTEM NAME$'
114044   INTEGER PWSBC:="5OPSEG"
114045   DISP -200
114045          INTEGER CHAR1,CHAR2,WORD,NNCHAR,CCSTRING,ROUTSWITCH,CFNO
114045          INTEGER POINTER WCSPNT
114045          INTEGER CSTDINDEX,CSTA=CHAR2,CRSTA=WORD,CSGNO=CFNO
114045          INTEGER CSGNAM=CHAR1
114045   PSID
114045
114045   SYMBOL 5REMXCHAR=23           % MAX 16 CHARS IN SUBSYSTEM NAME
114045
114045   55DFREENT: L=:D; CALL ENTER
114047          T=:CSTDINDEX; A=:OSTRING; D=:OPNT
114052          A:=3; GO FELLS
114054   DFPREENT: L=:D; CALL ENTER
114056          A:=2; GO FELLS
114060   DPREENT: L=:D; CALL ENTER
114062            A:=1; GO FELLS
114064   REEDUMP: L=:D; CALL ENTER
114066          A:=0
114067   FELLS: A=:ROUTSWITCH; D=:CFNO
114071          IF ROUTSWITCH><3 THEN "STRCOM"; CALL GPAR FI
114077          IF REEPNT="REECEND" GO FAR ERRT
114103          -5REMXCAR=:NNCHAR; "3BDUREE"=:"WCSPNT"
114107          DO
114107                CALL OREAD; MIN NNCHAR; GO LAB1; GO FAR ERTLONG
114113   LAB1:        A=:CHAR1 SH 10=:WORD
114116                CALL OREAD; MIN NNCHAR; GO LAB2; GO FAR ERTLONG
114122   LAB2:        A=:CHAR2+WORD=:WORD
114125                WORD=:WCSPNT; MIN "WCSPNT"
114130          WHILE CHAR1><##' AND CHAR2><T
114137          OD; NNCHAR+5REMXCAHR=:NNCHAR; GO L1; *)FILL
114155   L1:    IF ROUTSWITCH=0 OR A=2 THEN
114162             "STSTART"; CALL GPAR; A=:CSTA
114165             "STRESTART"; CALL GPAR; A=:CRSTA
114170          FI
114170          IF ROUTSWITCH=0 THEN
114172             "STRFILE"; CALL GPAR; CALL BPOPEN; GO FAR CFIERR; A=:CFNO
114177             "TSGNAM"; CALL KGPAR; A:=-1; A=:CSGNAM
114203          ELSE IF A=1 THEN
114207             "STRFILE"; CALL GPAR; CALL PROPEN; GO FAR CFIERR; A=:CFNO
114214             "TSGNAM"; CALL KGPAR; A:=-1; A=:CSGNAM
114220          ELSE IF A=2 THEN
114224             "TSGNN"; CALL GPAR; IF A=0 OR A>>SGMAX GO FAR ERRSS
114232             A=:CSGNO*5SEGSIZE+SEGSTART
114235             IF A.FLAG NBIT 5SREEP GO FAR ERRSS
114241          FI; FI; FI; CALL ESCOFF
114242          T:=0; "REESM"; CALL CCBRSRV; D/\0
114246          IF ROUTSWITCH<2 THEN CALL CRSRTL; GO FAR RETU FI
114254          "3BDUREE"=:CSTRING; D=:CPNT; X:="REECOMTAB"; CALL ABLOOK
114261          IF A=0 AND D=0 GO FAR ERRM
114264          IF "REETEND"-REECSPNT<<NNCHAR+1 SHZ -1 GO FAR ERRT
114273          REECSPNT=:"WCSPNT"; X:="3BDUREE"
114276          DO X.S0=:WCSPNT; MIN "WCSPNT"; X+1; T-1; WHILE T><0; OD
114306          GO L2; *)FILL
114341   L2:    IF ROUTSWITCH=0 THEN
114343             CSTA=:REEPNT.S2; CRSTA=:X.S3
```

```
114350              A:=CSGNAM; T:=CFNO; CALL VRTLOAD
114353              IF A=0 GO ERRTL; A=:REEPNT.S1
114356           ELSE IF A=1 THEN
114362              A:=CSGNAM; T:=CFNO; CALL PVRTLOAD; IF A=0 GO ERRTL
114366              T=:REEPNT.S2:=D=:X.S3; A=:X.S1
114373           ELSE IF A=2 THEN
114377              CSGNO=:REEPNT.S1; CSTA=:X.S2; CRSTA=:X.S3
114406           ELSE
114407              -1=:REEPNT.S1; CSTDINDEX=:X.S2
114414           FI; FI; FI
114414   OUT:    -1=:X.S4; REECSPNT=:X.S0; X+4=:REEPNT; "WCSPNT"=:REECSPNT
114424           "PWSBC"; *MON 2WSEG
114426   RETU:   "REESM"; CALL CCBRELEASE; CALL CRLRTL; 0=:CPNT
114432           IF ROUTSWITCH<2 THEN
114436              T:=CFNO; *MON 2CLOS; JMP * 1
114441           FI; GO LEAVE
114442   ERRT:   "TABF"; GO ERRF
114444   ERRSS:  "SEGF"; GO ERRF
114446   ERRTL:  "EDIRTL"; GO ERRF
114450   ERTLONG: "TXLONGNAME"; GO ERRF
114452   ERRM:   "NAMAL"
114453   ERRF:   CALL OUTTEXT; GO RETU
114455   CFIERR: *MON 64
114456           GO RETU
114457   RBUS
114474
114474
114474   %=============================================================================
114474   % 18.21       D L R E E N T   C R E S E G M
114474
114474   % COMMAND: DELETE-REENTRANT <NAME>
114474   %        : CLEAR-REENTRANT-SEGMENT <SEGMENT NAME>
114474
114474   SUBR DLREENT,CRESEGM,CRSRTL,CRLRTL
114474   DISP -200; INTEGER CSEGN,ROUTSWITCH; PSID
114474   INTEGER ERRSS:='$NO SUCH REENTRANT SEGMENT'
114512   INTEGER T1SNCL:='$SEGMENT NUMBER ',T2SNCL:=' IS NOT CLEARED'
114533   INTEGER STRCOM:='N NAME; ',ILLNAM:='NO SUCH NAME'
114547   INTEGER PAREE:=NULL,PWSBC:="50PSEG"
114551   INTEGER TI500:='$ND-500 STANDARD DOMAIN,
114551   USE THE ND-500-MONITOR COMMAND DELETE-STANDARD-DOMAIN'
114621
114621   DISP -200; INTEGER SVCSTRING; PSID
114621
114621   CRSRTL: L=:D; CALL ENTER
114623           T:=1; "RTLFIE"; CALL CCBRSRV; GO ERRRTL; GO LEAV2
114630   ERRRTL: "RTLRMESS"; CALL OUTTEXT; GO LEAVE
114633   CRLRTL: L=:D; CALL ENTER
114635           "RTLFIE"; CALL CCBRELEASE
114637           GO LEAVE
114640   *)FILL
114646
114646   CRESEGM: L=:D; CALL ENTER
114650           1=:ROUTSWITCH; GO FELLS
114653   DLREENT: L=:D; CALL ENTER
114655           0=:ROUTSWITCH
114656   FELLS:  "PAREE"; *MON 2REEN
114660           IF ROUTSWITCH=0 THEN
114662              "STRCOM"; CALL SGPAR; GO ERM64; A=:SVCSTRING
114666              "REESM"; T:=0; CALL CCBRSRV; 0/\0; CALL CRSRTL; GO RETU
```

```
114674              A:=SVCSTRING; X:="REECOMTAB"; CALL ABLOOK; T=:X
114700              IF A><0 THEN "ILLNAM"; CALL OUTTEXT; GO RETU FI
114704              IF X.S1=-1 THEN "TI500"; CALL OUTTEXT; GO RETU FI
114713              X.S1=:CSEGN
114715           ELSE
114716              "TSGNN"; CALL SGPAR; GO ERM64; A=:CSEGN
114722              "REESM"; CALL CCBRSRV; 0/\0; CALL CRSRTL; GO RETU
114727           FI; DO; CSEGN; CALL FREESGN; GO NOMORE; CALL DLRETAB OD
114734   NOMORE: IF CSEGN =0 OR A>>SGMAX GO ERSS
114741           A*5SEGSIZE+SEGSTART; IF A.FLAG NBIT 5SREEP GO ERSS
114747           CSEGN; CALL ZRTLOAD
114751           IF A><0 THEN
114752              "T1SNCL"; CALL OUTTEXT; CSEGN; CALL OCTU; "T2SNCL"; CALL OUTTEXT
114760           FI; "PWSBC"; *MON 2WSEG   % WRITE BACK SEGMENT
114762   RETU:   "REESM"; CALL CCBRELEASE; CALL CRLRTL
114765           GO LEAVE
114766   ERSS:   "ERRSS"; CALL OUTTEXT; GO RETU
114771   ERM64:  *MON 64
114772           GO LEAVE
114773   RBUS
115023
115023
115023   %=========================================================================
115023   %              B P O P E N    P R O P E N
115023   % SUBROUTINE TO OPEN A BPUN FILE OR A PROG FILE
115023   %
115023   % ENTRY:      A=FILE NAME POINTER
115023   % EXIT:       ERROR, A=ERROR CODE
115023   % EXIT+1:     A=OPEN FILE NUMBER
115023   %
115023   SUBR BPOPEN,PROPEN
115023   INTEGER BPTYP:='BPUN',PRTYP:='PROG'
115031   BPOPEN: K:="0"; GO FELLS
115033   PROPEN: K:=1
115034   FELLS: L=:D; CALL ENTER
115036          X:=A; T:=3; IF K THEN "PRTYP" ELSE "BPTYP" FI; *MON 2NOPE
115046          GO LEAVE
115047          GO LEAV2
115050   RBUS
115053
115053
115053   INTEGER TALREADY:='ALREADY IN USE'
115063   INTEGER RESRTL:=("5RTLSEM",NULL,"1")
115066   INTEGER RTLRMESS:='$RT-LOADER ALREADY IN USE'
115103   INTEGER ILLA:='ILLEGAL ADDRESS',ILLP:='ILLEGAL PARAMETER'
115124   INTEGER STSTART:='IOSTART ADDRESS: ',STRESTART:='IORESTART ADDRESS: '
115147   INTEGER STRHIGH:='IOHIGH ADDRESS: ',STRLOW:='IOLOW ADDRESS: '
115170   INTEGER STROFILE:='S OUTPUT FILE: ',STRTERM:='IDTERMINAL NUMBER: '
115212   INTEGER AMBSUSY:='AMBIGUOUS SUBSYSTEM$'
115225   INTEGER FTPPROG:='PROG'
115230   INTEGER GGSGM:='GGSEGMENT (NAME OR NUMBER(OCT)): '
115251   INTEGER DMACNOTLOAD:='NOT LOADED'
115257   *)FILL
115261
115261   SUBR EXHENTMODE
115261   RBUS
115261
115261   %=================================================================
115261   %
115261   %              B A C K G R O U N D   E N T R Y   P O I N T S
```

```
115261   %
115261   %=================================================================
115261   % 17.15        S T S U P E R   O P C O M   O P C O R
115261   %               E R O P C O M   E R R S   T F I L E R R
115261   %
115261
115261   SUBR STSUPER,OPCOM,OPCOR,BLOGOUT,EROPCOM,REMESC,REMRUB,ERRS,TFILERR,OPCFIL,5OPCOR
115261
115261   INTEGER SYS:='SYSTEM ERROR',SYSUN:='SYSTEM UNAVAILABLE$'
115302   INTEGER ABRPAR:=RTREF
115303   INTEGER USBR:='$USER BREAK AT ',MONINST(0); *MON
115314   INTEGER BASEM:=(BATINT,"1")
115316   INTEGER KTEN:=10
115317   % INTEGER UELOS:='UE-LOGIN'
115317   INTEGER 5XUSBR:='$ *** ND-500 USER BREAK ***$'
115336   DISP 0; DOUBLE DDNUM=RERNUM,DDN1=RN1; PSID
115336   DISP 30; INTEGER IDERO=DERO; PSID
115336   *)FILL
115337
115337   STSUPER:
115337           "DEMFIELD"; CALL CCBRELEASE
115341           "MLIDFIELD"; CALL CCBRELEASE
115343           "REESM"; CALL CCBRELEASE
115345           "BFIELD"=:B; "STBEG"=:STPNT; 1=:TDVN
115353           IF BCHFLAG NBIT "0" AND TTIFIELD.BSTATE<5REMOT THEN
115363              "RESIPAR"; *MON 2RESR
115365              "RESOPAR"; *MON 2RESR
115367           FI; CALL ESCOFF; GO L1; *)FILL
115403   L1:     IF "RTLFI".RTRES=RTREF THEN
115410              "KTEN"=:PANAMSR; A:="PANAMSR"; *MON 2WSEG
115414           FI; "BASEM"; *MON 2RELE
115416           0=:CPNT; "COMSTRING"=:CSTRING; 15; CALL CWRITE; 0=:CPNT
115424           "STACK"=:CSTCK
115426           X:=TTIFIELD
115427           IF X.FLAGB BIT 5ABJOB THEN
115432              IF BCHFLAG=1 AND X.MXTIME=77777 THEN
115442                 UEFLG BONE 5UETM
115444              ELSE
115445                 UEFLG BONE 5UEST
115447              FI; A=:UEFLG
115450           FI
115450           IF COBSTATE=5BUSER THEN A+1 ELSE X.BSTATE FI; A=:5COBSTATE; 0=:COBSTATE
115461           T:=5BCOMM=:X.BSTATE; IF A>>11 GO ERRS
115466           T:=X.FLAGB
115467           IF A><0 THEN
115470              IF T BIT 5LOGOUT AND UEFLG BIT 5UETE THEN
115475                 T:=:A; A BZERO 5LOGOUT BZERO 5ABJOB/\ESCMASK=:X.FLAGB
115502                 T BONE 5UEST=:UEFLG; X.BSTATE=:COBSTATE
115506                 "UECMD"; *MON 2CMND
115510                 COBSTATE; 0=:COBSTATE; GO L2; *)FILL
115532              FI; T:=X.FLAGB
115533              IF T BIT 5LOGOUT THEN CALL ABENTRY FI
115536   L2:        IF T BIT 5ABJOB THEN CALL UECHECK; T BZERO 5ABJOB=:X.FLAGB; CALL JABORT FI
115544           FI; T BZERO 5LOGOUT=:X.FLAGB
115546           GOSW DUM,FAR ESCOM,FAR USER,FAR ESC,FAR ERROUT,FAR REMESC,FAR REMRUB,FAR ERRS,FAR ESC,DUM
115561   DUM:    IF BCHFLAG=0 THEN
115563              T:=TTNO; *MON 2CIBU; JMP *1
115566           FI
115566           IF UNAFLAG><0 AND TTNO><LGCOLDSTART AND BCHFLAG>=0 THEN
115576              IF =2 THEN 0=:BCHFLAG FI
```

```
115602          5BPASSIVE=:X.BSTATE
115604          "SYSUN"; CALL OUTTEXT
115606          IF UNEXTRA SHZ -10><##' THEN "UNEXTRA"; CALL OUTTEXT FI
115615          T:=1; A:=1; *MON 2SYCN; JMP * 1      % (BAD)
115621          "ABRPAR"; *MON 2ABOR
115623       FI; O=:COBSTATE=:MSTPN=:CMDFFIELD
115626       1XUSTATUS BZERO 1ILLCONTINUE=:1XUSTATUS
115631       UEFLG; A BZERO 5UECM=:UEFLG
115634       CALL MBZMEMORY                          % ZEROING  BACKGROUND SEGMENT?
115635       IF UEFLG BIT 5UEAL THEN
115640          *MON 2DESC                                                  % LOG IN AS USER "USER-ENVI
115641          CALL UEPRELOGIN                                             % AND START THE UE LOGIN PR
115642       %  "UELOS"; *MON 2CMND
115642       ELSE
115643          IF TTIFIELD.FLAGB BIT 5MLGIN THEN       % CALLED BY MON LOGIN (MON 236)
115647             IF "MLIDFIELD".RTRES="STSIN" THEN
115654                CALL EXHENTMODE; GO OPCOM         % EXECUTE COLD-START MODE FILE
115656             FI; TTIFIELD.FLAGB BONE 5SPSIO=:X.FLAGB
115662          FI; 5LOGIN=:X.BSTATE                    % MARK NOT YET LOGGED IN
115664          CALL LOGIN; 5BCOMM=:X.BSTATE=:INLOGGED
115670          IF TTIFIELD.FLAGB BIT 5MLGIN THEN
115674             X:="USPAR"; "MLIDFIELD"+MLIPASET; T:=12=:D; *MOVB; JMP *
115703             CALL CHSMLGIN; A:=0; CALL MLGRSTART
115706          FI
115706          IF FLLIPCOM BIT "0" THEN          % DEFAULT SUBSYSTEM SHOULD BE STARTED
115711             "LIPBUF"; *MON 2CMND
115713          FI;
115713          IF TTIFIELD.FLAGB NBIT 5MLGIN THEN
115717             *MON 2EESC
115720          FI
115720       FI
115720       CALL ESCON; GO FAR OPCOM; *)FILL
115763 ERROUT: TTIFIELD.DER0=:DDNUM; X.DER2=:DDN1; CALL UECHECK
115771       IF X.FLAGB NBIT 5MLGIN THEN CALL ERSYS(BERR) FI
115776       IF FLBGTERM BIT 1 THEN            % FATAL-ERROR TERMINATION ENABLED
116001             A BZERO 1=:FLBGTERM;        % DISABLE WHILE EXECUTING COMMANDS
116003             CURUSER=:USPAR(0)           % USER NUMBER IN USERP.P1
116006             TTNO=:USPAR(1)              % LOGICAL NUMBER IN .P2
116011             RERNUM=:USPAR(2); CALL EXECC  % ERROR NUMBER IN USPAR.P3
116015             FLBGTERM BONE 1=:FLBGTERM
116020       FI; IF TTIFIELD.FLAGB BIT 5MLGIN GO LOGOUT
116024       GO OPCOR; *)FILL
116034 ESCOM: IF BCHFLAG=0 THEN
116036          T:=TTNO; *MON 2CIBU; JMP *1; MON 2COBU; JMP *1
116043          IF INLOGGED=0 GO FAR DUM
116046          GO FAR USER
116047       FI
116047 ESC:  UEFLG;
116050       IF A BIT 5UECM AND A BIT 5UEMO THEN
116054          IF BCHFLAG=1 OR A=2 THEN CALL UECHECK FI
116064       FI; IF BCHFLAG=1 OR =-1 GO FAR BLOGOUT
116073       IF A=2 THEN
116076          CALL CRLF; O=:BCHFLAG; T:=TTIFIELD.RIFIL; *MON 2CLOS; JMP *1
116104          O=:X.RIFIL; T:=X.DFOPP.ROFIL; O=:X.ROFIL; *MON 2CLOS; JMP *1
116112          DO WHILE MSTPN>0                  % CLEAR MODE STACK
116115             A-2=:MSTPN; AD:=MOSTK(MSTPN); T:=A; *MON 2CLOS; JMP *1
116124             T:=D; *MON 2CLOS; JMP *1
116127          OD; CALL UECHECK
116131          IF FLBGTERM BIT 0 THEN GO MOTHA FI
116135       ELSE
```

```
116136              CALL UECHECK
116137              IF TTIFIELD.FLAGB BIT 5MLGIN GO LOGOUT
116143              T:=TTNO; *MON 2CIBU; JMP *1;MON 2COBU; JMP * 1
116150              IF 5COBSTATE><5ND5ESC THEN                          % ND-100
116154                 "USBR"; CALL OUTTEXT; "ESCBLOCK".RPREG; CALL OCTU
116161              ELSE
116162                 "5XUSBR"; CALL OUTTEXT
116164              FI; 0=:5COBSTATE
116165              IF FLBGTERM BIT 0 THEN                          % USER-BREAK TERMINATION ENABLED
116170   MOTHA:        A BZERO "0"=:FLBGTERM                        % DISABLE WHILE EXECUTING COMMANDS
116172                 CURUSER=:USPAR(0)                            % USER NUMBER IN USPAR.P1
116175                 TTNO=:USPAR(1)                               % LOGICAL NUMBER IN .P2
116200                 -1=:USPAR(2);CALL EXECC                      % FLAG INDICATING USER-BREAK (ELSE ERNO)
116204                 FLBGTERM BONE "0"=:FLBGTERM
116207              FI
116207           FI; GO USER; *)FILL
116227   5OPCOR: "STBEG"=:STPNT                                      % RELEASE ALL DBH'S HELD BY THIS RT-PROGRAM
116231   USER:   CALL RELRT                                          % CLOSE ALL FILES EXCEPT PERMANENTLY OPEN ONES
116232           T:=-1; *MON 2CLOS; MON 2ERMS;                       % CLOSE ALL XMSG "USER" PORTS
116235           T:=1; *MON 2XMSG
116237           CALL RESESC
116240           IF FLQERM><0 THEN                                   % SAVE ERROR INFORMATION
116242               TTIFIELD.DERO=:DDNUM
116245           FI
116245           CALL UECHECK                                        % FATAL ERROR (MON 65)
116246           IF FLQERM><0 AND FLBGTERM BIT 1 THEN                % RESET MON 65 FLAG
116253              A BZERO 1=:FLBGTERM; 0=:FLQERM
116256              CURUSER=:USPAR(0)
116261              TTNO=:USPAR(1)
116264              TTIFIELD.IDERO=:USPAR(2); CALL EXECC
116271              FLBGTERM BONE 1=:FLBGTERM
116274           FI; GO OPCOM
116275   INTEGER POINTER CCUSER:=CUSER     %IN FILE SYSTEM
116276   *)FILL
116307   OPCOR: IF UEFLG BIT 5UECM THEN CALL UECHECK; FI
116313          IF BCHFLAG><0 GO JABORT                              % RELEASE ALL DBH'S HELD BY THIS RT-PROGRAM
116316          CALL RELRT
116317          T:=-1; *MON 2CLOS; MON 2ERMS;                   % CLOSE ALL XMSG "USER" PORTS
116322          T:=1; *MON 2XMSG
116324          CALL RESESC
116325          IF TTIFIELD.FLAGB BIT 5MLGIN GO LOGOUT
116331   OPCOM: IF TTIFIELD.BSTATE=5BPASSIVE GO FAR DUM   % NOT LOGGED IN
116336          "STBEG"=:STPNT; "STACK"=:CSTCK
116342          IF RTREF="RTERR" OR ="STSIN" THEN
116351             "ABRPAR"; *MON 2ABOR
116353          FI
116353          IF UEFLG BIT 5UECM THEN CALL UECHECK; FI
116357          1=:TDVN; CALL CRLF
116362   OPCF:  CALL OPCFIL
116363          "BFIELD"=:B
116365          IF BCHFLAG=0 THEN
116367                IF TTIFIELD.TYPRING BIT 5COM OR A BIT 5BAD THEN
116375                   T:=1; A;=12; *MON 2SYCN; JMP * 1     % (BAD)
116401                   "USTX4"; CALL OUTTEXT
116403                ELSE
116404                   LUSTX; CALL OUTTEXT
116406                FI
116406          FI CALL ESCOFF; CALL GCOM; CALL SINCOM; 0/\0; GO FAR OPCOM
116413          "STBEG"=:STPNT; GO OPCF      %IT WAS A FILE COMMAND
116416   *)FILL
```

```
116446   ERRS:  CALL ERRFATAL
116447   EROPCOM: "STBEG"=:STPNT; IF CCUSER=-1 GO FAR DUM; GO FAR OPCOM
116456   TFILERR: T:=-2=:CCUSER; GO FILERR
116461
116461   % SUBROUTINE TO RELEASE FILE SYSTEM SEMAPHORES:
116461
116461   OPCFIL: "BFIELD"=:B; L=:D; CALL ENTER; SPASTYPE=:PASSTYPE
116467   OPCLOOP: RTREF.BRESLINK
116471          DO WHILE A><RTREF
116474                IF A="DEMFIELD" GO OPCR
116477                IF A>="9SFIS" AND A<"9EFIS" THEN
116505   OPCR:           CALL CCBRELEASE; GO OPCLOOP
116507                FI; A.RESLINK
116511          OD; GO LEAVE
116513   *)FILL
116525
116525   %SUBROUTINE TO CHECK FOR USER ENVIRONMENT
116525
116525   UECHECK: L=:D; CALL ENTER
116527          IF UEFLG NBIT 5UECM AND A NBIT 5UETE THEN GO LEAVE FI
116535          CALL OPCFIL; "STBEG"=:STPNT; "STACK"=:CSTCK
116542          IF UEFLG BIT 5UECM THEN  % RETURN FROM COMMAND CALL
116545                A BZERO 5UECM=:UEFLG; X:=UEXREG; TAD:=UECMRET; D=:P
116552          FI
116552          IF A BIT 5UETE THEN % CALL USER ENVIRONMENT
116554                A BONE 5UEPRUN=:UEFLG; "UECMD"; *MON 2CMND
116560          FI; UEFLG; A BZERO 5UETE;A BZERO 5UECM=:UEFLG
116564          GO OPCOR
116565   *)FILL
116576
116576   REMESC: X:="ESCREM"; GO REMFL
116600   REMRUB: X:="RUBREM"
116601   REMFL: CALL NDNCOMMAND; CALL OUTUSTX; GO OPCOM
116604   RBUS
116611
116611   %==========================================================================
116611   % 17.16.1    C O M S B   S E T O L D   U E C O M S U B   U E L O G I N
116611   %
116611   % SUBROUTINE CALLED FROM THE COMND MONITOR CALL
116611   % A POINTS TO COMMAND STRING; T=SEGMENTS OF CALLING PROGRAM
116611
116611   SUBR COMSB,SETOLD,UECOMSUB,UELOGIN
116611   DISP -200; INTEGER TSEGM,T2SEG,WPNT,INDXX,COMM,APTAB,NPTAB,SVCPNT,CHAR; PSID
116611
116611   INTEGER PM60:=("16")
116612   SETOLD: L=:D; CALL ENTER; A=:WPNT; 0=:COMM; GO CCM
116617   COMSB: L=:D; CALL ENTER; A=:WPNT; 1=:COMM; GO CCM
116625   UECOMSB: L=:D; CALL ENTER; TAD=:UECMRET; X=:UEXREG; A=:WPNT; T=:TSEGM
116633          IF A:=T/\377>="F5DSG" AND A<="L5DSG" THEN A:=4 ELSE A:=2 FI
116646          A=:COMM; UEFLG BONE 5UECM=:UEFLG; GO CCM1
116653   UELOGIN: L=:D; CALL ENTER; TAD=:UECMRET; X=:UEXREG; A=:WPNT; 3=:COMM
116662          UEFLG BONE 5UECM=:UEFLG
116665   CCM:   T=:TSEGM
116666   CCM1:  X:=TTIFIELD
116667          X+"5BREGBLOCK-5REG"
116670          X.OLDPAG SHZ -7/\3=:APTAB; X.OLDPAG SHZ -11/\3=:NPTAB
116700          TSEGM/\377=:T2SEG*5SEGSIZE+SEGSTART=:X; X.LOGADR SHZ -10
116710          IF A=200 AND APTAB><NPTAB THEN T2SEG BONE 17=:T2SEG FI
116722          0=:INDXX
116723          FOR INDXX TO "5WCBUFSIZE-1" DO
```

```
116727                  X:=WPNT; T:=T2SEG; CALL GET1L; GO FAR ERR
116733                  A=:COMSTRING(INDXX); MIN WPNT; 0/\0
116737             OD
116743             FOR X:=0 TO "5CBUFSIZE-2" DO
116747                T:="COMSTRING"; *LBYT
116751             WHILE ><##'
116754             OD; 15; T:="COMSTRING"=:CSTRING; *SBYT
116762             0=:CPNT; GO L1; *)FILL
117002   L1:       IF COMM=3 THEN CALL UUELOGIN; GO FAR L3 FI
117010             IF A=4 THEN              % IF RECOVER THEN RELEASE ND-500 PROCESS
117013                CALL SCAB; IF A=##( GO C5RECOVER
117017                0=:CPNT; X:=COMTAB; CALL ABLOOK; T=:X
117023                IF A=-1 GO C5RECOVER
117026                IF A=0 AND X.S0="CNO" GO C5RECOVER
117033                GO L2
117034
117034   C5RECOVER: IF A="CNO" THEN CALL SCAB ELSE 0=:CPNT FI
117042                X:=CPNT=:SVCPNT; T:=CSTRING; *LBYT
117046                IF A><##( THEN                          % MAY BE REENTRANT SUBSYSTEM
117051                   X:="REECO"; CALL ABLOOK
117053                   IF A=0 GO OK                         % IT IS REENTRANT SUSBSYSTEM
117054                   IF A=-2 THEN CALL CRLF; "AMBSUSY"; CALL OUTTEXT; GO L3 FI
117063                FI                                      % MAY BE A PROG FILE
117063                "WORKA"=:OSTRING; SVCPNT=:CPNT; 0=:OPNT
117070                DO
117070                   CALL CREAD; A=:CHAR; CALL SRCHINT(DELIM)
117074                WHILE A>=3
117077                   CHAR; CALL OWRITE
117101                OD; CALL SCAB; ##'; CALL OWRITE
117105                X:="WORKA"; T:=3; "FTPPROG"; *MON 2NOPE
117111                GO ERROP
117112                T:=A; *MON 2CLOS; JMP *+1
117115   OK:          0=:CPNT
117116                "PM60"; *MON 60; JMP *1
117121                GO L2; *)FILL
117140             FI
117140             IF A><0 THEN
117141   L2:          0=:CPNT; CALL SINCOM; 0/\0; 0/\0
117145   LOOP:        RTREF.BRESLINK
117147                DO WHILE A><RTREF
117152                   IF A>="9SFIS" AND A<"9EFIS" THEN CALL CCBRELEASE; GO LOOP FI
117162                   A.RESLINK
117164                OD
117165             FI
117165   L3:       UEFLG BZERO 5UECM=:UEFLG; T:=TSEGM; GO LEAVE
117172   ERR:      "ILLA"; CALL OUTTEXT; GO FAR OPCOM
117175   ERROP:    A=:X:=##"; CALL TCO; "WORKA"; CALL OUTTEXT; ##"; CALL TCO
117204             A:=X; *MON 64
117206             GO L3
117207   RBUS
117221
117221
117221   %==========================================================================
117221   % 17.16.2     I O U T
117221   %
117221   % MONITOR CALL TO PRINT OCTAL OR DECIMAL NUMBER
117221   % A=VALUE, T=8 OR 10
117221
117221   SUBR IOUT
117221   DISP -200; INTEGER LREG; PSID
```

```
117221   IOUT:  X:=B; "BFIELD"=:B; A:=L=:LREG
117226          IF X.ZTREG-10=0 THEN X.ZAREG; CALL OCTU
117233          ELSE IF A-2=0 THEN X.ZAREG; CALL DECU FI
117240          FI; LREG=:L; B:=X; TAD:=ZTADREG; X:=ZXREG; EXIT
117246   RBUS
117251
117251
117251   %==========================================================================
117251   % 17.16.3     R S I O
117251   %
117251   %  MONITOR CALL TO GET I/O DEVICE NUMBERS
117251
117251
117251   SUBR RSIO
117251   INTEGER POINTER PCURUSER:=CURUSER
117252   RSIO: B=:D; "BFIELD"=:B
117255          IF BCHFLAG=0 THEN
117257             A:=TTNO=:T
117261          ELSE
117262             T:=TTIFIELD.RIFIL; A:=X.DFOPP.ROFIL
117266          FI; X:=BCHFLAG; D=:B
117270          A=:D; A:=X; X:=PCURUSER; EXIT
117274   RBUS
117275
117275
117275   %==================================================================
117275   % 17.16.4     G B G S Z
117275   %
117275   %  MONITOR CALL TO GET SIZE OF BACKGROUND SEGMENT
117275
117275   SUBR GBGSZ
117275   GBGSZ: B=:T; "BFIELD"=:B
117300          BCSEGM/\377*5SEGSIZE+SEGSTART=:B
117305          LOGADR SHZ -10; T=:B; T:=ZTREG; EXIT
117312   RBUS
117316
117316
117316   %==========================================================================
117316   %             M B E C H O
117316   % MONITOR CALL TO CONTROL INPUT/OUTPUT IN BATCH
117316   % ZAREG=VALUE TO SET TO BMECHO
117316
117316   SUBR MBECHO
117316   MBECHO: X:="BFIELD":=:B=:D
117321          IF X.ZAREG=-1 THEN BMECHO=:X.ZAREG; GO OUT FI
117330          IF BCHFLAG=1 THEN
117334             X.ZAREG/\1=:BMECHO
117337          ELSE IF A=2 THEN
117343             X.ZAREG/\7=:BMECHOD
117346             IF A BIT BITERM AND TTIFIELD.RIFIL=TTNO THEN
117355                BMECHO BZERO BITERM=:BMECHO
117360             FI
117360             IF BMECHO BIT BOTTERM AND TTIFIELD.DFOPP.ROFIL=TTNO THEN
117371                BMECHO BZERO BOTTERM=:BMECHO
117374             FI
117374          FI; FI
117374   OUT:   D=:B; TAD:=ZTADR; X:=ZXREG; EXIT
117400   RBUS
117404
117404
```

```
117404
117404
117404   % DEFINE FILESYSTEM COMMAND ENTRY POINTS
117404   @ICR
117404   SUBR CRDIR,RNDIR,ENDIR,RLDIR,SDDIR,LIDIR,DUDIR,CHDIR,CRUSE,DLUSE,
117404          RNUSE,GIUSE,TAUSE,LIUSE,DUUSE,CHUSE,CPASS,CLPASS,
117404          CRFRI,DLFRI,SFRIA,LIFRI,CRFIL,CRNVE,ALFIL,ALNVE,
117404          EXFIL,DLFIL,RNFIL,STERF,SPERF,SFILA,LIFIL,DUOBJ,
117404          CHOBJ,DIRST,USEST,FILST,OPENF,CONNF,CLOSF,LIOPF,
117404          SBLOS,SPERO,SBYTP,SBLOC,RESFI,RELFI,WHEFI,OPRTF,
117404          CORTF,CLRTF,OPENS,DUPAG,CHPAG,DUBIT,CHBIT,TESDI,
117404          REGDI,COPFI,RESTU,SAVDI,RELTU,LIRTO,STARS,SMDIR,
117404          STSPL,ABORS,LSPOQ,APPES,DELES,GIVES,TAKES,SPOPL,RESRS;
117404   RBUS;
117404   SUBR BERR,RERR,XERPRINT,SERTERR,ERLIST;
117404   @CR;
117404   RBUS
117404
117404   %==============================================================================
117404   %
117404   %              C O M M A N D   P R O C E S S O R
117404   %
117404   %==============================================================================
117404   % 17.17       S I N C O M
117404   %
117404
117404   % SUBROUTINE TO PROCESS A COMMAND IN THE COMMAND BUFFER
117404   % SKIP RETURN IF FILE SYSTEM COMMAND
117404
117404   SUBR SINCOM
117404
117404   DISP -200
117404          INTEGER POINTER FUNC; INTEGER ARRAY POINTER EACHPAR
117404          INTEGER CMELEM
117404   PSID
117404
117404   INTEGER ERRAB:='ERROR, RT-PROG.=0'
117415   INTEGER TNNDNET:='$ND-NET NOT IMPLEMENTED'
117431   INTEGER TCMNIMP:='$COMMAND NOT IMPLEMENTED'
117446   SINCOM: L=:D; CALL ENTER
117450          CALL SCAB; IF =15 GO FAR OPCOM; 0=:CPNT; IF =##( THEN CALL RECFILE FI
117461          X:=COMTAB;CALL ABLOOK; T=:X=:CMELEM
117465          IF =-1 THEN 0=:CPNT; X:="COMSTRING"
117472   RECO:        CALL RECOVER; GO LEAV2          %IF NOT COMMAND
117474          ELSE IF ><0 GO FAR ERRA FI
117477          IF X.CPROTECT>PASSTYPE GO FAR ERRC
117503          X.CMAND=:"FUNC"
117505          IF A=0 THEN
117506              IF X.CPARAM=-4 AND MXLIN=0 THEN "TNNDNET" ELSE "TCMNIMP" FI
117517              GO FAR ERR
117520          FI
117520          IF X.CPARAM><0 AND A><-3 AND A><-4 THEN
117530             X.CPARAM=:"EACHPAR"
117532             FOR X:=0 TO 5 DO WHILE EACHPAR(X)><0
117540                CALL SGPAR; GO FAR CERR; A=:ACTPAR(X)
117543             OD
117545          FI
117545          IF CMELEM.CMAND/\177400=0 THEN
117551             A:=153000+"FUNC"=:T; "CPLIST"; *EXR ST
117556          ELSE
```

```
117557              IF X.CPROTECT=-2 THEN X:="FUNC"; CALL SFILSYS(CMMON); GO FAR FFERR; GO L3EAVE FI    % FILSYS SPO
117570              IF A=-1 THEN X:="FUNC"; CALL FILSYS(CMMON); GO FAR FFERR; GO L3EAVE FI     % FILSYSTEM SEGMENT.
117600              IF X.CPARAM=-3 THEN X:="FUNC"; CALL SMSGCOMMAND; GO LEAV2; FI
117607              IF A=-4 THEN                    % ND-NET COMMAND
117612                 X:="FUNC"; CALL NDNCOMMAND; GO LEAV2
117615              FI; CALL FUNC
117616           FI
117616   OUT:    GO LEAV2
117617
117617   *)FILL
117647   CERR:   *MON 64
117650           GO LEAV2
117651
117651   % ERROR RETURN FROM A FILESYSTEM COMMAND
117651
117651   FFERR: CALL EOPCAL(ERLIST); GO L3EAVE
117654
117654   % SPECIAL CHECK FOR ABORT:
117654   CCABORT: IF ACTPAR(0)=0 GO ERRB; "CPLIST"; *MON 2ABOR
117661           GO LEAV2
117662
117662   % SPECIAL ENTRY FOR HOLD:
117662   HHOLD: CALL ESCON; "CPLIST"; *MON 2HOLD
117665           CALL ESCOFF; GO LEAV2
117667
117667   % SPECIAL ENTRY FOR UPDAT:
117667   UPDA:   IF ERUCL=0 AND PASSTYPE=0 GO ERRC  %PUBLIC ACCESS ONLY ONCE
117673           0=:ERUCL
117674           "CPLIST"; *MON 2UPDA
117676           GO LEAV2
117677
117677   % SPECIAL ENTRY FOR ENTSG:
117677   EENTSG: "CPLIST"; *MON 2ENTS
117701           IF <0 GO ERRAP; GO LEAV2
117703
117703   % ERROR EXITS:
117703   ERRAP: ER109; GO CERR                    % ILLEGAL PARAMETER
117705   ERRA:  ER210; GO CERR                    % AMBIGUOUS COMMAND
117707   ERRB:  "ERRAB"; GO ERR
117711   ERRC:  ER211; GO CERR                    % PROTECTED COMMAND
117713   ERR:   CALL OUTTEXT; GO LEAVE
117715   *)FILL
117727   @ICR
117727
117727   %==========================================================================
117727   % 17.18       3 C O M T A B
117727   %
117727   INTEGER ARRAY 3COMTAB:=(
117727   CN0,0,RECOVER,0,    CN1,0,CONTINUE,0,
117737   CN2,0,GOTOUSER,0,   CN3,0,DUMP,0,
117747   CN4,0,LOAD,0,       CN5,0,PLACE,0,
117757   CN6,0,LOGOUT,0,     CN7,-3,LOOKAT,0,
117767   CN8,0,CSTATUS,0,
117773                       CN13,0,DATCL,0,
117777   CN14,0,XRTLOAD,1,   CN15,0,COPYF,0,
120007   CN16,0,MEMORY,0,    CN17,0,STOPSYS,2,
120017   CN18,0,LISTTQ,0,    CN19,0,LISTXQ,0,
120027   CN50,0,LISTSEG,0,   CN51,-3,LISTRT,0,
120037   CN52,0,BATCH,1,     CN53,0,APPBATCH,0,
120047   CN54,0,ABBA,1,      CN55,0,ABJOB,0,
```

```
120057   CN56,0,LIBAT,0,      CN57,0,MODE,0,
120067   CN60,-3,TERMSTAT,0,CN61,0,STOPTERM,1,
120077   CN62,0,XMACD,1,      CN63,0,SCEDULE,0,
120107   CN64,0,ENTRT,1,      CN65,0,WHOISON,0,
120117   CN67,0,CTIMUS,0,     CN68,0,INITACC,2,
120127   CN69,0,CACCO,2,      CN70,0,SACCO,2,
120137   CN71,0,LIBQUE,0,     CN66,0,GETRN,1,
120147   CN72,-3,TAPEFU,0,
120153                        CN75,0,CCDUM,0,
120157   CN76,0,SETUN,2,      CN77,0,SETAV,2,
120167   CN78,0,SETERR,2,     CN79,0,INCOM,2,
120177   CN80,0,HELP,0,       CN81,CP28,HHOLD,0,
120207   CN82,0,TERMODE,0,    CN83,0,PRESYS,2,
120217   CN83,0,RESYS,2,      CN17,0,STOPSYS,2,
120227   CN84,0,CCDUM,0,       CN85,0,REEDUMP,2,
120237   CN86,0,LIREENT,0,    CN87,0,DLREENT,2,
120247   CN88,0,CMRFILE,0,    CN89,0,CMWFILE,0,
120257   CN90,0,OPERATOR,0,   CN91,0,WFOPERATOR,0,
120267   CN92,0,REUSER,1,     CN93,0,LOGSYST,1,
120277   CN94,0,DEFHISTO,0,   CN95,0,STAHIST,0,
120307   CN96,0,STOHIST,0,    CN97,0,PRHIST,0,
120317                        CN99,0,EXECIOX,1,
120323   CN100,-4,REMOTE,0,     CN101,-4,LOCAL,0,
120333   CN102,-4,STACOM,1,     CN103,-4,STOCOM,1,
120343   CN104,-4,LREM,1,       CN105,0,APREB,0,
120353   CN106,0,LIREQU,0,     CN107,-4,CMSTAT,0,
120363   CN108,0,DELBE,0,      CN109,0,DELRE,0,
120373   CN110,-4,REPAS,2,      CN111,-4,CLSTA,0,
120403   CN120,0,SERVSINT,2,CN121,0,NEXINCOM,2,
120413   CN122,0,LIDEV,0,     CN123,0,LRTPROG,1,
120423   CN124,0,SETMEM,0,    CN125,0,INBRPROG,2,
120433   CN126,0,CDVCOM,2,    CN127,-3,LDVFCOM,0,
120443   CN128,0,MAIL,0,      CN129,0,ERINIT,2,
120453   CN130,0,ERPRINT,2,   CN131,0,LIVERS,0,
120463                        CN133,0,LIICOM,2,
120467   CN134,0,CSTTYP,0,    CN135,0,CGTTYP,0,
120477   CN136,0,CDSPFMESS,0,CN137,0,CDESCFU,0,
120507   CN140,0,CEESCFU,0,   CN141,0,DEFESC,2,
120517   CN142,0,CSBSIZE,2,   CN143,0,DEFSHIST,2,
120527   CN144,0,STAPLOG,2,   CN145,0,STOPLOG,2,
120537   CN146,0,GETERROR,2,
120543                         CN149,0,COPCOM,2,
120547   CN150,0,DEFTERM,2,    CN151,0,LITERM,0,
120557   CN152,0,ENTRM,0,      CN153,0,DITRM,0,
120567   CN154,0,PASET,0,      CN155,0,NTERM,2,
120577   CN156,0,DEFLIP,2,     CN157,0,LISLIP,0,
120607   CN158,0,ENLIP,0,      CN159,0,DILIP,0,
120617   CN160,0,DEFLOC,2,     CN161,-3,XBADM,0,
120627   CN162,-3,STABA,2,     CN163,-3,STOBA,2,
120637   CN164,-3,SRMMO,0,     CN165,-3,SLCMO,0,
120647   CN166,0,CCOLDSTA,2,   CN166,0,CCOLDSTA,2,
120657                         CN167,0,MEMLIM,0,
120663   CN168,0,PLPROG,0,     CN169,0,DPREENT,2,
120673   CN170,0,DFPREENT,2,   CN171,0,LRESEG,2,
120703   CN172,0,CRESEG,2,     CN173,0,CAON,0,
120713   CN174,0,CAOFF,0,
120717
120717   CN20,CP20,2RT,1,     CN21,CP21,2SET,1,
120727   CN22,CP22,2ABSET,1,CN23,CP21,2INTV,1,
120737   CN24,CP24,2CONCT,1,CN25,CP20,2DSCNT,1,
```

```
120747   CN26,CP26,2PRIOR,1,CN27,CP27,UPDA,0,
120757   CN28,CP28,2CLADJ,1,CN29,CP29,2FIX,1,
120767   CN30,CP29,2UNFIX,1,CN31,CP20,CCABORT,1,
120777   CN32,CP32,2PRSRV,1,CN33,CP33,2PRLS,1,
121007   CN34,CP20,2RTON,1, CN35,CP20,2RTOF,1,
121017   CN36,CP36,2IOSE,0, CN37,CP37,EENTSG,1,
121027   CN38,CP38,2FIXC,1,
121033
121033   CN200,0,CRDIR,-2,  CN201,0,RNDIR,-2,
121043   CN202,0,ENDIR,-2,  CN203,0,RLDIR,-2,
121053   CN204,0,SDDIR,-2,  CN205,0,LIDIR,-2,
121063   CN206,0,DUDIR,-2,  CN207,0,CHDIR,-2,
121073   CN208,0,CRUSE,-2,  CN209,0,DLUSE,-2,
121103   CN210,0,RNUSE,-2,  CN211,0,GIUSE,-2,
121113   CN212,0,TAUSE,-2,  CN213,0,LIUSE,-2,
121123   CN214,0,DUUSE,-2,  CN215,0,CHUSE,-2,
121133   CN216,0,CHANP,-2,  CN217,0,CLPAS,-2,
121143   CN218,0,CRFRI,-2,  CN219,0,DLFRI,-2,
121153   CN220,0,SFRIA,-2,  CN221,0,LIFRI,-2,
121163   CN222,0,CRFIL,-1,  CN223,0,CRNVE,-1,
121173   CN224,0,ALFIL,-1,  CN225,0,ALNVE,-1,
121203   CN226,0,EXFIL,-1,  CN227,0,DLFIL,-1,
121213   CN228,0,RNFIL,-1,  CN229,0,STERF,-1,
121223   CN230,0,SPERF,-1,  CN231,0,SFLAC,-1,
121233   CN232,0,LIFIL,-1,  CN233,0,DUOBJ,-2,
121243   CN234,0,CHOBJ,-2,  CN235,0,DIRST,-2,
121253   CN236,0,USEST,-2,  CN237,0,FILST,-1,
121263   CN238,0,OPENF,-1,  CN239,0,CONNF,-1,
121273   CN240,0,CLOSF,-1,  CN241,0,LIOPF,-1,
121303   CN242,0,SBLOS,-1,  CN243,0,SPERO,-1,
121313   CN244,0,SBYTP,-1,  CN245,0,SBLOC,-1,
121323   CN246,0,RESFI,-2,  CN247,0,RELFI,-1,
121333   CN248,0,WHEFI,-1,  CN249,0,OPRTF,-1,
121343   CN250,0,CORTF,-1,  CN251,0,CLRTF,-1,
121353   CN252,0,OPENS,-1,  CN253,0,DUPAG,-2,
121363   CN254,0,CHPAG,-2,  CN255,0,DUBIT,-2,
121373   CN256,0,CHBIT,-2,  CN257,0,TESDI,-2,
121403   CN258,0,REGDI,-2,  CN259,0,COPDI,-2,
121413                      CN261,0,COPFI,-1,
121417                      CN263,0,RESTU,-2,
121423   CN264,0,RELTU,-2,  CN265,0,SAVDI,-2,
121433   CN266,0,SAVDI,-2,
121437   CN268,0,LIRTO,-1,  CN269,0,STARS,-2,
121447   CN270,0,STSPL,-2,  CN271,0,ABORS,-2,
121457   CN272,0,LSPOQ,-2,  CN273,0,APPES,-2,
121467   CN274,0,DELES,-2,  CN275,0,GIVES,-2,
121477   CN276,0,TAKES,-2,  CN277,0,SPOPL,-2,
121507   CN278,0,RESTS,-2,  CN279,0,SDFIA,-2,
121517   CN280,0,DEUFI,-1,
121523
121523   CN284,0,STMPF,-1,
121527
121527   CN285,0,SNSPCOPY,-2,CN286,0,FWSPRINT,-2,
121537   CN287,0,BSPRINT,-2, CN288,0,DSCOND,-2,
121547   CN289,0,STOPR,-2,   CN290,0,STAPR,-2,
121557   CN291,0,MSPQE,-2,   CN292,0,RMSPF,-2,
121567   CN293,0,SSFORM,-2,  CN294,0,LSFORM,-2,
121577   CN295,0,RSTAC,1,    CN296,0,RSTOC,1,
121607   CN297,0,LIRTL,1,    CN298,0,SMDIR,-2,
121617   CN299,0,CREDI,-2,   CN300,0,CRLDI,-2,
```

```
121627   CN301,0,SDRUI,-1,    CN302,0,RDRUI,-1,
121637   CN303,0,RSFAC,-2,    CN304,0,RLFAC,-2,
121647   CN305,0,RTSFA,-2,    CN306,0,RTLFA,-2,
121657   CN307,0,CDDIR,-2,    CN308,0,CMDIR,-2,
121667   CN309,-3,UELGN,2,    CN310,0,CLEBA,2,
121677
121677   CNDUM,0,CCDUM,0,    CNDUM,0,CCDUM,0,
121707   CNDUM,0,CCDUM,0,    CNDUM,0,CCDUM,0,-1),
121720
121720
121720
121720   %==========================================================================
121720   % 17.19        C O M M A N D   N A M E S
121720   %
121720   CN307:='CLEAR-DEFAULT-DIRECTORY', CN308:='CLEAR-MAIN-DIRECTORY',
121747   CN309:='UE-AUTOMATIC-LOGIN',      CN310:='CLEAR-BATCH-QUEUE',
121772   CN20:='RT',                       CN21:='SET',                      CN22:='ABSET',
122001   CN0:='RECOVER',                   CN1:='CONTINUE',                  CN2:='GOTO-USER',
122017   CN3:='DUMP',                      CN4:='LOAD-BINARY',               CN5:='PLACE-BINARY',
122037   CN6:='LOGOUT',                    CN7:='LOOK-AT',                   CN8:='STATUS',
122053   CN13:='DATCL',                    CN14:='RT-LOADER',                CN15:='COPY',
122066   CN16:='MEMORY',                   CN17:='STOP-SYSTEM',              CN18:='LIST-TIME-QUEUE',
122110   CN19:='LIST-EXECUTION-QUEUE',     CN50:='LIST-SEGMENT',             CN51:='LIST-RT-DESCRIPTION',
122144   CN23:='INTV',                     CN24:='CONCT',                    CN25:='DSCNT',
122155   CN26:='PRIOR',                    CN27:='UPDAT',
122163   CN28:='CLADJ',      CN29:='FIX',        CN30:='UNFIX',                CN31:='ABORT',
122176   CN32:='PRSRV',      CN33:='PRLS',       CN34:='RTON',                 CN35:='RTOFF',
122212   CN36:='IOSET',      CN37:='ENTSG',      CN38:='FIXC',
122223   CN52:='BATCH',      CN53:='APPEND-BATCH',                             CN54:='ABORT-BATCH',
122243   CN55:='ABORT-JOB',  CN56:='LIST-BATCH-PROCESS',                       CN57:='MODE',
122265   CN63:='SCHEDULE',   CN67:='TIME-USED',  CN68:='INIT-ACCOUNTING',
122307   CN69:='START-ACCOUNTING',              CN70:='STOP-ACCOUNTING',      CN71:='LIST-BATCH-QUEUE',
122341   CN72:='DEVICE-FUNCTION',
122351   CN75:='CC',                            CN76:='SET-UNAVAILABLE',
122363   CN77:='SET-AVAILABLE',                 CN78:='SET-ERROR-DEVICE',
122403   CN79:='INITIAL-COMMAND',               CN80:='HELP',                 CN81:='HOLD',
122421   CN82:='TERMINAL-MODE',                 CN83:='RESTART-SYSTEM',       CN84:='ENTER',
122443   CN85:='DUMP-REENTRANT',                CN86:='LIST-REENTRANT',       CN87:='DELETE-REENTRANT',
122474   CN88:='RFILE',                         CN89:='WFILE',                CN90:='OPERATOR',
122507   CN91:='WAIT-FOR-OPERATOR',             CN92:='RESTART-USER',         CN93:='RT-PROGRAM-LOG',
122537   CN94:='DEFINE-HISTOGRAM',              CN95:='START-HISTOGRAM',      CN96:='STOP-HISTOGRAM',
122570   CN97:='PRINT-HISTOGRAM',
122600   CN99:='EXECUTE-IOX',
122606   CN100:='REMOTE',    CN101:='LOCAL',    CN102:='START-COMMUNICATION',
122627   CN103:='STOP-COMMUNICATION',           CN104:='REMOTE-LOAD',
122647   CN105:='APPEND-REMOTE',                CN106:='LIST-REMOTE-QUEUE',CN107:='COMMUNICATION-STATUS',
122702   CN108:='DELETE-BATCH-QUEUE-ENTRY',     CN109:='DELETE-REMOTE-QUEUE-ENTRY',
122734   CN110:='REMOTE-PASSWORD',              CN111:='COMMUNICATION-LINE-STATUS',
122761
122761   CN120:='SINTRAN-SERVICE-PROGRAM',CN121:='NEXT-INITIAL-COMMAND',
123010   CN122:='LIST-DEVICE',CN123:='LIST-RT-PROGRAMS',CN124:='SET-MEMORY-CONTENTS',
123041   CN125:='INITIALIZE-BACKGROUND-PROGRAMS',CN126:='CLEAR-DEVICE',
123070   CN127:='LIST-DEVICE-FUNCTIONS',CN128:='MAIL',
123106   CN129:='INITIALIZE-ERROR-LOG',CN130:='PRINT-ERROR-LOG',
123131   CN131:='LIST-TITLE',
123137   CN133:='LIST-INITIAL-COMMANDS',CN134:='SET-TERMINAL-TYPE',
123163   CN135:='GET-TERMINAL-TYPE',CN136:='DEFINE-SPOOLING-FILE-MESSAGE',
123213   CN137:='DISABLE-ESCAPE-FUNCTION',CN140:='ENABLE-ESCAPE-FUNCTION',
123243   CN141:='DEFINE-ESCAPE-CHARACTER',
123257   CN142:='CHANGE-BACKGROUND-SEGMENT-SIZE',
```

```
123277   CN143:='DEFINE-SYSTEM-HISTOGRAM',
123313   CN144:='START-PROGRAM-LOG',CN145:='STOP-PROGRAM-LOG',
123335   CN146:='GET-ERROR-DEVICE',CN147:='GET-ALTERNATIVE-RT-LOADER',
123363   CN148:='MAKE-ALTERNATIVE-RT-LOADER',
123401   CN149:='OPCOM',
123404   CN150:='DEFINE-TERMINATION-HANDLING',
123422   CN151:='LIST-TERMINATION-HANDLING',
123437   CN152:='ENABLE-TERMINATION-HANDLING',
123455   CN153:='DISABLE-TERMINATION-HANDLING',
123474   CN154:='SET-USER-PARAMETERS',CN155:='NEXT-TERMINATION-COMMAND',
123523   CN156:='DEFINE-DEFAULT-SUBSYSTEM',CN157:='LIST-DEFAULT-SUBSYSTEM',
123554   CN158:='DEFAULT-SUBSYSTEM-ENABLE',CN159:='DEFAULT-SUBSYSTEM-DISABLE',
123606   CN160:='DEFINE-LOCAL-CHARACTER',CN161:='TADADM',
123626   CN162:='START-TADADM',CN163:='STOP-TADADM',
123643   CN164:='SET-REMOTE-MODE',CN165:='SET-LOCAL-MODE',
123663   CN166:='COLD-START',CN167:='MEMORY-LIMITS',
123700   CN168:='PLACE-PROGRAM',CN169:='DUMP-PROGRAM-REENTRANT',
123723   CN170:='DEFINE-REENTRANT-PROGRAM',CN171:='LOAD-REENTRANT-SEGMENT',
123754   CN172:='CLEAR-REENTRANT-SEGMENT',CN173:='ALTON',
123773   CN174:='ALTOFF',
123777
123777   CN60:='TERMINAL-STATUS',CN61:='STOP-TERMINAL',CN62:='DMAC',
124021   CN64:='RTENTER',CN65:='WHO-IS-ON',CN66:='GET-RT-NAME',
124040
124040   CN200:='CREATE-DIRECTORY',CN201:='RENAME-DIRECTORY',
124062   CN202:='ENTER-DIRECTORY',CN203:='RELEASE-DIRECTORY',
124103   CN204:='SET-DEFAULT-DIRECTORY',CN205:='LIST-DIRECTORIES-ENTERED',
124133   CN206:='DUMP-DIRECTORY-ENTRY',CN207:='CHANGE-DIRECTORY-ENTRY',
124162   CN208:='CREATE-USER',CN209:='DELETE-USER',CN210:='RENAME-USER',
124204   CN211:='GIVE-USER-SPACE',CN212:='TAKE-USER-SPACE',
124224   CN213:='LIST-USERS',CN214:='DUMP-USER-ENTRY',CN215:='CHANGE-USER-ENTRY',
124253   CN216:='CHANGE-PASSWORD',CN217:='CLEAR-PASSWORD',CN218:='CREATE-FRIEND',
124302   CN219:='DELETE-FRIEND',CN220:='SET-FRIEND-ACCESS',CN221:='LIST-FRIENDS',
124331   CN222:='CREATE-FILE',CN223:='CREATE-NEW-VERSION',CN224:='ALLOCATE-FILE',
124360   CN225:='ALLOCATE-NEW-VERSION',CN226:='EXPAND-FILE',CN227:='DELETE-FILE',
124407   CN228:='RENAME-FILE',CN229:='SET-TERMINAL-FILE',
124426   CN230:='SET-PERIPHERAL-FILE',CN231:='SET-FILE-ACCESS',CN232:='LIST-FILES',
124456   CN233:='DUMP-OBJECT-ENTRY',CN234:='CHANGE-OBJECT-ENTRY',
124501   CN235:='DIRECTORY-STATISTICS',CN236:='USER-STATISTICS',
124524   CN237:='FILE-STATISTICS',CN238:='OPEN-FILE',CN239:='CONNECT-FILE',
124550   CN240:='CLOSE-FILE',CN241:='LIST-OPEN-FILES',CN242:='SET-BLOCK-SIZE',
124576   CN243:='SET-PERMANENT-OPEN',CN244:='SET-BYTE-POINTER',
124621   CN245:='SET-BLOCK-POINTER',CN246:='RESERVE-FILE',CN247:='RELEASE-FILE',
124650   CN248:='WHERE-IS-FILE',CN249:='RTOPEN-FILE',CN250:='RTCONNECT-FILE',
124675   CN251:='RTCLOSE-FILE',CN252:='SCRATCH-OPEN',CN253:='DUMP-PAGE',
124720   CN254:='CHANGE-PAGE',CN255:='DUMP-BIT-FILE',CN256:='CHANGE-BIT-FILE',
124745   CN257:='TEST-DIRECTORY',CN258:='REGENERATE-DIRECTORY',
124770   CN259:='COPY-DIRECTORY',CN260:='*BACKUP-DIRECTORY',CN261:='COPY-FILE',
125016   CN262:='*RETRIEVE-DIRECTORY',CN263:='RESERVE-DEVICE-UNIT',
125042   CN264:='RELEASE-DEVICE-UNIT',CN265:='SAVE-DIRECTORY',CN266:='COPY-DEVICE',
125072   CN268:='LIST-RTOPEN-FILES',CN269:='START-SPOOLING',
125113   CN270:='STOP-SPOOLING',CN271:='ABORT-PRINT',CN272:='LIST-SPOOLING-QUEUE',
125142   CN273:='APPEND-SPOOLING-FILE',CN274:='DELETE-SPOOLING-FILE',
125170   CN275:='GIVE-SPOOLING-PAGES',CN276:='TAKE-SPOOLING-PAGES',
125214   CN277:='SPOOLING-PAGES-LEFT',CN278:='RESTART-PRINT',
125235   CN279:='SET-DEFAULT-FILE-ACCESS',CN280:='DELETE-USERS-FILES',
125263   CN281:='*CREATE-VOLUME',CN282:='*LIST-VOLUME',CN283:='*COPY-USERS-FILES',
125313   CN284:='SET-TEMPORARY-FILE',
125325   CN285:='SET-NUMBER-OF-PRINT-COPIES',CN286:='FORWARD-SPACE-PRINT',
125355   CN287:='BACKSPACE-PRINT',CN288:='DEFINE-SPOOLING-CONDITIONS',
```

```
125403   CN289:='STOP-PRINT',CN290:='START-PRINT',
125417   CN291:='MOVE-SPOOLING-QUEUE-ENTRY',CN292:='REMOVE-FROM-SPOOLING-QUEUE',
125452   CN293:='SET-SPOOLING-FORM',CN294:='LIST-SPOOLING-FORM',
125475   CN295:='START-RT-ACCOUNT',CN296:='STOP-RT-ACCOUNT',
125516   CN297:='LIST-RT-ACCOUNT',CN298:='SET-MAIN-DIRECTORY',
125540   CN299:='RESERVE-DIRECTORY',CN300:='UNRESERVE-DIRECTORY',
125563   CN301:='SET-DEFAULT-REMOTE-SYSTEM',CN302:='RESET-DEFAULT-REMOTE-SYSTEM',
125616   CN303:='RESERVE-OPEN-FILE-ENTRIES',CN304:='RELEASE-OPEN-FILE-ENTRIES',
125650   CN305:='RTRESERVE-OPEN-FILE-ENTRIES',CN306:='RTRELEASE-OPEN-FILE-ENTRIES',
125704
125704   CNDUM:='',
125705
125705   CP20:=(STRRT,0),CP21:=(STRRT,STNOU,STTIM,0),
125713   CP22:=(STRRT,STSEC,STMIN,STHOUR,0),CP24:=(STRRT,STLOG,0),
125723   CP26:=(STRRT,STPRI,0),CP27:=(STMIN,STHOUR,STDAY,STMONTH,STYEAR,0),
125734   CP28:=(STNOU,STTIM,0),CP29:=(GGSGM,0),
125741   CP32:=(STLOG,STIOFL,STRRT,0),CP33:=(STLOG,STIOFL,0),
125750   CP36:=(STLOG,STIOFL,STRRT,STCONT,0),
125755   CP37:=(GGSGM,STPAG,STINT,STSTART,0),CP38:=(GGSGM,STFP,0),
125765
125765   STNOU:='IDNO. OF UNITS: ',STTIM:='IDTIME UNIT: ',
126005   STSEC:='IDSECOND: ',STMIN:='IDMINUTE: ',STHOUR:='IDHOUR: ',
126026   STDAY:='IDDAY: ',STMONTH:='IDMONTH: ',STYEAR:='IDYEAR: ',
126044   STLOG:='IDLOG. UNIT: ',STPRI:='IDPRIORITY: ',
126062   STSGM:='IOSEGMENT(OCT): ',STIOFL:='IOINPUT/OUTPUT(0 OR 1): ',
126110   STCONT:='IOCONTROL CODE(OCT): ',STPAG:='IDPAGE TABLE: ',
126133   STINT:='IDINT.LEVEL: ',STFP:='IOFIRST PAGE(OCT): ',
126154   STRRT:='R RT NAME: ';
126162   @CR;
126162   RBUS
126162
126162   %=============================================================================
126162   % 17.20        A B E N T R Y
126162   %
126162   %ENTRY FOR ABORTING THE TERMINAL BY COMMAND "STOP-TERMINAL"
126162   SUBR ABENTRY
126162   INTEGER ABSY:=' *** ABORTED BY SYSTEM ***$'
126200   ABENTRY:0/\0;
126201           CALL OPCFIL
126202           "BFIELD"=:B;"STBEG"=:STPNT;CALL CRLF
126207           IF BCHFLAG BIT "0" GO BOBORT
126212           T:=TTNO; *MON 2CIBU
126214           0/\0; "ABSY"; CALL OUTTEXT; CALL LOGOUT; GO FAR OPCOM
126221   RBUS
126231
126231
126231   %=====================================================================
126231   % 17.21        O U T U S T X T
126231   %
126231   %  SUBROUTINE TO OUTPUT VERSION ID AND USER DEFINED TITLE
126231
126231   SUBR OUTUSTXT
126231
126231   OUTUSTXT: L=:D; CALL ENTER
126233           "SVERALL"; CALL OUTTEXT
126235           SINVERS(0) SHZ -10; SVTXT(A); CALL OUTTEXT
126243           "SVLETTER"; CALL OUTTEXT
126245           "USTXT"; CALL OUTTEXT
126247           GO LEAVE
126250   RBUS
```

```
126255
126255   %=========================================================================
126255   %       X O U T U S T X T
126255   %
126255   % ROUTINE CALLED FROM THE ND-NET SEGMENT ROUTINE RUBREM
126255   %
126255   SUBR XOUTUSTXT
126255   XOUTUSTXT: CALL OUTUSTXT; GO OPCOM
126257   RBUS
126261
126261
126261   @DEV 1
126261   @DEV (S-S-J)SINC-1
```

```
126261
126261   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% S I N C - 1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
126261
126261
126261   %==========================================================================
126261   %
126261   %   18.0      C O M M A N D S
126261   %
126261   %==========================================================================
126261   % 18.1        L O G I N
126261
126261   %       SUBROUTINE TO LOG IN, CALLED FROM STSUPER
126261
126261   INTEGER PASSW:='PASSWORD: '
126267   INTEGER PROJNAM:='PROJNAM:DATA'          % PROJ.PASS. AND PROJECTNAME
126276   INTEGER XRTPROJ:='RTPROJ:DATA'           % RTPROGNAME AND RTPROJECT
126304   INTEGER AFILNAM:='ACCOUNTS:DATA'         % ACCOUNTING INFORMATION
126313   INTEGER ARRAY IOACTAB=?
126313   INTEGER WPROP:='UNKNOWN PROJECT PASSWORD'          % ACCOUNTING
126330   *)FILL
126330
126330   SUBR LOGIN,PROJ3,CLACIO,ALOGOUT
126330
126330   INTEGER NSU:='$THERE IS NO USER ',NNA:='S '
126344   INTEGER ARRAY DUMY:=(##E,##N,##T,##E,##R),ENTTAB(0)
126351   INTEGER MTS:='ID',ENTR:=' ENTERED AT'
126361   INTEGER USENT:='$ --- SINTRAN III BATCH PROCESSOR ---$$ USER '
126410   INTEGER MTIM:='$$MAXIMUM TIME IS',MINUT:=' MINUTES$'
126426   INTEGER ENTUSER:='$ENTER '
126432   INTEGER NPROJ:='$PROJECT NAME: '                   % ACCOUNTING
126442   INTEGER PRNTX:='$PROJECT PASSWORD: '               % ACCOUNTING
126454   INTEGER WSPW:=' WITH THE SPECIFIED PASSWORD$'
126473   INTEGER GUPTX:='$$ TOO MANY ATTEMPTS TO ENTER$'
126513   INTEGER SGUPTX:=' TOO MANY ATTEMPTS TO ENTER ON TERMINAL:'
126540   INTEGER ETBERR:='$*****BATCH ERROR; BATCH NUMBER'
126560   INTEGER TERENT:='$@ENTER COMMAND NOT FIRST COMMAND$IN BATCH-INPUT FILE: '
126614   INTEGER HPAR:=(ENTCT,"2"),BHLPAR:=("24","2")
126620   INTEGER TNPASSW:='$USERS WITHOUT PASSWORD ARE NOT ALLOWED TO LOG IN$$'
126652   INTEGER TXBIFI:='$BATCH-INPUT FILE: '
126664   INTEGER TXWARN:='$REMEMBER TO DEFINE A PASSWORD BEFORE LOGGING OUT$$'
126716
126716          DISP -200
126716   INTEGER PASS                  % PASSWORD READ FROM THE TERMINAL DURING LOGIN.
126716   INTEGER USSA
126716   INTEGER REPCT                 % REPEAT COUNT: GIVE USER THREE ATTEMPTS AT SPECIFYING A VALID PROJECT PASSWO
126716   INTEGER ENTFIRST
126716   INTEGER CCPASSW
126716          PSID
126716
126716   *)FILL
126720
126720   LOGIN: L=:D; CALL ENTER
126722          0=:UEFLG=:USPAR(0)=:USPAR(1)=:USPAR(2)=:USPAR(3)=:USPAR(4)
126735          "STACK"=:CSTCK
126737          CALL CAOFF                                  % USER'S ALT.PIT OFF
126740          #''=:XSPFMESS; 0=:FXSPFMESS
126743          IF BCHFLAG=2 THEN 0=:BCHFLAG FI
126750          IF A=-1 GO FAR BOPSTART; T:=TTNO; *MON 2CIBU
126755          0/\0; A:=1; *MON 2BRKM; MON 2ECHO
126761          0=:REMLIN=:CHNR; CALL CRLF; CALL DATCL; 0=:PASSTYPE=:SPASTYPE
```

```
126767          CALL OUTUSTXT
126770          0=:ENTCT
126771  OPSTART:
126771          CALL ESCON
126772  XOPSTART: T:=1; A:=2; *MON 2SYCN; JMP * 1          % ENTER (BAD)
126776          "ENTUSER"; CALL OUTTEXT; CALL GCOM
127001          T:=1; A:=3; *MON 2SYCN; JMP * 1             % PASSWORD (BAD)
127005          *MON 2DESC
127006          "PASSW"; CALL OUTTEXT; 0=:CPNT=:PASS
127012          -1; *MON 2ECHO
127014          DO
127014                CALL TCI; IF =15 GO OUT; A=:T
127021                IF TTIFIELD.CESCP/\377=T GO XXER1
127026                A:=T; T:=PASS SHR 3+A=:PASS
127033          OD; *)FILL
127056
127056  XXER1: IF TTIFIELD.FLAGB BIT 5MLGIN GO EIMLOGIN
127062          GO OPSTART
127063  XXER2: IF T:=TTIFIELD.FLAGB BIT 5MLGIN GO EIMLOGIN
127067          *MON 64
127070          EXIT
127071  EIMLOGIN: A:=-1; CALL MLGRSTART
127073          GO ALOGOUT
127074
127074  OUT:    1;*MON 2ECHO
127076          IF UNAFLAG><0 AND TTNO><LGCOLDSTART AND BCHFLAG>=0 THEN
127106             *MON 2EESC
127107             GO STSUPER
127110          FI
127110          IF ENTCT=MAXCT THEN                                  % ATTEMPT TO ENTER
127114             T:=1; A:=7; *MON 2SYCN; JMP * 1                   % (BAD)
127120             "GUPTX"; CALL OUTTEXT                             % TOO MANY TIMES ?
127122             IF XERDEV=1 THEN 1206 FI; A=:TDVN
127130             CALL CRLF; CALL DATCL; "SGUPTX"; CALL OUTTEXT
127134             TTNO; CALL DECU; CALL CRLF; 1=:TDVN
127141             GO XOPSTART; *)FILL
127157          FI
127157          IF MAXCT><-1 THEN
127163              A:="HPAR"; *MON 2HOLD
127165              MIN ENTCT
127166          FI
127166          PASS;0=:PASS; X:="COMSTRING"
127171          CALL FILSYS(ENFUSR); GO XXER1; A=:PASSTYPE; T=:ENTFIRST
127176          A=:SPASTYPE;=D=:CCPASSW
127201          CUSER=:CURUSER
127203          IF EXSECURITY BIT 7NPASS AND CCPASSW=0 AND CURUSER><-1 THEN
127214             IF ENTFIRST=0 THEN
127216                "TXWARN"; CALL OUTTEXT
127220             ELSE
127221                T:=1; A:=21; *MON 2SYCN; JMP *+1
127225                "TNPASSW"; CALL OUTTEXT; CALL CRLF; GO LOGOUT
127231             FI
127231          FI; CALL CLACIO                                      % CLEAR ACCOUNTED I/O
127232          0=:A=:D; AD=:CT500=:CPUT5                            % CLEAR ND-500 USEAGE.
127236          0=:CURPROG.DTIN2=:X.DTIN1; *MON 2TIME
127242          AD=:TIMON; X:="SC100"; T:=2; "TYPD"
127246          CALL FILSYS(DOPSCR); CALL XXER2
127251          FOR X:=0 TO UEMAX DO 0=:UEDAT(X); OD                 % CLEAR UEDATA
127260          -3=:REPCT                                            % GIVE USER THREE ATTEMPTS AT GIVING CORRECT PROJECT PASS
127262          GO PRJIN
```

```
127263   *)FILL
127303   %
127303   %       IF ACCOUNTING IS ON THEN READ PROJECT PASSWORD FROM THE TERMINAL AND CHECK IT AGAINST VALUES IN PRO
127303   %
127303   PRJIN: IF ACCFLAG><0 THEN                                  % ACCOUNTING ON
127303                T:=1; A:=4; *MON 2SYCN; JMP * 1               % PROJECT PASSWORD (BAD)
127305                "PRNTX"; CALL OUTTEXT
127311                -1; *MON 2ECHO                                % NO ECHO FOR PROJECT PASSWORD
127313                CALL PROJ1                                    % CHECK IF PASSWORD IS NEEDED AND, READ AND VALIDATE IT I
127315                IF A = -1 THEN
127316                   T:=1; A:=10; *MON 2SYCN; JMP * 1           % (BAD)
127321                   "WPROP"; CALL OUTTEXT                      % NO SUCH PROJ.PASSWORD
127325                   IF TTIFIELD.FLAGB BIT 5MLGIN GO FAR EIMLOGIN
127327                   MIN REPCT; GO PRJIN
127333                   GO ALOGOUT                                 % LOGOUT: INCORRECT PASSWORD GIVEN 3 TIMES.
127335                FI
127336                1;*MON 2ECHO
127336                IF PRJN=0 GO CONT                             % NO PROJECTS IN PROJNAM:DATA.
127340                T:=1; A:=5; *MON 2SYCN; JMP * 1               % (BAD)
127342                "NPROJ"; CALL OUTTEXT
127346                A:="WORKA"+10; CALL OUTTEXT                   % PRINT PROJECT NAME ON TERMINAL.
127350                CALL CRLF
127353
127354             ELSE                                             % ACCOUNTING NOT RUNNING.
127355                0=:PRJN
127356             FI
127356   CONT:     #OK; CALL TCO2
127360             IF TTIFIELD.FLAGB NBIT 5MLGIN THEN
127364                CALL OP2SY(YOUHAVEMAIL); 0/\0; 0=:FLMAIL
127370             FI
127370   OUT2:     T:=1; A:=6; *MON 2SYCN; JMP * 1                  % (BAD)
127374             0=:LDADR; -1=:HDADR
127377             GO LEAVE
127400   *)FILL
127414   %
127414   %       LOG USER INTO AN IDLE BATCH PROCESSOR.
127414   %
127414   BOPSTART: TTIFIELD.BCHNUM SHZ 2-3=:X; 1=:BATAB(X)          % ZERO PRJN IN CASE OF ERROR DURING LOGIN
127423             0=:A=:TTIFIELD.USIDX=:PRJN
127427             X:="BAUSER"; CALL FILSYS(ENSYS); GO BSERR
127433   NYPAR:    DO WHILE UNAFLAG><0 AND EXUNAFLAG=0
127437                "BHLPAR"; *MON 2HOLD
127441             OD
127442             IF TTIFIELD.BCHISTS=0 THEN CALL NWBPAR
127446                IF A<0 AND T=0 THEN A BZERO 17; CALL 9ERR(#06); GO NYPAR FI
127455             ELSE CALL NW2PAR
127457                IF A<0 AND T=0 THEN A BZERO 17; CALL 9ERR(#06); GO NYPAR FI
127466                T:=TTIFIELD.RIFIL; AD:=SARDFILP; *MON 2SBYT; 0
127473             FI
127473             GO LOOP; *)FILL                                  % SKIP UNTIL <@>
127504   LOOP:     DO CALL TCI; IF A=##@  GO OUT1 OD                % READ TO COMSTRING
127511   OUT1:     "COMSTRING"=:CSTRING; 0=:CPNT                    % UNTIL <CR>
127514             DO CALL TCI; CALL CWRITE; IF A=15 GO UT OD
127522   UT:       T:=TTIFIELD.RIFIL; *MON 2RBYT; 0
127526             AD=:SARDFILP; 0=:CPNT
127530             FOR X:=-5 DO CALL CREAD                          % IF NOT EQUAL TO 'ENTER'
127532                IF A><ENTTAB(X) GO FAR ERRENT                 % TRY AGAIN
127535             OD; CALL CREAD; IF A><40 AND A><##, GO FAR ERRENT
127545             CALL FILSYS(RLUSE); A:=0; CALL BACK
127551             0=:TTIFIELD.RIFIL=:X.DFOPP.ROFIL
127555             "NNA"; CALL GPAR; A=:X=:USSA; 0=:PASS; CALL CREAD
```

```
127563          DO
127563             CALL CREAD; IF A=40 OR =##,  GO OUT3;
127572             T:=PASS SHR 3+A=:PASS
127576          OD
127577   *)FILL
127611
127611   BSERR: L=:T; CALL 9ERR(#08); GO BOBORT              % BATCH SYSTEM ERROR.
127615
127615   OUT3:  A:=PASS; 0=:PASS; CALL FILSYS(ENFUSR); GO FAR ENERR
127622          A=:PASSTYPE=:SPASTYPE:=D=:CCPASSW
127626          IF EXSECURITY BIT 7NPASS AND CCPASSW=0 GO FAR BPSWERR
127634          X:="SC100"; T:=2; "TYPD"; CALL FILSYS(DOPSCR); GO BSERR; 1=:BCHFLAG
127644          CALL NW2PAR; IF A<0 THEN IF T=0 GO IER; GO OBAERR FI
127651          CUSER=:TTIFIELD.USIDX=:CURUSER; T:=X.RIFIL; AD:=SARDFILP; *MON 2SBYT; 0
127661          IF ACCFLAG><0 THEN                          % ACCOUNTING ON
127663             "WORKA"=:OSTRING; 0=:OPNT                % BUFFER FOR PROJ.PASSWORD
127666             FOR X:=0 TO 17 DO                        % PROJ.PASSW. MAX 16 (DEC) BYTES
127672                CALL CREAD                            % READ FROM COMMAND BUFFER
127673                IF A=40 OR A=##, ;GO OUTFILL
127701                CALL OWRITE                           % WRITE INTO WORKA
127702             OD
127704             GO TESTCALL
127705   *)FILL
127725
127725   OUTFILL:  A:=##
127726             FOR X TO 17 DO; CALL OWRITE; OD          % PAD WITH SPACES
127734   TESTCALL: CALL PROJ3                               % CHECK PROJ.PASSWORD
127735             IF A = -1 THEN CALL FLOGOUT; FI          % NON-EXISTENT PASSWORD SPECIFIED: LOGOUT.
127741             CALL BACK
127742          ELSE
127743             0=:PRJN
127744             CALL BACK; A:="NNA"; CALL GPAR           % DUMMY READ.
127747          FI
127747          A:="MTS"; CALL GPAR                         % READS MAXIMUM TIME
127751          IF A=0 THEN A:=1 FI A=:TTIFIELD.MXTIME
127755          IF ACCFLAG><0 AND PRJN><0 THEN
127761   %
127761   %         ACCOUNTING RUNNING AND PROJECT NAMES / PASSWORDS IN PROJNAM:DATA
127761   %
127761             FOR D:=0 TO 7 DO
127765              X:=D+10;WORKA(X);X:=D;A=:COMSTR(X)      % PROJ.NAME IN COMSTRING
127772             OD
127774             A:=#''=:COMSTR(8)
127777          FI
127777          IF TTIFIELD.DFOPP.ROFIL><120 AND ><121 THEN A:=14; CALL TCO FI
130012          CALL OUTUSTXT; "USENT"; CALL OUTTEXT
130015          X:="WORKA"; T:=CURUSER SHZ -10; A:=CURUSER/\377
130022          CALL FILSYS(GUSEN); GO FAR BSERR
130025          #''=:WORKA(10)
130030          A:="WORKA"; CALL OUTTEXT; "ENTR"; CALL OUTTEXT; *MON 2TIME
130035          AD=:TIMON; 0=:CURPROG.DTIN2=:X.DTIN1; CALL DATCL; CALL CRLF
130043          GO NN1
130044   *)FILL
130071
130071   NN1:   CALL CLACIO
130072          0=:A=:D; AD=:CT500=:CPUT5                   % CLEAR ND-500 USEAGE.
130076          "MTIM"; CALL OUTTEXT
130100          A:=TTIFIELD.MXTIME; CALL DECU; "MINUT"; CALL OUTTEXT
130105          IF ACCFLAG><0 AND PRJN><0 THEN  % ACCOUNTING ON AND PROJ.PASS. IN FILE
130111            "NPROJ";CALL OUTTEXT;"COMSTRING";CALL OUTTEXT
```

```
130115          FI
130115          A:=14;CALL TCO;GO FAR OUT2
130120   *)FILL
130133
130133   BPSWERR: FIXEDEV=:TDVN; CALL CRLF; CALL DATCL
130137          "ETBERR"; CALL OUTTEXT; TTIFIELD.BCHNUM; CALL DECU
130144          "TNPASSW"; CALL OUTTEXT; "TXBIFI"; CALL OUTTEXT; GO ERRF
130151   ERRENT: FIXEDEV=:TDVN; CALL CRLF; CALL DATCL
130155          "ETBERR"; CALL OUTTEXT; TTIFIELD.BCHNUM; CALL DECU
130162          "TERENT"; CALL OUTTEXT
130164   ERRF:   "FIBUF"; CALL OUTTEXT; CALL CRLF; CALL CRLF; 1=:TDVN
130172          GO INENERR
130173   ENERR: X:="BAUSER"; CALL FILSYS(ENSYS); A:=0     %%%%; CALL NW2PAR
130177          FIXEDEV=:TDVN; CALL CRLF; CALL DATCL
130203          "ETBERR"; CALL OUTTEXT; TTIFIELD.BCHNUM; CALL DECU
130210          "NSU"; CALL OUTTEXT; A:=USSA; CALL OUTTEXT; "WSPW"; CALL OUTTEXT
130216          "TXBIFI"; CALL OUTTEXT; GO ERRF
130221   INENERR: T:=TTIFIELD.RIFIL; 0=:X.DFOPP.BCHOSTS=:X.ROFIL; AD:=SARDFILP; *MON 2SBYT; JMP *1
130231          T:=1; A:=-1; DO; *MON 2INBT; JMP *1
130235          OD
130236          GO BLOGOUT
130237   *)FILL
130255   %
130255   %       LOCAL SUBROUTINE TO READ, CHECK AND CONVERT PROJECT-PASSWORD TO
130255   %       INDEX OF PROJECT NAME IN FILE PROJNAM:DATA
130255   %
130255   %       RETURN: A = 0 AND PRJN = PROJECT INDEX IF SUCCESSFUL: PROJECT NAME IS RETURNED IN WORKA(10)-(20) &
130255   %               A = 0 AND PRJN = 0 IF NO PROJECTS DEFINED.
130255   %               A = -1 AND PRJN = 0 IF NON-EXISTENT PROJECT PASSWORD.
130255   %
130255          DISP -200
130255   INTEGER PGNUM                  % NUMBER OF THE PAGE (1K WORDS) JUST READ FROM PROJNAM:DATA.
130255   INTEGER BLMAX                  % NUMBER OF THE LAST BLOCK (EACH OF 20 (OCT) WORDS) ON PROJNAM:DATA.
130255   INTEGER OCUSER                 % SAVED CURRENT USER ENTERED. (FROM SYSTEM SEGMENT).
130255   INTEGER OUSDI                  % SAVED USER DEFAULT DIRECTORY. (FROM SYSTEM SEGMENT).
130255   INTEGER OUSNO                  % SAVED USER INDEX. (FROM SYSTEM SEGMENT).
130255   INTEGER SAERR                  % SAVED FILESYSTEM ERROR NUMBER.
130255   INTEGER CNT                    % LOCAL COUNTER IN PRJCH.
130255   INTEGER SYSFL                  % FLAG FOR USER SYSTEM LOGGING IN; = 0: NOT USER SYSTEM; = 1: USER SYSTEM.
130255          PSID
130255   INTEGER ARRAY ACBUF=?                          % BUFFER FOR 1 PAGE OF PROJNAM:DATA
130255   INTEGER AFIPAR:=(FILNO,NULL,ACBUF,V0,"2000")   % PARAMETER LIST FOR RFILE CALL.
130255                                                  % NOTE THAT V0 = 74300 WHICH WILL BE ,B-200 HERE.
130262
130262   *)FILL                                         % LOGIN FROM A TERMINAL: READ PROJECT PASSWORD.
130263   PROJ1: D:=L; CALL ENTER
130265          "WORKA"=:OSTRING; 0=:OPNT
130270          FOR X:=0 TO 17 DO                      % BACKGROUND:READ FROM TERMINAL
130274             CALL TCI
130275             IF A=15 OR A=40 OR A=##, GO PROJ2   % WRITE INTO WORKA
130306             CALL OWRITE
130307          OD
130311   %              CONTROL DROPS OUT OF THE LOOP ALL 20 (OCT) CHARACTERS HAVE BEEN READ.
130311          GO CONT2
130312   PROJ2: A:=## ;
130313          FOR X TO 17 DO          % PAD REST OF CHARACTERS WITH SPACE.
130316             CALL OWRITE
130317          OD; GO CONT2; *)FILL
130325   %
130325   %       ENTRY FROM BATCH LOGIN: PROJECT PASSWORD ALREADY READ FROM FIRST LINE OF FILE.
130325   %
```

```
130325   PROJ3: D:=L; CALL ENTER
130327   %
130327   %       SUBSTITUTE USER SYSTEM FOR CURRENT USER TO ACCESS PROJNAM:DATA.
130327   %
130327   CONT2: O=:SAERR=:SYSFL
130331          CUSER=:OCUSER; USDI=:OUSDI; USNO=:OUSNO   % SAVE SPECIFICATION OF THE USER LOGGED IN.
130337          CALL FILSYS(GSYSI); GO USERR; T=:CUSER
130343          IF CUSER = OCUSER THEN 1=:SYSFL; FI
130351          A:=T/\377=:USNO; T SHZ -10 =:USDI
130356          "ACCSEMRE"; *MON 2RESR                    % RESERVE ACCOUNTING SEMAPHORE.
130360          X:="PROJNAM"; T:=3; A:="TYPD"             % OPEN PROJNAM:DATA
130363          *MON 2NOPE; JMP ARERR
130365          A=:FILNO=:T:=2000; *MON 2SBLZ; JAF FCERR  % SET BLOCK-SIZE
130372          1=:CNT; O=:PGNUM
130375          DO
130375             "AFIPAR"; *MON 2RFIL; JAF FCERR        % READ PAGE PGNUM: FIRST BLOCK OF FIRST PAGE
130400                                                    % CONTAINS THE NUMBER OF RECORDS.
130400             IF PGNUM = 0 THEN
130402   %
130402   %            IF BLMAX=0 NO RECORDS ARE SPECIFIED
130402   %
130402               ACBUF(0)=:BLMAX                      % WORD 1 = NUMBER OF RECORDS IN THE FILE.
130405               IF A = 0  GO FOUND
130406             FI
130406   %
130406   %         THERE ARE 100 (OCT) BLOCKS EACH CONTAINING A PROJECT NAME / PROJECT PASSWORD PAIR ON EACH PAGE.
130406   %         ON THE FIRST PAGE CNT STARTS FROM 1 SINCE THE INDEX BLOCK IS THE FIRST RECORD.
130406   %
130406             FOR CNT TO 77 DO
130412                FOR D:=0 TO 7 DO
130416                   T:=WORKA(D)                      % COMPARE THE STRING READ FROM THE TERMINAL
130420                   A:=CNT*20+D=:X                   % WITH THE TEXT FROM THE FIRST 8 WORDS OF EVERY BLOCK.
130424                   A:=ACBUF(X)
130425                   IF A >< T GO NEXT                % TWO STRINGS DIFFER.
130427                OD
130431   %
130431   %            IF CONTROL FALLS OUT OF THE BOTTOM OF THE LOOP THE CORRECT PASSWORD HAS BEEN FOUND:
130431   %            COPY PROJECT NAME TO "WORKA"+10 AND RETURN BLOCK NUMBER IN THE A-REG. D-REG WILL NOW = 10.
130431   %
130431                FOR D TO 17 DO
130434                   A:=CNT*20+D=:X                   % COPY PROJECT NAME TO WORKA(10) - WORKA(20) AND
130440                   T:=ACBUF(X)                      % PRNAM
130441                   T=:WORKA(D); X-10; T=:PRNAM(X)
130445                OD
130447                A:=PGNUM*100+CNT                    % NUMBER OF BLOCK IN PROJNAM:DATA
130452                GO FOUND
130453   NEXT:
130453                IF A:=PGNUM*100+CNT >= BLMAX THEN
130461                   GO NOTFD
130462                FI
130462             OD
130466             O=:CNT; MIN PGNUM
130470          OD
130471   *)FILL
130510   NOTFD: -1=:PRJN; GO RETUR
130513   FOUND: A=:PRJN; A:=0
130515   FCERR: A=:SAERR                                  % ERROR IN SET-BLOCK-SIZE OR RFILE.
130516   RETUR: T:=FILNO; *MON 2CLOS                      % CLOSE PROJNA:DATA
130520   ARERR: A=:SAERR                                  % ERROR IN OPENING OR CLOSING PROJNAM:DATA.
130521          OCUSER=:CUSER; OUSDI=:USDI; OUSNO=:USNO   % RESTORE CURRENT USER
```

```
130527          A:=#''=:WORKA(20)                          % ' AT THE END OF PROJECT NAME FOR OUTTEXT.
130532          "ACCSEMRE"; *MON 2RELE; JMP TERR           % RELESE SEMAPHORE
130535   USERR: A=:SAERR
130536   TERR:  IF A:=SAERR >< 0 THEN                                % IF A FILESYSTEM ERROR MESSAGE OCCURS
130540             T:=1; A:=11; *MON 2SYCN; JMP * 1                 % (BAD)
130544             A:="ACER"; CALL OUTTEXT; A:=SAERR; *MON 2ERMS    % LOG THE USER OUT.
130550             O=:PRJN
130551             IF SYSFL = 1 THEN
130555                A:=0; GO LEAVE                          % ALLOW USER SYSTEM TO LOG IN TO RECTIFY ERROR.
130557             ELSE
130560                BCHFLAG+1; GOSW  ALOGOUT, BLOGOUT       % LOG OUT OTHER USERS.
130565             FI
130565          FI
130565          A:=PRJN; IF A=-1 THEN O=:PRJN; FI; GO LEAVE
130573   *)FILL
130604          INTEGER ARRAY ACBUF(2000)
132604
132604   %-----------------------------------------------------------------------
132604   %      LOCAL SUBROUTINE TO CLEAR ACCOUNTED I/O BLOCKS
132604
132604   CLACIO:D:=L; CALL ENTER
132606          IF "IOACTAB" = 0 THEN GO LEAVE FI          % OPTION NOT INCLUDED
132611          CURPROG-RTSTART=:D:=0; T:=5RTSIZE; *RDIV ST
132617          A SHZ 1 + "IOACTAB"=:X; O=:A=:D; CALL 1DWRITE
132625          GO LEAVE
132626
132626   CFILERR: A=:X; *MON 2EESC
132630          CALL ESCON; A:=X; GO FILERR
132633   RBUS
132640
132640   %=========================================================
132640   %            U U E L O G I N
132640   % MONITOR CALL TO LOG IN A NEW USER
132640   % PARAMETERS IN COMSTRING: 0-10B=USERNAME, 11B=PASSWORD, 12-21B=PROJECT
132640   SUBR UUELOGIN
132640   DISP -200; INTEGER PASS; INTEGER ARRAY POINTER UCOMSTRING; PSID
132640   INTEGER NOU:='NO USER'
132644   INTEGER NOSC:='NO SCRATCH FILE'
132654   UUELOGIN: L=:D; CALL ENTER
132656           COMSTRING(11)=:PASS; "COMSTRING"+12=:"UCOMSTRING"
132664           FOR X:=0 TO 7 DO UCOMSTRING(X)=:WORKA(X) OD
132674           CALL FILSYS(RLUSE); 0/\0
132677           X:="COMSTRING"; PASS; CALL FILSYS(ENUSE); GO ERR1
132704           A=:PASSTYPE=:SPASTYPE; CUSER=:CURUSER
132710           CALL CLACIO; O=:A=:D; AD=:CT500=:CPUT5
132715           O=:CURPROG.DTIN2=:X.DTIN1; *MON 2TIME
132721           AD=:TIMON; X:="SC100"; T:=2; "TYPD"; CALL FILSYS(DOPSCR); GO ERR2
132730           IF ACCFLAG><0 THEN
132732                 CALL PROJ3; IF A= -1 GO ERR3
132736           FI; GO LEAVE
132737   ERR1:   "NOU"; GO ERR
132741   ERR2:   "NOSC"; GO ERR
132743   ERR3:   "WPROP"
132744   ERR:    CALL OUTTEXT; UEFLG BZERO 5UECM=:UEFLG; CALL LOGOUT
132751   RBUS
132772
132772   %=============================================
132772   %          U E P R E L O G I N
132772   % SUBROUTINE TO LOG IN USER ENVIRONMENT
132772   SUBR UEPRELOGIN
```

```
132772   INTEGER UEPASS:=0
132773   INTEGER NOTUS:='USER USER-ENVIRONMENT DOES NOT EXIST$'
133016   INTEGER FELL1:='UE-LOGIN PROCEDURE IS MISSING$'
133036   INTEGER FELL2:='UE-LOGIN'
133043   UEPRELOGIN: L=:D; CALL ENTER; "STACK"=:CSTCK; 0=:PASSTYPE=:SPASTYPE
133051          X:="USEVT"; UEPASS; CALL FILSYS(ENSYS); GO ERR
133056          CUSER=:CURUSER; 0=:USPAR(0)=:USPAR(1)=:USPAR(2)=:USPAR(3)=:USPAR(4)
133072          FOR X:=0 TO UEMAX DO 0=:UEDATA(X) OD; 4000/\ UEFLG=:UEFLG
133104          5BCOMM=:INLOGGED
133106          A:="FELL2"; *MON 317
133110          GO ERR2; GO LEAVE
133112   ERR2:  0=:INLOGGED;
133113          "FELL1"; CALL OUTTEXT; CALL LOGOUT
133116   ERR:   "NOTUS"; CALL OUTTEXT; CALL LOGOUT
133121   RBUS
133137
133137   %==========================================================================
133137        %              U E A D M (MON 321)   CALLABLE FROM BACKGROUND ONLY          %
133137        %                                                                          %
133137        %  ENTRY POINT = MOFIA                                                     %
133137        %                                                                          %
133137        %  CALLING PROGRAM:                                                        %
133137        %  1.PARAMETER = LOG.DEV.          IN T-REG  IF LOG.DEV=1 THEN OWN TERMINAL %
133137        %  2.PARAMETER = FUNCTION CODE     IN A-REG                                %
133137        %  3.PARAMETER = ADDR. OF STRING   IN X-REG                                %
133137        %                                                                          %
133137        %                                *  ON RETURN:  - SKIP RETURN IF OK        %
133137        %                                *              - NORMAL RETURN IF ERROR AND %
133137        %                                *                A-REG = ERROR CODE      %
133137        %                                                                          %
133137        %  THIS MONITOR CALL IS PROTECTED BY A SEMAPHORE ( UESEM,0;0;*-2;0 )       %
133137        %  TO PREVENT SIMULTANEOUS UPDATING OF SYSTEM DATA.                        %
133137        %                                                                          %
133137        %  MONITOR CALL FOR ND SUBSYSTEMS (UE,JCL AND TRUE)                        %
133137        %                                                                          %
133137        %  THE ROUTINE WORKS ON A DATAFIELD RESIDING ON THE SYSTEM SEGMENT         %
133137        %  (INTEGER ARRAY UEDAT(30)), HOLDING INFORMATION UTILIZED BY ND SUBSYSTEMS. %
133137        %  AN ADDITIONAL  FLAGWORD (UEFLG) IS USED TO KEEP TRACK OF THE ENABLED    %
133137        %  FACILITIES.                                                             %
133137        %                                                                          %
133137        %      INTEGER UEFLG        (SYSTEM SEGMENT)                               %
133137        %                                                                          %
133137        %      SYMBOL 5UEMO=0       JEC IS ACTIVE                                  %
133137        %      SYMBOL 6UETE=1       UE TERMINATION HANDLING ENABLED                %
133137        %      SYMBOL 5UEIE=2       INHIBIT ERROR MESSAGES FROM S III ROUTINES     %
133137        %      SYMBOL 5UETLOG=3     TERMINATION HANDLING ENABLED FOR LOGOUT        %
133137        %      SYMBOL 5UERUN=4      USER ENVIRONMENT IS RUNNING                    %
133137        %      SYMBOL 5UEPRUN=5     USER ENVIRONMENT IS SOON RUNNING               %
133137        %      SYMBOL 5UECM=6       THE MON.CALL NCMD IS EXECUTING                 %
133137        %      SYMBOL 5UEST=7       STOP-TERMINAL HAS BEEN EXECUTED                %
133137        %      SYMBOL 5UEES=10      ESCAPE HAS BEEN PRESSED                        %
133137        %      SYMBOL 5UETM=11      CPU TIMEOUT IN BATCH JOB                       %
133137        %      SYMBOL 5UELOGD=12    USER WANTS TO LOG OUT                          %
133137        %      SYMBOL 5UEAL=13      AUTOMATIC LOGIN AND START OF SUBSYSTEM         %
133137        %                                                                          %
133137        %      FUNCTION CODES:                                                     %
133137        %                                                                          %
133137        %      1 = READ UEDAT                                                      %
133137        %      2 = SET UEDAT                                                       %
133137        %      3 = ENABLE           SETS BITS 0,1,2 OR 3                           %
```

```
133137          %       4 = DISABLE             RESETS BITS 0,1,2 OR 3                                  %
133137          %                               THE CALLING PROGRAM'S X-REGISTER HOLDS THE ADDRESS      %
133137          %                               OF THE BITMASK TO BE SET OR RESET                       %
133137          %                                                                                       %
133137          %       5 = DEFINE  UE TERMINATION HANDLING                                             %
133137          %       6 = LIST      "       "       "                                                 %
133137          %       7 = READ ERROR INFORMATION                                                      %
133137          %       10= TURN OFF "MODE-STATUS" FOR OWN TERMINAL                                     %
133137          %                                                                                       %
133137          %        FUNCTION 7:                                                                    %
133137          %        ===========                                                                    %
133137          %        OUTPUT FORM UEADM TO PROGRAMS ARRAY:                                           %
133137          %                                                                                       %
133137          %        WORD     1: ERROR NUMBER                                                       %
133137          %        WORD     2: ERROR FLAG        1=MON 65 ERROR                                   %
133137          %                                      2=SINTRAN III RUN TIME ERR                       %
133137          %                                      3=FILE SYSTEM ERROR                              %
133137          %                                      4=STOP TERMINAL HAS BEEN EXECUTED                %
133137          %                                      5=ESCAPE HAS BEEN PRESSED                        %
133137          %                                      6=BATCH JOB TIMEOUT                              %
133137          %                                      7=USER WANTS TO LOG OUT                          %
133137          %        WORD     3: P-REGISTER FOR FAILING PROGRAM                                     %
133137          %        WORD 3-12: REGISTER BLOCK (P,X,T,A,D,L,STS(BIT 0-7),B),                        %
133137          %                   ONLY WHEN ERROR FLAG =1                                             %
133137          %                                                                                       %
133137          %       AFTER THE ERROR INFORMATION HAS BEEN READ,  RERNUM & RERPREG ARE SET            %
133137          %       TO 0.                                                                           %
133137          %                                                                                       %
133137          %       THE BUFFER HOLDING THE TERMINATIONS COMMAND FOR USER ENVIRONMENT                %
133137          %       ON THE COMMAND SEGMENT:                                                         %
133137          %                                                                                       %
133137          %       INTEGER ARRAY UECMD(UEBSZ)     WHERE UEBSZ=37                                   %
133137          %                                                                                       %
133137          %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
133137  INTEGER ARRAY UECMD=?
133137  SUBR UEADM,T1P15
133137          SYMBOL UECNT=36                                         % THE LOGICAL DEVICE IS NOT A TERMINAL
133137          SYMBOL UEER1=311                                        % NO TERMINATION HANDLING DEFINED
133137          SYMBOL UEER2=312
133137
133137  % NOTE! DISPLACEMENTS T2SEG,WPNT AND SYARR ARE USED BY TAD ROUTINE BSCPC
133137  %       DO NOT CHANGE DISPLACEMENT VALUE.
133137
133137  DISP -200
133137          INTEGER TSEGM,T2SEG,WPNT,INDXX,COMM,APTAB,NPTAB,UEPNT=NPTAB
133137          INTEGER INDYY,UEFUN,LREG,BREG,SYSG=TSEGM,LOGN=NPTAB
133137          INTEGER UERERNUM=APTAB,UEWRK=APTAB
133137          INTEGER ARRAY POINTER SYARR
133137  PSID
133137  UEADM:  X:=B;A:="BFIELD"=:B;X=:BREG;A:=L=:LREG                  % ILLEGAL FUNCTION CODE
133145          IF X.ZAREG<1 OR >10 THEN
133154                201=:X.ZAREG; GO FAR UEEUT
133157          ELSE                                                    % FUNCTION,STRING ADDR.& SEGMENT NO. OF
133160                A=:UEFUN;X.ZXREG=:WPNT;T:=X.STRSEG=:TSEGM
133165          FI                                                      % RESERVE SEMAPHORE AND WAIT IF OCCUPIED
133165          "UESEM"; T:="0"; CALL CCBRSRV;0/\0
133171  @ICR
133171          UEFUN GOSW UEDUM,UEGET,UESET,UEENA,UEDIS,UEDEF,
133201                     UELIS,FAR UEREI,FAR OFFMD
```

```
133203   @CR;
133204   UEDUM:
133204   UEGET: 0=:COMM; GO FELL1
133206   UESET: 1=:COMM; GO FELL1
133211   UEDEF: 2=:COMM; GO FELL2
133214   UELIS: 3=:COMM; GO FELL2
133217   UEREI: 4=:COMM; GO FELLS
133222   UEENA: 5=:COMM; GO FELLS
133225   UEDIS: 6=:COMM; GO FELLS
133230   FELL1: "UEDAT"=:"SYARR";UEMAX=:INDYY;GO FELLS                    % INITIALIZE ARRAY POINTER WITH
133235   FELL2: "UECMD"=:"SYARR";UECNT=:INDYY;GO FELLS                    % UEDAT OR UECMD
133242          *)FILL
133253   FELLS: X.OLDPAG SHZ -7/\3=:APTAB; X.OLDPAG SHZ -11/\3=:NPTAB     % PREPARATIONS NECESSARY BEFORE
133263          TSEGM/\377=:T2SEG*5SEGSIZE+SEGSTART=:X; X.LOGADR SHZ -10  %    CALLING GET1L/PUT1L TO GET OR PUT
133273          IF A=200 AND APTAB><NPTAB THEN T2SEG BONE 17=:T2SEG FI    %    ONE LOCATION ON A 3RD SEGMENT
133305          "0"=:INDXX; BREG.ZTREG=:LOGN
133312          IF A=1 THEN GO UEJMP FI                                   % T=1: OWN TERMINAL
133316          IF A=0 THEN 1=:LOGN; GO UEJMP FI   % NOTE! THIS LINE SHOULD BE DELETED FOR K-VERS
133322          CALL LOGPH; IF A=0 THEN D=:A; FI
133325          IF =0 THEN A:=174; GO FAR ERRX1 FI                        % ILLEGAL PARAMETER
133330          A=:X;CALL CHTERM; GO FAR UENOT                            % CHECK IF TERMINAL
133333          T:="DBPROG"; CALL XGTDFADDR;
133335          A.SEGM/\177400 SHZ -10=:SYSG                              % GET NO. OF SYSTEMSEGMENT
133342   @ICR
133342   UEJMP: COMM GOSW FAR UEGT1,FAR UEST1,FAR UEST3,
133347                   FAR UEGTE,UERE1,UEEN1,UEDI1
133352   @CR;
133353   % ENABLE/DISABLE TERMINATION HANDLING:
133353   UEEN1: K:=1;GO FELL3
133355   UEDI1: K:="0"
133356   FELL3: T:=T2SEG; X:=WPNT; CALL GET1L; GO FAR ERR;                % GET ENABLE/DISABLE MASK
133362          A/\17=:D                % DONT CARE ABOUT OTHER BITS THAN 0,1,2 AND 3.
133364          IF A BIT 5UETE AND UECMD(0)=#'' THEN
133373                UEER2=:BREG.ZAREG; GO FAR UEERR                     % UE TERMINATION HANDLING NOT DEFINED
133377          FI
133377          IF LOGN><1 THEN
133403                T:=SYSG;X:="UEFLG";CALL GET1L; GO FAR ERR
133407          ELSE
133410                UEFLG
133411          FI
133411          IF K THEN A\/D GO FELL4 FI                                % ENABLED
133415          IF D BIT 5UETLOG THEN A BZERO 5UETLOG FI                  % DISABLE TERMINATION HANDLING ON LOGOUT
133420          IF D BIT 5UETE THEN A BZERO 5UETE FI                      % DISABLE TERMINATION HANDLING
133423          IF D BIT 5UEIE THEN A BZERO 5UEIE FI                      % SINTRAN III ERROR MESSAGES ARE OK
133426          IF D BIT 5UEMO THEN A BZERO 5UEMO FI                      % JEC IS NOT ACTIVE
133431          GO FELL4; *)FILL
133462   FELL4: A=:UEWRK;
133463          IF LOGN><1 THEN
133467                UEWRK;T:=SYSG;CALL PUT1L; GO FAR ERR; GO FAR UEOUT
133474          ELSE
133475                UEWRK=:UEFLG
133477          FI
133477          GO FAR UEEOK;
133500   %  GET ERROR INFO:
133500   UERE1: IF LOGN><1 THEN GO UERE3 FI
133505          % OWN TERMINAL:
133505          RERNUM;X:=WPNT;T:=T2SEG;CALL PUT1L;GO FAR ERR             % READ ERROR NUMBER FROM SYSTEM SEGMENT
133512          MIN WPNT;0/\0
133514   L1:    UEFLG
133515          IF A BIT 5UELOGD THEN A BZERO 5UELOGD=:UEFLG; 7; GO L2; FI % USER WANTS TO LOG OUT
```

```
133523          IF A BIT 5UEST THEN A BZERO 5UEST=:UEFLG; 4; GO L2; FI      % STOP TERMINAL
133531          IF A BIT 5UEES THEN A BZERO 5UEES=:UEFLG; 5; GO L2; FI      % ESCAPE HAS BEEN PRESSED
133537          IF A BIT 5UETM THEN A BZERO 5UETM=:UEFLG; A:=6; GO L2; FI   % CPU TIMEOUT
133545          IF RERNUM=0 THEN A:=0; GO L2 FI
133551          IF FLQERM=1 THEN GO UERE2 FI
133556          IF RERNUM/\177400>"0" THEN 2 ELSE 3 FI
133566          X:=WPNT;T:=T2SEG;CALL PUT1L;GO FAR ERR;MIN WPNT;0/\0
133574          A:=RERPREG
133575  L2:     X:=WPNT; T:=T2SEG;CALL PUT1L; GO FAR ERR; GO FAR UEOUT
133602          *)FILL
133611  UERE2:  X:=WPNT;T:=T2SEG;CALL PUT1L;GO FAR ERR;MIN WPNT;0/\0        % IF THE ERROR WAS CAUSED BY MON 65
133617          "ESCBLOCK"=:"SYARR";7=:INDYY;GO FAR UEGTE                   % HAND OVER THE ESC.BLOCK AS WELL
133624          *)FILL
133630  % OTHER TERMINAL:
133630  UERE3:  T:=SYSG;X:="RERNUM"; CALL GET1L; GO FAR ERR; A=:UERERNUM
133635          T:=T2SEG; X:=WPNT; CALL PUT1L; GO FAR ERR; MIN WPNT; 0/\0
133643  L3:     X:="UEFLG";T:=SYSG; CALL GET1L; GO FAR ERR
133647          IF A BIT 5UELOGD THEN
133651                A BZERO 5UELOGD; T:=SYSG;X:="UEFLG";CALL PUT1L
133655                GO FAR ERR; 7; GO L4
133660          FI
133660          IF A BIT 5UEST THEN
133662                A BZERO 5UEST; T:=SYSG;X:="UEFLG";CALL PUT1L
133666                GO FAR ERR; 4; GO L4
133671          FI
133671          IF A BIT 5UEES THEN
133673                A BZERO 5UEES;T:=SYSG;X:="UEFLG"
133676                CALL PUT1L; GO FAR ERR; 5; GO L4
133702          FI
133702          IF A BIT 5UETM THEN
133704                A BZERO 5UETM;T:=SYSG;X:="UEFLG"
133707                CALL PUT1L; GO FAR ERR; 6; GO L4
133713          FI
133713          IF UERERNUM=0 THEN GO L6 FI
133716          T:=SYSG; X:="FLQERM"; CALL GET1L; GO FAR ERR
133722          IF =1 THEN GO UERE4 FI
133726          IF UERERNUM/\177400>0 THEN 2 ELSE 3 FI
133735          X:=WPNT;T:=T2SEG;CALL PUT1L; GO FAR ERR; MIN WPNT; 0/\0
133743          X:="RERPREG"; T:=SYSG; CALL GET1L; GO FAR ERR
133747          X:=WPNT; T:=T2SEG; CALL PUT1L; GO FAR ERR; GO FAR UEOU1
133754          *)FILL
133765  UERE4:  T:=T2SEG; X:=WPNT; CALL PUT1L; GO FAR ERR; MIN WPNT;0/\0
133773          T=:D:=SYSG; X:="ESCBLOCK"=:UEPNT;7=:INDYY; GO FAR UEGT3
134002  L6:     A:=0
134003  L4:     X:=WPNT; T:=T2SEG; CALL PUT1L; GO FAR ERR; GO FAR UEOUT
134010          *)FILL
134015          %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
134015          %                                                            %
134015          %                                                            %
134015          % UEST1: GET INFORMATION FROM THE PROGRAM AND STORE IT       %
134015          %        ON THE SYSTEM SEGMENT (UEDAT) OR IN THE BUFFER      %
134015          %        FOR UE-TERMINATION COMMAND (UECMD) RESIDING         %
134015          %        ON THE COMMAND SEGMENT.                             %
134015          %                                                            %
134015          %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
134015  UEST1:  IF LOGN><1 THEN GO UEST2 FI
134022          % OWN TERMINAL:
134022          IF TTIFIELD.TYPRING BIT 5BAD THEN                           % TAD
134026  T1P15:     CALL BSCPC; 0/\0                                         % SEND CODE TO LOCAL SYSTEM
134030          FI
 34030  UEST3:  FOR INDXX TO INDYY DO
```

```
134034                X:=WPNT; T:=T2SEG; CALL GET1L; GO FAR ERR % IF ERR: ILLEGAL ADDR.REF IN MONITOR CALL
134040                A=:SYARR(INDXX); MIN WPNT; 0/\0
134044          OD
134050          IF COMM=2 THEN
134054                3=:PANAMSR;A:="PANAMSR"; *MON 2WSEG
134060          FI
134060          GO FAR UEEOK; *)FILL
134066          % OTHER TERMINAL:
134066   UEST2: IF COMM=1 THEN "UEDAT"=:UEPNT FI
134074          IF COMM=2 THEN "UECMD"=:UEPNT FI
134102          FOR INDXX TO INDYY DO
134106              T:=T2SEG;X:=WPNT; CALL GET1L; GO FAR ERR; MIN WPNT; 0/\0
134114              T:=SYSG;X:=UEPNT; CALL PUT1L; GO ERR; MIN UEPNT; 0/\0
134122          OD
134126          GO FAR UEEOK
134127          *)FILL
134135          %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
134135          %                                                                      %
134135          %      UEGT1: GET INFORMATION FROM UEDAT OR UECMD AND                  %
134135          %             HAND IT OVER TO THE CALLING PROGRAM.                     %
134135          %                                                                      %
134135          %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
134135   UEGT1: IF LOGN><1 THEN GO UEGT2 FI
134142          % OWN TERMINAL:
134142   UEGTE: FOR INDXX TO INDYY DO
134146                A:=SYARR(INDXX);
134150                X:=WPNT; T:=T2SEG; CALL PUT1L; GO ERR
134154                MIN WPNT; 0/\0
134156          OD
134162          GO UEOUT
134163   UEGT2: IF COMM=0 THEN "UEDAT"=:UEPNT FI
134167          IF COMM=3 THEN "UECMD"=:UEPNT FI
134175   UEGT3: FOR INDXX TO INDYY DO
134201                T:=SYSG;X:=UEPNT;CALL GET1L;GO ERR; MIN UEPNT; 0/\0
134207                T:=T2SEG;X:=WPNT; CALL PUT1L; GO ERR; MIN WPNT;0/\0
134215          OD
134221          GO UEOU1; *)FILL
134226   OFFMD: X=:B; X:="DFS1"; T:=X+45
134232          DO                                   % COPY FROM DFS2 TO DFS1
134232             *LDA ,B; STA ,X; AAX 1; AAB 1
134236          WHILE X<T
134240          OD
134241          "BFIELD"=:B; X:="DFS1"=:BREG
134245          IF X.ZTREG><0 THEN A:=25; GO ERRX1; FI
134251          IF BCHFLAG><2 THEN A:=174; GO ERRX1; FI
134257          0=:BCHFLAG; X:=TTIFIELD; T:=X.RIFIL; 0=:X.RIFIL; *MON 43
134264          0/\0; X:=X.DFOPP; T:=X.ROFIL; 0=:X.ROFIL; *MON 43; JMP *+1
134272          DO WHILE X:=MSTPN><0
134274             X-2=:MSTPN; AD:=MOSTK(X); A=:T; *MON 43
134301             0/\0; D=:T; *MON 43; JMP *+1
134305          OD; GO UEEOK
134307   ERR:   A:=153                          % ERROR EXIT FROM GET1L/PUT1L
134310   ERRX1: A=:BREG.ZAREG;GO UEERR
134313   UENOT: A:=UEER1; GO ERRX1                                % SPECIFIED DEVICE IS NOT A TERMINAL
134315   UEOUT: IF COMM=4 THEN                                    % ERROR INFORMATION HAS BEEN READ
134321               "0"=:RERNUM=:RERPREG=:FLQERM                 % AND ERROR NO. & P-REG WHEN ERROR OCCUR
134325               UEFLG; A BZERO 5UELOGD
134327               A BZERO 5UETM; A BZERO 5UEES; A BZERO 5UEST=:UEFLG
134333          FI                                                % IS SET TO 0.
134333          GO UEEOK
```

```
134334   UEOU1: IF COMM=4 THEN
134340                "0";T:=SYSG;X:="RERNUM"; CALL PUT1L; GO ERR
134345                X+1; CALL PUT1L; GO ERR;X:="FLQERM";CALL PUT1L
134352                GO ERR
134353                T:=SYSG; X:="UEFLG"; CALL GET1L; GO ERR
134357                A BZERO 5UETM; A BZERO 5UEES; A BZERO 5UEST; T:=SYSG; X:="UEFLG"
134364                CALL PUT1L; GO ERR
134366          FI
134366   UEEOK: MIN LREG
134367   UEERR: "UESEM"; CALL CCBRELEASE
134371   UEEUT: LREG=:L;=BREG=:B;TAD:=ZTADREG;X:=ZXREG;EXIT                    % ERROR EXIT
134400   RBUS
134412
134412   %=========================================================================
134412   % 18.2        B L O G O U T
134412
134412   % BATCH LOGOUT, USER LEVEL
134412   SUBR BLOGOUT,FLOGOUT
134412   INTEGER EXI:='$ BATCH USER LOGGED OFF AT'
134430   FLOGOUT: "WPROP"; CALL OUTTEXT; GO FEILP  % NO SUCH PROJECT PASSWORD
134433   BLOGOUT: "STACK"=:CSTCK; T:=TTIFIELD.RIFIL; *MON 2RBYT; JMP *2
134441          AD=:SARDFILP; 0=:MSTPN
134443          IF BCHFLAG=1 THEN
134447             "EXI"; CALL OUTTEXT; CALL DATCL; CALL CRLF; CALL CTIMUS
134454             IF TTIFIELD.DFOPP.ROFIL><120 AND ><121 THEN A:=14; CALL TCO FI
134467             CALL CRLF; CALL ACCOUNT
134471          FI
134471   FEILP: -1=:BCHFLAG
134473          0=:TTIFIELD.RIFIL=:X.DFOPP.ROFIL
134477          CALL MBZMEMORY;                                  % ZEROING MEMORY
134500          CALL FILSYS(RLUSE); A:=0; 0=:TTIFIELD.BSTATE=:X.USIDX
134506          GO XBLOGOUT            % CONTINUE ON FIRST PAGE
134507   RBUS
134523
134523   INTEGER BINPE:='$BATCH INPUT ERROR:'
134535   INTEGER BOUPE:='$BATCH OUTPUT ERROR'
134547   INTEGER JOBAB:='$ *** BATCH JOB ABORTED ***$'
134566
134566   %=========================================================================
134566   % 18.3        B I L C M N D   B I L P A R
134566   %
134566   % ILLEGAL COMMAND AND ILLEGAL PARAMETER IN BATCH
134566   SUBR BILCMND,BILPAR
134566   INTEGER ILMND:='$ILL. COMMAND'
134575   BILCMND:"ILMND"; GO BIL
134577   BILPAR:"ILLP"
134600   BIL:    CALL OUTTEXT; GO JABORT
134602   RBUS
134605
134605   %=========================================================================
134605   % 18.4        B O B O R T
134605   %
134605   SUBR BOBORT
134605   INTEGER BABO:=' *** BATCH PROCESS ABORTED ***$'
134625   BOBORT: "STBEG"=:STPNT
134627          CALL DATCL; "BABO"; CALL OUTTEXT; "RINPA"; *MON 2RESR
134634          T:=201; *MON 2XMSG
134636          T:=1; *MON 2XMSG
134640          T:=BINDV; *MON 2CIBU; JMP *1
134643          CALL MBZMEMORY;                                  % ZEROING MEMORY ?
```

```
134644          CALL FILSYS(RLUSE); A:=0; -1=:BCHFLAG
134651          0=:TTIFIELD.BSTATE=:X.BCHISTS=:X.RIFIL=:X.DFOPP.ROFIL
134657          X.BCHNUM SHZ 2-3=:X; 0=:BATAB(X); "ABRPAR"; *MON 2ABOR; JMP *
134667   RBUS
134677
134677   %=========================================================================
134677   % 18.5        J A B 2
134677   %
134677   SUBR JAB2
134677   JAB2:   "STBEG"=:STPNT
134701          "JOBAB"; CALL OUTTEXT; IF BCHFLAG=2 GO TOUS
134707          0=:TTIFIELD.DFOPP.BCHOSTS; T:=1; DO; *MON 2INBT; JMP *1
134715          OD
134716   RBUS
134721
134721
134721   %=========================================================================
134721   % 18.6        L O G O U T
134721   %
134721   % COMMAND: LOGOUT
134721   %
134721   % COMMAND FOR LOGGING OUT
134721
134721   SUBR LOGOUT,ALOGOUT
134721
134721   INTEGER EXI:='$--EXIT--$',ABRPAR:=RTREF,PAREE:=NULL
134731   LOGOUT: L=:D; CALL ENTER; "STACK"=:CSTCK
134735          IF UEFLG BIT 5UETLOG THEN        % TERMINATION HANDLING ON LOGOUT ENABLED
134740             A BONE 5UELODG=:UEFLG; "UECMD"; *MON 317; JMP *+1
134745          FI
134745          IF FLMAIL=0 AND BCHFLAG=0 AND TTIFIELD.FLAGB NBIT 5LOGOUT AND A NBIT 5MLGIN THEN
134757               CALL OP2SY(YOUHAVEMAIL); GO NOMAIL; MIN FLMAIL; GO LEAVE
134764          FI
134764   %%     IF TTIFIELD.FLAGB BIT 5MLGIN THEN A BONE 5SPSIO=:X.FLAGB FI
134764   NOMAIL:IF BCHFLAG BIT "0" GO BILCMND; 0=:BCHFLAG
134770          IF REMLIN><0 THEN CALL NDNCOMMAND(LOCAL) FI; CALL DATCLK
134775          CALL TIMUSED; CALL ACCOUNT       % TIME-USED, STORE DATA IN ACCOUNTS
134777          "0"; *MON 2BRKM; MON 2ECHO
135002   ALOGOUT: CALL MBZMEMORY;                          % ZEROING MEMORY
135003          CALL FILSYS(RLUSE);0/\0;
135006          "EXI"; CALL OUTTEXT; 0=:CURUSER
135011          T:=1; A:=13; *MON 2SYCN; JMP * 1           % (BAD)
135015          IF TTIFIELD.TYPRING BIT 5COM THEN A:=15; CALL TCO
135023          ELSE 0=:X.DFOPP.SCREEN      % RESET STOP ON PAGE
135026          FI; "PAREE"; *MON 2REEN
135030          0=:TTIFIELD.BSTATE; X.DFLAG/\137777 =: X.DFLAG
135035          IF A BIT 5IESC THEN              % IF ESCAPE DISABLED THEN
135037             *MON 2EESC                    % ENABLE-ESCAPE
135040          FI; X.FLAGB BZERO 5MLGIN BZERO 5SPSIO=:X.FLAGB
135044          0=:COBSTATE=:CMDFFIELD=:5COBSTATE=:PRJN
135050          CALL RESESC                                % RESET TO NORMAL ESCAPE MODE
135061          X.FLAGB BZERO 5ESC2SET BONE 5ESCON=:FLAGB % ESCAPE ON
135055
135055          IF LGCOLDSTART><1 AND A=TTNO AND XTMRTERM=0 THEN CALL RSCOLDSTART FI
135067          0=:INLOGGED
135070          IF USLOGOUT><0 THEN A=:L; L:=:P FI         % CALL USER DEFINED LOGOUT ROUTINE
135074          "ABRPAR"; *MON 2ABOR; JMP *
135077   RBUS
135130
135130   %=========================================================================
```

```
135130   %       M B Z M E M O R Y
135130   %
135130   % SUBROUTINE TO CLEAR THE BACKGROUND SEGMENT
135130   %
135130   SUBR MBZMEMORY
135130
135130   DISP -200; INTEGER CBACTPRI; PSID
135130
135130   MBZMEMORY: L=:D; CALL ENTER
135130          IF INLOGGED><0 AND EXSECURITY BIT 7ZMEMORY THEN
135132             BACTPRI=:CBACTPRI
135137             CALL CAON; CALL ZMEMORY; CALL CAOFF; CALL ZMEMORY
135141             CBACTPRI=:BACTPRI
135145
135147          FI; GO LEAVE
135150
135150   RBUS
135156
135156
135156   %=========================================================================
135156   %       R S C O L D S T A R T
135156   %
135156   % SUBROUTINE TO RESET LOGICAL DEVICE NUMBER 1 AS THE "MAIN TERMINAL"
135156   % AFTER A @COLD-START COMMAND.
135156   %
135156   SUBR RSCOLDSTART
135156
135156   DISP -200; INTEGER CSEG; PSID
135156   INTEGER PWSEG:=BFIELD+CSEG
135157   INTEGER PAPRSV:=("1","0")
135161
135161   RSCOLDSTART: L=:D; CALL ENTER
135163          CALL ESCOFF
135164          IF "BPTMP"><0 THEN
135166             T:=MBSPRTAB; X:=ASBPRTAB; "DT01R"; *STATX TXCBP
135172             "PAPRSV"; *MON 2PRLS
135174          FI; LGCOLDSTART; CALL LOGPH; A=:D
135177   @LIB CXCPU
135177          T:=A.TDFPHPAGE; DBPCOLDSTART.WINDOW/\177400\/T=:X.WINDOW
135206   @ELIB
135206          X:=D; T:="DBPROG"; DBPCOLDSTART; CALL XSTDFADDR
135212          1=:TTNO; X:="DT01R"
135215          "BAK01"; T:="DBPROG"; CALL XSTDFADDR
135220   @LIB CXCPU
135220          X.TDFLGADDR/\1777+"SUBFPAGE*2000"=:X
135224   @ELIB
135224          X=:TTIFIELD
135225          BCSEG SHZ -10=:CSEG; "PWSEG"; *MON 2WSEG
135232          IF LGCOLDSTART=XERDEV THEN A:=1; CALL SSETERROR FI
135240          1=:LGCOLDSTART
135242   @LIB CXCPU
135242          T:="DT01R".TDFPHPAGE; RTREF.WINDOW/\177400\/T=:X.WINDOW
135251   @ELIB
135251          GO LEAVE
135252   RBUS
135277
135277
135277   %=========================================================================
135277   % 18.7       C M R F I L E   C M W F I L E
135277
135277   % COMMANDS: RFILE FILENO,ADDRESS,FILE BLOCK NO,NO. OF WORDS
```

```
135277   %            WFILE FILENO,ADDRESS,FILE BLOCK NO,NO. OF WORDS
135277   SUBR CMRFILE,CMWFILE
135277   DISP -200; INTEGER POINTER RWSUBR; INTEGER BLNO; PSID
135277   INTEGER STRFNO:='IOFILE NUMBER: ',STRADD:='IOMEMORY ADDRESS: '
135321   INTEGER STRBLOCK:='IOBLOCK NUMBER: '
135332   INTEGER STRWORDS:='IONO. OF WORDS: '
135343   CMRFILE: "RECOX"; GO RWF
135345   CMWFILE: "DUMPX"
135346   RWF:    L=:D; CALL ENTER; A=;"RWSUBR"
135351           "STRFNO"; CALL GPAR; A=:FILNO; "STRADD"; CALL GPAR; A=:X
135357           "STRBLOCK"; CALL GPAR; A=:BLNO; "STRWORDS"; CALL GPAR
135364           T:=BLNO; CALL RWSUBR
135366           IF A><0 THEN
135367              *MON 64
135370           FI; GO LEAVE
135371   RBUS
135400
135400   %=========================================================================
135400   % 18.8        R E C O V E R   D U M P   C O N T I N U E
135400   %             G O T O U S E R   R E C F I L E   P L P R O G
135400
135400   % COMMANDS: RECOVER FILE
135400   %            DUMP FILE,START ADDRESS,RESTART ADDRESS
135400   %            CONTINUE
135400   %            GOTO-USER ADDRESS
135400   %            PLACE-PROGRAM
135400
135400   SUBR RECOVER,DUMP,CONTINUE,GOTOUSER,RECFILE,PLPROG,ST500,CHSMLGIN,EILCONT
135400
135400   INTEGER E1IN500:='$ERROR IN REENTRANT SUBSYSTEM TABLE$'
135423   INTEGER NF5FREE:='$NO ND-500 PROCESS AVAILABLE'
135442   INTEGER EIN500:='$ERROR DETECTED IN ND-500 SYSTEM, ERROR CODE:'
135471   INTEGER STRSUB:='S SUBSYSTEM NAME: '
135503   INTEGER PLIST:=(FILNO,NULL,BSTART,NULL,"12"),FILTYPE:='PROG'
135513   INTEGER RPLIST:=(FILNO,NULL,BSTART,NULL,"7")
135520   INTEGER PAREE:=NULL
135521   INTEGER TXILCONT:='$COMMAND ILLEGAL WHEN CURRENT PROGRAM IS AN ND-500 PROGRAM$'
135557
135557   DISP -200; INTEGER BEGAD,NMBR,BLADR,APNT,ROUTSWITCH,CSTDINDEX,CINDX,E5COD; PSID
135557   DISP 30; INTEGER IDERO=DERO; PSID
135557
135557   INTEGER PPM60:=("145",BFIELD+CSTDINDEX-1)
135561   *)FILL
135564
135564   ST500: IF PN500D><0 THEN
135566             1XUSTATUS BONE 1ILLCONTINUE=:1XUSTATUS
135571             X.S2=:CSTDINDEX; "PPM60"; *MON 60
135575             GO ERR5; GO 50PCOM
135577          FI; A:=-1=:D; "E1IN500"; GO E5FEL
135603   ERR5:  IF A=:D=2016 THEN "NF5FREE" ELSE "EIN500" FI
135612   E5FEL: T:=D=:TTIFIELD.IDERO=:E5COD:=1=:FLQERM; CALL LOUTTEXT
135621          E5COD; CALL OCTUT; ##B; CALL TCO; GO 50PCOM
135626
135626   CH2BANK: IF LDDAD<<=HDDAD AND 2BXAD=0 THEN ER185; EXIT FI
135636            EXITA
135637
135637   PLPROG:  L=:D; CALL ENTER; A:=2; GO FELLS
135643   RECFILE: L=:D; CALL ENTER; A:=0; GO FELLS
135647   RECOVER: L=:D; CALL ENTER; A:=1
135652   FELLS: A=:ROUTSWITCH
```

```
135653          "STRSUB"; CALL GPAR; A=:CSTRING
135656   FELL2: IF ROUTSWITCH=0 GO RECF
135660          T:=CSTRING; X:=0; *LBYT
135663          IF =##( GO RECF; CPNT=:APNT; 0=:CPNT
135671          X:="REECOMTAB"; CALL ABLOOK; T=:CINDX:=APNT=:CPNT
135676          IF =0 THEN      % FOUND, REENTRANT SUBSYSTEM
135677                UEFLG BZERO 5UECM BZERO 5UEIE=:UEFLG
135703                1XUSTATUS BZERO 1ILLCONTINUE=:1XUSTATUS
135706                CALL LAMDISCONNECT
135707                IF ROUTSWITCH=3 THEN CALL XMLGRSTART FI
135714                IF CINDX.S1><-1 THEN X.S2=:BSTART; X.S3=:BRESTART FI
135725                IF ROUTSWITCH=2 THEN
135731                   IF X.S1><-1 THEN CALL REEC FI; GO LEAVE
135737                ELSE
135740                   IF X.S1=-1 GO ST500                % ND-500 STANDARD DOMAIN
135744                   CALL SREEC                         % CONTINUE ON SYS.SEG.
135745                FI
135745          ELSE
135746             IF =-2 THEN
135751                IF ROUTSWITCH=3 GO MLGERR
135755                ER212; *MON 64
135757                GO LEAVE
135760             FI
135760          FI; GO RECF
135761          *)FILL
136011
136011   LFILERR:     A=:D
136012                IF ROUTSWITCH=3 THEN
136016   MLGERR:         -1; CALL MLGRSTART; GO LOGOUT
136021                FI; A:=D; GO FILERR
136023   XFILERR:     A=:D; IF ROUTSWITCH=3 GO MLGERR; A:=D; *MON 64
136032                GO LEAVE
136033   LOUTTEXT:    A=:D; IF ROUTSWITCH=3 GO MLGERR; A:=D; GO OUTTEXT
136042   XMLGRSTART; A:=0; GO MLGRSTART
136044
136044   % NOT REENTRANT, TRY A PROGFILE:
136044   RECF:  X:=CSTRING; "FILTYPE"; T:=3; *MON 2NOPE
136050          GO OPENERR; A=:FILNO; CALL LOGPH
136053          IF A=0 THEN A:=133; GO XFILERR FI
136056          IF A.TYPRING/\133770><0 THEN A:=133; GO XFILERR FI
136064          T:=FILNO; "PAREE"; *MON 2REEN
136067          CALL LAMDISCONNECT
136070          "RPLIST"; *MON 2RFIL
136072          IF><0 GO XFILERR; LDADR=:BEGAD; CALL CHMEM; HDADR; CALL CHMEM
136100          A:=FILNO; CALL LOGPH
136102          IF A.TYPRING BIT 5COM THEN
136106            CALL FAR CH2BANK; GO XFILERR
136110            T:=FILNO
136111            FOR X:=-754 DO
136112                *MON 2INBT
136113                GO XFILERR
136114            OD
136115          FI; UEFLG BZERO 5UECM BZERO 5UEIE=:UEFLG
136121          1XUSTATUS BZERO 1ILLCONTINUE=:1XUSTATUS
136124          IF ROUTSWITCH=3 THEN CALL XMLGRSTART FI
136131          T:=FILNO; X:="BSTART"
136133          IF ROUTSWITCH=2 THEN
136137             T:=FILNO; X:="BSTART"; CALL 2BRECOVER; GO LFILERR
136143          ELSE
136144             T:=FILNO; X:="BSTART"; CALL 2BSRECOVER; GO LFILERR
```

```
136150          FI; T:=FILNO; *MON 2CLOS
136152          GO LFILERR; GO LEAVE
136154
136154   OPENERR: A=:X; ##"; CALL TCO; "WORKA"; CALL OUTTEXT
136161          ##"; CALL TCO
136163          IF ROUTSWITCH=3 GO FAR MLGERR
136167          A:=X; *MON 64
136171          GO LEAVE
136172   *)FILL
136216
136216   CONTINUE: L=:D; CALL ENTER
136220             IF 1XUSTATUS BIT 1ILLCONTINUE GO EILCONT
136223             BRESTART; GO RCOM
136225   GOTOUSER: L=:D; CALL ENTER
136227             X:="ESCBLOCK"; "STRNUM"; CALL AGPAR; X.RPREG
136233             IF T:=1XUSTATUS BIT 1ILLCONTINUE GO EILCONT
136236             GO RCOM
136237
136237   EILCONT: "TXILCONT"; CALL OUTTEXT; GO LEAVE
136242   *)FILL
136252
136252   CHSMLGIN: L=:D; CALL ENTER
136254          IF "MLIDFIELD".MLICPNT<<X.MLIMXADDR-2 THEN
136262             D:=D; X:=A+1; 0=:CPNT
136265             DO WHILE D<<100
136270                T:="MLIDFIELD"; *LBYT
136272                X+1:=:D; T:="COMSTRING"; *SBYT
136276                X+1:=:D
136300                IF A=40 OR A=15 THEN
136306                    IF CPNT=0 THEN A:=D-1=:CPNT FI; FI
136313             OD; "COMSTRING"=:CSTRING; 3=:ROUTSWITCH; GO FAR FELL2
136321          FI; GO LEAVE
136322
136322   % DUMP COMMAND:
136322   DUMP:  L=:D; CALL ENTER; "STRFILE"; CALL GPAR; A=:X:="FILTYPE";T:=2;*MON 2NOPE
136332          GO CFILERR; A=:FILNO; CALL LOGPH
136335          IF D=0 THEN A:=133; GO CFILERR FI
136341          IF D.TYPRING/\44000=0 THEN A:=133; GO CFILERR FI
136347          "STSTART"; CALL GPAR; A=:BSTART
136352          "STRESTART"; CALL GPAR; A=:BRESTART
136355          T:=FILNO; "PLIST"; *MON 2WFIL
136360          IF ><0 GO CFILERR; LDADR=:BEGAD; -77=:BLADR
136365          A:=FILNO; CALL LOGPH
136367          IF D.TYPRING BIT 5COM THEN
136373            CALL FAR CH2BANK; GO CFILERR
136375            T:=FILNO; A:=-1
136377            FOR X:=-754 DO
136400                *MON 2OUTB
136401                GO CFILERR
136402            OD
136403          FI; T:=FILNO; X:="BSTART"; CALL 2BDUMP; GO CFILERR
136407          T:=FILNO; *MON 2CLOS
136411          GO CFILERR; GO LEAVE
136413
136413   CFILERR: *MON 64
136414          GO LEAVE
136415   RBUS
136434
136434
136434   %=========================================================================
```

```
136434   % 18.9        L O A D   P L A C E
136434   %
136434   % COMMANDS: LOAD-BINARY FILE
136434   %            PLACE-BINARY FILE
136434
136434   SUBR LOAD,PLACE
136434   INTEGER CHS:='CHECKSUM ERROR',FILTYPE:='BPUN'
136447   DISP -200; INTEGER LOADFLAG,CHAR,CHSUM; INTEGER POINTER LREG; PSID
136447
136447   % AUXILIARY ROUTINES:
136447   RCHAR: T:=FILNO; *MON 2INBT
136451          GO CFILERR; EXIT
136453   RWORD: T:=FILNO; *MON 2INBT
136455          GO CFILERR; A SH 10=:CHAR; *MON 2INBT
136461          GO CFILERR; A+CHAR; EXIT
136464   ROCT:  T:=L=:"LREG"; D:=0
136467          FOR X:=-6 DO
136470   LOOPR:        CALL RCHAR; A BZERO 7-60; IF <0 OR >7 GO LOOPR; D SH 3+A
136501          OD; D=:A; GO LREG
136504
136504   LOAD:  T:=1; GO LPL
136506   PLACE: T:=0
136507   LPL:   L=:D; CALL ENTER; T=:LOADFLAG
136512          IF 1XUSTATUS BIT 1ILLCONTINUE GO EILCONT
136515          "STRFILE"; CALL GPAR; A=:X:="FILTYPE"; T:=1; *MON 2NOPE
136523          GO CFILERR; A=:FILNO; CALL ESCON
136526          KBACTPRI=:BACTPRI
136530          IF LOADFLAG><0 THEN
136532                DO CALL ROCT WHILE ><125001; OD; CALL ROCT
136540                CALL CHMEM; A=:"ESCBLOCK".RPREG
136543                X.RSREG BZERO "0"=:X.RSREG
136546          FI; UEFLG BZERO 5UECM=:UEFLG
136551          CALL LAMDISCONNECT
136552          1XUSTATUS BZERO 1ILLCONTINUE=:1XUSTATUS
136555          DO CALL RCHAR WHILE A/\177><##!; OD
136563          CALL RWORD; A=:LDADR; CALL CHMEM
136566          CALL RWORD; A=:X+LDADR-1; CALL CHMEM; A=:HDADR
136574          -1=:LDDAD=:2BXADR; 0=:HDDAD
136600          X=:A:=LDADR; CALL LODX     % SYSTEM SEGMENT
136603          A=:CHSUM; CALL RWORD
136605          IF ><CHSUM THEN 0=:LOADFLAG; "CHS"; CALL OUTTEXT FI
136613          CALL ESCOFF; T:=FILNO; *MON 2CLOS
136616          GO CFILERR; IF LOADFLAG><0 GO STUSER; GO LEAVE
136623   CFILERR: *MON 64
136624          GO LEAVE
136625   RBUS
136647
136647   %==========================================================================
136647   % 18.10       M E M O R Y
136647   %
136647   % COMMAND:MEMORY LOWER ADDR.,UPPER ADDR.
136647
136647   %COMMAND TO SET MEMORY LIMITS FOR DUMP
136647   SUBR MEMORY
136647   MEMORY: L=:D; CALL ENTER; "STRLOW"; CALL GPAR; CALL CHMEM
136654           A=:LDADR; "STRHIG"; CALL GPAR; CALL CHMEM; A=:HDADR
136661           -1=:LDDAD=:2BXADR; 0=:HDDAD
136665           GO LEAVE
136666   RBUS
136672
```

```
136672   %=========================================================================
136672   % 18.11      S E T M E M     Z M E M O R Y
136672   %
136672
136672   % COMMAND: SET-MEMORY-CONTENTS <CONTENTS> <LOW ADDRESS> <HIGH ADDRESS>
136672   %
136672   SUBR SETMEM,ZMEMORY
136672   INTEGER STRCON:='IOCONTENTS: '
136701   DISP -200; INTEGER VALUE,CUPAD; PSID
136701   ZMEMORY: L=:D; CALL ENTER
136703           X:=0; -1=:CUPAD;0=:VALUE; GO FELLS
136710   SETMEM: L=:D; CALL ENTER
136712           "STRCOM"; CALL GPAR; A=:VALUE
136715           "STRLOW"; CALL GPAR; A=:X; "STRHIGH"; CALL AGPAR; -1; A=:CUPAD
136724   FELLS: BACTPRI=:D; CALL DALTON; *BSET ZRO
136730           T:=BCSEGM; CUPAD=:D; VALUE; CALL USET; CALL ALTOFF
136736           GO LEAVE
136737   RBUS
136750
136750   %=========================================================================
136750   % 18.12      X R T L O A D   Y R T L O A D   Z R T L O A D   W R T L O A D
136750   %           P Y R T L O A D   P Z R T L O A D
136750   %
136750   % XRTLOAD:   RT-LOADER COMMAND
136750   %      ENTRY:      NO PARAMETERS
136750   %      RETURN:     NO RETURN INFORMATION
136750   % YRTLOAD:   DUMP-REENTRANT COMMAND
136750   %      ENTRY:      A=POINTER TO SEGMENT NAME
136750   %                  T=OPEN FILE NUMBER
136750   %      RETURN:     A=SEGMENT NUMBER (=0: ERROR)
136750   % ZRTLOAD:   DELETE-REENTRAN AND CLEAR-REENTRANT-SEGMENT COMMANDS
136750   %      ENTRY:      A=SEGMENT NUMBER
136750   %      RETURN:     A=0: OK, A><0: ERROR
136750   % WRTLOAD:   CANGE-SEGMENT-SIZE COMMAND
136750   %      ENTRY:      A=SEGMENT NUMBER
136750   %                  X=SEGMENT SIZE
136750   %      RETURN:     A=0: OK; A><0: ERROR
136750   % PYRTLOAD:  DUMP-PROGRAM-REENTRANT COMMAND
136750   %      ENTRY:      T=OPEN FILE NUMBER
136750   %                  A=POINTER TO SEGMENT NAME
136750   %      RETURN:     A=SEGMENT NUMBER (=0: ERROR)
136750   %                  T=START SDDR.
136750   %                  D=RESTART ADDR.
136750   % PZRTLOAD:  LOAD-REENTRANT-SEGMENT COMMAND
136750   %      ENTRY:      T=OPEN FILE NUMBER
136750   %                  A=POINTER TO SEGMENT NAME
136750   %      RETURN:     A=SEGMENT NUMBER (=0; ERROR)
136750
136750   % RETURN: A=SEGMENT NO.
136750   SUBR XRTLOAD,YRTLOAD,ZRTLOAD,WRTLOAD,PYRTLOAD,PZRTLOAD
136750   INTEGER MNOS1:='$***WARNING: MAKE NO NEW SEGMENTS BEFORE THE '
136777   INTEGER MNOS2:='$   INITIALIZE-BACKGROUND-PROGRAMS COMMAND IS USED$'
137031   INTEGER MNOS3:='N CONTINUE THE COMMAND? '
137046
137046   DISP -200; INTEGER TREG,AREG,DREG; TRIPLE STAD=TREG; PSID
137046
137046   XRTLOAD:  L=:D; CALL ENTER; TAD=:STAD; T:="RTLOADER"; GO FELL
137053   ZRTLOAD:  L=:D; CALL ENTER; TAD=:STAD; T:="RRRTLOADER"; GO FELL
137060   WRTLOAD:  L=:D; CALL ENTER; TAD=:STAD; T:="RRRRTLOADER"; GO FELL
137065   PYRTLOAD: L=:D; CALL ENTER; TAD=:STAD; T:="RRPRDUMP"; GO FELL1
```

```
137072   PZRTLOAD: L=:D; CALL ENTER; TAD=:STAD; T:="RRPRLOAD"; GO FELL1
137077   YRTLOAD:  L=:D; CALL ENTER; TAD=:STAD; T:="RRTLOADER"
137103   FELL1: T=:DREG
137104          IF XTMRTERM><0 THEN
137106                "MNOS1"; CALL OUTTEXT; "MNOS2"; CALL OUTTEXT
137112                "MNOS3"; CALL GPAR; CALL YESNO; IF A><1 GO OPCOM
137120          FI; GO FELL2
137121   FELL:  T=:DREG
137122   FELL2: CALL CRSRTL; GO LEAVE
137124          TAD:=STAD; CALL TORTLOADER
137126          TAD=:STAD; CALL CRLRTL
137130          TAD:=STAD; GO LEAVE
137132   RBUS
137152
137152   %=============================================================================
137152   % 18.14        C S T A T U S
137152
137152   % COMMAND: STATUS
137152   %
137152   % COMMAND TO WRITE REGISTER STATUS
137152   SUBR CSTATUS
137152   INTEGER REGNAME:='PXTADLSB'
137157   CSTATUS: L=:D; CALL ENTER
137161          FOR X:=0 TO 7 DO
137165                T:="REGNAME"; *LBYT
137167                CALL TCO; ##=; CALL TCO; ESCBLOCK(X); CALL OCTU; CALL CRLF
137175          OD; CALL LEAVE
137200   RBUS
137205
137205   %=============================================================================
137205   % 18.15        D A T C L
137205
137205   % COMMAND: DATCL
137205   %
137205   % COMMAND FOR WRITING DATE AND CLOCK
137205   SUBR DATCL,PDATCL
137205   INTEGER JAN:=' JANUARY',FEB:=' FEBRUARY',MAR:=' MARCH',APR:=' APRIL'
137227   INTEGER MAY:=' MAY',JUN:=' JUNE',JUL:=' JULY',AUG:=' AUGUST'
137244   INTEGER SEP:=' SEPTEMBER',OCT:=' OCTOBER',NOV:=' NOVEMBER',DEC:=' DECEMBER'
137271   INTEGER ARRAY MONA:=(JAN,JAN,FEB,MAR,APR,MAY,JUN,JUL,AUG,SEP,OCT,NOV,DEC)
137306   DOUBLE POINTER TIMOF:=WORKA+11
137307   INTEGER PARAM:=V0
137310   DISP -200; INTEGER BUNI,SEC,MINUTE,HOUR,DAY,MNTH,YEAR; PSID
137310   PDATCL: L=:D; CALL ENTER;                  % ENTRY POINT TO PRINT TIME AND DATE
137312           0=:BUNI=:SEC; GENDAT(0)=:MINUTES; GENDAT(1)=:HOUR;
137322           GENDAT(2)=:DAY; GENDAT(3)=:MNTH; GENDAT(4)=:YEAR; GO L1
137334   DATCL: L=:D; CALL ENTER; "PARAM"; *MON 2CLOC
137340      L1: A:=40; CALL TCO; A:=HOUR; CALL DEC2; ##.; CALL TCO; A:=MINUTE; CALL DEC2
137350           ##.; CALL TCO; A:=SEC; CALL DEC2; A:=DAY; CALL DECU
137356           A:=MONA(MNTH); CALL OUTTEXT; A:=YEAR; CALL DECU
137363           D:=0; A:=YEAR-3554/\77; AD SHR 4; A+MNTH; AD SHR 5; A+DAY; AD SHR 5
137374           A+HOUR; AD SHR 6; A+MINUTE; AD SHR 6; A+SEC:=:D; AD=:TIMOF; GO LEAVE
137404   RBUS
137414
137414   %=============================================================================
137414   % 18.16        C O P Y F
137414
137414
137414   %        COMMAND: COPY DESTINATION DEVICE,SOURCE DEVICE
137414   %
```

```
137414   %       COMMAND TO COPY TO FILE FROM FILE
137414
137414   SUBR COPYF
137414
137414   INTEGER TODEV:='S TO DEVICE: ',FRDEV:='S FROM DEVICE: '
137433
137433   DISP -200
137433   INTEGER DSFILE                  % DESTINATION FILE NUMBER.
137433   INTEGER SRFILE                  % SOURCE FILE NUMBER.
137433   INTEGER ORIER                   % ORIGINAL, OR SAVED ERROR.
137433   PSID
137433
137433   COPYF: L=:D; CALL ENTER
137435          0=:DSFILE=:SRFILE
137437          "TODEV"; CALL GPAR; A=:X:="TYPS"; T:=0; *MON 2NOPE
137445          GO OPENERR; A=:DSFILE
137447          "FRDEV"; CALL GPAR; A=:X:="TYPS"; T:=1; *MON 2NOPE
137455          GO OPENERR; A=:SRFILE; CALL ESCON
137460          DO     T:=SRFILE; *MON 2INBT
137462                 GO OUTER; T:=DSFILE; *MON 2OUTB
137465                 GO OUTER
137466          OD
137467   OPENERR: A=:ORIER; ##"; CALL TCO; "WORKA"; CALL OUTTEXT
137474          ##"; CALL TCO; GO OUT
137477   OUTER: A=:ORIER
137500   OUT:   CALL ESCOFF; T:=DSFILE; *MON 2CLOS; JMP * 1
137504          T:=SRFILE; *MON 2CLOS; JMP * 1
137507          A:=ORIER;  *MON 2ERMS
137511          GO LEAVE
137512   RBUS
137522
137522
137522   %==========================================================================
137522   % 18.17       L I S T T Q   L I S T X Q
137522
137522
137522   % COMMAND: LIST-TIME-QUEUE
137522   %          LIST-EXEC-QUEUE
137522   %
137522   SUBR LISTTQ,LISTXQ
137522   LISTTQ: L=:D; CALL ENTER; CALL TQCOPY; GO OUTP
137526
137526   LISTXQ: L=:D; CALL ENTER; X:="BEXQU-5BWLINK"; CALL XQCOPY
137532   OUTP:  FOR X:=0 TO "5WORKSIZE-1" DO
137536                IF WORKA(X)=-1 GO LEAVE; CALL RTOUT;CALL CRLF
137544          OD
137546   RBUS
137554
137554
137554   %==========================================================================
137554   % 18.18       L I S T S E G
137554
137554   %COMMAND: LIST-SEGMENT <SEG. NO.>
137554   SUBR LISTSEG
137554   @ICR
137554   INTEGER XFPA:='$FIRST PAGE: ',XL:=' LENGTH: ',XMS:=' MASS. ADR: ',
137577           XWP:='WPM ',XRP:='RPM ',XFP:='FPM ',XWI:='WIP ',XPG:='PGU ',
137616           XR2:='RING2 ',XR1:='RING1 ',XIN:='INHIB ',XFIX:='FIX ',
137635           XDM:='DEMAND ',XOK:='OK',XUN:='SEG.FILE: ',XPRO:='PROTECT ',
137656           XREEP:='REE-SUB ';
```

```
137663   INTEGER ARRAY FBITS:=(XWP,XRP,XFP,XWI,XPG,XR2,XR1,0,
137673          0,XREEP,XPRO,0,XIN,XFIX,XDM,XOK);
137703   @CR;
137703   INTEGER BADSEG:='ILLEGAL SEGMENT NUMBER'
137717   DISP -200; INTEGER WORD; PSID
137717
137717   LISTSEG: L=:D; CALL ENTER
137721          "GGSGM"; CALL GPAR; IF <=0 OR >SGMAX GO ERR
137730          A*5SEGSIZE+SEGSTART=:X;"WORKA"; T:=5; CALL COPYB
137736          "XFPA"; CALL OUTTEXT; X.LOGADR/\377; CALL OCTU
137743          "XL";  CALL OUTTEXT; X.LOGADR SHZ -10; CALL OCTU; CALL CRLF
137751          "XUN"; CALL OUTTEXT; X.MADR SHZ -16; CALL OCTU
137756          "XMS"; CALL OUTTEXT; X.MADR /\37777; CALL OCTU; CALL CRLF
137764          X.FLAG=:WORD
137766          FOR X:=0 TO 17 DO
137772                IF WORD<0 AND FBITS(X)><0 THEN CALL OUTTEXT FI;
137777                WORD SH 1=:WORD
140002          OD
140004          CALL CRLF; GO LEAVE
140006   ERR:   "BADSEG"; CALL OUTTEXT; GO LEAVE
140011   RBUS
140032
140032   %=========================================================================
140032   % 18.23        T I M O U T
140032   %
140032   %SUBROUTINE TO WRITE MINUTES AND SECONDS; AD=TIME IN BASIC UNITS
140032   SUBR TIMOUT
140032   INTEGER XMIN:=' MINS',XSEC:=' SECS'
140040   DISP -200; INTEGER REM; PSID
140040   TIMOUT: D=:T:=L; CALL ENTER; T=:D:=9TIM2; *RDIV ST
140046          T:=D=:REM; IF A><0 THEN CALL DTDEC; "XMIN"; CALL OUTTEXT FI
140054          REM=:D:=0; T:=9TIM1; *RDIV ST
140061          CALL DTDEC; "XSEC"; CALL OUTTEXT; GO LEAVE
140065   RBUS
140072
140072   %=========================================================================
140072   % 18.24        L I D E V
140072
140072   % COMMAND: LIST-DEVICE <LOG. NO.> <I/O-FLAG>
140072   %
140072   SUBR LIDEV
140072   DISP -200; INTEGER LOGNO; PSID
140072   INTEGER CIOLOG:='IDLOG. UNIT: ',STRIO:='IOINPUT/OUTPUT(0 OR 1): '
140072   INTEGER RESBY:='RESERVED BY: ',WTPR:='WAITING RT-PROGRAMS:$'
140116   INTEGER NORE:='NOT RESERVED',NODEV:='NO SUCH DEVICE'
140140   LIDEV: L=:D; CALL ENTER; "CIOLOG"; CALL GPAR; A=:LOGNO
140157                                                  % ILLEGAL ON FILE
140164          IF A/\177700=100 GO ERR
140170          "STRIO"; CALL GPAR; A=:X; LOGNO; CALL LOGPH
140175          IF X><0 THEN D=:A FI; IF =0 GO ERR; A=:X
140201          IF X.RTRES><0 THEN
140203                "RESBY"; CALL OUTTEXT; X.RTRES; CALL RTOUT
140207          ELSE  "NORE"; CALL OUTTEXT
140212          FI; CALL CRLF
140213          X=:A; CALL XQCOPY
140215          IF WORKA(0)><-1 THEN "WTPR"; CALL OUTTEXT
140224                FOR X TO "5WORKSIZE-1" DO WHILE WORKA(X)><-1
140233                  CALL RTOUT; CALL CRLF
140235                OD
140237          FI; GO LEAVE
140240   ERR:   "NODEV"; CALL OUTTEXT; GO LEAVE
```

```
140243  RBUS
140260
140260  %=========================================================================
140260  % 18.25      L R T P R O G
140260
140260  % COMMAND: LIST-RT-PROGRAMS <OUTPUT FILE>
140260  %
140260  % ALL RT-PROGRAMS IN RT-FIL WILL BE LISTED
140260  SUBR LRTPROG
140260  @1CR
140260  INTEGER HEADING:=
140260  '  NAME  RT-DESC  PRIOR STATUS   P-REG T.LEFT INTERV   ACTUAL SEGM$$',
140322  RWAI:=' RTWAIT ',PASSIVE:=' PASSIVE',XIOW:=' IO-WAIT',XREA:=' READY  ',
140346  XWQU:=' W-QUEUE',SPACES:='        ',XRTOF:=' RTOFF';
140363  @CR;
140363  DISP -200
140363          INTEGER ELNO,RTD,WTDEVICE
140363          DOUBLE DACC2; INTEGER POINTER LREG
140363  PSID
140363
140363  % LOCAL SUBROUTINE FOR DOUBLE PREC. SUBTRACTION
140363  % AD-DACC2=:AD
140363  DSUB:   A:=:L=:"LREG"; D=:T; DACC2; D-; *COPY CM1 ADC SA DA
140371          D+T; A+L+C; GO LREG
140374
140374  % LOCAL SUBROUTINE TO PRINT NO.OF SECONDS
140374  % AD=BASIC UNITS
140374  TIMS:   D=:T:=L; CALL ENTER; T=:D:=9TIM1; *RDIV ST
140402          CALL DECU; GO LEAVE
140404
140404  % MAIN ENTRYPOINT:
140404  LRTPROG: L=:D; CALL ENTER
140406          CALL ROFIPAR; GO LEAVE
140410          "HEADING"; CALL OUTTEXT; RTSTART=:ELNO
140414          DO X:=ELNO WHILE X<SEGSTART
140420                "ELNO"+B; CALL RTFELEM; GO NEXT
140424                CALL ESCON; CALL ESCOFF
140426                X=:RTD; D=:X; CALL PRIRT; ELNO=:X; CALL OCTU
140434                CALL RTDCOPY; X=:WTDEVICE; "WORKA".STATUS/\377; CALL DECU
140442                IF X.WLINK=0 THEN
140444                  IF X.STATUS BIT 5RWAIT THEN "RWAI" ELSE "PASSIVE" FI
140452                ELSE
140453                  IF X.STATUS BIT 5WAIT THEN "XIOW"
140457                  ELSE IF WTDEVICE<="BEXQU" THEN "XREA" ELSE "XWQU" FI
140467                FI FI; CALL OUTTEXT
140470                X=:D:=X.RTDLGADDR; T:=0; *LDATX
140474                CALL OCTU
140475                IF RTD.TLINK=0 THEN "SPACES"; CALL OUTTEXT
140502                ELSE MTIME=:DACC2; X.DTIME; CALL DSUB; CALL TIMS
140510                FI
140510                IF X.STATUS NBIT 5INT THEN "SPACES"; CALL OUTTEXT
140515                ELSE X.DTINT; CALL TIMS
140520                FI
140520                X.ACTSEG SHZ -10; CALL OCTU; X.ACTSEG/\377; CALL OCTU
140526                IF X.ACTPRI BIT 5RTOF THEN "XRTOF"; CALL OUTTEXT FI
140533                CALL CRLF
140534  NEXT:         ELNO+5RTSIZE=:ELNO
140537          OD
140540  OUT:    T:=TDVN; *MON 2CLOS; 0
140543          GO LEAVE
```

```
140544   RBUS
140576
140576
140576   %=========================================================================
140576   % 18.27       S T O P T E R M
140576
140576   %COMMAND: STOP-TERMINAL <LOG. NO.>
140576   %
140576   SUBR STOPTERM
140576   INTEGER NOTERM:='$NO TERMINAL'
140605   STOPTERM: L=:D; CALL ENTER; "STRTERM"; CALL GPAR
140611           CALL LOGPH; CALL CHTERM; GO ERR
140614           IF A.TYPRING NBIT 5IOBT GO ERR
140620           T:="DBPROG"; CALL XGTDFADDR
140622           IF A=D GO ERR; X=:D                          % CONTINUE ON FIRST PAGE ON SEGMENT
140624           CALL XSTOPTERM; GO LEAVE
140626   ERR:    "NOTERM"; CALL OUTTEXT; GO LEAVE
140631   RBUS
140641
140641
140641   %=========================================================================
140641   % 18.28       C H T E R M
140641
140641   % SUBROUTINE TO CHECK LOG. NO. OF A TERMINAL
140641   % ENTRY:      A=DATAFIELD
140641   % RETURN:     ERROR
140641   % SKIPRETURN:OK
140641   SUBR CHTERM
140641   CHTERM: L=:D; CALL ENTER
140643           IF A><0 THEN X:="BACKTAB"; A=:D
140646                 DO WHILE X.S0><-1
140652                  IF A=D GO LEAV2; X+1    %OK
140655                 OD; D=:A
140657           FI; GO LEAVE       %NO TERMINAL
140660   RBUS
140662
140662
140662   %=========================================================================
140662   % 18.29       E N T R T
140662
140662   % COMMAND: RTENTER
140662   %
140662   % ENTER USER "RT" FOR RT-PROGRAMS
140662   %
140662   SUBR ENTRT
140662   DOUBLE RTUS:='RT'
140664   DISP -200; DOUBLE USRT; PSID
140664   ENTRT: L=:D; CALL ENTER
140666           RTUS=:USRT; X:=-200+B                        % MOD 25/6 FOR 4096 USERS
140672           CALL ENTRCORE
140673   ERR:    *MON 64
140674           GO LEAVE
140675   RBUS
140676
140676   %=========================================================================
140676   % 18.30       W H O I S O N
140676
140676   %COMMAND: WHO-IS-ON
140676   %
140676   SUBR WHOISON
```

```
140676   DISP -200
140676           INTEGER WHSEG,TERMNO,XUSNO,MYSELF; INTEGER POINTER TPNT
140676           INTEGER CDBPROG,CBSTATE
140676   PSID
140676   INTEGER THISTR:='===>'
140701   INTEGER NOTME:='    '
140704   WHOISON: L=:D; CALL ENTER; "BACKTAB"=:"TPNT"
140710           DO IF X:=TPNT=-1 GO OUT
140714              IF X><0 THEN
140715                 T:="BSTATE"; CALL XGTDFADDR; A=:CBSTATE
140720                 T:="DBPROG"; CALL XGTDFADDR; A=:CDBPROG
140723                 CALL CHTACTIVE; GO NXT
140725                 IF CBSTATE><0 AND A><5CFILTRA AND CDBPROG><0 THEN
140734                    IF A=CURPROG THEN 1=:MYSELF ELSE 0=:MYSELF FI
140743                    T:=CDBPROG.SEGM SHZ-10=:WHSEG; X:="TTNO"
140750                    CALL GET1L; GO OUT; A=:TERMNO
140753                    T:=WHSEG; X:="CUSER"; CALL GET1L; GO OUT
140757                    IF >=0 THEN
140760                       A=:XUSNO; CALL CRLF
140762                       IF MYSELF=0 THEN
140764                          "NOTME"; CALL OUTTEXT
140766                       ELSE
140767                          "THISTR"; CALL OUTTEXT
140771                       FI; TERMNO; CALL DECU; "20040"; CALL TCO2
140775                       #''=:WORKA(10); T:=XUSNO SHZ -10; A:=XUSNO/\377  % MOD FOR 4096
141004                       X:="WORKA"; CALL FILSYS(GUSEN); GO CFILERR
141010                       "WORKA"; CALL OUTTEXT
141012                    FI
141012                 FI
141012   NXT:        FI; MIN "TPNT"
141013           OD
141014   OUT:    GO LEAVE
141015   CFILERR: *MON 64
141016           GO LEAVE
141017   RBUS
141042
141042
141042
141042   %=========================================================================
141042   % 18.31        G E T R N
141042
141042   % COMMAND: GET-RT-NAME RT-DESCRIPTION ADDRESS
141042   %
141042
141042   SUBR GETRN
141042   GETRN: L=:D;CALL ENTER;"STRNUM";CALL GPAR
141046           IF =0 THEN RTREF FI; CALL RTOUT; GO LEAVE
141052   RBUS
141056
141056   %=========================================================================
141056   % 18.32        C C D U M
141056
141056   % COMMAND: CC - COMMENT
141056   %
141056   SUBR CCDUM
141056   CCDUM: EXIT
141057   RBUS
141057
141057
141057   %=========================================================================
```

```
141057   % 18.33      S E T U N   S E T A V   U N E X T R A
141057
141057   %`COMMAND: SET-UNAVAILABLE MESSAGE
141057   %           SET-AVAILABLE
141057   %
141057   INTEGER UNEXTRA:=#''
141060   SUBR SETUN,SETAV
141060   INTEGER DUMBUF(5WCBUFSIZE); DATA(#'')
141145   INTEGER STPAR:='N EXTRA TEXT: '
141155   INTEGER PWSBC:="5OPSEG"
141156   SETUN: L=:D; CALL ENTER; X:=0; "STPAR"; CALL AGPAR; GO OUT
141164          FOR X:=0 TO "5CBUFSIZE-1" DO
141170                CALL CREAD; T:="UNEXTRA"; *SBYT
141173                WHILE ><15
141176          OD
141200   OUT:   ##'; T:="UNEXTRA"; *SBYT
141203          1=:UNAFLAG; A:="PWSBC"; *MON 2WSBC
141207          GO LEAVE
141210
141210   SETAV: 0=:UNAFLAG=:EXUNAFLAG; EXIT
141213   RBUS
141223
141223
141223   %=========================================================================
141223   % 18.34      S E T E R R O R
141223
141223   % COMMAND: SET-ERROR-DEVICE <LOG. UNIT>
141223   %
141223   SUBR SETERROR,GETERROR,SSETERROR
141223   DISP -200; INTEGER LOGN,LUNADDR,CDFADDR; PSID
141223   INTEGER STLOG:='IDLOG. UNIT: ',ILLD:='ILLEGAL UNIT',PWSBC:="5ERRSEG"
141223   INTEGER IMPAR:="5CIMSEG",CERRD:='ERROR DEVICE:'
141242   SSETERROR: L=:D; CALL ENTER; GO FELLS
141252   SETERROR: L=:D; CALL ENTER
141255          "STLOG"; CALL GPAR
141257   FELLS: A=:LOGN
141261          CNVRT(0)+6=:LUNADDR
141262          LOGN; CALL LOGPH; X:=D=:CDFADDR
141266          IF X><0 AND X.TYPRING BIT 5IOBT THEN A BONE 5NORESRV=:X.TYPRING
141272                IF A BIT 5TERM THEN T:="TYPRING"; CALL XSTDFADDR FI
141300                A:=X+"TYPRING"=:X; T:=5CIMSEG; CALL GET1L; CALL ERRFATAL
141304                A BONE 5NORESERV; CALL PUT1L; CALL ERRFATAL
141312                X:="TTNO"; T:=5ERRSEG; CALL GET1L; CALL ERRFATAL
141315                IF ><LOGN AND A><1 THEN CALL LOGPH; IF D=0 THEN CALL ERRFATAL FI
141321                  D.TYPRING BZERO 5NORESRV=:X.TYPRING
141333                  IF A BIT 5TERM THEN T:="TYPRING"; CALL XSTDFADDR FI
141337                  A:=X+"TYPRING"=:X; T:=5CIMSEG; CALL GET1L; CALL ERRFATAL
141343                  A BZERO 5NORESERV; CALL PUT1L; CALL ERRFATAL
141351
141354                FI
141354                X:="TTNO"; LOGN; T:=5ERRSEG; CALL PUT1L; CALL ERRFATAL
141361                "PWSBC"; *MON 2WSEG
141363                LOGN=:XERDEV; T:=5CIMSEG; X:="XERDEV"; CALL PUT1L; CALL ERRFATAL
141371                X:=LUNADDR; T:=0; CDFADDR; *STATX
141375                T:=5CIMSEG; CALL PUT1L; CALL ERRFATAL
141400                "IMPAR"; *MON 2WSEG
141402          ELSE "ILLD"; CALL OUTTEXT
141405          FI; GO LEAVE
141406   *FILL
141407
141407   GETERROR: L=:D; CALL ENTER
```

```
141411          "CERRD"; CALL OUTTEXT; X:="TTNO"; T:=5ERRSEG; CALL GET1L; CALL ERRFATAL
141417          CALL DECU; GO LEAVE
141421
141421   RBUS
141443
141443   %==========================================================================
141443   % 18.35      I N C O M   N E X I N C O M   L I I C O M   I N I B U F
141443
141443   % COMMAND: INITIAL-COMMAND <STRING>
141443   %          NEXT-INITIAL-COMMAND <STRING>
141443   %          LIST-INITIAL-COMMANDS <OUTPUT FILE>
141443   %
141443   % MOVES THE PARAMETER STRING TO INIBUF (BELOW)
141443   INTEGER ARRAY INIBUF:=#''
141444   SUBR INCOM,NEXINCOM,LIICOM
141444
141444   DISP -200; INTEGER CBYTP; PSID
141444
141444   INTEGER ARRAY DUMBUF(200); DATA(#'')
141645   INTEGER BFULL:='BUFFER FULL'
141653   INTEGER BYTPNT,STPAR:='N COMMAND: ',PWSBC:="5OPSEG"
141663   INCOM: 0=:BYTPNT
141664   NEXINCOM: L=:D; CALL ENTER; "STPAR"; CALL GPAR
141670          FOR X:=BYTPNT TO 374 DO
141674                CALL CREAD; T:="INIBUF"; *SBYT
141677                IF =15 GO OUT
141702          OD; "BFULL"; CALL OUTTEXT
141706   OUT:   ##'; T:="INIBUF"; *SBYT
141711          X+2 BZERO "0"=:BYTPNT; *SBYT
141715          "PWSBC"; *MON 2WSEG              % WRITE OP.COM. SEG. BACK
141717          GO LEAVE
141720   LIICOM: L=:D; CALL ENTER; CALL ESCON
141723          CALL ROFIPAR; GO LEAVE
141725          IF INIBUF(0) SHZ -10/\177><##' THEN
141734                0=:CBYTP
141735   LOOP:        CALL CRLF
141736                FOR X:=CBYTP TO 374 DO
141742                   T:="INIBUF"; *LBYT
141744                   WHILE A/\177><##'
141750                OD; CBYTP SHZ -1+"INIBUF"; X+2 BZERO "0"=:CBYTP
141760                X:=A; CALL OUTTEXT
141762                IF CBYTPT<BYTPNT GO LOOP
141766          FI; T:=TDVN; *MON 2CLOS; 0
141771          GO LEAVE
141772   RBUS
142007
142007   %==========================================================================
142007   % 18.36      H E L P
142007
142007   % COMMAND: HELP <COMMAND> <OUTPUT FILE>
142007   %
142007   SUBR HELP
142007   INTEGER SPACES:='          '
142014   INTEGER RTC:=    'RT:      '
142021   INTEGER SYSTC:= 'SYSTEM: '
142026   INTEGER FICOM:= 'FILSYS: '
142033   INTEGER STRCOM:='S COMMAND: '
142041   HELP:  L=:D; CALL ENTER; "STRCOM"; CALL GPAR
142045          IF BCHFLAG><1 THEN
142051             FOR X:=0 TO 5WCBUFSIZE DO WORKA(X)=:FIBUF(X) OD
```

```
142061          FI; CALL ROFIPAR; GO LEAVE
142063          "FIBUF"=:CSTRING; X:=COMTAB; CALL ESCON
142067          DO WHILE X.CNAME><-1
142073             IF T:=BCHFLAG-1=0 GO L1
142077             0=:CPNT; CALL ABL1
142101             IF ><2 THEN
142104   L1:          IF X.CPROTECT=0 OR A=-3 THEN "SPACES"
142112                ELSE IF =1 THEN "RTC"
142117                     ELSE IF =2 THEN "SYSTC"
142124                          ELSE IF =-1 OR =-2 THEN "FICOM" ELSE CALL ERRFATAL FI
142136                FI   FI   FI
142136                CALL OUTTEXT; X.CNAME; CALL OUTTEXT; CALL CRLF
142142             FI
142142             X+4
142143          OD
142144          T:=TDVN; *MON 2CLOS; JMP * 1
142147          GO LEAVE
142150   RBUS
142166
142166   %=========================================================================
142166   % 18.37       T E R M O D E
142166   %
142166   %  COMMAND:  TERMINAL-MODE
142166   %       <CAPITAL>,<DELAY AFTER CR>,<STOP ON FULL PAGE>,<LOGOUT ON MISS. CARRIER>
142166   %
142166   SUBR TERMODE,T1P09
142166   INTEGER NOTERM:='USE LOCAL MODE'
142176   INTEGER STCAP:='N CAPITAL LETTERS? ',STDLAY:='N DELAY AFTER CR? '
142222   INTEGER STSCREEN:='N STOP ON FULL PAGE? '
142235   INTEGER STLOGOUT:='N LOGOUT ON MISSING CARRIER? '
142254   TERMODE: L=:D; CALL ENTER; IF TTIFIELD.TYPRING BIT 5COM GO ERR
142262          "STCAP"; CALL GPAR; CALL YESNO
142265          IF >=0 THEN T:=TTIFIELD.DFLAG BZERO 5CAPITAL
142271                IF ><0 THEN T BONE 5CAPITAL FI; T=:X.DFLAG
142274          FI
142274          "STDLAY"; CALL GPAR; CALL YESNO
142277          IF >=0 THEN
142300                T:=TTIFIELD.DFOPP.TYPRING BZERO 5CRDLY
142304                IF ><0 THEN T BONE 5CRDLY FI; T=:X.TYPRING
142307          FI
142307          "STSCREEN"; CALL GPAR; CALL YESNO
142312          IF >=0 THEN 0=:TTIFIELD.DFOPP.SCREEN
142316                IF ><0 THEN MIN X.SCREEN FI
142320          FI
142320          "STLOGOUT"; CALL GPAR; CALL YESNO
142323          IF A>=0 THEN T:=TTIFIELD.FLAGB
142326                IF A><0 THEN T BONE 5LBLOG ELSE T BZERO 5LBLOG FI
142332                T=:X.FLAGB
142333          FI
142333          IF T:=TTIFIELD.TYPRING BIT 5BAD THEN      % BAD
142337   T1P09:       CALL CTMOD; 0/\0
142341          FI
142341          GO LEAVE
142342   ERR:   "NOTERM"; CALL OUTTEXT; GO LEAVE
142345   RBUS
142355
142355   %=========================================================================
142355   % 18.38       Y E S N O
142355   %
```

```
142355   % SUBROUTINE TO CHECK FOR YES AND NO
142355   % A POINTS TO NAME ON START; RETURN WITH A=0:NO,A=1:YES,A=-1:ERROR
142355   SUBR YESNO
142355   INTEGER YNTAB:=(YES,0,1,0,   NO,0,0,0,   -1)
142366   INTEGER YES:='YES',NO:='NO'
142372   YESNO: L=:D; CALL ENTER; X:="YNTAB"; CALL ABLOOK
142376          IF ><0 THEN -1 ELSE T.CMAND FI; GO LEAVE
142404   RBUS
142406
142406
142406   %=========================================================================
142406   % 18.39      O P E R A T O R
142406
142406   % COMMAND: OPERATOR <TEXT>
142406   %
142406   SUBR OPERATOR
142406   SYMBOL 5OPTERM=1206
142406   INTEGER PARTEXT:='N MESSAGE: '
142414   INTEGER STAR:=('$***',' TERMINAL',':$')
142426   OPERATOR: L=:D; CALL ENTER; "PARTEXT"; CALL GPAR
142432          X:="TTNO"; T:=5ERRSEG; CALL GET1L; CALL ERRFATAL
142436          IF =1 THEN 5OPTERM FI; A=:TDVN
142443          "STAR"; CALL OUTTEXT; CALL CLFORM; CALL TEXTN; TTNO; CALL DECU
142451          CALL TEXTN
142452          DO CALL CREAD WHILE ><15; CALL TCO; OD; CALL CRLF; 1=:TDVN
142463          GO LEAVE
142464   RBUS
142502
142502   %=========================================================================
142502   % 18.40      W F O P E R A T O R
142502
142502   % COMMAND: WAIT-FOR-OPERATOR
142502   %
142502   SUBR WFOPERATOR
142502   SYMBOL 5OPTERM=1206
142502   DISP -200; INTEGER RTP; PSID
142502   INTEGER CMINUS:=('$---',' WAITING TERMINAL')
142516   INTEGER PLUS:=('$+++',' RESTARTED TERMINAL')
142533   INTEGER SRPAR:=RTP+BFIELD
142534   WFOPERATOR: L=:D; CALL ENTER
142536          X:="TTNO"; T:=5ERRSEG; CALL GET1L; CALL ERRFATAL
142542          IF =1 THEN 5OPTERM FI; A=:TDVN
142547          "CMINUS"; CALL OUTTEXT; CALL CLFORM; CALL TEXTN
142553          TTNO; CALL DECU; CALL CRLF; CALL ESCON
142557          TTNO; CALL LOGPH
142561          X:=A; T:="DBPROG"; CALL XGTDFADDR
142564          A=:RTP; "SRPAR"; *MON 2RTOF; MON 2RTWT
142570          CALL ESCOFF; "PLUS"; CALL OUTTEXT; CALL CLFORM; CALL TEXTN
142575          TTNO; CALL DECU; CALL CRLF; 1=:TDVN; GO LEAVE
142603   RBUS
142624
142624   %=========================================================================
142624   % 18.41      R E U S E R
142624
142624   % COMMAND: RESTART-USER <TERMINAL NUMBER>
142624   %
142624   SUBR REUSER
142624   DISP -200; INTEGER RTP; PSID
142624   INTEGER NOTERM:='NO TERMINAL',NOTUS:='NOT USED'
142637   INTEGER SRPAR:=RTP+BFIELD
```

```
142640   REUSER: L=:D; CALL ENTER
142642           "STRTERM"; CALL GPAR; CALL LOGPH; CALL CHTERM; GO ERR
142647           IF A.TYPRING NBIT 5IOBT GO ERR
142653           T:="BSTATE"; CALL XGTDFADDR
142655           IF A=0 GO NOUS
142656           T:="DBPROG"; CALL XGTDFADDR
142660           IF A=0 GO NOUS; A=:RTP
142662           CALL CHTACTIVE; GO NOUS
142664           "SRPAR"; *MON 2RTON; MON 2RT
142667           GO LEAVE
142670   ERR:    "NOTERM"; CALL OUTTEXT; GO LEAVE
142673   NOUS:   "NOTUS";  CALL OUTTEXT; GO LEAVE
142676   RBUS
142711
142711   %==========================================================================
142711   % 18.42      L O G S Y S T
142711
142711   % COMMAND: RT-PROGRAM-LOG <PROGRAM> <INTERVAL> <SAMPLE RATE>
142711   %
142711   INTEGER ARRAY LGARR=?                % FOUND IN OPTIONAL ROUTINE
142711   SUBR LOGSYST,STAPLOG,STOPLOG,XPRHIST
142711   DISP -200
142711           INTEGER XPERIOD,HEADCOUNT,LUU1,LUU2,CTSTPROG
142711           INTEGER ALPAR,DLPAR
142711           DOUBLE DCLAD=ALPAR
142711   PSID
142711   @ICR
142711   SYMBOL ACTC1,ACTC2,ACTP1,ACTP2,
142711           SWC1,SWC2,SWP1,SWP2,
142711           FILC1,FILC2,FILP1,FILP2,
142711           DIC1,DIC2,
142711           PASP1,PASP2,
142711           IOWP1,IOWP2,
142711           TOTL1,TOTL2,
142711           U1T1,U1T2,U1P1,U1P2,
142711           U2T1,U2T2,U2P1,U2P2,
142711           ADRSWP,ADRU1,ADRU2,
142711           LOGFIELD,TESTPROG,
142711           PTOT1,PTOT2,
142711           XINCR,CINCR;
142711   INTEGER NOTIMP:='NOT IMPLEMENTED',
142721           PLHEAD:='$          PERCENT  SAMPLES$',
142737           STRRT:='R RT NAME: ',STRNT:='IDINTERVAL(SEC): ',
142756           STRSAM:='IDINTERRUPTS/SAMPLE: ',
142771           SYSHEAD:='$$CPU  SWAP  FILES  DISC',
143006           RTHEAD:='$$  CPU     SWAP   FILES DISC PASSIVE IO-WAIT',
143035           PARIHOLD:=("1","1"),
143037           TNREMO:='$NOT ALLOWED FROM REMOTE TERMINAL$',
143061           TNBATCH:='$NOT ALLOWED IN BATCH$',
143075           PAR:=("5LOGSEM",NULL,"1"),PARHOLD:=(XPERIOD+BFIELD,"2"),
143102           PAR2:=("1352",NULL,"1"),
143105           TLOGUN:='  LOG.UNIT ',PLU1:='IDLOG.UNIT NO.: ',
143124           TNSTART:='$NOT STARTED',
143133           TILLUN:='$ILLEGAL UNIT',
143142           STRIO:='IOINPUT/OUTPUT (0 OR 1): '; *)FILL;
143163   INTEGER SYSFORMAT:=('$',ACTC1,'    ',SWC1,'    ',FILC1,'    ',DIC1),
143176   XUT1:=('    ',U1T1),
143201   XUT2:=('          ',U2T1,-1),
143210           RTFORMAT:=('$',ACTP1,'/',ACTC1,' ',SWP1,'/',SWC1,' ',FILP1,'/',
143223             FILC1,' ',DIC1,'    ',PASP1,'    ',IOWP1),
```

```
143234   YUT1:=('        ',U1P1,'/',U1T1),
143243   YUT2:=('    ',U2P1,'/',U2T1,-1),
143251   TXPERCENT:='$`      PERCENT SAMPLES$',
143265   TXOTHER:='OTHER LEVELS:',
143274   TXLEVEL:='LEVEL ';
143300   @CR;
143300   *)FILL
143300   GOSTPIOF: A:=:X; GO STPIOF
143302   LOGSYST: L=:D; CALL ENTER; IF "LGARR"=0 THEN "NOTIMP"; GO FAR ERR FI
143310           IF TTIFIELD.TYPRING BIT 5COM OR A BIT 5BAD THEN "TNREMO"; GO FAR ERR FI
143320           IF BCHFLAG=1 THEN "TNBATCH"; GO FAR ERR FI
143326           "PAR"; *MON 2RESR
143330           IF ><0 THEN "TALRES"; GO FAR ERR FI
143333           "STRRT"; CALL GPAR; A=:X=:CTSTPROG; "LGARR"+TESTPROG; CALL GOSTPIOF
143342           "STRNT"; CALL AGPAR; 74; A=:XPERIOD
143346           "STRSAM"; CALL AGPAR; 10; A-=:X; "LGARR"+XINCR; CALL GOSTPIOF
143356           A=:X; "LGARR"+CINCR; CALL GOSTPIOF
143362           X:="COMRTP"; A:=1; CALL STPIOF
143365           0=:LUU1=:LUU2
143367           "PLU1"; CALL GPAR; IF A=0 GO L1; A=:LUU1; CALL LOGPH; AD=:DCLADR
143375           "STRIO"; CALL GPAR; IF A=0 THEN ALPAR ELSE DLPAR FI
143403           IF A=0 GO FAR ERRUN; GO L1; *)FILL
143430   INTEGER POINTER PXUT1:=XUT1,PXUT2:=XUT2,PYUT1:=YUT1,PYUT2:=YUT2
143434   L1:     A=:X; "LGARR"+ADRU1; CALL GOSTPIOF
143440           IF A=0 THEN A-1=:PXUT1=:PYUT1; GO L3 FI; A:=#  ; A=:PXUT1=:PYUT1
143450           "PLU1"; CALL GPAR; IF A=0 GO L2; A=:LUU2; CALL LOGPH; AD=:DCLADR
143456           "STRIO"; CALL GPAR; IF A=0 THEN ALPAR ELSE DLPAR FI
143464           IF A=0 GO FAR ERRUN
143466   L2:     A=:X; "LGARR"+ADRU2; CALL GOSTPIOF
143472           IF A=0 THEN A-1=:PXUT2=:PYUT2; GO L3 FI
143477           A:=#  ; A=:PXUT2=:PYUT2; GO L3; *)FILL
143512   L3:     MASSNO(0); CALL LOGPH; A=:X; "LGARR"+ADRSWP; CALL FAR GOSTPIOF
143521           CALL ROFIPAR; GO LEAVE
143523   @LIB CXCPU
143523           IF TTIFIELD.TYPRING BIT 5TERM THEN X:=X.DFOPP.TDRADDR ELSE X:=X.DFOPP FI
143533   @ELIB
143533   @LIB CXCPU-,
143533           "LGARR"+LOGFIELD; CALL FAR GOSTPIOF
143536           20=:HEADCOUNT; "PARIHOLD"; *MON 2HOLD
143542           -2=:TTIFIELD.DFOPP.TMR          % SET TIMECOUNTER
143546           DO                        %UNTIL ESCAPE IS PUSHED
143546                 FOR X:="LGARR" TO A:="LGARR"+U2P2 DO
143553                     A:=0; CALL STPIOF
143555                 OD
143557                 CALL ESCON; "PARHOLD"; *MON 2HOLD
143562                 IF HEADCOUNT=20 THEN
143566                   0=:HEADCOUNT
143567                   IF CTSTPROG=0 THEN "SYSHEAD" ELSE "RTHEAD" FI; CALL OUTTEXT
143575                   IF LUU1><0 THEN
143577                      "TLOGUN"; CALL OUTTEXT; LUU1; D:=0; A:=:D; CALL FILSYS(DDECI)
143606                   FI
143606                   IF LUU2><0 THEN
143610                     "TLOGUN"; CALL OUTTEXT; LUU2; D:=0; A:=:D; CALL FILSYS(DDECI)
143617                   FI
143617                 FI; MIN HEADCOUNT
143620                 IF CTSTPROG=0 THEN "SYSFORMAT" ELSE "RTFORMAT" FI
143625                 A=:TEXTADR
143626                 DO WHILE TEXTADR.S0><-1
143633                   CALL TEXTN; TEXTADR.S0+"LGARR"
143637                   T:="LGARR"+TOTL1; CALL PERCENT; MIN TEXTADR
```

```
143643                OD
143644          OD; GO LEAVE
143646   ERRUN: "TILLUN"
143647   ERR:   CALL OUTTEXT; GO LEAVE
143651   *)FILL
143673
143673   STAPLOG: L=:D; CALL ENTER
143675          IF "LGARR"=0 THEN "NOTIMP"; GO ERR FI
143701          IF TTIFIELD.TYPRING BIT 5COM OR A BIT 5BAD THEN "TNREMO"; GO ERR FI
143711          IF BCHFLAG=1 THEN "TNBATCH"; GO ERR FI
143717          "PAR"; *MON 2RESR
143721          IF A><0 THEN "TALRES"; GO ERR FI
143724          "PAR2"; *MON 2RESR
143726          IF A><0 THEN "TALRES"; GO ERR FI
143731          "STRSAM"; CALL AGPAR; 10; A-=:X; "LGARR"+XINCR; CALL FAR GOSTPIOF
143741          A=:X; "LGARR"+CINCR; CALL FAR GOSTPIOF
143745   @LIB CXCPU
143745          IF TTIFIELD.TYPRING BIT 5TERM THEN X:=X.DFOPP.TDRADDR ELSE X:=X.DFOPP FI
143755   @ELIB
143755   @LIB CXCPU-,
143755          "LGARR"+LOGFIELD; CALL FAR GOSTPIOF
143760          FOR X:="PGARR" TO A:="LOTH2" DO A:=0; CALL STPIOF; OD
143770          "LGARR"+PTOT1=:X; A:=0; D:=0; CALL STDPIOF
143776          X:=RTREF; "LGARR"+TESTPRO; A:=:X; CALL STPIOF
144003          X:="ALLRTP"; A:=1; CALL STPIOF
144006          X:="COMRTP"; A:=0; CALL STPIOF
144011          GO LEAVE
144012   *)FILL
144033
144033   DISP -200; INTEGER INDXX,CRTDADDR; PSID
144033
144033   STOPLOG: L=:D; CALL ENTER
144035          X:="ALLRTP"; CALL LDPIOF; IF A=0 THEN "TNSTART"; GO ERR FI
144042          "PAR2"; *MON 2RESR
144044          IF A><0 THEN "TALRES"; GO ERR FI
144047          CALL ROFIPAR; GO LEAVE; 0=:INDXX
144052          X:="ALLRTP"; A:=0; CALL STPIOF
144055          "LGARR"+LOGFIELD; X:=0; CALL FAR GOSTPIOF
144061          "PLHEAD"; CALL OUTTEXT
144063          DO
144063                INDXX*5RTSIZE+RTSTART WHILE A<=SEGSTART-5RTSIZE
144072                A=:CRTDADDR:="CRTDADDR"+B; CALL RTFELEM; GO POCTRT
144077                D=:X; CALL PRIRT; GO FELLS
144102   POCTRT:      IF CRTDADDR.ACTPRI BZERO 5RTOFF=0 AND X.STADR=0 GO NEXT
144110                CRTDADDR; CALL OCTU
144112   FELLS:       #: ; CALL TCO2
144114                "LGARR"+PTOT1=:T; INDXX SH 1+"PGARR"; CALL PERCENT
144123                20040; CALL TCO2; INDXX SHZ 1+"PGARR"=:X; CALL LDDPIOF
144132                CALL FILSYS(DDECI); CALL CRLF
144135   NEXT:        MIN INDXX
144136          OD; GO OVER; *)FILL
144171   OVER:  5HSEM; CALL LOGPH
144173          IF A><0 THEN
144174             IF A.RTRES=RTREF THEN
144201                X:="HFLGHISTO"; CALL LDPIOF
144203                IF A><0 THEN
144204                   "TXPERCENT"; CALL OUTTEXT; 0=:INDXX
144207                   FOR INDXX DO WHILE INDXX<12
144213                      "TXLEVL"; CALL OUTTEXT; INDXX+##0; CALL TCO; # :; CALL TCO2
144222                      "LGARR"+PTOT1=:T; INDXX SH 1+"SLVTB"; CALL PERCENT
```

```
144231                    20040; CALL TCO2; INDXX SH 1 +"SLVTB"=:X
144237                    CALL LDDPIOF; CALL FILSYS(DDECI); CALL CRLF
144243                 OD; "TXOTHER"; CALL OUTTEXT
144247                 "LGARR"+PTOT1=:T; "LOTH1"; CALL PERCENT; 20040; CALL TCO
144256                 X:="LOTH1"; CALL LDDPIOF; CALL FILSYS(DDECI)
144262                 CALL CRLF; CALL CRLF; GO XPRHISTO  % IN PRHIST COMMAND
144265              FI
144265           FI
144265        FI
144265        "PAR2"; *MON 2RELE
144267        IF TDVN><1 THEN
144273              T:=A; *MON 2CLOS; JMP *+1
144276        FI; GO LEAVE
144277
144277   RBUS
144325
144325
144325   %=========================================================================
144325   % 18.43       P E R C E N T
144325   %
144325   % SUBROUTINE TO COMPUTE AND PRINT A PERCENTAGE
144325   % A=POINTER TO A DOUBLE VALUE; T=POINTER TO TOTAL (DOUBLE)
144325   SUBR PERCENT
144325   DISP -200; DOUBLE ADVAL,DDO=SO; PSID
144325   PERCENT: L=:D; CALL ENTER
144327        X:=A; CALL LDDPIOF; AD=:ADVAL; X:=T; CALL LDDPIOF; X:=0
144335        DO WHILE A><0; AD SHZ -1; X+1 OD; IF D<0 THEN AD SHZ -1; X+1 FI
144345        IF D BIT 16 THEN D SHZ -1; X+1 FI
144351        D=:T; ADVAL SH 2; FOR X-, DO AD SHZ -1 OD; 144; *RMPY SA DD; RDIV ST
144362        IF A+1 SHZ-1>=144 THEN ##1; CALL TCO; #00; CALL TCO2
144373        ELSE A=:X; 40; CALL TCO; X=:A; CALL DEC2
144401        FI; GO LEAVE
144402   RBUS
144407
144407   %=========================================================================
144407   % 18.44       D E F H I S T O   S T A H I S T O   S T O H I S T O
144407   %             P R H I S T O
144407
144407   % COMMANDS: DEFINE-HISTOGRAM <PROGRAM> <STARTADDRESS> <INTERVAL>
144407   %            START-HISTOGRAM
144407   %            STOP-HISTOGRAM
144407   %            PRINT-HISTOGRAM <OUTPUT FILE>
144407   %
144407   INTEGER ARRAY HISTO=?
144407   SUBR DEFHISTO,STAHISTO,STOHISTO,PRHISTO,DEFSHISTO,XPRHIST
144407   @ICR
144407   SYMBOL POUTSIDE=200,POUT2,TOTL1,TOTL2,STLT1,STLT2,
144407          TESTPROG,HISTART,FPREG,DELTA,HISTFLAG,CHILEVL,FLGHISTO;
144407   INTEGER NOTIMP:='NOT IMPLEMENTED',ALRES:='ALREADY IN USE',STRRT:='R RT NAME: ',
144435          PROT:='PROTECTED',
144442          STRDELTA:='IOINTERVAL: ',ILLINT:='ILL. INTERVAL',
144460          NOTRES:='NOT RESERVED',
144467      HEAD:=('                PERCENT SAMPLES$OUTSIDE:   .        ','  ',' OUT OF '),
144527          LFORM:=('$','-',':  ','  '),
144535          HISEM:=("5HSEM",NULL,"1"),
144540          ILEVL:=('IDLEVEL: '),
144545          ILLEVL:=('$ILL. LEVEL'),
144553          TXSYS:=('$SYSTEM:            ');
144565   @CR;
144565   *)FILL
```

```
144567
144567   DISP -200; INTEGER HFLAG; PSID
144567
144567   INTEGER CIRR(0); *IRR DP
144570   DEFSHISTO: L=:D; CALL ENTER; A:=1; GO L1
144574   DEFHISTO: L=:D; CALL ENTER; A:=0
144577   L1:     A=:HFLAG; IF "HISTO"=0 THEN "NOTIMP"; GO FAR ERR FI
144604           "HISEM"; *MON 2RESR
144606           IF ><0 THEN "ALRES"; GO FAR ERR FI; IF HFLAG><0 THEN X:=RTREF; GO L2 FI
144615           "STRRT"; CALL GPAR
144617           IF ><0 THEN A=:X; IF PASSTYPE=0 THEN "PROT"; GO FAR ERR FI; ELSE X:=RTREF FI
144627   L2:     "HISTO"+TESTPROG; A:=:X; CALL STPIOF
144633           IF HFLAG><0 THEN
144635                 "ILEVL"; CALL GPAR; IF A<0 OR A>14 THEN "ILLEVL"; GO ERR FI
144645                 A SH 3+CIRR=:X; "HISTO"+CHILEVL; A:=:X; CALL STPIOF
144654           FI; "STSTART"; CALL GPAR; A=:X:="HISTO"+FPREG; A:=:X; CALL STPIOF
144663           "STRDELTA"; CALL GPAR; IF <=0 OR >2000 THEN "ILLINT"; GO ERR FI
144674           A=:X:="HISTO"+DELTA; A:=:X; CALL STPIOF; "HISTO"=:D
144703           FOR X:=0 TO STLT2 DO
144707                 X:=:D; A:=0; CALL STPIOF; X:=:D; D+1
144714           OD; "HISTO"+HISTART=:X; A:=0; CALL STPIOF
144723           "HISTO"+FLGHISTO; X:=HFLAG; A:=:X; CALL STPIOF
144730           GO LEAVE
144731   *)FILL
144760
144760   STAHIST: L=:D; CALL ENTER; X:=1; GO STAO
144764   STOHIST: L=:D; CALL ENTER; X:=0
144767   STAO:   "HISEM"; *MON 2WHER
144771           IF ><RTREF THEN "NOTRES"; GO ERR FI
144776           "HISTO"+HISTART; A:=:X; CALL STPIOF
145002           IF A=1 THEN
145005                 "HISTO"+FLGHISTO=:X; CALL LDPIOF; A=:X
145012                 "HISTO"+HISTFLAG; A:=:X; CALL STPIOF
145016           ELSE
145017                 "HISTO"+HISTFLAG=:X; A:=0; CALL STPIOF
145024           FI; GO LEAVE
145025
145025   ERR:    CALL OUTTEXT; "HISEM"; *MON 2RELE
145030           GO LEAVE
145031   *)FILL
145042
145042   DISP -200; INTEGER COUNT; PSID
145042   DOUBLE ARRAY POINTER DHISTO:=HISTO
145043   PRHIST: L=:D; CALL ENTER; "HISEM"; *MON 2WHER
145047           IF ><RTREF THEN "NOTRES"; GO ERR FI
145054           "HISTO"+HISTART=:X; A:=0; CALL STPIOF
145061           CALL ROFIPAR; GO LEAVE; CALL ESCON
145064   XPRHIST: "HEAD"; CALL OUTTEXT; "HISTO"+POUTSIDE; T:=A+"TOTL1-POUTSIDE"
145072           CALL PERCENT; CALL TEXTN; "HISTO"+POUTSIDE=:X; CALL LDDPIOF
145100           CALL FILSYS(DDECI)
145102           CALL TEXTN; "HISTO"+TOTL1=:X; CALL LDDPIOF; CALL FILSYS(DDECI); 0=:COUNT
145112           "HISTO"+TESTPROG=:X; CALL LDPIOF
145116           IF A.ACTPRI BIT 17 THEN
145122             "HISTO"+FLGHISTO=:X; CALL LDPIOF
145126             IF A=0 THEN
145127                 "TXSYS"; CALL OUTTEXT
145131                 "HISTO"+TOTL1=:T; "HISTO"+STLT1; CALL PERCENT
145137                 20040; CALL TCO2
145141                 "HISTO"+STLT1=:X; CALL LDDPIOF; CALL FILSYS(DDECI)
145147             FI
```

```
145147          FI
145147          FOR COUNT STEP 2 TO 176 DO
145153                "LFORM"; CALL OUTTEXT; "HISTO"+FPREG=:X; CALL LDPIOF
145161                CALL OCTU; CALL TEXTN
145163                "HISTO"+DELTA=:X; CALL LDPIOF; A=:D:="HISTO"+FPREG=:X; CALL LDPIOF
145174                A+D; CALL STPIOF; A-1; CALL OCTU
145200                CALL TEXTN; "HISTO"+TOTL1=:T; "HISTO"+COUNT; CALL PERCENT
145207                CALL TEXTN; "HISTO"+COUNT=:X; CALL LDDPIOF; CALL FILSYS(DDECI)
145216          OD; IF TDVN><1 THEN 14; CALL TCO FI        % CTRL L
145230          T:=TDVN; *MON 2CLOS; MON 2QERM
145233          "HISEM"; *MON 2RELE
145235          GO LEAVE
145236   RBUS
145272
145272   %==========================================================================
145272   % 18.45       E X E C I O X
145272
145272   % COMMAND: EXECUTE-IOX VALUE,DEVICE-NO.
145272   %
145272   SUBR EXECIOX
145272   DISP -200; INTEGER VAL,DEV; PSID
145272   INTEGER PAR:=(VAL+BFIELD,DEV+BFIELD),ILLD:='ILL. DEVICE NO.'
145304   INTEGER VALUE:='IOVALUE: ',DEVNO:='IODEVICE NO: '
145320   EXECIOX: L=:D; CALL ENTER; "VALUE"; CALL GPAR; A=:VAL
145325           "DEVNO"; CALL GPAR; A=:DEV; "PAR"; *MON 2EXIO
145332           IF ><0 THEN "ILLD"; CALL OUTTEXT
145335           ELSE VAL; CALL OCTU
145340           FI; GO LEAVE
145341   RBUS
145347
145347   %==========================================================================
145347   % 18.46       C S T T Y P   C G T T Y P
145347   %
145347   % COMMAND: GET-TERMINAL-TYPE
145347   %          SET-TERMINAL-TYPE <TERM. TYPE>
145347   %
145347   SUBR CSTTYP,CGTTYP,T1P14
145347   INTEGER PTTYP:='IDTERMINAL TYPE: '
145360   DISP -200; INTEGER CLOGU; PSID
145360
145360   CSTTYP: L=:D; CALL ENTER
145362           "STRTERM"; CALL AGPAR; GO OK; A=:CLOGU
145366           CALL LOGPH; IF A=0 GO CILLP
145370           IF A.TYPRING NBIT 5TERM AND A NBIT 5COM AND A NBIT 5BAD GO CILLP
145400           IF CLOGU><TTNO AND PASSTYPE><2 THEN
145410                 25; *MON 64
145412                 GO LEAVE
145413           FI; GO L1
145414   @LIB CXCPU
145414   OK:     IF TTIFIELD.TYPRING BIT 5TERM THEN X:=X.TDRADDR FI
145421   @ELIB
145421   @LIB CXCPU-,
145421   L1:     "PTTYP"; CALL GPAR
145423           T:="CTTYP"; CALL XSTDFADDR
145425           IF X.TYPRING BIT 5BAD THEN
145430   T1P14:     CALL BSTTY; 0/\0
145432           FI
145432           GO LEAVE
145433
145433   CGTTYP: L=:D; CALL ENTER
```

```
145435          "STRTERM"; CALL AGPAR; GO OK1
145440          CALL LOGPH; IF A=0 GO CILLP
145442          IF A.TYPRING NBIT 5TERM AND A NBIT 5COM AND A NBIT 5BAD GO CILLP
145452          GO L2
145453   @LIB CXCPU
145453   OK1:   IF TTIFIELD.TYPRING BIT 5TERM THEN X:=X.TDRADDR FI
145460   @ELIB
145460   @LIB CXCPU-,
145460   L2:    CALL CRLF; "PTTYP"+1; CALL OUTTEXT
145464          T:="CTTYP"; CALL XGTDFADDR; CALL DECU
145467          GO LEAVE
145470   CILLP: "ILLP"; CALL OUTTEXT; GO LEAVE
145473   RBUS
145507
145507
145507   %==========================================================================
145507   % 18.47       C D S P F M E S S
145507   %
145507   % COMMAND: DEFINE-SPOOLING-FILE-MESSAGE <MESSAGE>
145507   %
145507   SUBR CDSPFMESS
145507   INTEGER TDSPF:='MESSAGE: '
145514   INTEGER TPMIALL:='N PRINTING MESSAGE INDEPENDANT OF SPOOLING CONDITIONS? '
145550
145550   CDSPFMESS: L=:D; CALL ENTER
145552          X:=0; T:=0; *MON 1; JPL I (LOGOU
145556          IF A/\177=15 THEN
145562                "TDSPF"; CALL OUTTEXT
145564          FI
145564          FOR X TO 137 DO
145567                T:=0; *MON 1; JPL I (LOGOU
145572                T:="XSPFMESS"; *SBYT
145574                IF A=15 THEN
145577                   X+1; 12; T:="XSPFMESS"; *SBYT
145603                FI
145603          WHILE A><##'
145606          OD
145610          "TPMIALL"; CALL GPAR; CALL YESNO
145613          IF A><1 THEN A:=0 FI; A=:FXSPFMESS
145620          GO LEAVE
145621   RBUS
145631
145631
145631
145631   %==========================================================================
145631   % 18.48       D E F E S C
145631   %
145631   % COMMAND: DEFINE-ESCAPE-CHARACTER  <VALUE OF ESCAPE CHAR.>
145631   %
145631   SUBR DEFESC,T1P10
145631   INTEGER PESCVAL:='IOVALUE OF "ESCAPE" CHAR. (OCT): '
145652   DISP -200; INTEGER CCESC,CLOGU; PSID
145652
145652   DEFESC: L=:D; CALL ENTER
145654          "STRTERM"; CALL AGPAR; GO OK; A=:CLOGU
145660          CALL LOGPH; IF A=0 GO CILLP
145662          IF A.TYPRING NBIT 5TERM AND A NBIT 5COM AND A NBIT 5BAD GO CILLP
145672          IF CLOGU><TTNO AND PASSTYPE><2 THEN
145702                25; *MON 64
145704                GO LEAVE
```

```
145705          FI; GO L1
145706  @LIB CXCPU
145706  OK:     IF TTIFIELD.TYPRING BIT 5TERM THEN X:=X.TDRADDR FI
145713  @ELIB
145713  @LIB CXCPU-,
145713  L1:     "PESCVAL"; CALL AGPAR; 33
145716          A=:CCESC
145717          T:="CESCP"; CALL XGTDFADDR; A/\177400\/CCESC; T:="CESCP"; CALL XSTDFADDR
145725          IF X.TYPRING BIT 5BAD THEN
145730  T1P10:     CALL BSDAE; 0/\0
145732          FI
145732          GO LEAVE
145733  CILLP: "ILLP"; CALL OUTTEXT; GO LEAVE
145736  RBUS
145750
145750  %=========================================================================
145750  % 18.48.2    D E F L O C A L
145750  %
145750  % COMMAND: DEFINE-LOCAL-CHARACTER  <VALUE OF LOCAL CHAR.>
145750  %
145750  SUBR DEFLOC
145750  INTEGER PDISVAL:='IOVALUE OF "LOCAL" CHAR. (OCT): '
145771  DISP -200; INTEGER CCLOC,CLOGU; PSID
145771
145771  DEFLOC: L=:D; CALL ENTER
145773          "STRTERM"; CALL AGPAR; GO OK; A=:CLOGU
145777          CALL LOGPH; IF A=0 GO CILLP
146001          IF A.TYPRING NBIT 5TERM AND A NBIT 5BAD GO CILLP
146007          IF CLOGU><TTNO AND PASSTYPE><2 THEN
146017                25; *MON 64
146021                GO LEAVE
146022          FI; GO L1
146023  @LIB CXCPU
146023  OK:     IF TTIFIELD.TYPRING BIT 5TERM THEN X:=X.TDRADDR FI
146030  @ELIB
146030  @LIB CXCPU-,
146030  L1:     "PDISVAL"; CALL AGPAR; A:=0
146033          A SHZ 10=:CCLOC
146035          T:="CESCP"; CALL XGTDFADDR; A/\377\/CCLOC; T:="CESCP"; CALL XSTDFADDR
146043          GO LEAVE
146044  CILLP: "ILLP"; CALL OUTTEXT; GO LEAVE
146047  RBUS
146060
146060
146060  %=========================================================================
146060  % 18.49      C D E S C F U   C E E S C F U
146060  %
146060  % COMMAND: ENABLE-ESCAPE-FUNCTION
146060  %          DISABLE-ESCAPE-FUNCTION
146060  %
146060  SUBR CDESCFU,CEESCFU,T1P11
146060
146060  DISP -200; INTEGER ROUFLG,CLOGU; PSID
146060
146060  CDESCFU: L=:D; CALL ENTER
146062          1=:ROUFLG; GO FELLS
146065  CEESCFU: L=:D; CALL ENTER
146067          0=:ROUFLG
146070  FELLS: "STRTERM"; CALL AGPAR; GO OK; A=:CLOGU
146074          CALL LOGPH; IF A=0 GO CILLP
```

```
146076          IF A.TYPRING NBIT 5TERM AND A NBIT 5COM AND A NBIT 5BAD GO CILLP
146106          IF CLOGU><TTNO AND PASSTYPE><2 THEN
146116                25; *MON 64
146120                GO LEAVE
146121          FI; GO L1
146122   @LIB CXCPU
146122   OK:    IF TTIFIELD.TYPRING BIT 5TERM THEN X:=X.TDRADDR FI
146127   @ELIB
146127   @LIB CXCPU-,
146127   L1:    T:="DFLAG"; CALL XGTDFADDR
146131          IF T:=ROUFLG=0 THEN A BZERO 5IESC ELSE A BONE 5IESC FI
146137          T:="DFLAG"; CALL XSTDFADDR
146141          IF X.TYPRING BIT 5BAD THEN
146144   T1P11:    CALL BCESC; 0/\0
146146          FI
146146          GO LEAVE
146147   CILLP: "ILLP"; CALL OUTTEXT; GO LEAVE
146152   RBUS
146162
146162
146162   %=========================================================================
146162   % 18.50       C S B S I Z E
146162   %
146162   % COMMAND: CHANGE-BACKGROUND-SEGMENT-SIZE <SEGMENT>
146162   %
146162   SUBR CSBSIZE
146162   DISP -200; INTEGER CSEG,CSIZE,CDBPROG; PSID
146162   INTEGER PABSIZE:='IDBACKGROUND SEGMENT SIZE IN PAGES: '
146205   INTEGER ERT1:='$ILLEGAL SEGMENT NUMBER$'
146222   INTEGER ERT2:='$NOT BACKGROUND SEGMENT$'
146237   INTEGER ERT3:='$NO BACKGROUND PROGRAM WILL USE THIS BACKGROUND SEGMENT$'
146274   INTEGER ERT4:='$SEGMENT WILL NOT BE USED BY ANY BACKGROUND PROGRAMS$'
146327   INTEGER ERT5:='$ILLEGAL SEGMENT SIZE$'
146343   INTEGER ETCSEG:='$(NOT ENOUGH SPACE IN SEGMENT FILES?$SEGMENT SET TO 64K)$'
146400   CSBSIZE: L=:D; CALL ENTER
146402          RTREF.SEGM SHZ -10+1=:CSEG                % OWN BACKGROUND SEGMENT IS DEFAULT
146407          "GGSGM"; CALL AGPAR; CSEG; A=:CSEG
146413          IF A=0 OR A>>SGMAX GO ERR1                % SEGMENT NUMBER OUT OF RANGE
146417          A*5SEGSIZE+SEGSTART=:X
146422          IF X.FLAG/\177776><160002 GO ERR2         % ILLEGAL SEGMENT TYPE
146427          X-5SEGSIZE; IF X.FLAG NBIT 5SYSEGM GO ERR2      % SYSTEM SEGM IS NOT PREVIOUS SEGMENT
146433          FOR X:="BAK01" STEP 5RTSIZE TO "9LBPR-5RTSIZE" DO
146437             IF X.SEGM SHZ -10+1=CSEG THEN          % FIND PROGRAM USING THIS BACKGROUND SEGMENT
146445                IF X><RTREF THEN CALL XSRTOF; GO ERR3 FI  % IF PROG IS INACTIVE SET RTOF
146452                X=:CDBPROG; GO L1
146454             FI
146454          OD; GO ERR4                               % SEGMENT NOT USED BY ANY BACKGROUND PROGRAM
146457   L1:    "PABSIZE"; CALL GPAR
146461          IF A><100 AND A><200 GO ERR5; A=:CSIZE
146470          "RESRTL"; *MON 2RESR                      % RESERVE RT-LOADER
146472          IF A<0 THEN "RTLRMESS"; CALL OUTTEXT; GO LEAVE FI
146476          A:=CSEG; X:=CSIZE; CALL WRTLOAD           % CALL RT-LOADER
146501          IF A=-2 THEN                              % IF A><0 THEN ERROR FROM RTLOADER
146504                "EDIRTL"; CALL OUTTEXT
146506                "ETCSEG"; CALL OUTTEXT
146510          ELSE IF A><0 THEN "EDIRTL"; CALL OUTTEXT FI
146514          FI
146514          IF X:=CDBPROG><0 THEN CALL XRRTOF FI      % RESET RTOFF BIT (SET RTON)
146517          GO LEAVE
146520   ERR1:  "ERT1"
```

```
146521   ERRF:  CALL OUTTEXT; GO LEAVE
146523   ERR2:  "ERT2"; GO ERRF
146525   ERR3:  "ERT3"; GO ERRF
146527   ERR4:  "ERT4"; GO ERRF
146531   ERR5:  "ERT5"; GO ERR5
146533   RBUS
146564
146564
146564   @LIB OLD
146564
146564   %==========================================================================
146564   %  18.51      D E F T E R M   L I T E R M   E N T R M   D I T R M
146564   %             N T E R M
146564   %
146564   % COMMAND: DEFINE-TERMINATION-HANDLING <RT OR BACKGROUND> <RT-PROG/COMND>
146564   %          NEXT-TERMINATION-COMMAND <COMMAND>
146564   %          LIST-TERMINATION-HANDLING <RT/BACKGROUND> <OUTPUT-FILE>
146564   %          ENABLE-TERMINATION-HANDLING (RT/BACKGROUND> <TERM. NO.> <UB/FE>
146564   %          DISABLE-TERMINATION-HANDLING <RT/BACKGROUND> <TERM. NO.> <UB/FE>
146564   %
146564   % THESE TEXT STRINGS ARE ALSO USED BY THE "DEFAULT-SUBSYSTEM" COMMANDS
146564   INTEGER TNOBGD:='NO BACKGROUND PROGRAM DEFINED!$'
146604   INTEGER XTXEROP:='$ERROR IN OPERATION WHEN ACCESSING SEGMENT '
146632
146632   SYMBOL CBSIZ=200
146632   INTEGER ARRAY TECBUF(CBSIZ); DATA(#'') % CLEAN-UP COMMAND BUFFER
147033   INTEGER PETECOM:=TECBUF                 % BYTE POINTER TO END OF TE-COMMANDS
147034
147034   SUBR DEFTERM,NTERM,LITERM,ENTRM,DITRM
147034
147034   DISP -200
147034   INTEGER CBYTP,LIFLAG,ENFLAG,SGNO,UBFEFLAG,TDAFI,TLDN,ROUTSWITCH
147034   INTEGER DYNAL,CINDX=CBYTP,CADDR=TLDN
147034   PSID
147034   INTEGER TRTNAM:='R RT NAME: '
147042   INTEGER TUSBR:='$USER-BREAK '
147051   INTEGER TFATA:='$FATAL-ERROR '
147060   INTEGER TTERE:='TERMINATION ENABLED$'
147073   INTEGER TTERD:='TERMINATION DISABLED$'
147106   INTEGER TCOMA:='N COMMAND: '
147114   INTEGER TCOINU:='COMMAND IN USE!'
147124   INTEGER TBUFUL:='BUFFER FULL'
147132   INTEGER TNOTDEF:='$$NO TERMINATION DEFINED!$'
147150
147150   INTEGER BYTPNT
147151   INTEGER ARRAY PRESP:=("1222",NULL,"1")     % TERMINATION CMND-BUFFER SEMAPHORE
147154   INTEGER PWSBC:="50PSEG"
147155   INTEGER BMAX=?
147155   SYMBOL 7RTBG=0, 7UBFE=1
147155
147155   NTERM: L=:D; CALL ENTER; GO NETERM
147160   DEFTERM: L=:D; CALL ENTER
147162          "PRESP"; *MON 2RESR
147164          IF A><0 THEN "TCOINU"; CALL OUTTEXT; GO LEAVE; FI
147170          7RTBG; CALL RBGUF; GO LEAVE
147173          IF A=0 THEN            % RT-TERMINATION
147174                "TRTNAM"; CALL GPAR; A=:RTTERM
147177                T:=2; X:="RTTERM"; CALL PUT1L; GO FAR ERROP
147203          ELSE                   % BACKGROUND-TERMINATION
147204                0=:BYTPNT
```

```
147205   NETERM:        "TCOMA"; CALL GPAR          % ENTRY POINT FOR "NEXT-TERM-COM"
147207                  FOR X:=BYTPNT TO BMAX DO
147213                      CALL CREAD; T:="TECBUF"; *SBYT
147216                      IF A=15 GO FINI
147221                  OD; "TBUFUL"; CALL OUTTEXT
147225   FINI:          ##'; T:="TECBUF"; *SBYT
147230                  X+2 BZERO "0"=:BYTPNT=:PETECOM; *SBYT
147235                  "PWSBC"; *MON 2WSEG         % WRITE OP.COM. SEG BACK
147237           FI; "PRESP"; *MON 2RELE
147241           GO LEAVE; *)FILL
147263
147263   %  COMMAND TO LIST TERMINATION-HANDLING COMMANDS
147263
147263   INTEGER BMAX:=CBSIZ*2-4
147264   LITERM: L=:D; CALL ENTER
147266           "PRESP"; *MON 2RESR
147270           IF A><0 THEN "TCOINU"; CALL OUTTEXT; GO LEAVE; FI
147274           7RTBG; CALL RBGUF; GO LEAVE; A=:LIFLAG; CALL ESCON; CALL ROFIPAR; GO LEAVE
147303           IF LIFLAG=0 THEN                    % RT TERMINATION
147305                  IF RTTERM=0 THEN
147307                     "TNOTDEF"; CALL OUTTEXT
147311                  ELSE
147312                     CALL RTOUT
147313                  FI
147313           ELSE                                % BACKGROUND TERMINATION
147314                  IF PETECOM="TECBUF" THEN
147320                     "TNOTDEF"; CALL OUTTEXT
147322                  ELSE
147323                     0=:CBYTP; CALL CRLF
147325   LOOP:             CALL CRLF
147326                     FOR X:=CBYTP TO BMAX DO
147332                         T:="TECBUF"; *LBYT
147334                     WHILE A><##'
147337                     OD; CBYTP SHZ -1+"TECBUF"
147344                     X+2 BZERO "0"=:CBYTP
147347                     X:=A; CALL OUTTEXT
147351                     IF CBYTP<PETECOM GO LOOP
147355                  FI
147355           FI; T:=TDVN; *MON 2CLOS; JMP *+1
147360           "PRESP"; *MON 2RELE
147362           GO LEAVE; *)FILL
147377
147377   %  COMMANDS TO ENABLE/DISABLE TERMINATION-HANDLING
147377   INTEGER PCWSGB:=BFIELD+SGNO
147400   ENTRM: A:=1; GO FELLS
147402   DITRM: A:="0"
147403   FELLS: L=:D; CALL ENTER; A=:ROUTSWITCH
147406           7RTBG; CALL RBGUF; GO LEAVE; A=:ENFLAG     % DEFAULT=NO OPERATION
147412           IF A=0 THEN            % RT
147413                  2=:SGNO                             % IMAGE
147415                  IF PASSTYPE=0 THEN GO FAR UNAUTH FI % LEGAL ONLY FOR SYSTEM AND RT
147420                  IF RTTERM=0 THEN
147422                     "TNOTDEF"; CALL OUTTEXT; GO LEAVE
147425                  FI
147425                  IF ROUTSWITCH><0 THEN
147427                     1=:FLRTTERM; T:=2; X:="FLRTTERM"
147433                     CALL PUT1L; GO FAR ERROP; "TTERE"
147436                  ELSE
147437                     A:=0=:FLRTTERM; T:=2; X:="FLRTTERM"
147443                     CALL PUT1L; GO FAR ERROP; "TTERD"
```

```
147446                  FI; CALL OUTTEXT; GO FAR OUT
147450            FI
147450            % ELSE BACKGROUND:
147450                  "STRTERM"; CALL AGPAR; TTNO; A=:TLDN
147454                  IF A><TTNO AND PASSTYPE<2 THEN GO FAR UNAUTH FI
147464                  A:=TLDN; CALL LOGPH; CALL CHTERM; GO FAR NOBDEF; A=:TDAFI
147471                  0=:DYNAL; X:=TDAFI; CALL MBDYNALLOC; GO NOTDYN; A=:CINDX
147476                  MIN DYNAL; 5PT3S=:SGNO; GO LABL1
147502    NOTDYN:       T:="DBPROG"; CALL XGTDFADDR
147504                  IF A=0 THEN GO FAR NOBDEF FI              % IF NO BACKGROUND PROGRAM
147506                  A.SEGM SHZ -10=:SGNO
147512    LABL1:        IF TDAFI.TYPRING BIT 5TERM OR A BIT 5COM OR A BIT 5BAD GO OK
147522                  IF T:=BATAB(0)=0 THEN GO FAR ILLPA FI; D:=0; X:=3
147531                  DO WHILE T>D
147533                     IF A:=BATAB(X)-TLDN=0 THEN GO OK FI
147537                     X+4; D+1
147541                  OD; GO FAR ILLPA; *)FILL
147566
147566    OK:           IF PETECOM="TECBUF" GO NOTDEF
147572                  7UBFE; CALL RBGUF; A:=3; A=:UBFEFLAG
147576                  IF A><0 THEN
147577                     IF DYNAL=0 THEN
147601                        X:="FLBGTERM"
147602                      ELSE
147603                        A:=CINDX*5PRVTT+"PRVTTABLE"+"PRVFBGTERM"=:X
147610                      FI; X=:CADDR
147611                  FI; T:=SGNO; CALL GET1L;  GO ERROP; A=:LIFLAG
147615                  IF UBFEFLAG BIT "0" THEN  % USER-BREAK
147620                     "TUSBR"; CALL OUTTEXT
147622                     A:=LIFLAG
147623                     IF T:=ROUTSWITCH><0 THEN A BONE "0" ELSE A BZERO "0" FI
147631                     A=:LIFLAG; X:=CADDR; T:=SGNO; CALL PUT1L; GO ERROP
147636                     IF LIFLAG BIT "0" THEN "TTERE" ELSE "TTERD" FI
147644                     CALL OUTTEXT
147645                  FI; GO L1; *)FILL
147662
147662    L1:           IF UBFEFLAG BIT 1 THEN    % FATAL-ERROR
147665                     "TFATA"; CALL OUTTEXT
147667                     A:=LIFLAG
147670                     IF T:=ROUTSWITCH><0 THEN A BONE 1 ELSE A BZERO 1 FI
147676                     A=:LIFLAG;T:=SGNO; X:=CADDR; CALL PUT1L; GO ERROP
147703                     IF LIFLAG BIT 1 THEN "TTERE" ELSE "TTERD" FI
147711                     CALL OUTTEXT
147712                  FI
147712            GO OUT; *)FILL
147717    NOBDEF: "TNOBGD"; GO EOUT                % NO BACKGROUND DEFINED
147721    NOTDEF:"TNOTDEF"; GO EOUT                % NO TERMINATION DEFINED
147723    ILLPA: "ILLP"                            % ILLEGAL PARAMETER
147724    EOUT:  CALL OUTTEXT; GO OUT
147726    ERROP: "XTXEROP"; CALL OUTTEXT; SGNO; CALL OCTU; GO OUT
147733    UNAUTH: 25; *MON 2ERMS        % YOU ARE NOT AUTHORIZED TO DO THIS
147735    OUT:   "PCWSGB"; *MON 2WSEG
147737           GO LEAVE
147740
147740    RBUS
147746
147746    %=================================================================
147746    %  18.52     D E F L I P   L I S L I P   E N L I P   D I L I P
147746    %
147746    % COMMAND: DEFINE-DEFAULT-SUBSYSTEM <SUBSYSTEM-NAME>
```

```
147746   %          LIST-DEFAULT-SUBSYSTEM <OUTPUT-FILE>
147746   %          DEFAULT-SUBSYSTEM-ENABLE <LOGICAL-NUMBER>
147746   %          DEFAULT-SUBSYSTEM-DISABLE <LOGICAL-NUMBER>
147746   %
147746   SYMBOL LIBSIZ=64
147746   INTEGER ARRAY LIPBUF(LIBSIZ); DATA(#'') % DEFAULT SUBSYSTEM NAME BUFFER
150033   INTEGER PELIPBUF:=LIPBUF                % BYTE POINTER TO END OF SUB.SYS. NAME
150034
150034   SUBR DEFLIP,LISLIP,ENLIP,DILIP
150034
150034   DISP -200
150034   INTEGER CBYTP,SGNO,TDAFI,ROUTSWITCH,CINDX,DYNAL,CADDR
150034   PSID
150034   INTEGER TSSNAM:='N SUBSYSTEM NAME: '
150046   INTEGER TBUFUL:='BUFFER FULL'
150054   INTEGER TNOTDEF:='$$NO SUBSYSTEM DEFINED!$'
150071   INTEGER PWSGB:=BFIELD+SGNO
150072   INTEGER BYTPNT
150073   INTEGER PWSBC:="5OPSEG"
150074   INTEGER BMAX=?
150074
150074   DEFLIP: L=:D; CALL ENTER
150076           0=:BYTPNT; "TSSNAM"; CALL GPAR
150101           FOR X:=BYTPNT TO BMAX DO
150105                    CALL CREAD; T:="LIPBUF"; *SBYT
150110                    IF A=15 GO FINI
150113           OD; "TBUFUL"; CALL OUTTEXT
150117   FINI:   ##'; T:="LIPBUF"; *SBYT
150122           X+2 BZERO "0"=:BYTPNT=:PELIPBUF; *SBYT
150127           "PWSBC"; *MON 2WSEG        % WRITE OP.COM. SEG BACK
150131           GO LEAVE; *)FILL
150142
150142   %  COMMAND TO LIST COMMAND STRING TO BE EXECUTED AFTER LOGIN
150142
150142   INTEGER BMAX:=LIBSIZ*2-4
150143   LISLIP: L=:D; CALL ENTER
150145           CALL ESCON; CALL ROFIPAR; GO LEAVE
150150           IF PELIPBUF="LIPBUF" THEN
150154                    "TNOTDEF"; CALL OUTTEXT
150156           ELSE
150157                    0=:CBYTP; CALL CRLF
150161                    FOR X:=CBYTP TO BMAX DO
150165                        T:="LIPBUF"; *LBYT
150167                    WHILE A><##'
150172                    OD; CBYTP SHZ -1+"LIPBUF"
150177                    X+2 BZERO "0"=:CBYTP
150202                    X:=A; CALL OUTTEXT
150204           FI; T:=TDVN; *MON 2CLOS; JMP *+1
150207           GO LEAVE; *)FILL
150217
150217   %  COMMANDS TO ENABLE/DISABLE START OF DEFAULT SUBSYSTEM (ON LOGIN)
150217
150217   ENLIP: A:=1; GO FELLS
150221   DILIP: A:="0"
150222   FELLS: L=:D; CALL ENTER; A=:ROUTSWITCH
150225           "STRTERM"; CALL AGPAR; TTNO; A=:X        % DEFAULT IS OWN TERMINAL
150231           IF A><TTNO AND PASSTYPE<2 THEN GO UNAUTH FI
150241           A:=X; CALL LOGPH; CALL CHTERM; GO NOBDEF; A=:TDAFI
150246           X:=TDAFI; 0=:DYNAL; CALL MBDYNALLOC; GO NOTDYN; A=:CINDX
150253           MIN DYNAL; 5PT3S=:SGNO; GO MBOK
```

```
150257
150257   NOTDYN: T:="DBPROG"; CALL XGTDFADDR
150261           IF A=0 THEN GO NOBDEF FI  % IF NO BACKGROUND PROGRAM
150263           A.SEGM SHZ -10=:SGNO
150267   MBOK:   IF TDAFI.TYPRING NBIT 5TERM AND NBIT 5COM AND NBIT 5BAD GO ILLPA
150277           IF PELIPBUF="TECBUF" GO NOTDEF
150303           IF DYNAL=0 THEN
150305              X:="FLLIPCOM"
150306           ELSE
150307              CINDX*5PRVT+"PRVTTABLE"+"PRVFLLIPCOM"=:X
150314           FI; X=:CADDR; T:=SGNO; CALL GET1L; GO ERROP
150320           IF T:=ROUTSWITCH><0 THEN A BONE "0" ELSE A BZERO "0" FI
150326           T:=SGNO; CALL PUT1L; GO ERROP
150331           "PWSGB"; *MON 2WSEG
150333           GO OUT; *)FILL
150354
150354   NOBDEF: "TNOBGD"; CALL OUTTEXT; GO OUT % NO BACKGROUND DEFINED
150357   NOTDEF:"TNOTDEF"; CALL OUTTEXT; GO OUT % NO TERMINATION DEFINED
150362   ILLPA: "ILLP"; CALL OUTTEXT; GO OUT    % ILLEGAL PARAMETER
150365   ERROP: "XTXEROP"; CALL OUTTEXT; SGNO; CALL OCTU; GO OUT
150372   UNAUTH: 25; *MON 2ERMS        % YOU ARE NOT AUTHORIZED TO DO THIS
150374   OUT:    GO LEAVE
150375
150375   RBUS
150402
150402   *)KILL TNOBG XTXER
150402
150402   %=========================================================================
150402   %  18.53      S E T P A
150402   %
150402   %  COMMAND: SET-USER-PARAMETERS <P1> <P2> <P3> <P4> <P5>
150402   %
150402   %            DEFAULT VALUES IS OLD CONTENT
150402   %
150402
150402   SUBR PASET
150402
150402   INTEGER TP1:='IDP1: ',TP2:='IDP2: ',TP3:='IDP3: ',TP4:='IDP4: ',TP5:='IDP5: '
150426
150426   PASET: L=:D; CALL ENTER
150430          "TP1"; CALL AGPAR; GO RP2; A=:USPAR(0)
150435   RP2:   "TP2"; CALL AGPAR; GO RP3; A=:USPAR(1)
150442   RP3:   "TP3"; CALL AGPAR; GO RP4; A=:USPAR(2)
150447   RP4:   "TP4"; CALL AGPAR; GO RP5; A=:USPAR(3)
150454   RP5:   "TP5"; CALL AGPAR; GO OUT; A=:USPAR(4)
150461   OUT:   GO LEAVE
150462
150462   RBUS
150471
150471
150471   SUBR ERINIT,ERPRINT
150471   INTEGER WSPAR:=LOGRSEG,LOGRSEG:=5LOGRSEG
150473   ERINIT: L=:D; CALL ENTER
150475           T:=LOGRSEG; X:=0; 5; CALL PUT1L; CALL ERRFATAL; "WSPAR"; *MON 2WSEG
150504           GO LEAVE
150505
150505   ERPRINT: L=:D; CALL ENTER
150507           CALL ROFIPAR; GO LEAVE; CALL ERSYS(XERPRINT)
150513           GO LEAVE
150514   RBUS
```

```
150521
150521  %=========================================================================
150521  %              M E M L I M
150521  %
150521  % COMMAND: MEMORY-LIMITS
150521  %
150521  SUBR MEMLIM
150521  INTEGER TLPAD:='IOLOWER PROGRAM ADDRESS: '
150536  INTEGER TUPAD:='IOUPPER PROGRAM ADDRESS: '
150553  INTEGER TLDAD:='IOLOWER DATA ADDRESS: '
150567  INTEGER TUDAD:='IOUPPER DATA ADDRESS: '
150603
150603  DISP -200; INTEGER CLPAD,CUPAD,CLDAD; PSID
150603
150603  MEMLIM: L=:D; CALL ENTER
150605          "TLPAD"; CALL AGPAR; A:=0; A=:CLPAD
150611          "TUPAD"; CALL AGPAR; A:=-1; A=:CUPAD
150615          "TLDAD"; CALL AGPAR; A:=0; A=:CLDAD
150621          "TUDAD"; CALL AGPAR; A:=-1
150624          A=:HDDAD:=CLDAD=:LDDAD:=CUPAD=:HDADR:=CLPAD=:LDADR; 0=:2BXAD
150634          GO LEAVE
150635  RBUS
150642
150642
150642  %=========================================================================
150642  %              C A O N   -   C A O F F
150642  %
150642  % COMMAND:    ALTON
150642  %             ALTOFF
150642  %
150642  SUBR CAON,CAOFF
150642  CAON:  L=:D; CALL ENTER; KABACTPRI=:BACTPRI; GO LEAVE
150647  CAOFF: L=:D; CALL ENTER; KBACTPRI=:BACTPRI; GO LEAVE
150654  RBUS
150657
150657  %=========================================================================
150657  %              M B D Y N A L L O C
150657  %
150657  % TEST IF A TERMINAL (TAD ETC) IS TEMPORARLY OR FIXED CONNECTED
150657  % TO A BACKGROUND PROGRAM
150657  %
150657  % ENTRY:      X=DATAFIELD
150657  %
150657  % EXIT:       NOT TEMPORARLY CONNECTION
150657  %
150657  % EXIT+1:     A=INDEX IN BACKGROUND TABLE
150657  %
150657  SUBR MBDYNALLOC
150657  MBDYNALLOC: L=:D; CALL ENTER
150661          IF "BPTMP"=0 GO LEAVE
150664          CALL GBTINDX; GO LEAVE; A=:D
150667          IF X.TYPRING NBIT 5TERM AND A NBIT 5BAD THEN GO LEAVE FI
150675          A:=D; GO LEAV2
150677  RBUS
150702
150702  %=========================================================================
150702  %              C H T A C T I V E
150702  %
150702  % SUBROUTINE TO CHECK IF A TERMINAL (OR TAD) WITH TEMPORARLY CONNECTION
150702  % TO A BACKGROUND PROGRAM IS ACTIVE OR NOT
```

```
150702   %
150702   % ENTRY:      X=DATAFIELD
150702   %             A=DBPROG
150702   %
150702   % EXIT:       NOT ACTIVE
150702   %
150702   % EXIT+1:     ACTIVE
150702   %
150702   SUBR CHTACTIVE
150702   CHTACTIVE: A=:T
150703          IF "BPTMP"=0 THEN EXITA FI
150706          IF X.RTRES=T THEN EXITA FI
150712          EXIT
150713   RBUS
150714
150714   %==========================================================================
150714   %             C C O L D S T A R T
150714   %
150714   %       COLD-START
150714   %
150714   SUBR CCOLDSTART,PCOLDSTART
150714   DISP -200; INTEGER LOGN; PSID
150714   CCOLDSTART: L=:D; CALL ENTER
150716          "STRTERM"; CALL KGPAR; A:=1; A=:LOGN
150722          CALL LOGPH
150723          IF A=0 OR A.TYPRING NBIT 5TERM THEN
150730   ERR:       "ILLP"; CALL OUTTEXT; GO LEAVE
150733          FI; T:="DBPROG"; CALL XGTDFADDR; IF A=0 GO ERR
150736          LOGN; GO PCOLDSTART
150740   RBUS
150747
150747
150747
150747   % BUFFER FOR USER ENVIRONMENT SUBSYSTEM NAME
150747   SYMBOL UEBSZ=37
150747   INTEGER ARRAY UECMD(0); *#''; *+UEBSZ/
151007
151007
151007   @DEV 1
151007   @DEV (S-S-J)SINC-2
```

```
151007
151007   %==========================   SINC-2   ========================================
151007   %
151007   %              S U B R O U T I N E S
151007   %
151007   %=========================================================================
151007   % 19.0         A B L O O K
151007
151007   %SUBROUTINE FOR ABBREVIATION LOOKUP, ENTRY: X=TABLE TOP
151007   %RETURN IN A-REG:0=OK,-1=NOT FOUND,-2=AMBIGUOUS; T=TABLE ELEMENT ADDRESS
151007   %D=0 IF EXACT MATCH
151007
151007   SUBR ABLOOK
151007   DISP -200; INTEGER RESULT,FNDADR,ACPNT; PSID
151007   ABLOOK: L=:D; CALL ENTER; -1=:RESULT; CPNT=:ACPNT
151015           DO IF X.CNAME=-1 THEN RESULT; X:=FNDADR; GO OUT FI
151024                T:=ACPNT=:CPNT; CALL ABL1; IF =0 THEN D:=0; GO OUTD FI
151032                IF =1 THEN       % SUBSET
151035                  IF RESULT=0 THEN -2=:RESULT
151041                  ELSE IF =-1 THEN 0=:RESULT; X=:FNDADR
151047                  FI FI
151047                FI
151047                X+4
151050           OD
151051   OUT:    D:=-1                    % NOT EXACT MATCH
151052   OUTD:   X=:T; GO LEAVE
151054   RBUS
151055
151055   %=========================================================================
151055   % 19.1         A B L 1
151055
151055   %ABBREVIATION CHECK ROUTINE, COMPARING THE COMMAND STING TO A STRING
151055   %ENTER WITH A=STRING; EXIT WITH A: 0=OK, 1=SUBSET, 2=NO MATCH
151055
151055   SUBR ABL1
151055   DISP -200; INTEGER PCPNT, CHAR,OCHAR,SUBSFLAG; PSID
151055   %AUXILIARY SUBROUTINE RETURNS A=1 IF LETTER OR DIGIT, ELSE 0
151055   ALPHANUM:IF >=##A AND <=##Z GO ALP; IF =### GO ALP
151066           IF >=##0 AND <=##9 THEN
151074   ALP:    1 ELSE A:=0 FI; EXIT
151100
151100   ABL1:   L=:D; CALL ENTER; A=:OSTRING; 0=:OPNT=:SUBSFLAG
151105           DO     CPNT=:PCPNT      %FOR EACH PART
151107                  DO               %FOR EACH CHARACTER
151107                    X:=0; CALL CREAD; A=:CHAR; CALL ALPHANUM; X+A; X+A
151115                    CALL OREAD; A=:OCHAR; CALL ALPHANUM; X+A
151121                    X GOSW NCNO,NCYO,YCNO,YCYO
151126   YCYO:            IF CHAR><OCHAR AND ><### GO YCNO2
151135           OD
151136   YCNO:          GO YCNO2      % NOT FOUND
151137   NCYO:          DO CALL OREAD; A=:OCHAR; CALL ALPHANUM WHILE ><0; OD
151144                  MIN SUBSFLAG
151145   NCNO:          X:=0; IF CHAR=##- THEN X+2 FI; IF OCHAR≠T THEN X+1 FI
151157                  X GOSW NCNO2,NCYO2,YCNO2,YCYO2
151164   YCYO2: OD
151165   NCYO2: X:=1 GO OUT
151167   NCNO2: IF SUBSFLAG><0 THEN X:=1 FI; GO OUT
151173   YCNO2: X:=2; CALL CREAD; IF =##- GO YCNO2; CALL ALPHANUM; IF ><0 GO YCNO2
151202   OUT:   CALL BACK; X=:A; GO LEAVE
151205   RBUS
```

```
151210
151210   %=================================================================
151210   % 19.2        G C O M
151210
151210   %SUBROUTINE TO PUT TT LINE INTO COMMAND BUFFER
151210   SUBR GCOM
151210   DISP -200; INTEGER CHAR; PSID
151210   GCOM:  L=:D; CALL ENTER
151212          IF BCHFLAG=0 THEN CALL EDIT
151215          ELSE  T:="COMSTRING"=:CSTRING; 0=:CPNT
151221                IF ><2 THEN TTIFIELD.BCHNUM SHZ 2-3;IF BATAB(A)<0 GO JAB2 FI
151234                DO CALL TCI; WHILE A=15 OD   IF A><##@ GO BILCMND
151244                DO CALL TCI IF A=11 THEN A:=## ; FI CALL CWRITE WHILE ><15 OD
151256          FI; 0=:CPNT; GO LEAVE
151260   RBUS
151266
151266   %=================================================================
151266   % 19.3        E D I T   M O N E D I T
151266
151266   % SUBROUTINE TO DO QED-COMPATIBLE LINE-EDITING
151266   SUBR EDIT,MONEDIT
151266   INTEGER ARRAY TABTA:=(10,16,36,50,62,74,106,120,0,0,0)  %TAB TABLE
151301   @ICR
151301   INTEGER ARRAY CCH:=(3NEXTI,CTA,CTS,CTC,CTD,CTE,CTF,BELL,CTH,CTI,3NEXTI,CTQ,
151315   CTL,CTM,BELL,CTO,CTP,CTQ,CTR,CTS,CTT,CTU,CTV,CTW,CTX,CTY,CTZ,BELL,BELL,BELL);@CR;
151337   DISP -200; INTEGER TCHAR; INTEGER POINTER NEXTI; PSID
151337   MONEDIT: K:=1; GO FELLS
151341   EDIT:    K:="0"
151342   FELLS: L=:D; CALL ENTER; T:="3NEXTI"=:"NEXTI"
151346          T:="WORKA"=:OSTRING; T:="COMSTRING"=:CSTRING
151352          IF K AND A><0 THEN
151355             A=:L; 1; *MON 2BRKM; MON 2ECHO
151361             GO ETCI              % CONTINUE INPUT AFTER NOWAIT RETURN
151362          FI
151362          0=:EMOD; 1; *MON 2BRKM; MON 2ECHO
151366   NEDIT: 0=:CPNT=:OPNT; DO CALL CREAD; CALL OWRITE WHILE ><15 OD
151376          0=:CPNT=:OPNT; GO NEXTI
151401   BELL:  7; CALL TCO
151403          DO
151403   3NEXTI:       CALL ETCI; A=:TCHAR
151405                IF =177 THEN 1 FI                    %DEL AS CTA
151411                IF <36 AND X:=CCH(A)><0 THEN X=:P FI %CTRL CHAR.
151420   STORE:        CALL CWRITE; IF EMOD=0 THEN CALL GETOLD FI
151424          OD
151425   OUT:   0=:NCOMPL; GO LEAVE   % RETURN, EDITING FINISHED
151427   *)FILL
151441
151441   % PROCESSING CONTROL CHARACTERS:
151441   CTA:   IF CPNT=0 GO BELL; CALL BACK;
151444          CALL VDUTTY; IF ><0 THEN CALL VDUBS
151447          ELSE ##^; CALL TCO FI; GO NEXTI
151453   CTC:   CALL GETOLD; IF =15 GO BELL; CALL CWRITE; CALL TCO; GO NEXTI
151462   CTD:   DO CALL OREAD; CALL CWRITE WHILE ><15; CALL TCO OD; CALL CRLF; GO OUT
151473   CTE:   IF EMOD=0 THEN 1=:EMOD; ##< ELSE 0=:EMOD; ##> FI; CALL TCO; GO NEXTI
151505   CTF:   DO CALL OREAD; CALL CWRITE WHILE ><15 OD; GO OUT
151514   CTH:   X:=OPNT; T:=OSTRING; *LBYT
151517          IF =15 GO BELL
151522          DO CALL GETOLD WHILE><15; CALL CWRITE; CALL TCO OD; GO NEXTI
151532   CTI:   FOR X:=0 TO 12 DO IF TABTA(X)=0 GO BELL WHILE A-1-CPNT<=0 OD
151546          FOR X:=A- DO CALL GETOLD; 40; CALL CWRITE; CALL TCO; OD; GO NEXTI
```

151235/ 125001
172366

151425/ 125001
173003

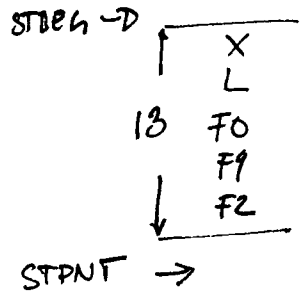
```
151556   CTL:    14; CALL CWRITE; CALL CRLF
151561   CTM:    15; CALL CWRITE; GO OUT
151564   CTO:    CALL ETCI; A=:TCHAR; CALL SRCHOLD
151567           DO CALL OREAD WHILE ><TCHAR; CALL CWRITE; CALL TCO OD
151576           OPNT-1=:OPNT; GO NEXTI; *)FILL
151615   CTP:    CALL ETCI; A=:TCHAR; CALL SRCHOLD
151620           DO CALL OREAD WHILE ><TCHAR; ##%; CALL TCO OD;OPNT-1=:OPNT;GO NEXTI
151633   CTQ:    0=:OPNT; IF CPNT=0 GO FAR BELL;
151637           CALL VDUTTY; IF ><0 THEN
151641                 DO CALL VDUBS; CPNT-1=:CPNT; WHILE ><0; OD
151647           ELSE 0=:CPNT; ##_; CALL TCO; CALL CRLF FI
151654           GO  NEXTI
151655   CTR:    12; CALL TCO
151657   CTRO:   FOR X:=OPNT TO 177 DO T:=OSTRING; *LBYT
151665                 WHILE ><15; CALL TCO
151671           OD; CALL CRLF
151674           FOR X:=0 TO CPNT-1 DO T:=CSTRING; *LBYT
151703                 CALL TCO; OD; GO NEXTI
151707   CTS:    CALL GETOLD; IF =15 GO FAR BELL; ##%; CALL TCO; GO NEXTI
151716   CTT:    CALL CRLF; FOR X:=0 TO CPNT-1 DO 40; CALL TCO OD; GO CTRO
151731   CTU:    FOR X:=0 TO 12 DO IF TABTA(X)=0 GO FAR BELL WHILE A-1-CPNT<=0 OD
151746           FOR X:=A- DO CALL GETOLD; IF =15 GO FAR BELL; CALL CWRITE;CALL TCO OD
151757           GO NEXTI; *)FILL
151773   CTV:    CALL ETCI; IF A=:TCHAR>=40 THEN CALL TCO
152001           ELSE ##&; CALL TCO; TCHAR+100; CALL TCO;TCHAR
152010           FI; GO FAR STORE
152011   CTW:    DO IF CPNT=0 GO FAR BELL; CALL BACK; CALL CREAD WHILE =40
152021                 CALL BACK; CALL VDUTTY; IF ><0 THEN CALL VDUBS FI; OD
152026           DO WHILE CPNT><0; CALL BACK; CALL CREAD WHILE ><40;
152035                 CALL BACK; CALL VDUTTY; IF ><0 THEN CALL VDUBS FI; OD
152042           CALL VDUTTY; IF =0 THEN ##\; CALL TCO FI; GO NEXTI
152047   CTX:    CALL ETCI; A=:TCHAR; CALL SRCHOLD
152052           DO CALL OREAD; A=:X; ##%; CALL TCO WHILE X><TCHAR OD; GO NEXTI
152063   CTY:    DO CALL GETOLD; CALL CWRITE WHILE ><15; OD; CALL CRLF; 0=:EMOD
152073           GO FAR NEDIT
152074   CTZ:    CALL ETCI; A=:TCHAR; CALL SRCHOLD
152077           DO CALL OREAD; A=:X; CALL CWRITE; CALL  TCO WHILE X><TCHAR OD
152107           GO NEXTI
152110   *)FILL
152126
152126   % SUBROUTINE TO TEST FOR VDU TERMINAL WITH BACKSPACE
152126   VDUTTY:IF TTIFIELD.CTTYP BIT 15 AND A BIT 16 THEN 1    %VDU WITH BS
152135           ELSE "0" FI; EXIT                                %HARD COPY
152140
152140   % SUBROUTINE TO OUTPUT BS SPACE BS TO THE VDU
152140   VDUBS: L=:X; 10; CALL TCO; 40; CALL TCO; 10; CALL TCO; X=:P
152150
152150   % SUBROUTINE TO GET A BYTE FROM OLDSTRING, AND BACKSPACE IF CR
152150   % RETURN: A=CHARACTER
152150   GETOLD: L=:D; CALL ENTER; CALL OREAD
152153           IF =15 THEN T:=OPNT-1=:OPNT FI; GO LEAVE
152162
152162   % SUBROUTINE TO SEARCH IN OLDSTRING FOR A CHARACTER
152162   % A=CHARACTER; RETURN ONLY IF CHARACTER IS FOUND
152162   SRCHOLD: FOR X=:D:=OPNT TO "5WCBUFSIZE+5WCBUFSIZE" DO
152167                 T:=OSTRING; *LBYT
152171                 IF A=TCHAR THEN X:=D; EXIT FI; IF =15 GO FAR BELL
152201           OD; GO FAR BELL
152204   RBUS
152207
```

```
152207   SUBR LEAV3
152207   RBUS
152207
152207   %==========================================================================
152207   % 19.4        G P A R   A G P A R   S G P A R
152207
152207   %SUBROUTINE TO GET PARAMETER
152207   % IF GPAR:
152207   %       ENTRY: A=DESCRIPTOR STRING
152207   %       EXIT:  A=INTEGER,FILE STRING,RT-PROG OR NAME STRING
152207   %
152207   % IF AGPAR:
152207   %       ENTRY: A=DESCRIPTOR STRING
152207   %       EXIT:  NO PARAMETER FOUND (USE DEFAULT VALUE)
152207   %       EXIT AD1: A=INTEGER,FILE STRING,RT-PROG OR NAME STRING
152207   %
152207   % IF SGPAR
152207   %       ENTRY:     A=DESCRIPTOR STRING
152207   %       EXIT:      ILLEGAL PARAMETER
152207   %       EXIT AD1:  A=PARAMETER
152207   %
152207
152207   SUBR GPAR,AGPAR,SGPAR
152207   %%INTEGER ILSNA:='$ILLEGAL SEGMENT NAME'
152207   INTEGER DELIM:=(#  ,#,,,15\15,-1),DESCR:=(#II,#SS,#RR,#NN,#GG,-1)
152221   INTEGER ALPN:=(#AZ,#09,-1),XSALPN:=(#AZ,#09,#--,#__,-1)
152231   DISP -200
152231          INTEGER OVAL,DVAL,MINFLAG,CHAR,TYPCHAR,RTN1,RTN2,RTN3,NIFLAG,AGPFLG
152231          INTEGER SERVFLG
152231          DOUBLE DRTN1=RTN1,DRTN2=RTN2; REAL RTNAME=RTN1
152231          INTEGER PCPNT=DVAL
152231          INTEGER NNCHAR=NIFLAG,TYP2CHAR=MINFLAG
152231   PSID
152231
152231   SGPAR: L=:D; CALL ENTER; T:=1=:SERVFLG; GO GPF
152236
152236   AGPAR: L=;D; CALL ENTER; 0=:SERVFLG
152241   AGPF:  T:=1=:AGPFLG; GO FELLS
152244   GPAR:  L=:D; CALL ENTER; 0=:SERVFLG
152247   GPF:   0=:AGPFLG
152250   FELLS: A=:OSTRING; 0=:OPNT=:NIFLAG
152253          CALL CREAD; CALL SRCHINT(DELIM); GOSW DEL,DEL,CRET,ALNUM
152263   ALNUM: 0=:CPNT; GO DEL
152265   CRET:  IF BCHFLAG><0 THEN 0=:CPNT; T:=CSTRING; X:=0; A:=15; *SBYT
152274          ELSE OSTRING=:X+1; CALL OUTTEXT; CALL GCOM; X=:OSTRING; 0=:OPNT
152304          FI
152304   DEL:   CALL SCAB
152305          IF T:=AGPFLG><0 AND A=15 OR A=##, THEN
152316                IF SERVFLG=0 GO LEAVE; GO LEAV2
152322          FI
152322          CALL OREAD; CALL SRCHINT(DESCR)
152325          GOSW INT,FAR SYMFIL,RTPROG,FAR NAME,FAR GSGNAM,FAR ERR; *)FILL
152346   INT:   CALL OREAD
152347   INTRT: A=:TYPCHAR; 0=:OVAL=:DVAL=:NIFLAG
152353          CALL CREAD; IF =##- THEN 1 ELSE CALL BACK;"0" FI; A=:MINFLAG
152364          DO     CALL CREAD; A=:CHAR; A-60; IF <0 OR >11 GO OUT
152373                 IF >7 THEN MIN NIFLAG FI
152377                 A=:T:=10*OVAL+T=:OVAL:=12*DVAL+T=:DVAL
152410          OD
152411   OUT:   IF CHAR=##D OR =##B THEN A=:TYPCHAR ELSE CALL BACK FI
```

```
152423          IF TYPCHAR=##D THEN DVAL ELSE IF NIFLAG><0 GO OERR; OVAL FI
152434          IF X:=MINFLAG><0 THEN A- FI; GO RETURN
152440
152440   PACKCHAR: DRTN1 SH 6; A=:RTN1; T:=CHAR BZERO 6
152445             DRTN2 SH 6; D+T; AD=:DRTN2
152451             EXIT
152452
152452   RTPROG: 0=:RTN1=:RTN2=:RTN3; CPNT=:PCPNT
152457          DO CALL CREAD; CALL SRCHINT(ALPN) WHILE =1 OD; T:=PCPNT=:CPNT
152470          IF =2 THEN ##0; GO INTRT FI % NUMBER, NOT NAME
152475          DO
152475                CALL CREAD; A=:CHAR; CALL SRCHINT(ALPN); IF =2 GO OUT2
152504                CALL PACKCHAR
152505          OD
152506   OUT2:  CALL BACK; RTNAME; CALL NAMSR; GO ERR; GO RETURN
152513
152513   RETURN: A=:OVAL; CALL CREAD; CALL SRCHINT(DELIM); IF =3 GO ERR
152522          CALL BACK; OVAL
152524   RETU1: IF T:=AGPFLG=0 THEN IF T:=SERVFLG=0 GO LEAVE; GO LEAV2 FI
152533          IF T:=SERVFLG=0 GO LEAV2; GO LEAV3
152537
152537   *)FILL
152546   SYMFIL: "WORKA"=:OSTRING=:OVAL; 0=:OPNT
152552          DO CALL CREAD; A=:CHAR; CALL SRCHINT(DELIM); IF <3 GO OUT3; CHAR; CALL OWRITE OD
152564   OUT3:  CALL BACK; ##'; CALL OWRITE; 0=:OPNT; OVAL; GO RETURN
152572
152572   NAME:  CPNT; GO RETU1
152574   OERR:  ER208; GO ERRR                    % NOT OCTAL NUMBER
152576
152576   ERR:   ER109                             % ILLEGAL PARAMETER
152577   ERRR:  IF X:=SERVFLG=0 THEN
152602             *MON 64
152603             X:=STPNT-STDELTA; "ERRRR"=:X.S1; GO LEAVE        % RETURN TO ERRRR
152610   ERRRR: FI; GO LEAVE
152611   ERRRS: ER207; GO ERRR
152613   *)FILL
152623
152623   GSGNAM: CALL OREAD; A=:TYP2CHAR
152625           "WORKA"=:OSTRING; 0=:NNCHAR; CPNT=:PCPNT
152632           DO CALL CREAD; A=:CHAR; WHILE A-60<<=11; MIN NNCHAR; OD
152642           IF NNCHAR><0 AND CHAR=##D OR A=##B THEN CALL CREAD; A=:CHAR FI
152655           PCPNT=:CPNT; CHAR; CALL SRCHINT(DELIM)
152662           IF A<<3 THEN
152665              IF TYP2CHAR=##G THEN ##0; GO FAR INTRT FI
152673           FI; 0=:RTN1=:RTN2=:RTN3; 0=:OPNT=:NNCHAR
152700           DO
152700                CALL CREAD; A=:CHAR; CALL SRCHINT(XSALPN)
152704           WHILE A><4
152707                IF A=1 AND NNCHAR=0 GO ERRRS
152714                MIN NNCHAR; CHAR; CALL OWRITE; CALL FAR PACKCHAR
152720                IF NNCHAR>7 GO ERRRS
152724          OD; CHAR; CALL SRCHINT(DELIM); IF A=3 GO ERRRS
152733          CALL BACK; ##'; CALL OWRITE
152736          IF TYP2CHAR=##S THEN
152742             IF NNCHAR=0 THEN A:=-1 ELSE A:=OSTRING FI; GO FAR RETURN
152750          FI; RTNAME; CALL SNAMSR; GO FAR ERRRS; GO FAR RETURN
152754
152754   RBUS
152767
152767   %STANDARD PARAMETERS:
```

173

```
152767
152767   INTEGER STRFILE:='S FILE NAME:'
152776   INTEGER STRNUM:='IONUMBER:'
153003
153003   INTEGER OILLP:='$NOT OCTAL NUMBER'
153014
153014   %==========================================================================
153014   % 19.5        C L F O R M   D E C 2
153014
153014   %SUBROUTINE TO OUTPUT CLOCK SETTING:HH.MM.SS
153014   SUBR CLFORM,DEC2
153014   INTEGER PARCL                    % B - 200 = BUNI.
153015   DISP -200; INTEGER BUNI,SEC,MINUTE,HOUR; PSID
153015   DEC2:  L=:D; CALL ENTER; A=:D:=0; T:=12; *RDIV ST
153023          D=:X; A+60; CALL TCO; 60+X; CALL TCO; GO LEAVE
153032
153032   CLFORM: L=:D; CALL ENTER; A:=B-200=:PARCL; "PARCL"; *MON 2CLOC
153041          HOUR; CALL DEC2; ##.; CALL TCO
153045          MINUTE; CALL DEC2; ##.; CALL TCO
153051          SEC; CALL DEC2; GO LEAVE
153054   RBUS
153057
153057   %==========================================================================
153057   % 19.6        R T O U T   P R I R T
153057
153057   % SUBROUTINE TO PRINT AN RT-NAME SYMBOLIC OR OCTAL
153057   SUBR RTOUT,PRIRT
153057   DISP -200
153057          INTEGER RTN1; DOUBLE DRTN2
153057          DOUBLE DRTN1=RTN1; REAL RTNAME=RTN1
153057   PSID
153057   RTOUT: L=:D; CALL ENTER
153061          A=:RTN1:="RTN1"+B; CALL INAMSR; GO OCTN; TAD=:RTNAME
153067          FOR X:=-10 DO
153070                IF RTN1 SHZ -12><0 THEN
153073                  IF A NBIT 5 THEN A BONE 6 FI; CALL TCO
153077                FI
153077                DRTN1 SH 6; A=:RTN1; DRTN2 SH 6=:DRTN2
153105          OD; GO LEAVE
153107   OCTN:  RTN1; CALL OCTU; GO LEAVE
153112
153112   % SUBROUTINE TO PRINT RT-NAME, FIXED FORMAT
153112   % T,A,X=NAME
153112   PRIRT: L=:D; CALL ENTER; X=:D; TAD=:RTNAME
153116          FOR X:=-10 DO IF RTN1 SHZ -12=0 THEN 40 FI
153123                IF A NBIT 5 THEN A BONE 6 FI; CALL TCO
153127                DRTN1 SH 6; A=:RTN1; DRTN2 SH 6=:DRTN2
153135          OD; GO LEAVE
153137   RBUS
153143
153143   %==========================================================================
153143   % 19.7        C R E A D   O R E A D   C W R I T E   O W R I T E
153143   %             B A C K S P   S C A B
153143   %
153143   %       S T R I N G  R O U T I N E S
153143
153143   %STRING I/O ROUTINES
153143   SUBR CREAD,OREAD,CWRITE,OWRITE,SCAB,CRLF,BACK
153143   INTEGER ILLST:='$TOO LONG STRING$'
153154
```



```
153154   CREAD: X=:D:=CPNT; MIN CPNT; T:=CSTRING; GO LBY
153161   OREAD: X=:D:=OPNT; MIN OPNT; T:=OSTRING
153165   LBY:   *LBYT
153166          GO OUT
153167
153167   CWRITE: X=:D:=CPNT; MIN CPNT; T:=CSTRING; GO SBY
153174   OWRITE: X=:D:=OPNT; MIN OPNT; T:=OSTRING
153200   SBY:   *SBYT
153201   OUT:   T=:XCSTRING
153202          IF X>>=5CBUFSIZE THEN
153205                T:=D; "ILLST"; X:=L; CALL OUTTEXT; D:=T; X=:L:=0
153214                T:=XCSTRING; 15; *SBYT
153217                IF CSTRING=T THEN X=:CPNT ELSE X=:OPNT FI
153225          FI
153225          D=:X; EXIT
153227   BACK: IF CPNT><0 THEN A-1=:CPNT FI; EXIT
153234
153234   SCAB: L=:D; CALL ENTER
153236   LOOP:  CALL CREAD; IF =40 GO LOOP
153242          T:=CPNT-1=:CPNT; GO LEAVE
153246   RBUS
153250
153250   %==========================================================================
153250   % 19.8        3 O U T T E X T
153250
153250   %SUBROUTINE TO OUTPUT TEXT ON TELETYPE
153250   %A-REG. POINTS TO STRING TERMINATED WITH'
153250   %THE VARIABLE "TEXTADR" IS MOVED TO THE LOCATION AFTER THE STRING
153250
153250   SUBR 3OUTTEXT
153250   DISP -200; INTEGER PNT,XREG; PSID
153250   3OUTTEXT: L=:D; CALL ENTER; A=:PNT; X=:XREG
153254          FOR X:=0 TO 1000 DO
153260                T:=PNT; *LBYT
153262                WHILE ><##' AND A NBIT 7
153267                IF A=##$ THEN CALL CRLF ELSE CALL TCO FI
153275          OD; A:=X SHZ -1+1+PNT=:TEXTADR
153304          X:=XREG; GO LEAVE
153306   RBUS
153311
153311   %==========================================================================
153311   % 19.9        T E X T N
153311
153311   % SUBROUTINE TO OUTPUT NEXT STRING
153311   SUBR TEXTN
153311   TEXTN: L=:D; CALL ENTER; TEXTADR; CALL OUTTEXT; GO LEAVE
153316   RBUS
153316
153316   %==========================================================================
153316   % 19.10       O C T U   D E C U   D T D E C
153316
153316   %SUBROUTINES FOR INTEGER OUTPUT; VALUE IN A-REG.
153316   SUBR OCTU,DECU,DTDEC
153316   @DEC
153316   DATA(10000,1000,100,10,1); INTEGER ARRAY CONST(0)
153323   @OCT
153323   DISP -200; DOUBLE DVAL; INTEGER VALUE,NULLFLAG; PSID
153323   CHBLANK: IF =0 AND X><-1 AND T:=NULLFLAG=0 THEN 40
153333           ELSE A+60; MIN NULLFLAG FI; EXIT
153337
```

1224

```
153337   OCTO: L=:D; CALL ENTER; AD SHZ -17=:DVAL; 40; CALL TCO; 0=:NULLFLAG
153346         FOR X:=-6 DO DVAL; CALL CHBLANK; CALL TCO
153352               DVAL SH 20 SHZ -15=:DVAL
153356         OD; GO LEAVE
153360
153360   DTDEC: L=:D; CALL ENTER; A=:VALUE; GO DOUT
153364   DECU: L=:D; CALL ENTER; A=:X; 40;CALL TCO
153371         IF X<0 THEN X-; ##- ELSE 40 FI; CALL TCO; X=:VALUE
153401   DOUT:  0=:NULLFLAG
153402         FOR X:=-5 DO VALUE=:D; A:=0; T:=CONST(X); *RDIV ST
153410               T:=D=:VALUE; CALL CHBLANK; CALL TCO
153414         OD; GO LEAVE
153416   RBUS
153420
153420   %=================================================================
153420   % 19.11      C R L F   T C O   T C I   T C O 2   W I N B   E T C I
153420   %
153420   %      T E R M I N A L   I / O
153420
153420   SUBR CRLF,TCO,TCI,TCO2,WINB,ETCI
153420   INTEGER POINTER CCUSER:=CUSER
153421   CRLF: L=:D; CALL ENTER; 15; CALL TCO; 12; CALL TCO; GO LEAVE
153430   TCO2: L=:D; CALL ENTER; A=:X SHZ -10; CALL TCO
153435         377/\X; CALL TCO; GO LEAVE
153441   TCO:  L=:D; X=:T
153443         IF X:=TTIFIELD.FLAGB NBIT 5SPSIO THEN
153447           T=:X:=TDVN; CALL SETPARITY; *MON 20UTB; JMP * 1; COPY SD DP
153455         FI; X:=T; D=:P
153457   TCI:  L=:D; CALL ENTER
153461         IF X:=TTIFIELD.FLAGB NBIT 5ESCON AND CCUSER>=0 THEN
153467           CALL ESCON
153470         FI
153470         IF X BIT 5SPSIO THEN
153472            IF "MLIDFIELD".MLICPNT=X.MLIMXADDR GO ERR
153477            X=:T:=A; *LBYT
153502            MIN "MLIDFIELD".MLICPNT; X:=TTIFIELD.FLAGB; GO L1
153507         FI; T:=1
153510   WINB: *MON 2INBT
153511   ERR:  CALL LOGOUT
153512   L1:   IF X NBIT 5ESCON THEN CALL ESCOFF FI; A/\177
153516         A=:T; A:=TTIFIELD.CESCP/\377; A:=:T
153523         IF A BIT 6 AND BIT 6 AND A><T AND A><177 AND A><X.CESCP SHZ -10 THEN
153540            A BZERO 5; FI
153541         IF =15 THEN CALL CRLF; 15 FI
153546         GO LEAVE
153547   *)FILL
153564
153564   % SPECIAL INPUT ROUTINE WHEN READING FROM 0 IN USER PROGRAMS
153564   % THE ROUTINE TESTS ON 5LSTA AND GIVES ERROR RETURN 316 IF LINE DOWN
153564   % IF NOT NOWAIT THE ROUTINE CONTINUES IN TCI.
153564
153564   ETCI:  L=:D
153565         IF COBSTATE><5BUSER GO TCI                    % NOT USER PROGRAM
153571         IF TTIFIELD.FLAGB BIT 5LSTA THEN A:=TER02; GO ERETU FI   % TERMINAL NOT CONNECTED
153577   @LIB CXCPU
153577         IF X.TYPRING BIT 5TERM THEN X:=X.TDRADDR FI
153603   @ELIB
153603         IF X.ISTATE>=0 GO TCI
153606         IF X:=TTIFIELD.FLAGB NBIT 5ESCON AND CCUSER>=0 THEN
153614            CALL ESCON FI
```

ZTDVN

COPY SD DP

{TCO/ JMP I*+1
172547

Bank 7 mem. error

```
153615          T:=1; *MON 2INBT
153617          GO ERETU; IF X NBIT 5ESCON THEN CALL ESCOFF FI; A/\177
153624          A=:T; A:=TTIFIELD.CESCP/\377; A:=:T
153631          IF A BIT 5 AND BIT 6 AND A><T AND A><177 AND A><X.CESCP SHZ -10 THEN
153646             A BZERO 5; FI
153647          D=:X
153650          IF =15 THEN CALL CRLF; 15 FI
153655          X=:P
153656
153656   % ERROR RETURN FROM INBT.
153656   % RETURN TO EDINBT WITH SKIPRETURN AND ERROR CODE IN D-REG
153656   % IF NOWAIT RETURN A-REG = CALL ADDRESS, ELSE A-REG = 0
153656   ERETU: A:=:D; IF D><3 THEN A:="0" FI; GO LEAV2
153664   RBUS
153675
153675
153675   %==========================================================================
153675   % 19.12       S R H I N T  -  S R C H I N T
153675
153675   % SUBROUTINE TO SEARCH FOR LEGAL INTERVAL
153675   %CALL+1 POINTS TO LIMIT TABLE, TERMINATED BY -1; A=VALUE
153675   %EXIT: A=INTERVAL NUMBER
153675
153675   SUBR SRHINT,SRCHINT
153675   DISP -200; INTEGER ARRAY POINTER ARR; PSID
153675   SRCHINT:
153675   SRHINT: L=:D; CALL ENTER; X:=D.S0=:"ARR"; X:=-1; A=:T
153704           DO     X+1; IF ARR(X)<0 GO OUT
153707                  AD SHZ -10; D SHZ -10
153711                  IF T>=A AND T<=D GO OUT
153715           OD
153716   OUT:    X=:A; GO LEAV2
153720   RBUS
153721
153721   %==========================================================================
153721   % 19.13       3 E N T E R   L 3 E A V E   L E A V 2   L E A V X   L E A V 3
153721
153721   %SUBROUTINES FOR SUBROUTINE ENTER AND LEAVE
153721   SUBR 3ENTER,L3EAVE,LEAV2,LEAVX,LEAV3
153721   INTEGER STKERR:='STACK ERROR'
153727   DISP 0; DOUBLE ADSV; REAL V0SV,V1SV,V2SV; PSID
153727
153727   3ENTER: TAD=:SAVTAD; X=:INDEX
153731           IF STPNT-"STEND">=0 GO ERR
153734           X=:A; AD=:STPNT.ADSV
153737           FV0=:X.V0SV; FV3=:X.V1SV; FV6=:X.V2SV
153745           X+STDELTA=:STPNT; SAVTAD; X:=INDEX; EXIT
153752
153752   L3EAVE:TAD=:SAVTAD; X=:INDEX; STPNT-STDELTA=:STPNT=:X
153760           IF A-"STBEG"<0 GO ERR
153762           MIN X.S1; MIN X.S1; GO LEV
153765   LEAVX:  TAD=:SAVTAD; T:=X; STPNT-STDELTA=:STPNT=:X
153773           IF A-"STBEG"<0 GO ERR
153775           MIN X.S1; T=:X.S0; GO LEV
154000   LEAV2:  TAD=:SAVTAD; STPNT-STDELTA=:STPNT=:X
154005           IF A-"STBEG"<0 GO ERR; MIN X.S1; GO LEV
154011   3LEAVE: TAD=:SAVTAD; STPNT-STDELTA=:STPNT=:X
154016           IF A-"STBEG"<0 GO ERR
154020   LEV:    X.V0SV=:FV0; X.V1SV=:FV3; X.V2SV=:FV6
154026           X.ADSV; A=:X; D=:L; SAVTAD; EXIT
```

```
154033   ERR:    "STBEG"=:STPNT; "STKERR"; CALL OUTTEXT; GO OPCOR
154040   RBUS
154044
154044
154044   %=========================================================================
154044   % 19.14       R O F I P A R
154044   %
154044   % SUBROUTINE TO READ AND OPEN THE OUTPUT-FILE
154044   %
154044   % EXIT:       ERROR
154044   %
154044   % EXIT+1:     OK, FILE NUMBER IN TDVN
154044   %
154044   SUBR ROFIPAR
154044   ROFIPAR: L=:D; CALL ENTER
154046           "STROFILE"; CALL AGPAR; A:=1
154051           IF A><1 THEN
154054                 A=:X:="TYPS"; T:=0; *MON 2NOPE
154060                 GO CFILERR
154061           FI; A=:TDVN
154062           GO LEAV2
154063   CFILERR: *MON 64
154064           GO LEAVE
154065   RBUS
154072
154072
154072   @LIB 8S3C-,
154072   %=========================================================================
154072   %  19.15      R B G U F
154072   %
154072   %  SUBROUTINE TO ASK FOR PARAMETER:
154072   %  ENTRY :
154072   %    A=0 :   'RT OR BACKGROUND?'
154072   %    A=1 :   'USER-BREAK OR FATAL-ERROR?'
154072   %
154072   %  RETURN: OK - EXIT AD1; A=0 (RT/USER-BREAK), A=1 (BACKGROUND/FATAL-ERROR)
154072   %          ERROR - EXIT
154072
154072   SUBR RBGUF
154072
154072   @ICR;
154072   INTEGER PARTAB:=(TRT,0,0,0,  BACGR,0,1,0,
154102                    TUB,0,1,0,  TFER,0,2,0,
154112                    -1);
154113   @CR;
154113
154113   INTEGER TRT:='RT',BACGR:='BACKGROUND',TUB:='USER-BREAK'
154131   INTEGER TFER:='FATAL-ERROR'
154137   INTEGER TRTOBG:='N RT OR BACKGROUND? '
154152   INTEGER TUBOFE:='N USER-BREAK OR FATAL-ERROR? '
154171
154171   RBGUF: L=:D; CALL ENTER
154173          IF A=0 THEN            % RT OR BACKGROUND?
154174             "TRTOBG"; CALL GPAR
154176          ELSE IF A=1 THEN       % USER-BREAK OR FATAL-ERROR?
154202             "TUBOFE"; CALL GPAR
154204          FI;FI
154204          X:="PARTAB"; CALL ABLOOK
154206          IF ><0 THEN GO ILLPA ELSE T.CMAND FI; GO LEAV2
154213   ILLPA: 174                    % ILLEGAL PARAMETER
```

```
154214          GO LEAVE
154215   .RBUS
154223
154223
154223
154223   %=========================================================================
154223   %  19.16      E X E C C
154223   %
154223   %  SUBROUTINE TO EXECUTE COMMANDS IN TERMINATION-HANDLING BUFFER
154223   %  MUST NOT BE CALLED FROM A ROUTINE USING ENTER/LEAVE
154223
154223   SUBR EXECC
154223
154223   DISP -200; INTEGER CBUFP; INTEGER POINTER LREG; PSID
154223
154223   EXECC: A:=L=:"LREG"
154225          IF PETECOM><"TECBUF" THEN
154231             2=:PASSTYPE          % ENABLE EXECUTION OF SYSTEM COMMANDS
154233             A:=T=:CBUFP; *MON 2CMND
154236             DO                   % SKIP TO NEXT COMMAND
154236                2=:PASSTYPE
154240                FOR X:=0 TO 177 DO T:=CBUFP; *LBYT
154246                WHILE ><##'
154251                OD; A:=X+2 SHZ -1+CBUFP=:CBUFP=:T; X:=0; *LBYT
154263             WHILE ><##'; CBUFP; *MON 2CMND
154270             OD; SPASTYPE=:PASSTYPE
154273          FI; GO LREG
154274
154274   RBUS
154276   @ELIB
154276   @LIB X8S3C-.
```

```
154276
154276  %==========================================================================
154276  %
154276  %   20.0          A C C O U N T I N G   S Y S T E M   ( SINC-2 )
154276  %
154276  %==========================================================================
154276  % 20.1        T I M U S E D   C T I M U S   T M T I M U S
154276  %
154276  % COMMAND: TIME-USED
154276  %
154276  % LISTS CPU TIME AND TOTAL TIME
154276
154276  SUBR TIMUSED,CTIMUS,TMTIMUS
154276
154276  INTEGER TIUIS:='$ TIME USED IS',OUTO:=' OUT OF'
154312  DISP -200; INTEGER 1TIMX,2TIMX,PRG,SEG; PSID
154312
154312  TIMUSED:IF ACCFLAG=0 OR PRJN=0 THEN EXIT FI        % ACCOUNTING OFF OR NO PROJ.PASSWORDS
154317  CTIMUS: L=:D; CALL ENTER; 1TIMON=:1TIMX; 2TIMON=:2TIMX; CURPROG=:PRG; GO TIMCOM
154330  TMTIMUS:L=:D; CALL ENTER; A=:PRG; T=:SEG
154334          X:="1TIMON"; CALL GET1L; CALL ERRFATAL; A=:1TIMX; T:=SEG
154341          X:="2TIMON"; CALL GET1L; CALL ERRFATAL; A=:2TIMX
154345  TIMCOM: "TIUIS"; CALL OUTTEXT; *MON 2TIME
154350          A:=:D-2TIMX; *RDCR ADC DD
154353          A:=:D-1TIMX; AD=:TIOOF; AD:=PRG.DTINT; CALL TIMOUT
154361          "OUTO"; CALL OUTTEXT; AD:=TIOOF; CALL TIMOUT; GO LEAVE
154366  RBUS
154377  @ELIB
154377  @LIB 8S3C-,
154377  %==========================================================================
154377  % 20.2        A C C O U N T   I N I T A C C   C A C C O   S A C C O
154377  %
154377  %       ACCOUNTING ROUTINE
154377  %
154377  %       THIS ROUTINE ASSUMES THAT LOG OFF TIME IS PACKED IN WORKA(11) BY DATCL.
154377  %
154377  INTEGER ACER:='$ACCOUNTING ERROR: FILE SYSTEM ERROR:'
154422  INTEGER RTBES=?
154422
154422  SUBR ACCOUNT,INITACC,CACCO,SACCO
154422  %
154422  % IF ANY ACCOUNTING FILES HAVE TO BE CREATED THE FOLLOWING DEFINITIONS ARE USED
154422  %
154422  INTEGER AFIL2:='"ACCOUNTS"'                          % TO CREATE ACCOUNTS
154430  INTEGER PRON2:='"PROJNAM"'                           % TO CREATE PROJNAM
154435  INTEGER RTPR2:='"RTPROJ"'                            % TO CREATE RTPROJ
154442  SYMBOL DEFDES=1750                                   % DEFAULT NUMBER OF ACCOUNTS BEFORE A WARNING IS GIVEN.
154442  SYMBOL DEFMAX=2260                                   % DEFAULT MAXIMUM NUMBER OF ACCOUNTS
154442  INTEGER DESIR:='IDNUMBER OF ACCOUNTS BEFORE WARNING: '
154465  INTEGER MAXA:='IDMAXIMUM NUMBER OF ACCOUNTS: '
154505  INTEGER AFOPAR:=(FILNO,NULL,WORKA,V0,"ACCRL")        % WRITE ACCOUNTS:DATA RECORD
154512  INTEGER AFIPAR:=(FILNO,NULL,V0,NULL,"3")             % READ/WRITE BLOCK 0 ACCOUNTS
154517  INTEGER AJOPAR:=(FILNO,NULL,WORKA+20,V0,"10")        % READ PROJNAM
154524  INTEGER PARA1:=(ICCRT,V3+2,"2")                      % GET RT DATA ( ICCRT )
154527  INTEGER PUBAC:='N'                                   % PUBLIC ACCESS TO ACCOUNTING FILES
154530  INTEGER FR1AC:='RWA'                                 % FRIEND ACCESS TO ACCOUNTS:DATA (FOR RT)
154532  INTEGER FR2AC:='R'                                   % FRIEND ACCESS TO PROJNAM:DATA AND RTPROJ:DATA (FOR RT)
154533  INTEGER OWNAC:='RWAD'                                % OWN ACCESS TO ACCOUNTING FILES
154536
154536  *)FILL
```

```
154542
154542   DISP -200; INTEGER NMBRR,BLNUM=NMBRR,DSNMR,MXNMR,SAERR,ACERR; PSID
154542
154542   ACCOUNT:IF ACCFLAG=0 OR PRJN=0 THEN EXIT FI        % ACCOUNTING OFF OR NO PROJECT PASSWORD SPECIFIED
154547           L=:D; CALL ENTER;
154551           X:="WORKA"                                 % USER-NAME INTO WORKA
154552           CALL FILSYS(GMAIN); GO FAR ERR
154555           CALL FILSYS(TUSEN); GO FAR ERR
154560           CALL FILSYS(GUSEN); GO FAR ERR
154563           X=:T:=0;17=:D                              % USER-NAME = 16 (DEC) CHARACTERS
154567           FOR X TO D DO; *LBYT
154572             IF A-47=0 GO OUT
154574           OD
154576   OUT:    A:=40; FOR X TO D DO; *SBYT                % REPLACE ' WITH SPACE AND FILL IN REST OF NAME WITH SPAC
154602           OD
154604           4=:WORKA(10)                               % FLAG RECORD AS A COMBINED ND-100/ND-500 ACCOUNTING RECO
154607           AD:=TIOOF=:DWORKA(13)                      % LOG-ON DURATION (TERMINAL TIME): BASIC TIME UNITS
154612           TTNO=:WORKA(15)                            % TERMINAL NUMBER
154615           AD:=CURPROG.DTINT=:DWORKA(16)              % CPU-TIME : BASIC TIME UNITS
154621           T:=20=:D                                   % 20 BYTES
154623           X:="WORKA"+T; A:="PRNAM"                   % COPY PROJECT NAME FROM SYSTEM SEGMENT TO RECORD TO
154626           *140131; JMP *                             % BE COPIED TO ACCOUNTS:DATA
154630           IF "IOACTAB" = 0 THEN
154632              0=:A=:D
154634           ELSE
154635              CURPROG-RTSTART=:D:=0;
154641              T:=5RTSIZE; *RDIV ST
154643              A SHZ 1 + "IOACTAB"=:X; CALL 1DREAD     % READ NUMBER OF BLOCK I/O TRANSFERS THROUGH THE FILE SYS
154647           FI
154647           AD=:DWORKA(30)                             % I/O ACCESSES.
154651           AD:=CT500=:DWORKA(32)                      % ND-500 CONNECT TIME.
154654           AD:=CPUT5=:DWORKA(34)                      % ND-500 CPU TIME.
154657           GO NN1
154660   *)FILL
154673
154673   NN1:    CALL FILSYS(RLUSE) 0/\0                    % LOG OUT USER
154676           X:="BAUSER";                               % AND ENTER SYSTEM
154677           CALL FILSYS(ENSYS); GO ERR                 % TO OPEN SYSTEM-FILE
154702           A:="ACCSEMRE"; *MON 2RESR                  % RESERVE ACCOUNTING SEMAPHORE
154704
154704           X:="AFILNAME"; T:=2; A:="TYPD"             % OPEN ACCOUNTS:DATA
154707           *MON 2NOPE;     JMP ERR
154711           A=:FILNO=:T; A:=ACCRL; *MON 2SBLZ; JAF ERR% SET BLOCK-SIZE
154716           "AFIPAR"; *MON 2RFIL; JAF ERR              % READ BLOCK 0
154721           0=:ACERR; MIN NMBRR
154723           IF NMBRR >= MXNMR THEN
154727             244=:ACERR                               % ERROR : NO. > MAXNO.
154731             GO RETA                                  % NO BLOCK TO BE WRITTEN
154732           FI
154732           IF A>=DSNMR THEN 243=:ACERR; FI            % ERROR : NO. > DESNO.
154737           T:=FILNO; "AFOPAR"; *MON 2WFIL; JAF ERR    % WRITE NEW BLOCK
154743           T:=FILNO; "AFIPAR"; *MON 2WFIL; JAF ERR    % WRITE BLOCK 0
154747   RETA:   T:=FILNO; *MON 2CLOS; JMP *1               % CLOSE ACCOUNTS
154752           "ACCSEMRE";*MON 2RELE                      % RELEASE SEMAPHORE
154754           IF ACERR >< 0 THEN
154756              *MON 2ERMS                              % APPROACHING OR AT THE END OF THE ACCOUNTING FILE.
154757           FI
154757           GO LEAVE
154760   ERR:    A=:SAERR:="ACER"; CALL OUTTEXT             % ERROR
154763           A:=SAERR; *MON 2ERMS
```

```
154765          0=:ACERR                                % CLEAR ACERR: THIS IS SFLAG FROM START- AND STOP-ACC.
154766          GO RETA
154767  *)FILL
155001  %-----------------------------------------------------------------------------
155001  %       COMMAND :  @INIT-ACCOUNTING
155001  %
155001  INITACC: L=:D; CALL ENTER
155003  ERRST: "DESIR"; CALL GPAR; IF A=0 THEN DEFDES FI  % DESIRED NO. ACCOUNTS: ENTRY FROM ERROR IN @START-ACCOU
155007          A=:DSNMR
155010          "MAXA";  CALL GPAR; IF A=0 THEN DEFMAX FI  % MAX NO. ACCOUNTS
155014          A=:MXNMR; 0=:NMBRR                        % BLOCK NO. VALUES
155016          "ACCSEMRE"; *MON 2RESR                    % RESERVE ACCOUNTING SEMAPHORE
155020          X:="AFILNAME"; T:=2; A:="TYPD"            % OPEN ACCOUNTS:DATA
155023          *MON 2NOPE
155024          GO ERR1; A=:FILNO                         % IF ERROR CHECK THAT THE FILE EXISTS.
155026  CONT1: T:=FILNO; "AFIPAR"; *MON 2WFIL; JAF ERR    % WRITE BLOCK 0
155032          *MON 2CLOS; JMP ERR                       % CLOSE ACCOUNTS
155034          "ACCSEMRE"; *MON 2RELE                    % RELESE SEMAPHORE
155036          1; GO SUIT1                               % GO TO START-ACCOUNTING: A=1 GIVES START-ACC.
155040  ERR1:   IF A >< 56 GO ERR
155043          X:="AFIL2"; T:=2; A:="TYPD"               % CREATE FILE
155046          *MON 2NOPE; JMP ERR
155050          A=:FILNO; X:="AFILNAME"
155052          T:="PUBAC"; A:="OWNAC"=:D:="FR1AC"
155056          *MON 2SFAC; JMP ERR                       % SET FILE ACCESS
155060
155060          GO CONT1
155061  *)FILL
155076  %-----------------------------------------------------------------------------
155076  %       COMMANDS :  @START-ACCOUNTING
155076  %                   @STOP-ACCOUNTING
155076  %
155076  INTEGER ACCRT:='ACCRT',                        PARA2:=ACCRT
155102  INTEGER INTERV:='IDLOGING INTERVAL (SEC.): ',NCCRT:='$ACCRT NOT LOADED$'
155132  INTEGER ACBACK:='N BACKGROUND :',             ACRT:='N RT: '
155146  INTEGER CFRTL:='N CLEAR LOGGED INFORMATION: '
155165  INTEGER ACSPOOL:='N SPOOLING :'
155174  INTEGER CNNTS:='$BACKGROUND ACCOUNTING MUST BE STARTED FIRST'
155223
155223  DISP -200; INTEGER CPT1,CPT2,CPT3,STERR,SFLAG,XPERIOD,BART ;PSID
155223
155223  CACCO: 1; GO SUITO                                % START-ACCOUNTING
155225  SACCO: "0"                                        % STOP-ACCOUNTING
155226  SUITO: L=:D; CALL ENTER
155230  SUIT1: A=:SFLAG                                   % START/STOP FLAG
155231          IF A = 0 GO STPOK                         % IF WE ARE STOPPING ACCOUNTING WE DO NOT NEED TO MAKE CH
155232          "PARA2"; *MON 151                         % RT DESCRIPT. ADDRESS OF ACCRT-PROGRAM
155234          IF A < 0 THEN "0"; FI                     % ACCRT NOT LOADED
155236          A=:ICCRT
155237  %
155237  %       CHECK TO SEE IF ACCOUNTS:DATA EXISTS AND HAS AT LEAST 1 PAGE.
155237  %
155237          1=:STERR                                  % SET START-ACC ERROR FLAG
155241          "ACCSEMRE"; *MON 2RESR                    % RESERVE ACCOUNTING SEMAPHORE
155243          T:=1; X:="AFILNAME"; A:="TYPD";
155246          *MON 2NOPE                                % OPEN ACCOUNTS:DATA FOR SEQUENTIAL READ
155247          GO STER1                                  % ERROR: IT DOES NOT EXIST
155250          A=:T; *MON 2INBT                          % TRY TO READ 1 BYTE FROM IT
155252          GO STER2                                  % FILE HAS NO PAGES
155253          0=:STERR                                  % ACCOUNTS:DATA CORRECT
```

```
155254   STER2: *MON 2CLOS; JMP *+1
155256   STER1: "ACCSEMRE"; *MON 2RELE                    % RELEASE ACCOUNTING SEMAPHORE
155260          IF STERR >< 0 GO FAR ERRST                % INITIALISE ACCOUNTS:DATA
155263   %
155263   %      BACKGROUND ACCOUNTING FOR ND-100/ND-500
155263   %
155263   STPOK: IF ACCFLAG = SFLAG GO SUIRT                % ALREADY STARTED/STOPPED
155267          "ACBACK"; CALL GPAR; CALL YESNO           % BACKGROUND ?
155272          IF A=-1 THEN
155275             "ILLP"; CALL OUTTEXT; GO LEAVE
155300          FI
155300          IF A = 1 THEN                             % YES
155303             IF SFLAG=1 THEN
155307                1=:ACCFLAG;                         % ACCOUNTING FLAG "ON"
155311                0=:BART; CALL OPCHECK;              % (SYSTEM)PROJNAM ?
155313             ELSE
155314                0=:ACCFLAG;                         % ACCOUNTING FLAG "OFF"
155315             FI
155315          FI
155315          GO SUIRT
155316   *)FILL
155332   %
155332   %      RT-ACCOUNTING
155332   %
155332   SUIRT: IF RTACCFLAG = SFLAG GO SUISP              % ALREADY STARTED/STOPPED
155336          IF PACTAB=0 GO SUISP                      % OPTION NOT INCLUDED IN THIS SINTRAN
155340          "ACRT"; CALL GPAR; CALL YESNO             % RT ?
155343          IF A = -1 THEN
155346             "ILLP"; CALL OUTTEXT; GO LEAVE
155351          FI
155351          IF A=1 THEN                               % YES
155354            IF SFLAG=0 THEN                         % STOP-ACCOUNTING
155356               IF ICCRT><0 THEN
155360                  "PARA1"; *MON 2ABOR               % ABORT ACCRT
155362               FI
155362               0=:RTACCFLAG                         % RT-ACCOUNTING FLAG "OFF"
155363            ELSE                                    % START-ACCOUNTING
155364               T:=5ACCS; X:=117776; CALL GET1L; *JMP*+1 % CONTENT OF START ADDRESS
155370               IF A=153134 THEN
155373                  "NCCRT";CALL OUTTEXT              % ACCRT NOT LOADED
155375                  GO SUISP
155376               FI
155376               "CFRTL";CALL GPAR;CALL YESNO                % CLEAR LOGGED ?
155401               IF A=-1 THEN "ILLP";CALL OUTTEXT;GO LEAVE;FI
155407               IF A= 1 THEN                                % YES
155412                 NBRTP-=:CPT1;PACTAB+1=:CPT2               % USER RT PROGRAMS
155420                 NBSRT SHZ 1 + PIOACTAB=:CPT3              % SYSTEM RT PROGRAMS
155424                 FOR CPT1 DO                               % CLEAR LOGGED INFORMATION
155424                      0=:A=:D                              % FOR ALL USER RT PROGRAMS
155426                      X:=CPT2; CALL 1DWRITE                % CPUTIME
155430                      IF X:=PIOACTAB >< 0 THEN
155432                         X:=CPT3                           % CLEAR LOGGED IO FOR
155433                         CALL 1DWRITE                      % USER RT PROGRAMS
155434                      FI
155434                      CPT2+3=:CPT2; CPT3+2=:CPT3
155442                 OD
155444               FI
155444               1=:RTACCFLAG; 1=:BART; CALL OPCHECK  % (SYSTEM)RTPROJ ?
155451               "INTERV"; CALL GPAR; A=:XPERIOD       % SAMPLING-FREQUENCY
155454               IF A>0 THEN "PARA1"; *MON 2INTV      % EXECUTION PERIOD
```

```
155460                  "PARA1";*MON 2RT                    % EXECUTE ACCRT
155462               FI
155462            FI
155462         FI
155462         GO SUISP
155463   *)FILL
155506   %
155506   %     SPOOLING ACCOUNTING
155506   %
155506   SUISP: IF SPACCFLAG=SFLAG GO LEAVE                 % SPOOLING ACCOUNTING
155512         "ACSPOOL"; CALL GPAR; CALL YESNO
155515         IF A=-1 THEN
155520            "ILLP"; CALL OUTTEXT; GO LEAVE
155523         FI
155523         IF A = 1 THEN
155526            IF SFLAG = 1 THEN
155532                  1=:SPACCFLAG
155534            ELSE
155535               0=:SPACCFLAG
155536            FI
155536         FI
155536         GO LEAVE
155537
155537   %-----------------------------------------------------------------------
155537   %     SUBROUTINE TO CHECK IF THE FILES EXIST
155537   %
155537   OPCHECK: "ACCSEMRE"; *MON 2RESR                    % RESERVE SEMAPHORE
155541         IF BART=0 THEN
155543            X:="PROJNAM"                             % PROJNAM OR
155544         ELSE
155545            X:="XRTPROJ"                             % RTPROJ
155546         FI
155546          T:=2; A:="TYPD"; *MON 2NOPE; JMP OPERR     % IF NOT EXIST : CREATE
155552          A=:FILNO=:T; *MON 2CLOS; JMP I (ERR        % CLOSE FILE
155556          "ACCSEMRE"; *MON 2RELE                     % RELESE SEMAPHORE
155560          EXIT                                       % RETURN SUIT1
155561
155561   %-----------------------------------------------------------------------
155561   %     SUBROUTINE TO CREATE THE FILE IF IT DOES NOT EXIST, INITIALISE
155561   %     THE INDEX BLOCK (BLNUM=NMBRR=0) AND SET THE FILE ACCESS.
155561   %
155561   OPERR: IF A><56 GO FAR ERR                         % NOT "NO SUCH FILE" ERROR
155564         IF BART=0 THEN X:="PRON2"                   % PROJNAM:DATA
155567                   ELSE X:="RTPR2"                   % RTPROJ:DATA
155571         FI
155571         T:=2; A:="TYPD"; *MON 2NOPE; JMP I (ERR      % OPEN FILE
155575         A=:FILNO=:T; 0=:NMBRR; "AFIPAR"; *MON 2WFIL % WRITE INDEX BLOCK
155602         IF A><0 GO FAR ERR                           % ERROR
155604         T:=FILNO; *MON 2CLOS; JMP I (ERR             % CLOSE FILE
155607         IF BART=0 THEN
155611            X:="PROJNAM"
155612         ELSE
155613            X:="XRTPROJ"
155614         FI
155614         T:="PUBAC"; A:="OWNAC"=:D:="FR2AC"
155620         *MON 2SFAC; JMP I (ERR
155622         "ACCSEMRE"; *MON 2RELE                      % RELESE SEMAPHORE
155624         EXIT                                        % RETURN SUIT1
155625
155625   RBUS
```

```
155645
155645  %=================================================================
155645  % 20.3        R S T A C   R S T O C
155645  %
155645  % COMMAND : START-RT-ACCOUNT   ;   STOP-RT-ACCOUNT
155645  %           START/STOP ACCOUNTING OF A SPECIFIC RT-PROGRAM
155645
155645  INTEGER NIMPL:='$NOT IMPLEMENTED'
155656
155656
155656  SUBR RSTAC,RSTOC
155656
155656  DISP -200;INTEGER BLNUM,BLMAX,SFLAG,PROGR,SAERR,SRTAD;PSID
155656
155656  INTEGER LERR:='$ILLEGAL RT REFERENCE',          STTRT;='S RT PROGRAM:'
155700  INTEGER PARAM:=(FILNO,NULL,WORKA+10,V0,"14"), PARA1:=(WORKA)
155706  INTEGER NOINFIL:='$RT PROG NOT IN FILE'
155721  INTEGER EMFILE:='$NO RT PROJECTS DEFINED'
155735
155735  *)FILL
155736
155736  RSTAC: 1;GO SUITE                          % START-RT-ACCOUNT
155740  RSTOC: "0"                                 % STOP-RT-ACCOUNT
155741  SUITE: L=:D; CALL ENTER; A=:SFLAG
155744         IF PACTAB=0 THEN "NIMPL"; CALL OUTTEXT;          % NOT IMPLEMENTED
155750            GO LEAVE
155751         FI
155751         "ACCSEMRE"; *MON 2RESR                           % RESERVE SEMAPHORE
155753         X:="XRTPROJ"; T:=3; A:="TYPD"; *MON 2NOPE; JMP FERR % OPEN RTPROJ
155760         A=:FILNO=:T:=14;   *MON 2SBLZ; JMP FERR          % BLOCK-SIZE = 14
155765         O=:BLNUM; "PARAM"; *MON 2RFIL; JAF FERR          % READ INDEX-BLOCK
155771         WORKA(10)=:BLMAX; IF A<=0 GO ERR1                % NO. BLOCKS IN FILE
155776         "STTRT"; CALL GPAR                               % RT-PROGRAM
156000         "PARA1"; *MON 151                                % RT DESCRIPTION
156002         IF A<0 OR A<"RTBES" GO ERR; A=:PROGR             % LEGAL RT-PROGRAM
156007         "WORKA"=:OSTRING; O=:OPNT                        % NAME INTO WORKA (GPAR)
156012         FOR X;=-10 DO; CALL OREAD; WHILE ><##'; OD       % PAD WITH SPACES
156020         IF X<0 THEN
156022           OPNT-1=:OPNT; A:=##  ; FOR X  DO; CALL OWRITE; OD
156030         FI
156030         1=:BLNUM                                         % LOOK UP ENTRY IN RTPROJ
156032         DO WHILE BLMAX>=BLNUM                            % READ NEW BLOCK
156036           "PARAM"; *MON 2RFIL; JAF FERR                  % 8 BYTES
156041            FOR D:=0 TO 3 DO
156045                 WORKA(D); X:=X+10; T:=WORKA(X)            % COMPARE
156051                 IF A><T GO NEXT
156053            OD                                            % FOUND
156055            GO TROUVE
156056  NEXT:     BLNUM+1=:BLNUM
156061         OD                                               % NOT FOUND
156062         "NOINFIL"; CALL OUTTEXT; GO FIN
156065  TROUVE:
156065         PROGR-"RTBES"=:D:=0; T:=5RTSIZE; *RDIV ST
156073         A=:D SHZ 1 + D + PACTAB=:X                       % INDEX IN ACTAB (ENTRY=3 W)
156100         IF SFLAG ><0 THEN                                % START-RT-ACCOUNT
156102            CALL 1IREAD
156103            IF A=0 THEN                                   % FLAG IN ACTAB
156104               T:=BLNUM; O=:A=:D
156107               CALL 1RWRITE
156110            FI
```

```
156110          ELSE                                                    % STOP-RT-ACCOUNT
156111             0=:A=:T=:D; CALL 1RWRITE                             % RESET FLAG IN ACTAB
156115          FI
156115   FIN:   T:=FILNO; *MON 2CLOS; JAF FERR                          % CLOSE RTPROJ
156120   FIN1:  "ACCSEMRE"; *MON 2RELE                                  % RELESE SEMAPHORE
156122          GO LEAVE
156123   ERR1:  "EMFIL"; CALL OUTTEXT; GO FIN                           % NO RT PROJECTS DEFINED.
156126   ERR:   "LERR";  CALL OUTTEXT; GO FIN                           % ONLY USER-RT-PROG.-ACC.
156131   FERR:  A=:SAERR; T:=FILNO; *MON 2CLOS; RAND                    % CLOSE FILE
156135          "ACER"; CALL OUTTEXT; A:=SAERR; *MON 2ERMS              % ERROR
156141          GO FIN1
156142   RBUS
156165
156165
156165   %==========================================================================
156165   % 20.4        L I S T - R T - A C C O U N T
156165   %
156165   %       LIST THE RT-PROGRAM NAME AND PROJECT NAME DEFINED IN RTPROJ:DATA
156165   %       AND THE RT-PROGRAM NAME FROM SINTRAN III TOGETHER WITH THE CPU
156165   %       TIME USED BY THIS PROGRAM.
156165   %
156165
156165   SUBR LIRTL
156165   INTEGER PARAM:=(FILNO,NULL,WORKA,V0,"14")
156172
156172   DISP -200; INTEGER BLNUM,CPT1,CPT2,SAERR,RTADR; PSID
156172
156172   INTEGER PARA1:=(V3+1)
156173   *)FILL
156174
156174   LIRTL: L=:D; CALL ENTER
156176          IF PACTAB=0 THEN
156200             "NIMPL"; CALL OUTTEXT; GO LEAVE
156203          FI
156203          "ACCSEMRE"; *MON 2RESR
156205          X:="XRTPROJ"; T:=3; A:="TYPD"; *MON 2NOPE; JMP FERR
156212          A=:FILNO=:T:=14; *MON 2SBLZ; JAF FERR
156217          0=:CPT1; PACTAB=:CPT2
156222          FOR CPT1 TO NBRTP DO
156226             X:=CPT2; CALL 1IREAD
156230             IF A><0 THEN
156231                A=:BLNUM; "PARAM"; *MON 2RFIL; JAF FERR          % THIS RT-PROGRAM IS BEING ACCOUNTED
156235                CALL CRLF                                        % READ THIS BLOCK NUMBER FROM RTPROJ:DATA.
156236                FOR X:=0 TO 3 DO; WORKA(X); CALL TCO2; OD
156246                A:=#  ; CALL TCO2; CALL TCO2                     % WRITE OUT RT-PROGRAM NAME FROM FILE
156251                GO SUITE
156252   *)FILL
156266   INTEGER INCONNU:='ACTAB INCOHERENT'
156277
156277   SUITE:        FOR X:=4 TO 13 DO; WORKA(X); CALL TCO2; OD      % WRITE OUT PROJECT NAME.
156307                 A:=#  ; CALL TCO2; CALL TCO2
156312                 CPT1*5RTSIZE+"RTBES"=:RTADR; "PARA1"; *MON 152
156320                 IF D=0 THEN
156322                    "INCONNU"; CALL OUTTEXT; GO TERMINE
156325                 FI
156325                 D=:X; CALL PRIRT; A:=#  ; CALL TCO2              % WRITE RT-NAME FROM SINTRAN.
156331                 X:=CPT2; CALL 1RREAD; CALL TIMOUT                % TIME-USED
156334              FI
156334              CPT2+3=:CPT2
156337           OD
```

```
156343   TERMINE:T:=FILNO; *MON 2CLOS; JAF FERR
156346   FIN:     "ACCSEMRE"; *MON 2RELE
156350            GO LEAVE
156351   FERR:    A=:SAERR; T:=FILNO; *MON 2CLOS; RAND
156355            "ACER"; CALL OUTTEXT; A:=SAERR; *MON 2ERMS
156361            GO FIN
156362
156362   RBUS
156376
156376   @ELIB
156376   @LIB X8S3C-,
156376   %=========================================================================
156376   % 20.5        G L P A R
156376
156376   % DESCRIPROR STRINGS FOR FILE SYSTEM COMMANDS
156376
156376   INTEGER ARRAY DDIRN:=('SADIRECTORY NAME: ')
156410   INTEGER ARRAY DOLDD:=('SAOLD DIRECTORY NAME: ')
156424   INTEGER ARRAY DNEWD:=('SANEW DIRECTORY NAME: ')
156440   INTEGER ARRAY DDEVN:=('SADEVICE NAME: ')
156450   INTEGER ARRAY DLOGU:=('IDDEVICE UNIT: ')
156460   INTEGER ARRAY SDLOG:=('IDDEVICE SUB-UNIT: ')
156472   INTEGER ARRAY DFRFL:=('SAFIXED(F) OR REMOVABLE(R): ')
156511   INTEGER ARRAY DUSEN:=('SUUSER NAME: ')
156520   INTEGER ARRAY DSPAC:=('IDNUMBER OF PAGES: ')
156532   INTEGER ARRAY DNUSE:=('SANEW USER NAME: ')
156543   INTEGER ARRAY DUSEI:=('IDUSER INDEX: ')
156553   INTEGER ARRAY DPASS:=('P PASSWORD: ')
156562   INTEGER ARRAY DOPAS:=('P OLD PASSWORD: ')
156573   INTEGER ARRAY DNPAS:=('P NEW PASSWORD: ')
156604   INTEGER ARRAY DFRIN:=('SAFRIEND NAME: ')
156614   INTEGER ARRAY DACCE:=('ALACCESS (R,W,A,C,D IN COMBINATIONS OR N): ')
156642   INTEGER ARRAY DFNAM:=('SFFILE NAME: ')
156651   INTEGER ARRAY DFSNA:=('SSFILE NAME: ')
156660   INTEGER ARRAY DPAGE:=('IBPAGE ADDRESS: ')
156671   INTEGER ARRAY DNFIL:=('SUNEW FILE NAME AND TYPE: ')
156707   INTEGER ARRAY DDVNU:=('IBDEVICE NUMBER: ')
156720   INTEGER ARRAY DPACC:=('ALPUBLIC ACCESS (R,W,A,C,D IN COMBINATIONS OR N): ')
156752   INTEGER ARRAY DFACC:=('ALFRIEND ACCESS (R,W,A,C,D IN COMBINATIONS OR N): ')
157004   INTEGER ARRAY DOACC:=('ALOWN ACCESS (R,W,A,C,D IN COMBINATIONS OR N): ')
157034   INTEGER ARRAY DOBJI:=('IDOBJECT INDEX: ')
157045   INTEGER ARRAY DACCM:=('AOACCESS MODE (R,W,RW,RX,WX,WA,RC,WC): ')
157071   INTEGER ARRAY DFNUM:=('IBFILE NUMBER: ')
157101   INTEGER ARRAY DBLSZ:=('IBBLOCK SIZE: ')
157111   INTEGER ARRAY DBYTP:=('IBBYTE POINTER: ')
157122   INTEGER ARRAY DBLOP:=('IDBLOCK NUMBER: ')
157133   INTEGER ARRAY DOUTF:=('SFOUTPUT FILE: ')
157143   INTEGER ARRAY DSOUD:=('SASOURCE DIRECTORY: ')
157156   INTEGER ARRAY DDESD:=('SADESTINATION DIRECTORY: ')
157173   INTEGER ARRAY DSOUF:=('SFSOURCE FILE: ')
157203   INTEGER ARRAY DDESF:=('SFDESTINATION FILE: ')
157216   INTEGER ARRAY DSDEV:=('SADESTINATION DEVICE: ')
157232   INTEGER ARRAY SCDEV:=('SASOURCE DEVICE: ')
157243   INTEGER ARRAY DPFNA:=('SFPERIPHERAL FILE NAME: ')
157260   INTEGER ARRAY DNUCO:=('IDNUMBER OF COPIES: ')
157273   INTEGER ARRAY BFAD:=('IBBIT FILE ADDRESS: ')
157306   INTEGER ARRAY MNCHK:=('SAMANUAL CHECK? ')
157317   INTEGER ARRAY VONA:=('SAVOLUME NAME: ')
157327   INTEGER ARRAY STYPE:=('SASOURCE TYPE: ')
157337   INTEGER ARRAY DSGEN:=('SAFILE GENERATION: ')
```

```
157351   INTEGER ARRAY DTYPE:=('SADESTINATION TYPE: ')
157364   INTEGER ARRAY FINRD:=('IDDESTINATION TAPE FILE NUMBER(1,2,3,...): ')
157412   INTEGER ARRAY FINRS:=('IDSOURCE TAPE FILE NUMBER(1,2,3,...): ')
157436   INTEGER ARRAY PRSFN:=('SAPRINTING NAME OF SPOOLING FILES? ')
157460   INTEGER ARRAY PSTSP:=('SASTOP AND WAIT FOR START-PRINT BEFORE PRINTING FILE? ')
157514   INTEGER ARRAY PNLPA:=('IDNUMBER OF LINES PER PAGE: ')
157533   INTEGER ARRAY PNPAG:=('IDNUMBER OF PAGES: ')
157545   INTEGER ARRAY PANLI:=('IDNUMBER OF LINES: ')
157557   INTEGER ARRAY PIBSF:=('SFINSERT BEFORE FILE: ')
157573   INTEGER ARRAY PAASF:=('SFAPPEND AFTER FILE: ')
157606   INTEGER ARRAY PINAPP:=('SAINSERT OR APPEND? ')
157621   INTEGER ARRAY PPIACC:=('SAPRINTING MESSAGE INDEPENDANT OF SPOOLING CONDITIONS? ')
157655   INTEGER ARRAY PFOID:=('X SPOOLING FORM IDENTIFICATION: ')
157676   INTEGER ARRAY DUFQU:=('SA? ')
157701   INTEGER ARRAY DRSND:=('SAREMOTE SYSTEM NAME: ')
157715   INTEGER ARRAY DRUND:=('SUREMOTE USER NAME: ')
157730   INTEGER ARRAY DPPAW:=('PPPROJECT PASSWORD: ')
157743   INTEGER ARRAY DNENT:=('IDNUMBER OF ENTRIES: ')
157756
157756
157756   % ROUTINE TO COLLECT PARAMETERS FOR FILE SYSTEM COMMANDS:
157756   % CALLED FROM FILE SYSTEM SEGMENT
157756
157756   % COLLECT PARAMETER FROM COMMAND STRING
157756   % IF NO MORE PARAMETERS ARE FOUND IN THE STRING, THE PARAMETER IS
157756   % ASKED FOR WITH THE SECOND PART OF THE DESCRIPTOR STRING
157756   % FIRST PART OF THE DESCRIPTOR STRING DEFINES PARAMETER TYPE
157756   % S MEANS SYMBOLIC PARAMETER (A: PURE ALFANUMERIC, U: USER NAME,
157756   % F: FILE NAME, S: FILE NAME WITH *'S)
157756   % I MEANS INTEGER NUMBER (B: OCTAL DEFAULT, D: DECIMAL DEFAULT)
157756   % P MEANS PASSWORD
157756   % A MEANS ACCESS WORD (L :LEGAL ACCESS, O: OPEN ACCESS)
157756   % X MEANS ALL CHARACTERS EXEPT (CR) AND '
157756
157756   % A-REG: ADDRESS OF DESCRIPTOR STRING
157756   % X-REG: ADDRESS OF STRING RETURNED IF SYMBOLIC PARAMETER
157756   % T-REG: MAX CHARACTERS IN RETURNED STRING
157756
157756   % RETURN - A-REG: ERROR CODE (PARAMETER TYPE ERROR)
157756
157756   % SKIPRETURN - AD-REG: VALUE IF INTEGER PARAMETER OR PASSWORD OR ACCESSWORD
157756   %               X-REG: -1 IF INTEGER, PASSWORD OR ACCESSWORD, 0 OTHERWISE
157756
157756   SUBR GLPAR
157756
157756   INTEGER ILDST:='$ILLEGAL DESCRIPTOR STRING'
157774
157774
157774   DISP -200
157774   INTEGER DESPO                  % DESCRIPTOR STRING POINTER
157774   INTEGER DESC1,DESC2            % TWO FIRST CHARACTERS OF DESCR. STRING
157774   INTEGER PNT1,PNT2              % STRING POINTERS
157774   INTEGER NUMB1,NUMB2
157774   DOUBLE NUMB=NUMB1              % DOUBLE PRECISION NUMBER
157774   INTEGER NEGFL                  % NEGATIVE NUMBER FLAG
157774   INTEGER CHAR                   % CHARACTER BUFFER
157774   INTEGER TREG,AREG,DREG,XREG; DOUBLE ADREG=AREG; REAL TADREG=TREG
157774   PSID
157774
157774   GLPAR: L=:D; CALL ENTER; TAD=:TADREG; X=:XREG
```

```
160000          0=:DESPO
160001          X:=AREG; T:=DESPO; CALL GETCH; A=:DESC1   % FIRST CH. IN DESCR.
160005          MIN DESPO; T:=DESPO; CALL GETCH; A=:DESC2
160011          IF DESC1=##P  GO TSPAC
160015          IF DESC1=##S THEN IF DESC2><##A AND ><##U AND ><##F AND ><##S GO ILDES
160036             GO TSPAC; FI
160037          IF DESC1=##I THEN IF DESC2><##D AND ><##B GO ILDES; GO TSPAC; FI
160053          IF DESC1=##A THEN IF DESC2><##L AND ><##O  GO ILDES; GO TSPAC; FI
160067          IF DESC1=##X GO TSPAC
160073   ILDES: "ILDST"; CALL OUTTEXT; GO LEAVE
160076
160076   *)FILL
160100
160100   TSPAC:
160100   IGNOR: X:="COMST"; T:=CPNT; CALL GETCH; A=:CHAR % READ FROM COM. STRING
160104          IF =15 GO FAR ASK                        % ASK FOR PAR. IF EMPTY
160107          IF X:=DESC1><##X THEN
160113             IF =##  THEN MIN CPNT; GO IGNOR; FI
160120          FI
160120   NPLGS: 0=:PNT1
160121          FOR PNT1 TO 120 DO                       % MOVE PARAMETER TO WORKA
160125                IF CHAR=15 GO ENDC                 % END OF STRING
160131                IF X:=DESC1><##X THEN
160135                   IF =##   OR =##, GO END         % END OF PARAMETER
160143                FI
160143                X:="WORKA"; T:=PNT1; CALL PUTCH; MIN CPNT
160147                X:="COMST"; T:=CPNT; CALL GETCH; A=:CHAR
160153          OD
160157   END:   MIN CPNT
160160   ENDC:  ##'; X:="WORKA"; T:=PNT1; CALL PUTCH      % INSERT '
160164          GO PAROK
160165
160165   *)FILL
160172
160172   % PARAMETER TRANSFERED TO PARAS
160172
160172   PAROK: IF DESC1=##S THEN
160176                0=:PNT2
160177                FOR PNT2 TO TREG-1 DO
160204                     X:="WORKA"; T:=PNT2; CALL GETCH
160207                     A=:CHAR
160210                     IF A<##A OR A>##Z THEN
160216                        IF <##0 OR >##9 THEN
160224                           IF ><##- AND ><##' THEN
160232                                 IF DESC2=##A GO FAR ILLC
160236                                 IF CHAR><##: THEN
160242                                    IF DESC2=##U GO FAR ILLC
160246                                    IF CHAR><##( AND ><##) AND ><##;
160256                                       AND ><##" AND ><##. THEN
160266                                       IF DESC2=##F GO FAR ILLC
160272                                       IF CHAR><##* GO FAR ILLC
160276                                 FI;FI;FI;FI;FI
160276                     X:=XREG; T:=PNT2; A:=CHAR; CALL PUTCH
160302                     IF A=##' GO FAR RETXY
160305                OD
160311                X:="WORKA"; T:=TREG; CALL GETCH; IF A=##' GO FAR RETXY
160317                ER19; GO FAR ERET
160321          ELSE IF A=##P GO FAR PASWO
160325                IF A=##A GO FAR ACCEW
160330                IF A=##X GO XSTRING
```

```
160333                  IF PNT1=0 THEN 0=:A=:D; AD=:NUMB; GO FAR RETNU; FI
160341                  X:="WORKA"; T:=PNT1-1; CALL GETCH   % NUMERIC PARAMETER
160345                  IF =##B THEN PNT1-1=:PNT1; GO OCTNU; FI  % LAST CH. B, OCTAL
160354                  IF =##D THEN PNT1-1=:PNT1; GO DECNU; FI        % LAST CH. D
160363                  IF DESC2=##B THEN GO OCTNU ELSE GO DECNU ; FI  % DEFAULT MODE
160371          FI
160371
160371   *)FILL
160402
160402   % COPY ALL CHARACTERS UNTIL '
160402
160402   XSTRING: 0=:PNT2
160403          FOR PNT2 TO TREG-1 DO
160410                  X:="WORKA"; T:=PNT2; CALL GETCH; X:=XREG; CALL PUTCH
160415                  IF A=##' GO FAR RETXY
160420          OD
160424          X:="WORKA"; T:=TREG; CALL GETCH; IF A=##' GO FAR RETXY
160432          ER19; GO FAR ERET
160434
160434   *)FILL
160441
160441   % DECIMAL NUMBER EXPECTED
160441
160441   DECNU: 0=:PNT2=:NUMB1=:NUMB2=:NEGFL
160445          T:=0; X:="WORKA"; CALL GETCH
160450          IF A=##- THEN MIN NEGFL; MIN PNT2; FI
160455          FOR PNT2 TO PNT1-1 DO
160462                  X:="WORKA"; T:=PNT2; CALL GETCH
160465                  IF <##0 OR >##9 THEN A:=ER2; GO ERET; FI
160475                  A-##0=:T; AD:=NUMB SHZ 3; A=:L; D=:X
160503                  AD:=NUMB SHZ 1; D+X; A:=A+C+L; D+T; A:=A+C
160512                  AD=:NUMB; OD
160517          GO RETNU
160520
160520   *)FILL
160522
160522
160522   % OCTAL NUMBER EXPECTED
160522
160522   OCTNU: 0=:PNT2=:NUMB1=:NUMB2=:NEGFL
160526          T:=0; X:="WORKA"; CALL GETCH
160531          IF A=##- THEN MIN NEGFL; MIN PNT2; FI
160536          FOR PNT2 TO PNT1-1 DO
160543                  T:=PNT2; CALL GETCH
160545                  IF <##0 OR >##7 THEN A:=ER3; GO ERET; FI
160555                  A-##0=:T; AD:=NUMB SHZ 3; D+T; AD=:NUMB; OD
160567   RETNU: AD:=NUMB; IF T:=NEGFL><0 THEN
160573              D=:T; A=:L; 0=:A=:D; D-T; A:=A+C-1-L; FI
160603          X:=-1; GO LEAVX
160605   RETXY: X:=0; GO LEAVX
160607
160607   % ILLEGAL CHARACTER
160607
160607   ILLC:  X:=XREG; T:=PNT2;   A:=CHAR; CALL PUTCH   % ADD APOSTROPHE AT THE END OF THE STRING.
160613          X:=XREG; T:=PNT2+1; A:=##';  CALL PUTCH
160620          A:=ER1
160621   ERET:  A=:AREG                                   % RETURN ERROR CODE
160622          GO LEAVE
160623
160623   % PASWORD EXPECTED
```

```
160623
160623   PASWO: IF DESC2=##P GO FAR XSTRING                    % FOR PROJECT PASSWORD
160627          0=:PNT2=:NUMB1=:NEGFL
160632          FOR PNT2 TO PNT1-1 DO
                     X:="WORKA"; T:=PNT2; CALL GETCH
                     T:=NUMB1 SHR 3 + A=:NUMB1; OD
160642          GO RETNU
160652
160653
160653   *)FILL
160660
160660   % ACESS WORD EXPECTED
160660
160660   ACCEW: 0=:PNT2=:NUMB1=:NEGFL
160663          FOR PNT2 TO PNT1-1 DO
160670             X:="WORKA"; T:=PNT2; CALL GETCH; A=:D
160674             IF A=##R THEN NUMB1 BONE BITR ELSE
160702                IF A=##W THEN NUMB1 BONE BITW ELSE
160710                   IF A=##C THEN NUMB1 BONE BITC ELSE
160716                      IF DESC2=##L AND D=##D THEN NUMB1 BONE BITD ELSE
160730                      IF DESC2=##O AND D=##X THEN NUMB1 BONE BITX ELSE
160742                         IF D=##A THEN NUMB1 BONE BITA ELSE
160750                         IF DESC2=##L AND D=##N THEN NUMB1 BONE BITN ELSE
160762                         A;=ER1; GO ERET; FI;FI;FI;FI; FI; FI; FI
160764          A=:NUMB1; OD
160771          GO FAR RETNU
160772
160772   % ASK FOR PARAMETER
160772
160772   ASK:    IF BCHFLAG><0 THEN X:="WORKA"; T:=0; ##'; CALL PUTCH; GO FAR PAROK FI
161001   ASKL:   AREG+1; CALL OUTTEXT
161004           IF DESC1=##P THEN -1; *MON 2ECHO
161012             0=:CPNT DO CALL TCI; CALL CWRITE; WHILE A><15 OD 0=:CPNT
161022           ELSE
161023             TTIFIELD.FLAGB BONE 5FFGPAR; CALL GCOM; X.FLAGB BZERO 5FFGPAR=:X.FLAGB
161032           FI
161032           X:="COMST"; DO T:=CPNT; CALL GETCH; A=:CHAR; WHILE A=## ; MIN CPNT OD
161043           GO FAR NPLGS
161044
161044   RBUS
161056
161056
161056   @ELIB
161056
```

```
161056
161056   @DEV 1
161056   @DEV (S-S-J)SIND
161056
161056   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%% S I N D %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
161056
161056   %==========================================================================
161056   % 21.0        M O D E
161056   %
161056
161056   % COMMAND: MODE <INPUT FILE> <OUTPUT FILE>
161056   %
161056   INTEGER 9MOCOLDSTART          % ><0 THEN RUN COLD-START MODE
161057   INTEGER 9IHENTFILE(36)        % NAME OF "COLD-START INPUT-FILE"
161115   INTEGER 9OHENTFILE(36)        % NAME OF "COLD-START OUTPUT-FILE"
161153   INTEGER 9HNTCOMMAND:='ENTER-DIRECTORY  '; * *-1/
161163   INTEGER X9HNTCOMMAND(12)      % PARAMETERS TO THE @ENTER-DIRECTORY COMMAND
161175
161175   INTEGER IPFI:='S INPUT FILE: ',OPFI:='S OUTPUT FILE: '
161215
161215   SUBR MODE,EXHENTMODE
161215
161215   INTEGER MOSTF:='$ MODE STACK OVERFLOW'
161230
161230   DISP -177
161230   INTEGER SVINF                          % SAVED INPUT FILE LDN.
161230   PSID
161230
161230   CSUB1: A=:SVINF
161231          IF A=1 THEN
161234             IF X:=BCHFLAG><1 THEN
161240                A:=TTNO
161241             ELSE
161242                A:=TTIFIELD.RIFIL
161244             FI
161244          FI; EXIT
161245
161245   CSUB2: IF A=1 THEN
161250             IF X:=BCHFLAG><1 THEN
161254                A:=TTNO
161255             ELSE
161256                A:=TTIFIELD.DFOPP.ROFIL
161261             FI
161261          FI; EXIT
161262
161262   MODE:  L=:D; CALL ENTER
161264          O=:BMECHO
161265          IF MSTPN>=MSTEN THEN "MOSTF"; CALL OUTTEXT; GO JABORT FI
161274          IF BCHFLAG><0 THEN
161276             A:=TTIFIELD.DFOPP.ROFIL=:D:=X.DFOPP.RIFIL
161304             AD=:MOSTK(MSTPN); X+2=:MSTPN
161310          FI
161310          "IPFI"; CALL AGPAR; GO NOIPA                    % GET INPUT FILE NAME.
161313          A=:X; A:="TYPS"; T:=1; *MON 2NOPE               % OPEN FILE FOR SEQUENTIAL READ.
161317          GO FILERR
161320          IF A=0 THEN A:=ER72; GO FILERR FI               % "NO END OF FILE FOUND" ??????
161323          CALL CSUB1; GO INOK
161325   NOIPA: IF BCHFLAG=0 THEN A:=TTNO ELSE A:=TTIFIELD.RIFIL FI % NO INPUT FILE SPECIFIED.
161333   INOK:  A=:TTIFIELD.RIFIL; 1=:X.BCHISTS
161337          "OPFI"; CALL AGPAR; GO NOOPA                    % GET OUTPUT FILE NAME.
```

```
161342          A=:X; A:="TYPS"; T:=0; *MON 2NOPE
161346          GO OUTERR; CALL FILSYS(SETPO); GO FILERR          % SET OUTPUT FILE PERMANENTLY OPENED.
161352          CALL CSUB2; GO UTOK
161354   NOOPA: IF BCHFLAG=0 THEN A:=TTNO ELSE A:=TTIFIELD.DFOPP.ROFIL FI
161363   UTOK:  A=:TTIFIELD.DFOPP.ROFIL; 1=:X.BCHOSTS
161370          IF BCHFLAG=0 THEN 2=:BCHFLAG FI
161374          A:=SVINF
161375          CALL FILSYS(SETPO); GO FILERR                     % SET INPUT FILE PERMANENTLY OPEN.
161400          GO LEAVE
161401
161401   OUTERR: T:=TTIFIELD.RIFIL; *MON 2CLOS; JMP *1
161405          GO FILERR
161406   *)FILL
161417
161417   EXHENTMODE: L=:D; CALL ENTER
161421          X:="BAUSER"; CALL FILSYS(ENSYS); GO ALOGOUT; 2=:PASSTYPE=:SPASTYPE
161430          5BCOMM=:TTIFIELD.BSTATE; -1=:INLOGGED
161435          X:="SC100"; T:=2; A:="TYPD"; CALL FILSYS(DOPSCR); GO ERR
161443          A:="TYPS"; T:=1; X:="9IHENTFILE"; *MON 2NOPE; JMP ERR
161450          IF A=0 GO ERR
161451          CALL FAR CSUB1; A=:TTIFIELD.RIFIL; 1=:X.BCHISTS
161456          A:="TYPS"; T:=0; X:="9OHENTFILE"; *MON 2NOPE; JMP ERR
161463          CALL FILSYS(SETPO); GO ERR
161466          CALL FAR CSUB2; A=:TTIFIELD.DFOPP.ROFIL; 1=:X.BCHOSTS
161474          2=:BCHFLAG
161476          SVINF; CALL FILSYS(SETPO); GO ERR
161502          A:=0; CALL MLGRSTART
161504          TTIFIELD.FLAGB BZERO 5MLGIN=:X.FLAGB
161510          GO LEAVE
161511   ERR:   -1; CALL MLGRSTART; GO ALOGOUT
161514
161514   RBUS
161530
161530   %=========================================================================
161530   % 21.1        B A T C H
161530   %
161530   % COMMAND: BATCH
161530   %
161530   INTEGER ARRAY RBPAR:=(V0,V3,V6)
161533
161533   SUBR BATCH
161533
161533   INTEGER BAFU:='$NO BATCH AVAILABLE',BSNIT:='$BATCH SEGMENT NOT INITIALIZED'
161565   INTEGER BANUM:='$BATCH NUMBER ='
161575   INTEGER BANAVA:='$BATCH NOT AVAILABLE',XBPNU:='IO'
161612
161612   DISP -177; INTEGER COUNT,CDFIELD; PSID
161612
161612   CSUBR: MIN BATAB(X); X+1; BATAB(X)=:V6; 0=:V3; X+1; BATAB(X)=:V0
161622          "RBPAR"; *MON 2PRSR
161624          EXIT
161625
161625   BATCH: L=:D; CALL ENTER
161627          IF BATAB(0)-=:COUNT=0 GO UT0
161634          "XBPNU"; CALL KGPAR; GO L1; A=:D
161640          IF A+COUNT>0 GO ERR1
161643          D-1 SH 2; X:=D+1
161646          IF BATAB(X)=0 THEN
161650             CALL CSUBR; IF A><0 GO ERR1
161652             GO UT1
```

```
161653          FI; GO LEAVE
161654   L1:    X:=1
161655          FOR COUNT DO
161655            IF BATAB(X)=0 THEN
161657              CALL CSUBR; IF A=0 GO UT1; X-2
161662            FI; X+4
161663          OD
161665   UT0:   "BAFU"; CALL OUTTEXT; GO LEAVE
161670   UT1:   MIN V3; "RBPAR"; *MON 2PRSR
161673          A:=X+1 SHZ "-2"=:COUNT; X=:V3; A:=BATAB(X); CALL LOGPH; A=:X
161702          X=:CDFIELD:=V3-1; X:=BATAB(X); CALL S5ESCF; X:=CDFIELD
161710          100=:X.MXTIM; 0=:X.USIDX=:X.IDLE=:X.BCHISTS=:X.DFOPP.BCHOSTS
161717          X:=V3-1; BATAB(X)=:V0; T:=A.SEGM SHZ -10; X:="BCHFLAG"; A:=-1
161730          CALL PUT1L; GO SERRS; X:="MSTPN"; A:=0; CALL PUT1L; GO SERRS
161736          "RBPAR"; *MON 2RT
161740          "BANUM"; CALL OUTTEXT; COUNT; CALL DECU; GO LEAVE
161745   SERRS: "BSNIT"; CALL OUTTEXT; COUNT*4+1=:X; 0=:BATAB(X); GO LEAVE
161755   ERR1:  "BANAVA"; CALL OUTTEXT; GO LEAVE
161760
161760   RBUS
161776
161776   INTEGER BANUM:='IDBATCH NUMBER: ',REMCO:='S REMOTE COMPUTER: ',TYREM:='REM'
162023
162023   %=========================================================================
162023   %              C H F T R E M
162023   %
162023   % SUBROUTINE TO CHECK IF A OPEN FILE IS A PERIPHERAL FILE WITH FILE TYPE :REM
162023   %
162023   % ENTRY:       A=T=OPEN FILE NUMBER
162023   %
162023   % EXIT:        ERROR, THE FILE IS CLOSED
162023   %
162023   % EXIT+1:      OK
162023   %
162023   SUBR CHFTREM
162023
162023   DISP -170; INTEGER CFILNO; PSID
162023
162023   INTEGER TXNREM:='$ERROR, THE "REMOTE COMPUTER" FILE MUST BE A PERIPHERAL FILE'
162062   INTEGER TXNR2:='$WITH FILE TYPE :REM'
162075
162075   CHFTREM: L=:D; CALL ENTER
162077          T=:CFILNO; CALL LOGPH; IF A=0 THEN A:=D FI
162103          IF A=0 THEN CALL ERRFATAL FI
162105          IF A.OFFTP NBIT OPBIT GO ERR                    % CHECK IF PERIPHERAL
162111          A:="WORKA"; T:=CFILNO; *MON 41; JMP ERM64       % READ OBJECT ENTRY
162115          X:="WORKA+11"; X.S0/\77577
162120          IF A><#RE GO ERR                                % CHECK FOR FILE TYPE :REM
162123          X.S0/\77577
162125          IF A><#M' GO ERR
162130          GO LEAV2
162131
162131   ERM64: *MON 64
162132          GO ERRF
162133   ERR:   "TXNREM"; CALL OUTTEXT; "TXNR2"; CALL OUTTEXT
162137   ERRF:  T:=CFILNO; *MON 2CLOS; JMP *1
162142          GO LEAVE
162143   RBUS
162155
162155
```

```
162155   %=================================================================
162155   % 21.2        A P P B A T C H   A P R E B
162155   %
162155   % COMMAND: APPEND-BATCH BATCH NO., BATCH INPUT FILE, BATCH OUTPUT FILE
162155   %          APPEND-REMOTE
162155   %
162155   SUBR APPBATCH,APREB,XAPPBATCH
162155
162155   INTEGER NMRQ:='$ BATCH QUEUE FULL'
162167   INTEGER BNOA:='$ BATCH PASSIVE'
162177   INTEGER TEIN:='$INPUT FILE ERROR: '
162211   INTEGER TFOUT:='$OUTPUT FILE ERROR: '
162224   INTEGER TWARN:='$WARNING, FILE DOES NOT EXIST; '
162244
162244   DISP -200
162244   INTEGER CLOGU
162244   INTEGER COUN                  % COUN SERVES AS A COUNTER TO GET THE RIGHT NUMBER OF PARAMETERS: IT DEPENDS
162244                                 % ON THE MIN INSTRUCTION. THIS PRACTICE SHOULD NOT BE EMULATED!
162244   INTEGER QSKIP                 % USED TO SKIP OVER THE " IN THE OUTPUT FILE NAME.
162244   INTEGER CDFELT
162244   INTEGER VOCFREE
162244          INTEGER CLFYLLE=VOCFREE
162244   INTEGER VOFYLLE
162244   INTEGER INFYLLE
162244   INTEGER OUTFYLLE
162244   INTEGER LASTFYLLE
162244   INTEGER CX
162244          INTEGER CHFIRST=CX
162244   INTEGER LRG
162244   INTEGER CCFLG
162244   PSID
162244
162244   APREB: L=:D; CALL ENTER; "REMCO"; CALL GPAR; T:=0; A=:X:="TYREM"; *MON 2NOPE
162254          GO CFILERR; A=:V0=:T; CALL CHFTREM; GO LEAVE; *MON 2CLOS; JMP *1
162263          T:=-1; GO FELL
162265
162265   APPBATCH:L=:D; CALL ENTER; "BANUM"; CALL GPAR; CALL CHBAT
162272          A SHZ 2-3=:X; IF BATAB(X)=0 THEN "BNOA"; CALL OUTTEXT; GO LEAVE FI
162302          T:=-2; X+3; BATAB(X)=:V0=:BATINT                % V0 (=200) IS USED TO HOLD AN ADDRESS
162307   FELL:  1=:V3; 0=:V6; "RBPAR"; *MON 2RESR
162314          V0; CALL XAPPBATCH
162316          "RBPAR"; *MON 2RELE
162320          GO LEAVE
162321   CFILERR: *MON 64
162322          GO LEAVE
162323   *)FILL
162334
162334   XAPPBATCH: L=:D; CALL ENTER
162336          0=:QSKIP; T=:COUN=:CCFLG; A=:CLOGU; CALL LOGPH
162343          A.CFREE=:VOCFREE; X.FYLLE=:VOFYLLE=:INFYLLE; X=:CDFELT
162352          "IPFI"; CALL GPAR                                % GET INPUT FILE NAME.
162354          A=:T=:D; X:=0
162357   TUSNA: *LBYT                                            % TEST USER NAME IS SPECIFIED IN FILE NAME.
162360          IF A >< ##( THEN                                % IF THE FIRST CHARACTER OF THE INPUT FILE NAME
162363            CALL FILSYS(TUSEN); GO NOUSE                  % NAME IS NOT A ( THEN WE MUST PUT IN A USER NA
162366            CALL FILSYS(GMAIN); GO CFILERR
162371            X:="WORKA+5WCBUFSIZE-10"
162372            CALL FILSYS(GUSEN); GO CFILERR                % GET USER NAME.
162375            A:=##(; CALL SIM2OUTB; GO FAR NMR             % *MON 2OUTB; JMP NMR
162400            FOR X:=0 TO 17 DO
```

```
162404                T:="WORKA+5WCBUFSIZE-10"; *LBYT                 % COPY USER NAME TO BATCH INPUT FILE NAME.
162406            WHILE A >< ##'; CALL SIM2OUTB; GO FAR NMR           % *MON 2OUTB; JMP NMR
162413            OD
162415            A:=##); CALL SIM2OUTB; GO FAR NMR                   % *MON 2OUTB; JMP NMR
162420          FI; GO NOUSE; *)FILL
162430  NOUSE:  X:=QSKIP
162431  LOOP:   T:=D; *LBYT
162433          IF A><##' THEN
162436            CALL SIM2OUTB; GO NMR                              % *MON 2OUTB; JMP NMR
162440            X+1; GO LOOP
162442          FI
162442          CALL SIM2OUTB; GO NMR                                % *MON 2OUTB; JMP NMR
162444          MIN COUN; GO FORT
162446
162446          VOFYLLE=:LASTFYLLE
162450          IF CCFLG=-1 THEN LASTFYLLE=:OUTFYLLE FI
162456          OUTFYLLE=:CLFYLLE
162460          0=:QSKIP; CALL CHIOPAR
162462          IF CCFLG><-1 THEN
162466             1=:QSKIP; OUTFYLLE=:INFYLLE; LASTFYLLE=:CLFYLLE
162474             CALL CHIOPAR
162475          FI; GO LEAVE
162476
162476  FORT:   VOFYLLE=:OUTFYLLE
162500          "OPFI"; CALL GPAR                                    % OUTPUT FILE NAME.
162502          A=:T=:D; X:=0; *LBYT
162506          IF A = ##" THEN
162511             CALL SIM2OUTB; GO NMR                             % *MON 2OUTB; JMP NMR
162513             1=:QSKIP; T:=D; X+1; GO TUSNA
162520          FI
162520          GO LOOP
162521
162521  NMR:    "NMRQ"; CALL OUTTEXT; GO LEAVE                       % NO MORE ROOM IN QUEUE.
162524  *)FILL
162531
162531  SIM2OUTB: T:=L=:LRG
162533          IF T:=VOCFREE=0 THEN EXIT FI
162537          T-1=:VOCFREE; X=:CX
162542          T:=CDFELT.BUFST; X:=VOFYLLE; CALL PIOFSBYT
162546          IF A:=X+1=CDFELT.MAX THEN A:=0 FI; A=:VOFYLLE
162555          X:=CX; LRG+1=:P
162561
162561  CHIOPAR: A:=L=:LRG
162563          D:=0; T:=INFYLLE; X:=CDFELT.BUFST:=:T
162570          CALL PIOFLBYT; A=:CHFIRST
162572          DO WHILE CLFYLLE><INFYLLE
162576             X:=CDFELT.BUFST; X:=:T; CALL PIOFLBYT
162602             IF T:=X+1=X:=CDFELT.MAX THEN T:=0 FI; T=:INFYLLE
162611             X:=D; T:="WORKA"; *SBYT
162614             X+1=:D
162616          OD; A:=D=:CLFYLLE
162621          IF CCFLG><-1 AND QSKIP=0 THEN
162627             IF CHFIRST/\177><##" THEN
162634                X:="WORKA"; "TYPS"; T:=1; *MON 2NOPE
162640                GO ERR1; IF A=0 THEN A:=127; GO ERR1 FI
162644                T:=A; *MON 2CLOS; JMP * 1
162647             FI
162647          FI
162647  L1:     X:=0
162650          DO WHILE X><CLFYLLE
```

```
162653              T:="WORKA"; *LBYT
162655              T:=CLOGU; *MON 2OUTB; JMP I (NMR
162660              X+1
162661           OD; LRG=:P
162664   ERR1:   A=:X
162665           IF A=56 THEN "TWARN"; CALL OUTTEXT; "WORKA"; CALL OUTTEXT; GO L1 FI
162675           IF QSKIP=0 THEN "TEIN" ELSE "TFOUT" FI; CALL OUTTEXT
162703           A:=X; *MON 64
162705           GO L1
162706   RBUS
162717
162717   %=============================================================================
162717   % 21.3        L I B A T
162717   %
162717   % COMMAND: LIST-BATCH-PROCESS
162717   %
162717   SUBR LIBAT
162717
162717   INTEGER PSIV:=' PASSIVE', IDLT:=' IDLE,   NO USER',ACTT:=' ACTIVE, '
162742   INTEGER  USE:='USER ',     LON:=' LOGGED ON'
162753
162753   DISP -200; INTEGER COUN,BATE; PSID
162753
162753   LIBAT: L=:D; CALL ENTER; IF BATAB(0)-=:COUN=0 GO UT2; 0=:BATE
162763          FOR COUN DO
162763             BATE+1=:BATE; CALL CRLF; A:=BATE; CALL DECU; A:=BATE SHZ 2-3=:X
162775             IF BATAB(X)=0 THEN "PSIV"; CALL OUTTEXT; GO UT1 FI
163002             X+2; A:=BATAB(X); CALL LOGPH; A=:X
163006             IF X.IDLE=0 THEN
163010                "IDLT"; CALL OUTTEXT                          % BATCH PROCESS IDLE
163012             ELSE
163013                "ACTT"; CALL OUTTEXT
163015                A:=X.USIDX SHZ -10=:T                          % MOD. 25/6/80
163020                A:=X.USIDX/\377                                % FOR 4096 USERS
163022                X:="WORKA"; CALL FILSYS(GUSEN); GO UT1
163026                "USE"; CALL OUTTEXT
163030                #''=:WORKA(10); A:="WORKA"; CALL OUTTEXT
163035             FI
163035             "LON"; CALL OUTTEXT
163037   UT1:    OD
163041   UT2:    CALL CRLF; GO LEAVE
163043
163043   RBUS
163057
163057   %=============================================================================
163057   % 21.4        L I B Q U E   L I R E Q U
163057   %
163057   % COMMAND: LIST-BATCH-QUEUE BATCH QUEUE NO.
163057   %           LIST-REMOTE-QUEUE REMOTE COMPUTER
163057   %
163057   SUBR LIBQUE,LIREQU
163057
163057   DISP -177; INTEGER ANCH,REMFLAG,DDD,ENNO,LRSW; PSID
163057
163057   INTEGER XEMPT:='EMPTY'
163062
163062   LIREQU: L=:D; CALL ENTER; "REMCO"; CALL GPAR; A=:X:="TYREM"; *MON 2NOPE
163071           GO CFILERR; A=:V0=:T; CALL CHFTREM; GO LEAVE; *MON 2CLOS; JMP *1
163100           1=:REMFLAG; GO FELL
163103
```

```
163103   CFILERR: *MON 64
163104          GO LEAVE
163105
163105   LIBQUE:L=:D; CALL ENTER; "BANUM"; CALL GPAR; CALL CHBAT
163112          A SHZ 2=:X; T:=BATAB(X)=:V0; 0=:REMFLAG
163117   FELL:  0=:V3=:V6; *MON 2XIBU; SAA 0
163123          IF A=0 GO TOM; "RBPAR"; *MON 2RESR
163126          MIN V3; "RBPAR"; *MON 2RESR; MON 2XIBU; SAA 0
163133          IF A-=:ANCH=0 GO TOM; 0=:LRSW; 1=:ENNO; CALL DECU; #  ; CALL TCO2
163144          FOR ANCH DO
163144            T:=V0; *MON 2INBT; SAT 0; MON 2OUTB; SAA 0
163151            IF A=##' THEN MIN LRSW
163155              IF ANCH><-1 AND LRSW NBIT "0" OR REMFLAG><0 THEN
163166                CALL CRLF; MIN ENNO; A:=ENNO; CALL DECU; #  ; CALL TCO2
163174              ELSE A:=40; CALL TCO FI
163177            ELSE CALL TCO FI
163201          OD
163203   RETU:  "RBPAR"; *MON 2RELE
163205          0=:V3; "RBPAR"; *MON 2RELE
163210          GO LEAVE
163211   TOM:   "XEMPT"; CALL OUTTEXT; GO RETU
163214
163214   RBUS
163231
163231   %=========================================================================
163231   % 21.5        D E L B E   D E L R E
163231   %
163231   % COMMAND: DELETE-BATCH-QUEUE-ENTRY INPUT PART OF QUEUE, OUTPUT PART OF QUEUE
163231   %          DELETE-REMOTE-QUEUE-ENTRY REMOTE COMPUTER, INPUT PART OF QUEUE ENTRY
163231   %
163231   SUBR DELBE,DELRE
163231
163231   DISP -200
163231          INTEGER LLOGNO,ANENT,CURR,LRWFL,SKPANT,ANCH,CHANR,MXCUR,ANCHS,SIST=LRWFL
163231   PSID
163231   INTEGER INPPA:='S INPUT PART OF QUEUE ENTRY (EXACT MATCH REQUIRED): '
163264   INTEGER OUTPA:='S OUTPUT PART OF QUEUE ENTRY (EXACT MATCH REQUIRED): '
163317   INTEGER NOSEN:='NO SUCH ENTRY'
163326
163326   COPS:  A=:T; A:="WORKA+20"=:D; 0=:ANCH
163332          FOR ANCH TO X:=37 DO
163336            X:=ANCH; *LBYT
163340            T:=:D; X:=MXCUR; *SBYT
163343            MIN MXCUR; T:=:D
163345          WHILE A><X:=##' OD
163354          EXIT
163355          *)FILL
163356   CFILERR: *MON 64
163357          GO LEAVE
163360
163360   DELBE: L=:D; CALL ENTER; "BANUM"; CALL GPAR; CALL CHBAT
163365          A SHZ 2=:X; T:=BATAB(X)=:LLOGNO; -2=:ANENT; 0=:MXCUR
163374          "INPPA"; CALL GPAR; CALL COPS; "OUTPA"; CALL GPAR; CALL COPS; GO FELL
163403
163403   DELRE: L=:D; CALL ENTER; "REMCO"; CALL GPAR; T:=0; A=:X:="TYREM"; *MON 2NOPE
163413          GO CFILERR; A=:LLOGNO=:T; CALL CHFTREM; GO LEAVE; *MON 2CLOS; JMP *1
163422          -1=:ANENT; 0=:MXCUR; "INPPA"; CALL GPAR; CALL COPS
163430
163430   FELL:  0=:LRWFL=:V6; T:=LLOGNO; *MON 2XIBU; SAA 0
163435          IF A=0 GO TOM; "RBPAR"; *MON 2RESR
```

```
163440          MIN LRWFL; "RBPAR"; *MON 2RESR; MON 2XIBU; SAA 0
163445          IF A-=:ANCH=:ANCHS=0 GO TOM
163451          0=:CURR=:SKPANT=:CHANR; ##'=:SIST
163456          FOR ANCH DO
163456            T:=LLOGNO; *MON 2INBT; SAA 0
163461            A=:D; MIN CHANR
163463            IF CURR><-1 THEN
163467              T:="WORKA+20"; X:=CURR; *LBYT
163472              IF A=D THEN
163474                IF SKPANT=0 THEN IF SIST><##' GO IBEG; CHANR-=:SKPANT FI MIN CURR
163506              ELSE
163507   IBEG:        0=:CURR=:SKPANT
163511              FI
163511            FI
163511            T:=LLOGNO; A:=D=:SIST; *MON 2OUTB; 0
163516            IF CURR=MXCUR THEN -1=:CURR FI
163524          OD
163526          IF CURR><-1 THEN
163532   TOM:     "NOSEN"; CALL OUTTEXT; GO RETU
163535          FI
163535          MIN SKPANT; GO COP1; GO SKIP; *)FILL
163553   COP1:  FOR SKPANT DO
163553            T:=LLOGNO; *MON 2INBT; SAT 0; MON 2OUTB; 0
163560            MIN ANCHS
163561          OD
163563   SKIP:  X:=0; FOR ANENT DO
163564            FOR ANCHS DO
163564              T:=LLOGNO; *MON 2INBT; SAA ##'
163567              T:="WORKA"; X+1; *SBYT
163572            WHILE A><##' OD
163577            MIN ANCHS; 0/\0
163601          OD
163603          CALL FILSYS(TUSSY); GO L1; GO OKUS      % CHECK USER
163607   L1:      X-=:ANCH:=1
163612            DO
163612              T:="WORKA"; *LBYT
163614              IF A=##' GO ERTU2
163617              IF A=##) THEN
163622                A:=##'; T:="WORKA"; *SBYT
163625                X=:A:=0; *SBYT
163630                X:="WORKA+1"; CALL FILSYS(GMUSI); GO ERTU  % MOD. FOR 4096 USERS
163634                IF CURUSER><T THEN                         % WRONG USER
163637   ERTU:          T:="WORKA"; X:=0; *LBYT
163642                  A=:X:=##); *SBYT
163645   ERTU2:         X:=1; FOR ANCH DO
163646                    T:="WORKA"; *LBYT
163650                    X+1; T:=LLOGNO; *MON 2OUTB; 0
163654                  OD  A:=25; *MON 2ERMS
163660                FI GO OKUS
163661              FI  X+1
163662            OD
163663   OKUS:  IF ANCHS<0 THEN
163665            FOR ANCHS DO
163665              T:=LLOGNO; *MON 2INBT; SAT 0; MON 2OUTB; 0
163672            OD
163674          FI
163674   RETU:  1=:LRWFL; "RBPAR"; *MON 2RELE
163700          0=:LRWFL; "RBPAR"; *MON 2RELE
163703          GO LEAVE
163704   RBUS
```

```
163710
163710
163710   %=========================================================================
163710   % 21.6        A B J O B   A B B A
163710   %
163710   % COMMAND: ABORT-JOB BATCH NO.
163710   %           ABORT-BATCH BATCH NO.
163710   %
163710   SUBR ABJOB,ABBA
163710   INTEGER BU:='S USER NAME: ',UNLO:='$ USER NOT LOGGED ON',BPA:='$ BATCH PROCESS PASSIVE'
163746   DISP -200; INTEGER BNUM; PSID
163746   ABJOB: L=:D; CALL ENTER; "BANUM"; CALL GPAR; CALL CHBAT; A SHZ 2-3=:X=:BNUM
163757   BPASS: IF BATAB(X)=0 THEN "BPA"; CALL OUTTEXT; GO LEAVE FI
163764          "BU"; CALL GPAR; A=:X; CALL FILSYS(GMUSI); GO CFILERR % MOD. 25/6/80
163772          X:=BNUM+2; A:=BATAB(X); T=:X; CALL LOGPH; T:=X          % FOR 4096 USERS
164000          IF A.USIDX><T THEN "UNLO"; CALL OUTTEXT; GO LEAVE FI
164007          IF CURUSER><T THEN                          % MOD. FOR 4096 USERS
164012             CALL FILSYS(TUSSY); GO CFILERR; FI
164015          X:=BNUM+1; X:=BATAB(X);
164020          CALL S5ESCF; -1=:BATAB(BNUM)
164024          GO LEAVE
164025
164025   % ABORT-BATCH COMMAND
164025   ABBA:  L=:D; CALL ENTER; "BANUM"; CALL GPAR; CALL CHBAT; A SHZ 2-3=:X
164035          IF BATAB(X)=0 GO BPASS; CALL NSTART(BABORT); GO LEAVE
164042
164042   CFILERR: *MON 64
164043          GO LEAVE
164044   RBUS
164056
164056
164056   %=========================================================================
164056   % 21.7        C H B A T
164056   %
164056   % CHECK BATCH NUMBER
164056   SUBR CHBAT
164056   INTEGER NSBA:='$ NO SUCH BATCH'
164066   CHBAT: IF A>BATAB(0) OR <=0 THEN "NSBA"; CALL OUTTEXT; GO LEAVE FI EXIT
164100   RBUS
164101
164101
164101   %=========================================================================
164101   % 21.8        S C E D U L E
164101   %
164101   % COMMAND: SCHEDULE <LOG. DEV. NO.> <INPUT OR OUTPUT>
164101   %
164101   SUBR SCEDULE
164101   INTEGER DVAR:='$ANOTHER DEVICE ALREADY RESERVED'
164122   INTEGER DRINC:='$DEVICES MUST BE SCHEDULED IN ASCENDING ORDER'
164151   INTEGER DVOR:='$ DEVICE NUMBER OUT OF RANGE',GPSC:='IO'
164172   DISP -200; INTEGER LOGU,BINFIL,BOFIL,RWFLS,1LOG,2LOG,RETFL,3LOG,4LOG; PSID
164172
164172
164172   SCEDULE:L=:D; CALL ENTER
164174          TTIFIELD.RIFIL=:BINFIL; X.DFOPP.ROFIL=:BOFIL
164202          0=:1LOG=:2LOG=:3LOG=:4LOG
164206          IF BINFIL><0 AND A/\177700><100 THEN BINFIL; CALL LOGPH; A=:1LOG FI
164217          IF BOFIL><0 AND A/\177700><100 THEN BOFIL; CALL LOGPH; A:=D=:2LOG FI
164231          IF TTNO/\177700><100 THEN TTNO; CALL LOGPH; A=:3LOG:=D=:4LOG FI
164243          X:=RTREF.BRESLINK
```

```
164245          DO WHILE X.RESLINK><RTREF
164251             IF A><1LOG AND A><2LOG AND A><3LOG AND A><4LOG THEN
164265                "DVAR"; CALL OUTTEXT; GO LEAVE
164270             FI; X:=A
164271          OD; 0=:LOGU=:RETFL
164274
164274   LOOP:  A:="GPSC"; CALL KGPAR; GO UT
164277          IF A=0 GO ERRDVOR
164300          IF A<LOGU THEN "DRINC"; CALL OUTTEXT; GO OPCOR FI A=:LOGU; CALL LOGPH
164310          IF D=0 THEN
164312             IF A=0 GO ERRDVOR; 0=:RWFLS
164314          ELSE
164315             1=:RWFLS
164317          FI; A:="RBPAR"; *MON 2RESR; JMP LOOP
164322   UT:    GO LEAVE
164323   ERRDVOR: "DVOR"; CALL OUTTEXT; GO OPCOR
164326   RBUS
164340
164340
164340   %==========================================================================
164340   % 21.9        K G P A R
164340   %
164340   % ROUTINE TO GET PARAMETER IF THERE IS ONE
164340   % IF IT IS, SKIPRETURN, ELSE NO SKIP
164340   % PARAMETERS AS GPAR
164340   SUBR KGPAR
164340   DISP -200; INTEGER OST; PSID
164340   KGPAR: L=:D; CALL ENTER; A=:OST; X:=0
164344   LOOP:  CALL CREAD; IF A=40 THEN X:=1; GO LOOP FI
164352          IF A=15 GO LEAVE; IF X><0 THEN CALL BACK FI
164357          CALL BACK; A:=OST; CALL GPAR; GO LEAV2
164363   RBUS
164367
164367
164367   %==========================================================================
164367   % 21.10       C L E B A
164367   %
164367   % COMMAND: CLEAR-BATCH-QUEUE <BATCH NUMBER>
164367   %
164367   SUBR CLEBA
164367   CLEBA: L=:D; CALL ENTER;
164371          IF PASSTYP><2 THEN A:=25; GO ERERP; FI
164377          "BANUM"; CALL GPAR; CALL CHBAT; A SHZ 2=:X; A:=BATAB(X)=:V0
164406          CALL LOGPH; A=:D; T:=V0; *MON 2ISIZ; SAA 0
164413          IF A=0 GO OKRET
164414          0=:V3=:V6; "RBPAR"; *MON 2RESR
164420          1=:V3; "RBPAR"; *MON 2RESR
164424          D:=:B;  CALL CLBUF; D:=:B; "RBPAR"; *MON 2RELE
164431          0=:V3; "RBPAR"; *MON 2RELE
164434   OKRET: GO LEAVE
164435   ERERP: *MON 64; JMP OKRET
164437   RBUS
164445
164445
```

```
164445
164445   @LIB SSCOM
164445
164445
164445   %=========================================================================
164445   % 22.7        L I V E R S
164445   %
164445   % COMMAND: LIST-VERSION
164445   %
164445   %  SUBROUTINE TO OUTPUT VERSION ID, USER DEFINED TITLE AND OTHER
164445   %  SYSTEM CHARACTERISTICS.
164445
164445   SUBR LIVERS
164445
164445   INTEGER REV33:='REVISION:   '
164453   INTEGER CPU33:='CPU (SYSTEM NUMBER):   '
164467   INTEGER GEN33:='GENERATED:   '
164476
164476   LIVERS: L=:D; CALL ENTER                                  % PRINT VERSION
164500           CALL OUTUSTXT
164501           "REV33"; CALL OUTTEXT                             % REVISION
164503           REVLEV; CALL OCTU; CALL CRLF
164506           "CPU33"; CALL OUTTEXT                             % CPU NUMBER
164510           SYSNO; CALL DECU; CALL CRLF
164513           "GEN33"; CALL OUTTEXT; CALL PDATCL                % GENERATION DATE
164516           GO LEAVE
164517   RBUS
164531
164531
164531   %=========================================================================
164531   % 22.11       N A M S R   I N A M S   R T F E L E M   S N A M S R
164531   %
164531   %    - - - 4.3.7  R T F I L   O P E R A T I O N S
164531
164531   % SUBROUTINE TO SEARCH RTFIL FOR RT-DESCRIPTION ADDRESSES AND
164531   % RT PROGRAM NAMES
164531   %
164531   % IF NAMSR:
164531   %       ENTRY:      TAD=RT-PROGRAM NAME
164531   %       EXIT:       NO SUCH RT-PROGRAM NAME IN RTFIL
164531   %       EXIT AD1:   A=RT-DESCRIPTION ADDRESS
164531   % IF INAMS OR RTFELEM:
164531   %       ENTRY:      A=ADDR. OF RT-DESCRIPTION ADDRESS
164531   %       EXIT:       NO RT-PROGRAM NAME FOUND
164531   %       EXIT AD1:   TAD=RT-PROGRAM NAME
164531   % IF SNAMSR:
164531   %       ENTRY:      TAD=SEGMENT NAME
164531   %       EXIT:       NO SUCH SEGMENT NAME FOUND
164531   %       EXIT AD1:   A=SEGMENT NUMBER
164531
164531   SUBR NAMSR,INAMS,RTFELEM,SNAMSR
164531
164531   DISP -200
164531   INTEGER CCNOC,RTN1; DOUBLE DRTN2,DRTN1=RTN1
164531   INTEGER ARRAY ARTNAM(5)
164531   REAL RTNAME=RTN1
164531   PSID
164531
164531   INTEGER CM151(0); *MON 151
164532   INTEGER CM322(0); *MON 322
```

```
164533   SNAMSR: X:=D; L=:D; CALL ENTER; D:=X; TAD=:RTNAME; CM322; GO FELLS
164542   NAMSR:  X:=D; L=:D; CALL ENTER; D:=X; TAD=:RTNAME; CM151
164550   FELLS: A=:NAROUTSWITCH; 0=:CCNOC
164552          CALL ESCOFF; RTREF.RSEGM=:REESS; 0=:X.RSEGM
164557          "NAMSARR-BXBITMAP"=:B; CALL BMFRTD; "BFIELD"=:B
164564          FOR X:=-10 DO
164565                IF RTN1 SHZ -12><0 THEN
164570                   IF A NBIT 5 THEN A BONE 6 FI; CALL STCH
164574                FI; DRTN1 SH 6; A=:RTN1; DRTN2 SH 6=:DRTN2
164602          OD; ##'; CALL STCH
164605          "ARTNAM"+B=:PANAMSR
164610          A:=NAROUTSWITC=:D:="PANAMSR"; *EXR SD; JMP *+1
164615          TAD=:RTNAME; "NAMSARR-BXBITMAP"; X:=RTREF; A=:B; CALL BMTRTD
164622          "BFIELD"=:B; REESS=:RTREF.RSEGM; CALL ESCON; TAD:=RTNAME
164631          IF A<0 GO LEAVE
164633          GO LEAV2
164634   *)FILL
164647
164647   RTFELEM:
164647   INAMS: L=:D; CALL ENTER
164651          A=:PANAMSR; X=:XXSAV
164653          CALL ESCOFF; RTREF.RSEGM=:REESS; 0=:X.RSEGM
164660          "NAMSARR-BXBITMAP"=:B; CALL BMFRTD; "BFIELD"=:B
164665          A:="PANAMSR"; *MON 152
164667          TAD=:RTNAME
164670          TAD=:RTNAME; "NAMSARR-BXBITMAP"; X:=RTREF; A=:B; CALL BMTRTD
164675          "BFIELD"=:B; REESS=:RTREF.RSEGM; CALL ESCON; TAD:=RTNAME
164704          X:=XXSAV; TAD:=RTNAME
164706          IF D=0 GO LEAVE; GO LEAV2
164711
164711   STCH:  X=:D:=CCNOC; T:="ARTNAM"+B; *SBYT
164716          MIN CCNOC; X:=D; EXIT
164721
164721   RBUS
164733
164733
```

```
164733
164733   %==========================================================================
164733   %
164733   %  23.0      S P E C I A L   R T - P R O G R A M S
164733   %
164733   %==========================================================================
164733
164733   INTEGER NOBLK=?,ADR2B=?,LDRADR=?,KLOLOG=?,KLHDEV=?,KLIOX=?,KLRC1=?
164733   INTEGER BLSTY=?,EIOBU=?,BGSYS=?,SYSST=?,SEGFNO
164734   DOUBLE DYBLST=?
164734   INTEGER YBCSEG                                    % SYSTEM AND USER SEGMENTS
164735   INTEGER ARRAY N5SEG=?,N5LOG=?
164735   INTEGER XBLPAGE                                   % NUMBER OF SECTORS PER PAGE ?
164736   INTEGER X5DSK,DRUMF,FRSEG=?,TMRTERM=?,XSWTP=?
164740   INTEGER MXTS0:=0,MXTS1:=0,MXTS2:=0,MXTS3:=0
164744
164744   @ICR
164744   INTEGER ARRAY ENDVTAB:=(EN000,0,EN200,EN300,EN400,EN500,EN600,EN700,
164754          E1000,E1100,E1200,E1300,E1400,E1500,E1600,E1700,
164764          E2000,E2100,E2200,E2300,E2400);
164771   @CR;
164771
164771   INTEGER IEXI(0); *EXIT
164772   %==========================================================================
164772   % 23.1       O L D S T A R T    ( S T A R T   R T - P R O G R A M )
164772   %
164772   SUBR OLDSTART
164772
164772   DISP -200
164772          INTEGER NPAGE,SFILNO,SADR1,SADR2,SADR3,MAXDN,BATFLG
164772          INTEGER POINTER MXAD,PADRI=SADR1,PADR2=SADR2
164772          DOUBLE POINTER PADR1=SADR1,PADR3=SADR3
164772          INTEGER CCOUNT=NPAGE
164772   PSID
164772   DISP 33; INTEGER N50NO; PSID
164772
164772   INTEGER SINRUN:='$SINTRAN III RUNNING -$$'
165007   INTEGER STX1:='$OCTAL NO. OF PAGES THE SYSTEM WILL USE ON THE SEGMENT FILE(S):'
165047   INTEGER STX2:='$FIRST SYSTEM SEGMENT STARTS ON PAGE (OCT.):'
165076   INTEGER STX3:='$NUMBER OF BACKGROUND PROCESSES (DEC.)      :'
165125   INTEGER STX4:='$EACH BACKGROUND PROCESS NEEDS (OCT. PAGES):'
165154   INTEGER PGSWAP:='PAGES FOR SWAPPING(OCT.):'
165171   INTEGER PARERR:="RTERR"
165172   INTEGER PARRTRFA:="RTRFA"
165173   INTEGER PARSLICE:=("RTSLI","31","1")
165176   INTEGER ABRPAR:=RTREF,OPPAR:="5OPSEG",IMPAR:="5CIMSEG"
165201   INTEGER PARHOLD:=("5","2")
165203   INTEGER PMENTSG:=("5PT3S","3","2","ERRFATAL")
165207   INTEGER PPBPTMP:=("BPTMP","17","2")                % INTERVAL=15 SECS.
165212   INTEGER PAPRSV:=("1","0","DUMMY")
165215   *)FILL
165233
165233   OLDSTART: 1=:EXUNAFLAG
165235          "BFIELD"=:B; "STBEG"=:STPNT; 1=:TDVN
165243          "PARRTRFA"; *MON 2RT
165245          "RESOPAR"; *MON 2RESR
165247          "RESIPAR"; *MON 2RESR
165251
165251   % COMPUTE MASS STORAGE CONSTANTS:
165251          X:=BUFASTART=:SBUF; T:=BUFBANK=:SBANK; X=:D; X:=RTREF.ACTPRI; D:=:X
```

```
165261          CALL DBTRANS; X=:LSBUF
165263          ABLPAGE(0)=:XBLPAGE; IF =20 THEN MIN DRUMF FI
165272          MASSNO(X)=:X5DSK; 0=:SEGFNO
165275
165275   % COMPUTE GENERATION CONSTANTS:
165275          FOR X:=0 TO 24 DO
165301                IF X><1 THEN
165304                  T:=CNVRT(X); ENDVTAB(X)-T-1 SHZ-1
165311                  X=:BATFLG:=T; T:=0; CALL PUT1L; CALL ERRFATAL; X:=BATFLG
165317                FI
165317          OD
165321          X:="BCHTA"; "8NOBA"-X-1 SHZ-2=:X.S0      % BATCH
165327          FOR X:="RTBES" STEP 5RTSIZE TO SEGSTART-5RTSIZE DO
165334                IF X.STADR=0 AND X.ACTPRI=0 THEN 40000=:X.ACTPRI FI
165342                IF X.BRESLINK=0 THEN X=:X.BRESLINK FI
165345          OD; GO BYP1
165350   *)FILL
165405
165405   %      SAVE CONSTANTS FOR "LOAD" PROGRAM, IF NOT A SYSTEM USING VDD/VDP.
165405
165405   BYP1:  IF VDDFLAG = 0 THEN
165407             XBLPAGE*77=:NOBLK                        % NUMBER OF PAGES/BLOCKS TO READ WITH THE BOOSTRAP
165412             X5DSK=:KLOLOG; CALL LOGPH
165415             IF A=0 THEN CALL ERRFATAL FI; A=:X
165420             "EIOBU"=:LDRADR+1400=:ADR2B; X.HDEV=:KLHDEV; A+"IOX+4"=:KLIOX
165430             DDBLST=:DYBLST; DSKTYP=:XSWTP; A:=-4=:KLRC1
165436
165436   %         MOVE BOOTSTRAP TO BLOCK 0:
165436
165436             "RELOA"=:SADR1; A:=0=:D; AD=:DFBLCK=:DTBLCK; X:=LSBUF
165445             CALL RTRANS
165446             DO WHILE SADR1<"LDEND"; PADRI=:X.S0; MIN SADR1; X+1 OD
165457             LDRADR=:SADR1
165461             X=:BATFLG
165462
165462   % INITIALIZE DISC-LAYOUT-DESCRIPTION ELEMENT FOR THE SWAP-DRIVER
165462
165462             MASSNO(0); CALL LOGPH; IF A=0 THEN CALL ERRFATAL FI
165467             IF A."TRNSF"="BDISK" THEN
165474                "HTABL"; X+A; X:=X.S0; "QDLAY"=:B; T:=0
165502                *LDDTX 00; STD 0,B; LDDTX 20; STD 2,B
165506                *LDDTX 40; STD 4,B; LDDTX 60; STD 6,B
165512                DO WHILE B><"QDLAY+10"
165515                   T:=0; *LDATX
165517                   X+1:=:B; T:=5CIMSEG; CALL PUT1L; CALL ERRFATAL
165524                   X+1:=:B
165526                OD; "BFIELD"=:B
165531             FI
165531
165531   %         MOVE MASS STORAGE DRIVER TO DEVICE BUFFER
165531
165531             DO WHILE SADR1<ADR2B
165535                X:=SADR1; T:=0; CALL GET1L; CALL ERRFATAL
165541                A=:BATFLG.S0; MIN SADR1; X+1=:BATFLG
165546             OD
165547             CALL WTRANS;  "5FCOMSEG" SHZ -10=:FRSEG
165553          FI
165553          IF HENTFLAG><0 THEN
165555             A:=1; A=:TDVN=:LGCOLDSTART
165560             T:=BCSEG SHZ -10; X:="TTNO"; CALL PUT1L; CALL ERRFATAL
```

```
165565              X:="DT01R"
165566   @LIB CXCPU
165566              X.TDFLGADDR/\1777+"5UBFPAGE*2000"=:X
165572   @ELIB
165572              A:=X; X:="TTIFIELD"; CALL PUT1L; CALL ERRFATAL
165576              GO FAR NOTFIRST; *)FILL
165655   INTEGER CDFADDR
165656           FI
165656
165656   % MOVE SYSTEM SEGMENT FOR TERMINAL ONE
165656   L1:     BCSEG=:YBCSEG; X:="BACKT"; A:=0; T:=1; CALL MOVSYSEG
165664
165664           IF LGCOLDSTART><1 THEN
165670              CALL LOGPH
165671              IF A><0 AND A.TYPRING BIT 5TERM THEN
165676                 X=:CDFADDR
165677   @LIB CXCPU
165677                 X.TDFLGADDR/\1777+"5UBFPAGE*2000"=:X
165703   @ELIB
165703                 A:=X; T:=BCSEG SHZ -10; X:="TTIFIELD"; CALL PUT1L; CALL ERRFATAL
165711                 LGCOLDSTART; X:="TTNO"; CALL PUT1L; CALL ERRFATAL
165715                 LGCOLDSTART; CALL SSETERROR
165717                 "BAK01"=:TTIFIELD.DBPROG
165722                 LGCOLDSTART=:TDVN
165724              FI
165724           FI
165724
165724   % MOVE ERROR PROGRAM SEGMENT
165724           AD:=DDASA
165725           "5FYER"*XBLPAGE; D+A; A:=DASAX; A:=A+C; AD=:DFBLCK
165733           AD:=DDBLST
165734           "SG14".MADR*XBLPAGE; D+A; A:=SBLST; A:=A+C; AD=:DTBLCK
165743           X.LOGADR SHZ -10-=:NPAGE
165747           FOR X:=NPAGE DO CALL RTRANS; CALL WTRANS OD
165753
165753   % "ACTIVATE" THE TERMINALS IN THE TIMER TABLE
165753           X:="M1TMRTAB"; A:=0; T:=0; CALL PUT1L; CALL ERRFATAL
165760
165760   % INITIALIZE DMAC SEGM., FILE USER SEGM., TADADM SEGMENT AND ACCRT SEGMENT
165760           X:=117776; A:=153134; T:=5ACCS; CALL PUT1L; CALL ERRFATAL
165765           X:="MACD"; IEXI;T:=5MACDSEG; CALL PUT1L; CALL ERRFATAL
165772           IF "5FDSZ"><0 THEN
165774              162003=:"SG22".FLAG   % SET WRITE PERMITTED
165777              X:="BCSTA"; A:=170777; T:=5FIUS; CALL PUT1L; CALL ERRFATAL    % FILE USER REENTRANT SEGMENT
166004              X:="BCSTA"+1; A:=IEXI; T:=5FIUS; CALL PUT1L; CALL ERRFATAL
166012              62003=:"SG22".FLAG    % SET WRITE PROTECT
166015              162003=:"SG26".FLAG   % SET WRITE PERMITTED
166020              X:="BCSTA"; A:=170777; T:=5FIU2; CALL PUT1L; CALL ERRFATAL    % FILE USER REENTRANT SEGMENT 2
166025              X:="BCSTA"+1; A:=IEXI; T:=5FIU2; CALL PUT1L; CALL ERRFATAL
166033              62003=:"SG26".FLAG    % SET WRITE PROTECT
166036           FI; GO LABL1; *)FILL
166102   INTEGER IEXIA(0); *EXIT AD1
166103   LABL1: IF PN500D><0 THEN                % INITIALIZE THE ND-500 MONITOR SEGMENTS
166105              162003=:"SG30".FLAG; X.LOGADR/\377 SH 12+3=:X
166115              T:=30; IEXIA; CALL PUT1L; CALL ERRFATAL; 62003=:"SG30".FLAG
166124              162003=:"SG31".FLAG; X.LOGADR/\377 SH 12+3=:X
166134              T:=31; IEXIA; CALL PUT1L; CALL ERRFATAL; 62003=:"SG31".FLAG
166143              GO LABL2; *)FILL
166154           FI
166154   LABL2: IF X:="BADM"><0 THEN
```

```
166156              IEXI; T:=5BADM; CALL PUT1L; CALL ERRFATAL
166162              X:="BADM"-1; A:=153134; T:=5BADM; CALL PUT1L; CALL ERRFATAL
166170           FI
166170           GO OUT50;*)FILL
166176   OUT50:
166176   %MOVE RT-LOADER
166176           AD:=DDBLST
166177           "SG13".MADR*XBLPAGE; D+A; A:=SBLST; A:=A+C; AD=:DFBLCK
166206           AD:=DDBLST
166207           "SG37".MADR*XBLPAGE; D+A; A:=SBLST; A:=A+C; AD=:DTBLCK
166216           "SG4".LOGADR/\77=:T
166222           "SG13".LOGADR/\77; T-A-=:NPAGE
166230           FOR X:=NPAGE DO; CALL RTRANS; CALL WTRANS; OD
166234           "SG4".LOGADR SHZ -10 -=:NPAGE
166241           AD:=DDBLST
166242           X.MADR*XBLPAGE; D+A; A:=SBLST; A:=A+C; AD=:DTBLCK
166250           FOR X:=NPAGE DO; CALL RTRANS; CALL WTRANS; OD
166254
166254   % MOVE SERVICE & MAIL
166254           AD:=DDBLST
166255           "SG15".MADR*XBLPAGE; D+A; A:=SBLST; A:=A+C; AD=:DFBLCK
166264           "SG23".LOGADR SHZ -10 -=:NPAGE
166271           AD:=DDBLST
166272           X.MADR*XBLPAGE; D+A; A:=SBLST; A:=A+C; AD=:DTBLCK
166300           FOR X:=NPAGE DO; CALL RTRANS; CALL WTRANS; OD
166304
166304   % MOVE NORD-NET SEGMENT
166304           IF "SG16".LOGADR SHZ -10><0 THEN
166310              A-=:NPAGE
166312              AD:=DDBLST
166313              "SG16".MADR*XBLPAGE; D+A; A:=SBLST; A:=A+C; AD=:DFBLCK
166322              AD:=DDBLST
166323              "SG27".MADR*XBLPAGE; D+A; A:=SBLST; A:=A+C; AD=:DTBLCK
166332              FOR X:=NPAGE DO; CALL RTRANS; CALL WTRANS; OD
166336           FI; GO LABL3; *)FILL
166355
166355   % MOVE FILESYSTEM REENTRANT SEGMENT #2
166355   LABL3: "SG12".LOGADR SHZ -10 -=:NPAGE
166362           AD:=DDBLST
166363           "SG12".MADR*XBLPAGE; D+A; A:=SBLST; A:=A+C; AD=:DFBLCK
166372           AD:=DDBLST
166373           "SG25".MADR*XBLPAGE; D+A; A:=SBLST; A:=A+C; AD=:DTBLCK
166402           FOR X:=NPAGE DO; CALL RTRANS; CALL WTRANS; OD
166406
166406   % MOVE FIRST PAGE OF FILESYSTEM REENTRANT SEGMENT #2 TO SPOOLING QUEUE SEGMENTS
166406           X:="SPTAB"
166407           DO WHILE X<=A:="ENDSP"-SPLEN
166413              A:=X.SQUEU; X=:NPAGE
166415              A*5SEGSIZE+"XSEGS"=:X;        % ADDRESS OF SPOOLING QUEUE SEGM ENTRY
166420              AD:=DDBLST
166421              X.MADR*XBLPAGE; D+A; A:=SBLST; A:=A+C; AD=:DTBLCK
166427              AD:=DDBLST
166430              "SG12".MADR*XBLPAGE; D+A; A:=SBLST; A:=A+C;. AD=:DFBLCK
166437              CALL RTRANS; CALL WTRANS; X:=NPAGE+SPLEN
166443           OD
166444
166444   % MOVE SPOOLING SEGMENT
166444           "SG42".LOGADR SHZ -10 -=:NPAGE
166451           AD:=DDBLST
166452           "SG42".MADR*XBLPAGE; D+A; A:=SBLST; A:=A+C; AD=:DFBLCK
```

```
166461          AD:=DDBLST
166462          "SG43".MADR*XBLPAGE; D+A; A:=SBLST; A:=A+C; AD=:DTBLCK
166471          FOR X:=NPAGE DO; CALL RTRANS; CALL WTRANS; OD
166475          GO LABL4; *)FILL
166514
166514   % MOVE FILESYSTEM COMMON SEGMENT
166514   LABL4: AD:=DDASA
166515          "5FYFS"*XBLPAGE; D+A; A:=DASAX; A:=A+C; AD=:DFBLCK
166523          "SG6".LOGADR SHZ -10 -=:NPAGE
166530          AD:=DDBLST
166531          X.MADR*XBLPAGE; D+A; A:=SBLST; A:=A+C; AD=:DTBLCK
166537          FOR X:=NPAGE DO; CALL RTRANS; CALL WTRANS; OD
166543
166543   % MOVE FILESYSTEM REENTRANT SEGMENT #1
166543          "SG24".LOGADR SHZ -10 -=:NPAGE
166550          AD:=DDBLST
166551          X.MADR*XBLPAGE; D+A; A:=SBLST; A:=A+C; AD=:DTBLCK
166557          FOR X:=NPAGE DO; CALL RTRANS; CALL WTRANS; OD
166563          GO LOP2
166564   *)FILL
166576
166576   % MOVE REMOTE FILE TRANSFER SEGMENTS
166576   LOP2:  "5FCOMSEG" SHZ -10=:SADR2; BCSEG SHZ -10-1=:SADR3
166605          FOR SADR2 TO SADR3 DO
166611            SADR2*5SEGSIZE+"XSEGS"=:SADR1; SADR1.LOGADR SHZ -10-=:NPAGE
166622
166622   %  1ST PAGE IS DATA AREA (OVERLAPPING WITH COMMAND SEGMENT)
166622            AD:=DDASA
166623            "5FCRL"-1*XBLPAGE; D+A; A:=DASAX; A:=A+C; AD=:DFBLCK
166632            AD:=DDBLST
166633            SADR1.MADR*XBLPAGE; D+A; A:=SBLST; A:=A+C; AD=:DTBLCK
166642            FOR X:=NPAGE DO CALL RTRANS; CALL WTRANS OD
166646          OD; X:=6; T:=5RTFIL; A:=0; CALL PUT1L; CALL ERRFATAL   % INIT. RTFIL.SEG
166657          0=:CCOUNT
166660          IF "BPTMP"=0 THEN                % IF NOT BACKGROUND PROCESS ALLOCATION SYSTEM
166662             "BACKT"=:SADR1
166664             DO WHILE PADRI><-1
166670                IF A><0 THEN
166671                   X:=A; T:="DBPROG"; CALL XGTDFADDR
166674                   IF A><0 THEN MIN CCOUNT FI
166676                FI
166676                MIN SADR1
166677             OD; A:=CCOUNT
166701          ELSE
166702             A:=1NBPP                      % BACKGROUND PROCESS ALLOCATION SYSTEM
166703          FI; A=:SADR1; BCSEG SHZ -10*5SEGSIZE+"XSEGS"=:X
166711          X.MADR/\37777=:SADR2; X+5SEGSIZE; X.LOGADR SHZ -10+"5SSSZ"=:SADR3
166721          "STX1"; CALL OUTTEXT; SADR3*SADR1+SADR2; CALL OCTU
166727          "STX2"; CALL OUTTEXT; SADR2; CALL OCTU
166733          "STX3"; CALL OUTTEXT; SADR1; CALL DECU
166737          "STX4"; CALL OUTTEXT; SADR3; CALL OCTU; CALL CRLF
166744          A:=1; X:="HENTFLAG"; T:=5CIMSEG; CALL PUT1L; CALL ERRFATAL
166751          GO FAR RSTART; *)FILL
167007
167007   NOTFIRST:
167007
167007   % "ACTIVATE" THE TERMINALS IN THE TIMER TABLE
167007          X:="M1TMRTAB"; A:=0; T:=0; CALL PUT1L; CALL ERRFATAL
167014          0=:SADR2;
167015          IF "BPTMP"=0 THEN                % NOT BACKGROUND PROCESS ALLOCATION SYSTEM
```

```
167017              "BACKT"=:SADR1
167021              DO WHILE PADRI><-1
167025                 IF A><0 THEN
167026                    X:=A; T:="DBPROG"; CALL XGTDFADDR
167031                    IF A><0 THEN
167032                       X:=SADR1; SADR2; T:=0; CALL MOVSYSEG; MIN SADR2
167037                       IF XTMRTERM=-1 GO RSTART
167043                    FI
167043                 FI; MIN SADR1
167044              OD
167045           ELSE
167046              FOR SADR2 DO WHILE SADR2><1NBPP; T:=0; CALL MOVSYSEG
167054              IF XTMRTERM=-1 GO RSTART;OD
167062           FI; GO RSTART; *)FILL
167075   RSTART:
167075   RSTA2: NOPGS-4=:FIXMAX
167100          X:="SG41".BPAGLINK; T:=CORMBANK; *LDATX DPAGP
167104   RSTA3: A=:L SHZ -6=:MBPRVTTABLE          % MEMORY BANK OF PRVTTABLE
167107          A:=L/\77 SH 12=:L                % L=16 LEAST SIGNIFICANT BITS OF PHYSICAL ADDR OF PIT3-SEG.
167113          "SG41".LOGADR/\77 SH 12=:D       % D=LOGICAL START ADDR OF PIT3-SEGMENT
167120          A:="PRVTTABLE"-D+L=:APRVTTABLE   % 16 LEAST SIGNIFICANT BITS IN PHYSICAL ADDR OF PRVTTABLE
167124          A:="SBPRTAB"-D+L=:ASBPRTAB       % 16 LEAST SIGNIFICANT BITS IN PHYSICAL ADDR OF SBPRTAB
167130          A:="EBPRTAB"-D+L=:AEBPRTAB       % 16 LEAST SIGNIFICANT BITS IN PHYSICAL ADDR OF EBPRTAB
167134          A:="EPRVTTABLE"-D+L=:AEPRVTTABL  % 16 LEAST SIGNIFICANT BITS IN PHYSICAL ADDR OF EPRVTTABLE
167140          "PARSLICE"; *MON 2INTV           % START TIMESLICER
167142          "PARSLICE"; *MON 2RT
167144          IF "BPTMP"><0 THEN               % START BACKGROUND PROCESS TIMOUT PROGRAM
167146             % INITIALIZE THE BPRFLG AND TSLSTATUS IN PRVTTABLE
167146             FOR X:=APRVTTABLE STEP 5PRVT TO AEPRVTTABL-5PRVT DO
167153                T:=MBPRVTTABL; *LDDTX TXXSV; STDTX TXSVB
167156             OD
167160             FOR X:=ASBPRTAB STEP BPRTSIZE TO AEBPRTAB-BPRTSIZE DO
167165                T:=MBSPRTAB; *LDDTX TXICB; STDTX TXCBP
167170             OD
167172             IF HENTFLAG=0 AND LGCOLDSTART><1 THEN
167200                CALL LOGPH; T:=MBSPRTAB; X:=ASBPRTAB; *STATX TXCBP
167204                "PAPRSV"; *MON 2PRSR
167206             FI; "PPBPTMP"; *MON 2INTV
167210             "PPBPTMP"; *MON 2RT
167212          FI
167212   @LIB CXCPU
167212          IF "BPTMP"=0 THEN                % NO BACKGROUND PROCESS ALLOCATION SYSTEM
167214             "BACKT"=:SADR1
167216             DO WHILE X:=SADR1.SO><-1
167223                IF X><0 AND X.TYPRING BIT 5TERM THEN
167227                   T:="DBPROG"; CALL XGTDFADDR
167231                   IF A><0 THEN T:=X.TDFPHPAGE; A.WINDOW/\177400\/T=:X.WINDOW FI
167240                FI; MIN SADR1
167241             OD
167242          FI
167242   @ELIB
167242          GO FILL1; *)FILL
167275   INTEGER CNEX,CHENTFLG=?
167276   FILL1: CALL SFILSYS(INITF); HENTFLAG=:CHENTFLG; 1=:HENTFLAG
167304          IF "BUFIN"><0 THEN CALL BUFIN FI
167307          IF BYPINIT><0 THEN                                  % ENTER-DIRECTORY
167311             IF INIBUF(0) SHZ-10><##' THEN
167317                   "INIBUF"; *MON 2CMND                       % MOD. 25/6/80 FOR 4096 USERS
167321                   A:=0
167322                   X:="BAUSER"; CALL FILSYS(ENSYS); GO ERRF; 2=:PASSTYPE
```

```
167330                    "INIBUF"=:CNEX
167332                    DO                % SKIP TO NEXT COMMAND
167332                      FOR X:=0 TO 177 DO T:=CNEX; *LBYT
167340                        WHILE ><##'
167343                      OD; A:=X+2 SHZ -1+CNEX=:CNEX=:T; X:=0; *LBYT
167355                      WHILE ><##'; CNEX; *MON 2CMND
167362                    OD
167363                    IF X:=USTART><0 THEN; *JPL ,X
167366                       FI
167366                    CALL FILSYS(RLUSE); 0/\0; CALL ENTRT        % ENTER USER RT
167372                 FI
167372              FI
167372              CALL SET5NON                                      % SET SEGM. 5 NONDEMAND
167373              "PARHOLD"; *MON 2HOLD                % WAIT 5 SECS FOR UPPDATING THE CLOCK
167375              CALL CRLF; CALL DATCL; CALL LIVERS
167400              "SINRUN"; CALL OUTTEXT
167402              "PGSWAP"; CALL OUTTEXT; NOPGS; CALL OCTU; CALL CRLF
167407              "OPPAR"; *MON 2WSBC
167411              "IMPAR"; *MON 2WSBC
167413              GO RETU; *)FILL
167440
167440      INTEGER CHENTFLAG
167441      INTEGER PM326:=(LGCOLDSTART,CUSN,CPSW,CPSW,CPSW,CPSW,CCCST)
167450      INTEGER CUSN:='SYSTEM',CPSW:='',CCCST
167456      RETU:   CALL INBX21                                  % INIT X21 BUFFERS
167457              IF ERUCL><0 THEN
167461                 *MON 2ERMS                                % PANEL CLOCK INCORRECT (SET BY TIMER)
167462              FI; "PARERR"; *MON 2RT
167464              IF CHENTFLAG=0 AND 9MOCOLDSTART><0 AND BYPINIT><0 THEN
167472                 "9HNTCOMM"; *MON 2CMND
167474                 "RESIPAR"; *MON 2RELE
167476                 "RESOPAR"; *MON 2RELE
167500                 A:="PM326"; *MON 326
167502              FI; 1=:TDVN
167504              GO STSLUTT
167505      ERRF:   *MON 2ERMS
167506              GO RETU
167507      RBUS
167522
167522
167522      %=================================================================
167522      % 23.2        M O V S Y S E G   R T R A N S   W T R A N S
167522      %
167522      %       SUBROUTINE TO COMPUTE THE MASS STORAGE ADDRESS OF THE
167522      %       SYSTEM SEGMENTS AND THE BACKGROUND SEGMENTS
167522      %       THE SYSTEM SEGMENTS WILL BE INITIALIZED AND MOVED TO THE SEGMENT FILES
167522      %
167522      %       ENTRY: X=ADDRESS IN BACKGROUND TABLE
167522      %              A=BACKGROUND-PROGRAM-IN-USE INDEX
167522      %              T=ROUTINE SWITCH
167522      %
167522
167522      INTEGER MOVFLG                          % GLOBAL ROUTINE SWITCH FOR MOVSYSEG
167523
167523      SUBR MOVSYSEG,RTRANS,WTRANS
167523
167523      DISP -200
167523      INTEGER CTABADDR                        % ADDRESS IN BACKGROUND TABLE
167523      INTEGER BCGINDEX,BATFLG,MAXDN
167523      INTEGER TSIZE                           % SIZE IN PAGES OF BACKGROUND AND SYSTEM SEGMENTS FOR A TERMINAL.
```

```
167523   INTEGER SGF1ADDR                          % MASS STORAGE ADDRESS OF SEGMENT
167523   INTEGER CSGFNO,CSGFINDEX
167523   INTEGER SG1TADDR                          % ADDRESS IN THE SEGMENT FILE OF THE SYSTEM SEGMENT FOR TERMINAL 1.
167523   INTEGER CSEG,CADDR,CCOUNT
167523   INTEGER POINTER PCADDR=CADDR
167523   PSID
167523   DISP 10; DOUBLE DS10; PSID
167523   INTEGER CMXT0                             % NUMBER OF BACKGROUND + SYSTEM SEGMENTS WHICH WILL FIT INTO SEG. F
167524   INTEGER CMXT1                             %                                  ""
167525   INTEGER CMXT2                             %                                  ""
167526   INTEGER CMXT3                             %                                  ""
167527
167527
167527   MOVSYSEG: L=:D; CALL ENTER
167531          X=:CTABADDR; A=:BCGINDEX; T=:MOVFLG
167534          YBCSEG SHZ -10*5SEGSIZE+SEGSTART=:SG1TADDR          % SEG.TAB. ENTRY OF FIRST SYS.SEG
167541          A.MADR/\37777=:SGF1ADDR                             % MASS.STOR.ADDR OF FIRST SYS.SEG
167545          X+5SEGSIZE; X.LOGADR SHZ -10+"5SSSZ"=:TSIZE         % NO. OF PAGES IN SYS.SEG + BACKG.SEG.
167552          IF BCGINDEX=0 THEN
167554             0=:CSGFINDEX=:CSGFNO; GO SETSADDR                % BAK01
167557          FI
167557          IF MXTS0><0 THEN                                    % NOT DEFAULT NO. OF SYS.SEG ON EACH SEGMENT FI
167561             A=:CMXT0; MXTS1=:CMXT1; MXTS2=:CMXT2; MXTS3=:CMXT3 % YES, USE THE SPECIFIED NUMBERS
167570          ELSE
167571                IF SIZF(0)-SGF1ADDR<<0 GO FAR ERR             % COMPUTE NO. OF SYS.SEG + BACKG.SEG ON EACH SE
167576                A=:D:=0; T:=TSIZE; *RDIV ST
167602                A=:CMXT0; SIZF(1)=:D:=0; *RDIV ST
167610                A=:CMXT1; SIZF(2)=:D:=0; *RDIV ST
167616                A=:CMXT2; SIZF(3)=:D:=0; *RDIV ST
167624                A=:CMXT3
167625          FI
167625          IF CMXT0+CMXT1+CMXT2+CMXT3 < BCGINDEX+1 GO FAR ERR  % SEGMENT FILES ARE TOO SMALL.
167635          IF CMXT0+CMXT1+CMXT2 < T THEN T:=3                  % DETERMINE WHICH SEGMENT FILE THESE SYSTEM AND
167643          ELSE  IF CMXT0+CMXT1 < T THEN T:=2                  % AND BACKGROUND SEGMENTS WILL LIE IN.
167651          ELSE  IF CMXT0 < T       THEN T:=1
167656          ELSE                          T:=0
167660          A:=0
167661          FI; FI; FI
167661          T=:CSGFNO:=BCGINDEX-A=:CSGFINDEX          % CURRENT SEG.FILE NUMBER AND CURRENT INDEX ON THIS SEG.F
167665          GO SETSADDR
167666   *)FILL
167702
167702   SETSADDR:
167702          "BGSYS"=:CADDR
167704          IF "BPTMP"=0 THEN CTABADDR-"BACKT" ELSE BCGINDX FI
167712          A=:D*"5SDSP"+YBCSEG=:CSEG                           % SEGMENT NUMBER OF CURRENT SYS.SEG + BACKG.SEG
167716          T:=YBCSEG SHZ -10; A:=D*"5NBSG"+T*5SEGSIZE+"XSEGS"=:X
167726          IF X.FLAG/\177766><162022 OR "5SSSZ" SH 10+"SYSST"><X.LOGADR GO FAR XERRS
167741          X+5SEGSIZE; IF X.FLAG/\177766><160002 GO FAR XERRS
167747          IF MOVFLG=1 AND BCGINDEX><0 THEN                    % INITIALIZE AND MOVE SYS.SEG
167755                X-5SEGSIZE; CSGFINDX*TSIZE
167760                IF T:=CSGFNO=0 THEN A+SGF1ADDR FI; T=:SEGFNO
167765                T SH 16; A\/T; A=:X.MADR; X+5SEGSIZE;.A+"5SSSZ"=:X.MADR
167773                X=:D; IF A/\37777+TSIZE-"5SSSZ">SIZF(SEGFNO) GO FAR ERR; X:=D                    % DATA SEGM
170004          FI
170004          X.FLAG BZERO 5INHB=:X.FLAG; X-5SEGSIZE; X.FLAG BZERO 5INHB=:X.FLAG
170013          IF MOVFLG=0 THEN                                    % ONLY RESET CUSER ON THE SYS.SEG.
170015                T:=CSEG SHZ -10; X:="CUSER"; -1
170021                CALL PUT1L; CALL ERRFATAL
170023                GO LEAVE; *)FILL
```

```
170052          FI; CALL UPDSIMAGE
170053          X=:T:=SEGFNO+X; AD:=DBLST(X); X:=:T
170060          X.MADR/\37777*XBLPAGE; D+A; A:=BLST(T); A:=A+C; AD=:DTBLCK
170070          AD:=DDASA
170071          SG1TADDR.LOGADR/\77*XBLPAGE; D+A; A:=DASAX; A:=A+C; AD=:DFBLCK % HARDWARE ADDRESS OF FIRST PAGE ON
170101          CALL RTRANS
170102          0=:BATFL
170103          IF X:=CTABADDR><0 THEN
170105             X.S0; CALL FLOGDV                                  % FLOGDV RESETS BATFL IF BATCH
170107             "TTNO"/\1777+LSBUF=:X
170113             T=:A; AD=:X.DS0                                    % PATCH IN THE CORRECT VALUES OF TTNO, TTIFIELD
170115   @LIB CXCPU
170115             CALL LOGPH; IF A=0 THEN CALL ERRFATAL FI
170120             IF A.TYPRING BIT 5TERM THEN
170124                X.TDFLGADDR/\1777+"5UBFPAGE*2000"=:X:="TTIFIELD"/\1777+LSBUF
170133                A:=;X=:X.S0
170135             FI
170135   @ELIB
170135          FI; GO FILL1; *)FILL
170160   FILL1: "BCSEG"/\1777+LSBUF=:X; CSEG=:X.S0                   % SCRATCH FILE NAME ETC.
170166          "SC100"/\1777+LSBUF=:X; BCGINDEX+1=:D; A:=0; T:=144; *RDIV ST
170200          IF A>0 THEN                 % THREE DIGIT NUMBER
170202             A SHZ 10=:L; A:=0
170205             IF D>=12 THEN T:=12; *RDIV ST
170212                L\/A
170213             FI; L:=:A; A+#00; L:=:A
170216             IF D><0 THEN D SHZ 10; T:=#0'+D
170223             ELSE T:=#0' FI; L=:A; T=:D
170227          ELSE                        % TWO DIGIT NUMBER
170230             T:=12; *RDIV ST
170232             IF A><0 THEN A SHZ 10+#00 ELSE A:=#00 FI
170237             A+D; T:=#' =:D
170242          FI; AD=:X.DS10              % UPDATE FILENAME STRING
170243          IF BATFLG><0 THEN
170245                A+1=:D; "BCHFLAG"/\1777+LSBUF=:X; A:=-1; AD=:X.DS0
170255          FI
170255          CALL WXTRANS                                        % WRITE THIS PAGE TO DISK
170256          CALL RTRANS; CALL WXTRANS                           % READ AND WRITE SECOND PAGE
170260          CALL RTRANS                                         % READ THIRD PAGE
170261          "OFLCK"/\1777+LSBUF=:X; 0=:X.S0                     % PATCH OPEN FILE TABLE LOCK ENTRY
170266          CALL WXTRANS                                        % WRITE THIRD PAGE TO DISK.
170267          CALL RTRANS; CALL WXTRANS                           % READ AND WRITE FOURTH PAGE
170271          CALL RTRANS; CALL WXTRANS                           % READ AND WRITE FIFTH PAGE
170273          CSEG SHZ -10; GO LEAVE
170276   +)FILL
170311
170311   INTEGER TXT1:='$NOT ENOUGH SPACE ON THE SEGMENT FILE(S)$'
170336   ERR:    "TXT1"; CALL OUTTEXT; GO OPCOR
170341   INTEGER TELL,CDFLT; INTEGER POINTER PPLREG,PPPLREG
170345   FLOGDV: A=:CDFLT:=L=;"PPLREG"
170350          0=:BATFLG
170351          X:=0;  CALL FFLOGDV
170353          X:=6;  CALL FFLOGDV
170355          X:=10; CALL FFLOGDV
170357          X:=12; CALL FFLOGDV
170361          X:=14; CALL FFLOGDV
170363          X:=15; CALL FFLOGDV
170365          X:=20; CALL FFLOGDV
170367          GO ERRL
170370   FFLOGDV: A:=L=:"PPPLREG"
```

```
170372          A:=X SH 6=:TELL; X:=CNVRT(X)
170376          T:=0; CALL GET1L; CALL ERRFATAL; A+TELL=:MAXDN; X+1
170404          DO
170404                T:=0; CALL GET1L; CALL ERRFATAL
170407                IF A=CDFLT THEN
170412                   IF TELL SHZ -6=12 THEN TELL=:BATFLG FI
170421                   T:=CDFLT=:D:=TELL; GO PPLREG
170425                FI
170425                IF MAXDN=TELL GO PPPLREG
170431                X+2; MIN TELL
170433          OD
170434   *)FILL
170450   INTEGER ARRAY REGBLK(10)
170460   INTEGER SXRG,REGNA:='PXTADLSB'
170466   INTEGER TERRR:='$TRANSFER ERROR IN READ'
170502   INTEGER TERRW:='$TRANSFER ERROR IN WRITE'
170517   INTEGER TERRL:='$ILL. LOGICAL DEVICE NO.'
170534   INTEGER TERRS:='$ILL. SYSTEM/BACKGROUND SEGMENT'
170554   XERRS: X=:SXRG:="REGBLK";*SRB ALEVB
170557          "TERRS"; GO ERRF
170561   ERRL:  X=:SXRG:="REGBLK";*SRB ALEVB
170564          "TERRL"; GO ERRF
170566   ERRW:  X=:SXRG:="REGBLK"; *SRB ALEVB
170571          "TERRW"; GO ERRF
170573   ERRR:  X=:SXRG:="REGBLK"; *SRB ALEVB
170576          "TERRR"
170577   ERRF:  CALL OUTTEXT;CALL CRLF
170601          SXRG=:REGBLK(1)
170604          FOR X:=0 TO 7 DO
170610                T:="REGNA";*LBYT
170612                CALL TCO;##=;CALL TCO;REGBLK(X);CALL OCTU;CALL CRLF
170620          OD
170622          IF BCGINDEX><0 GO OPCOR; *JMP *
170626
170626   %      SUBROUTINE TO READ A PAGE FROM DISK: THE FIRST BLOCK TO BE TRANSFERRED
170626   %            IS INCREMENTED BY THE NUMBER OF SECTORS PER PAGE.
170626   %
170626   RTRANS:X=:T; MASSUNIT(0) SH 6+60=:XFUNC; X:=T
170635          T:=X5DSK; "RPAR"; *MON 2ABST
170640          IF A<0 GO ERRR
170641          AD:=DFBLCK; A:=:D+XBLPAGE; D:=D+C; A:=:D; AD=:DFBLCK
170647          EXIT
170650   %
170650   %      SUBROUTINE TO WRITE A PAGE TO DISK: THE FIRST BLOCK TO BE TRANSFERRED
170650   %            IS INCREMENTED BY THE NUMBER OF SECTORS PER PAGE.
170650   %
170650   WTRANS: X=:T; MASSUNIT(SEGFNO) SH 6+61=:XFUNC
170656          X:=MASSNO(X); X:=:T; "WPAR"; *MON 2ABST
170662          IF A<0 GO ERRW
170663          AD:=DTBLCK; A:=:D+XBLPAGE; D:=D+C; A:=:D; AD=:DTBLCK
170671          EXIT
170672   INTEGER XXRG,LLRG
170674   WXTRANS: IF MOVFLG><2 GO WTRANS
170700          A:=L=:LLRG; X=:XXRG
170703          0=:CCOUNT
170704          FOR CCOUNT TO 1777 DO
170710                LSBUF+CCOUNT=:X; X.S0; T:=CSEG SHZ -10; X:=CADDR
170717                CALL PUT1L; GO ERRW; MIN CADDR
170722          OD; X:=XXRG; LLRG=:P
170731   *)FILL
```

```
170762
170762   INTEGER LIMIT
170763   UPDSIMAGE: A:=X+11=:LIMIT:=L=:LLRG; X=:XXRG
170771          FOR X TO LIMIT DO
170774                X.S0; T:=5CIMSEG; CALL PUT1L; CALL ERRFATAL
171000          OD; X:=XXRG; LLRG=:P
171005
171005   RBUS
171007
171007
171007   %=========================================================================
171007   % 23.3        I N B R P R O G
171007   %
171007   % COMMAND: INITIALIZE-BACKGROUND-PROGRAMS
171007   %
171007   SUBR INBRPROG
171007   DISP -200; INTEGER CADDR,CINDEX; INTEGER POINTER PCADDR=CADDR; PSID
171007   INTEGER IMPAR:="5CIMSEG"
171010   INBRPROG: L=:D; CALL ENTER
171012          IF XTMRTERM><-1 GO LEAVE
171016          T:=MASSNO(0); 2000; CALL G3BUF; CALL ERRFATAL; A=:T=:SBANK; A:=D=:SBUF
171027          RTREF.ACTPRI=:D; X:=SBUF; CALL DBTRANS; X=:LSBUF
171035          1=:CINDEX
171037          IF "BPTMP"><0 THEN                  % BACKGROUND PROCESS ALLOCATION SYSTEM
171041             FOR CINDEX DO WHILE CINDEX><2NBPP; T:=1; X:=0; CALL MOVSYSEG; OD
171052             "BBCHT"=:CADDR;GO L1             %INITIALIZE BATCH
171055          ELSE
171056             "BACKT"+1=:CADDR
171061   L1:         DO WHILE PCADDR><-1
171065                IF A><0 THEN
171066                   X:=A; T:="DBPROG"; CALL XGTDFADDR
171071                   IF A><0 THEN X:=CADDR; CINDEX; T:=1; CALL MOVSYSEG; MIN CINDEX FI
171077                FI; MIN CADDR
171100             OD
171101          FI; X:="XTMRTERM"; T:=5CIMSEG; A:=0; A=:X.S0; CALL PUT1L; CALL ERRFATAL
171107          X:="TMRTERM"; T:=0; A:=0; CALL PUT1L; CALL ERRFATAL
171114          T:=MASSNO(0); CALL R3BUF; 0/\0; "IMPAR"; *MON 2WSBC
171122          GO LEAVE
171123   RBUS
171151
171151   %=========================================================================
171151   %  24.3       C D V C O M
171151   %
171151   % COMMAND: CLEAR-DEVICE <LOG. DEV. NO.>
171151   %
171151   SUBR CDVCOM
171151   INTEGER PLOGU:='IOLOG. UNIT: ',TILUN:='$ILLEGAL DEVICE'
171170   CDVCOM: L=:D; CALL ENTER
171172          "PLOGU"; CALL GPAR; CALL LOGPH; IF A=0 THEN A:=D FI
171177          IF A=0 GO ERR
171200          IF A.TYPRING NBIT 5CLDV THEN
171204                IF A NBIT M144B GO ERR
171206                X.CLOGDV; CALL LOGPH; IF A=0 THEN A:=D FI
171212                IF A=0 OR A.TYPRING NBIT 5CLDV GO ERR
171217          FI; X=:B; CALL CLEDEV; "BFIELD"=:B; GO OPCOM
171224   ERR:   "TILUN"; CALL OUTTEXT; GO LEAVE
171227   RBUS
171235
171235
171235
```

```
171235   %=========================================================================
171235   % 24.5        O P C O M
171235   %
171235   % COMMAND: OPCOM
171235   %
171235   SUBR COPCOM
171235
171235   INTEGER TXILC:='$NOT POSSIBLE ON N-10 COMPUTERS'
171255   INTEGER TXILT:='$NOT ALLOWED FROM THIS TERMINAL'
171275
171275   COPCOM: L=:D; CALL ENTER
171277           IF CPSTA NBIT 5N100 THEN "TXILC"; CALL OUTTEXT; GO EREX;FI
171305           IF TTNO><1 THEN "TXILT"; CALL OUTTEXT; GO EREX;FI
171314           "150400"; *EXR SA
171316   EREX:   GO LEAVE
171317   RBUS
171323
171323   %=========================================================================
171323   % 24.6        C P U F I   L I V O L   C R E V O
171323   %
171323   %  COMMAND: COPY-USERS-FILES
171323   %          : LIST-VOLUME
171323   %          : CREATE-VOLUME
171323   %
171323   SUBR CPUFI,LIVOL,CREVO
171323   @ICR;
171323   INTEGER NOTPR:='$THIS COMMAND IS NO LONGER PRESENT IN SINTRAN$
171323   THE SUBSYSTEM CALLED BACKUP-SYSTEM SHOULD BE USED INSTEAD';
171410   @CR;
171410
171410   LIVOL:
171410   CREVO:
171410   CPUFI: L=:D; CALL ENTER
171412          "NOTPR"; CALL OUTTEXT; GO LEAVE
171415   RBUS
171416
```

```
171416
171416   %=========================================================================
171416   % 25.3        O S C P R O T
171416   %
171416   %  ROUTINE CALLED FROM SINTRAN-SERVICE-PROGRAM ON OP2SEG
171416   %  TO EXECUTE A PART OF THE COMMAND:
171416   %     SET-COMMAND-PROTECTION <COMMAND> <PROTECTION>
171416   %     REMOVE-SINTRAN-COMMAND <COMMAND>
171416   %
171416
171416   SUBR OSCPROT
171416
171416   INTEGER P6COM:='N COMMAND: '
171424   INTEGER TNSCOMM:='$NO SUCH COMMAND'
171435   INTEGER TNFILCOM:='$NOT ALLOWED ON FILESYSTEM COMMANDS'
171457
171457   DISP -200; INTEGER CCOMADR,XCPROT,LEGAREA; PSID
171457
171457   OSCPROT: L=:D; CALL ENTER
171461            "P6COM"; CALL SGPAR; GO LEAVE
171464            X:=COMTAB; CALL ABLOOK; T=:CCOMADR
171467            IF A><0 THEN "TNSCOM"; CALL OUTTEXT; GO LEAVE FI
171473            IF T.CPROTECT=-1 OR A=-2 THEN "TNFILCOM"; CALL OUTTEXT; GO LEAVE FI
171506            T:=CCOMADR; GO LEAV2
171510   RBUS
171516
171516
171516
171516   %=========================================================================
171516   % 26.1        M A I L
171516   %
171516   %  MAIL SYSTEM ENTRYPOINT (THE MAIL SYSTEM IS LOCATED ON SEGMENT 23)
171516   %
171516
171516   SUBR MAIL
171516   INTEGER PMAILP:=(SMMAIL,377\5OP2SEG)
171520   INTEGER MLSEM:=("5MLSEM",NULL,"1"),ALREADY:='MAIL ALREADY IN USE'
171535
171535   MAIL:  L=:D; CALL ENTER; "MLSEM"; *MON 2RESR
171541          IF A><0 THEN "ALREADY"; CALL OUTTEXT; GO LEAVE; FI
171545          T:="PMAILP"; *MON 2MCAL
171547          GO LEAVE
171550
171550   RBUS
171555
171555
171555   %=========================================================================
171555   %    T R T E R
171555   %    TERMINAL STATUS MONITOR CALL (ON SINTRAN SERVICE AND MAIL SEGMENT)
171555   SUBR TRTER
171555   DISP -200
171555            INTEGER BREG,LREG
171555   PSID
171555   INTEGER PTRSP:=(SMTRS,377\5OP2SEG)
171557   TRTER: X:="BFIELD"; X:=:B; L=:A;
171562          % ENTERS SIN SERV SEG WITH X=BREG AND A=LREG
171562          T:="PTRSP"; *MON 2MCAL
171564   RBUS
171566
171566
```

```
171566
171566   %=========================================================================
171566   % 25.1        S E R V S I N T
171566   %
171566   %  SINTRAN SERVICE PROGRAM ENTRYPOINT
171566   %  THE SERVICE PROGRAM IS LOCATED ON SEGMENT 23
171566
171566   SUBR SERVSINT
171566   INTEGER PSERP:=(SINSERV,377\50P2SEG)
171570   INTEGER RESPAR:=("SEMSERV",NULL,"1"),TINUSE:='$ALREADY IN USE'
171603
171603   SERVSINT: "RESPAR"; *MON 2RESR
171605           IF A><0 THEN "TINUSE"; CALL OUTTEXT; GO OPCOR; FI
171611           T:="PSERP"; *MON 2MCAL
171613           GO EROPCOM; GO OPCOM; GO OPCOR
171616   RBUS
171626
171626
171626
```

```
171626
171626   %==========================================================================
171626   %  27.1        L O A D   P R O G R A M
171626   %
171626
171626   @MAC

)9SCLC
171626  %
171626  % "LOAD" PROGRAM
171626  %
171626
171626  LOLOG=-2
171626  LOUNI=-1
171626  LKONS=177000
171626
171626
171626  % SET CACHE INHIBIT LIMIT AND CLEAR CACHE
171626  RELOA, PIOF; TRA STS; BSKP ONE 140 DA; JMP *+5; SAA 0
171633         TRR 11; SAA 77; JMP *+2; LDA (37400; TRR 12; TRR 10
171641  % MOVE RELOAD PROGRAM TO ADDRESS LKONS-WORD2
171641         LDX ADR3; LDA I ,X ADR1; STA I ,X ADR2; JNC *-2
171645         LDA ADR1; ADD (1400; STA ADR1
171650  % MOVE DRIVER TO THE CORRECT ADDRESS
171650         LDX (-1400; LDA I ,X ADR1; STA I ,X ADR2B; JNC *-2
171654         JMP I *+1; LKONS-WORD2
171656  ADR1,  TRANS+WORD2-RELOA
171657  ADR2,  LKONS
171660  ADR3,  -WORD2
171661
171661  )FILL
171664  TRANS, LDA (LKONS-21; COPY SA DB
171666         LDA ALOAF; JAZ NALOA              % DO NOT ASK FOR "LOAD" DEVICE
171670  ALO1,  LDA (-WORD3; JPL TOUT; STA LOLOG,B          % LOG. DEVICE
171673         LDA (-WORD4; JPL TOUT; STA HDEV,B           % HARDWARE DEVICE NO.
171676         ADD (IOX 4; STA KLIOX
171700         LDA (-WORD5; JPL TOUT; STA LOUNI,B          % UNIT NO.
171703         JMP NALOA
171704  TX0,'
LOG. DEVNO:'
171713  TX1,'
HDW. DEVNO:'
171722  TX2,'
UNIT:'
171726  ALOAF, XALOF                    % ALTERNATIVE/NOT ALTERNATIVE LOAD FLAG
171727  NOBLK, 0
171730  ADR2B, 0
171731  DYBLS, 0;0
171733  LDRAD, 0
171734  XSWTP, 0
171735
171735  NALOA, LDX NOBLK; LDT LOUNI,B; SHT 6
171740         LDA XSWTP; RADD SA DT; LDD DYBLS
171743  NALO4, JPL I LDRAD; JMP NALOA; JMP *+2; JMP NALO3    % CALL DRIVER
171747         STA NALO5
171750  NALOY, MCL PID; MCL PID; MCL PID
171754  KLIOX, IOX 4; BSKP ZRO 20 DA; JMP NALOY; LDA NALO5; JMP NALO4
171761  NALO3, LDA ALOAF; JAZ NALO2
171763         LDX (MASSN; LDT LOLOG,B; JPL CHSWD
171766         LDX (MASSU; LDT LOUNI,B; JPL CHSWD
```

```
171771  NALO2, IDENT PL11; JMP I (SINTR
171773  CHSWD, LDA ,X; COPY SA DD; STT ,X
171776         LDA 1,X; SKP DA UEQ SD; STT 1,X
172001         LDA 2,X; SKP DA UEQ SD; STT 2,X
172004         LDA 3,X; SKP DA UEQ SD; STT 3,X
172007         EXIT
172010  NALO5, 0
172011  )FILL
172021
172021  % INPUT/OUTPUT ROUTINES FOR LOAD-PROGRAM (ONLY TTY1)
172021
172021  TOUT,  COPY SL DD; SAX 0; ADD (LKONS; COPY SA DT
172025  TOUT1, LBYT; AAA -##'; JAF *+2; JMP TINN
172031         JPL CHOUT; JPC TOUT1
172033  CHOUT, IOX 306; BSKP ONE 30 DA; JMP *-2
172036         LBYT; IOX 305; EXIT
172041  TINN,  COPY SD DX; COPY DD
172043  TINN1, IOX 302; BSKP ONE 30 DA; JMP TINN2
172046         LDA (44004; IOX 303
172050  TINN2, IOX 302; BSKP ONE 30 DA; JMP *-2
172053         IOX 300; COPY SA DT; LDA (44004; IOX 303
172057         IOX 306; BSKP ONE 30 DA; JMP *-2; COPY ST DA; IOX 305
172064         BSET ZRO 70 DA; AAA -60; JAP *+2; JMP OUTR
172070         AAA -10; JAN TINN3
172072  OUTR,  COPY SD DA; COPY SX DP
172074  TINN3, AAA 10; SHA 15; SAD ROT 3
172077         LDA (44004; IOX 303; JMP TINN1
172102  )FILL
172104
172104  DFELT=*+060
172104  KLOLO=DFELT+LOLOG
172104  KLHDE=DFELT+HDEV
172104  KLRC1=DFELT+ERRC1
172104  *<DFELT+20
172104  )ZERO
172104  % POSS. FLT MESSAGE COMES NOW
172104  DFELT+21/
172205  WORD2=*-TRANS
172205  UPADR=*+1400
172205  LDEND=*
172205  WORD3=*-TX0
172205  WORD4=*-TX1
172205  WORD5=*-TX2
172205
172205  )KILL NOBLK DYBLS TOUT TOUT1 CHOUT TINN TINN1 TINN2 OUTR TINN3 NALOY
172205  )KILL NALOA NALO4 KLIOX NALO3 NALO2 CHSWD NALO5 KLOLO KLHDE WORD2
172205  )KILL UPADR DFELT WORD3 WORD4 WORD5
172205  )KILL ADR1 ADR2 ADR3 TRANS ALO1 TX0 TX1 TX2 LOLOG LOUNI LKONS
172205  )9RCLC
)9SLPL172205
172205    *OPEND=*
172205    *BCSTA/BCSTA;173777;OPEND    % LOWER LIMIT, UPPER LIMIT AND FIRST FREE LOCATION
110003    *OPEND/
172205    @EOF
172205
172205
```

```
172205
172205   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% S I N E - 2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
172205   *OP2BG/;*<*+3;)ZERO;*+3/
110003
110003   SUBR TOOPCSEG
110003   RBUS
110003
110003   INTEGER RETUAD=?
110003   %==========================================================================
110003   %
110003   %  28.0        S I N T R A N - S E R V I C E - P R O G R A M    A N D
110003   %
110003   %              M A I L   S Y S T E M
110003   %
110003   %==========================================================================
110003   %
110003   %              P T S I N T R A N
110003   %        GIVE A PAGE FROM LAMU AREA TO SINTRAN
110003   %
110003   %        ENTRY:  A = PHYSICAL ND-100 PAGE NUMBER
110003   %
110003   SUBR PTSINTRAN
110003   PTSINTRAN;
110003           A SH 2+CORMSTART=:D; SEGSTART+5SEGSIZE=:X; T:=CORMBANK          %%%%%
110012           MLEV; *MCL PIE                                                  %%%%%
110014           X.BPAGLINK; X:=:D; *STATX DPAGL                                 %%%%%
110017           A:=1; *STATX DPGPR                          %%%%%%%%%%%%%%%%%%%%%%%%%%%%
110021           A:=X=:D.BPAGLINK                            %%%%%%%%%%%%%%%%%%%%%%%%%%%%
110024           IF X.SEGLINK = 0 THEN                       %%%%%   MUST NOT      %%%%%
110026                 A:="BSEGLINK"                         %%%%%   CROSS A       %%%%%
110027                 DO WHILE A.SEGLINK >< -1 OD           %%%%%   PAGE LIMIT    %%%%%
110035                 A:=D=:X.SEGLINK; -1=:D.SEGLINK        %%%%%%%%%%%%%%%%%%%%%%%%%%%%
110042           FI; MLEV; *MST PIE                          %%%%%%%%%%%%%%%%%%%%%%%%%%%%
110044           EXIT                                        %%%%%%%%%%%%%%%%%%%%%%%%%%%%
110045   RBUS
110051
110051
110051   %==========================================================================
110051   %              T S L A N D O R
110051   %
110051   % SUBROUTINE FOR THE INSER-IN-TIME-SLICE/REMOVE-FROM-TIMESLICE COMMANDS
110051   %
110051   SUBR TSLANDOR
110051
110051   TSLANDOR: *IOF
110052           A=:D;=TSLSTATUS(X)/\D\/T=:TSLSTATUS(X)
110057           *ION; EXIT
110061   RBUS
110062
110062   %==========================================================================
110062   %        S T A G P I B   -   S T O G P I B
110062   %
110062   % ENTRY:      A=VALUE TO WRITE INTO B-REGISTER ON LEVEL 11 (DATAFIELD ADDR)
110062   %             T=VALUE TO WRITE INTO T-REGISTER ON LEVEL 11 (DRIVER START ADDR)
110062   %
110062   SUBR STAGPIB,STOGPIB
110062
110062   STAGPIB: *IOF
110063           *IRW 130 DB
110064           A:=T; *IRW   130 DT                                   % TO T-REG LEV. 11D
```

```
110066          "SLV11";*IRW   130 DP                                   % ROUTINE TO START POF-DRIVER
110070          LV11;*MST   PID
110072          *ION
110073          EXIT
110074
110074   STOGPIB: *IOF
110075          *IRW   130 DB                                           % TO B-REG. LEV. 11D
110076          A:=T; *IRW   130 DT                                     % TO T-REG LEV. 11D
110100          "SLV11";*IRW   130 DP                                   % ROUTINE TO START POF-DRIVER
110102          LV11;*MST   PID
110104          *ION
110105          EXIT
110106   RBUS
110110
110110   %=========================================================================
110110   %             X S B P R T A B
110110   %
110110   % PART OF THE BACKGROUND-ALLOCATION-UTILITIES COMMAND
110110   % TO OPERATE ON THE SBPRTAB ON THE PIT3 SEGMENT
110110   %
110110   % ENTRY:      A=ROUTINE SWITCH
110110   %             T=ACTUAL TERMINAL INPUT DATAFIELD
110110   %
110110   % EXIT:       ERROR
110110   %
110110   % EXIT+1:     OK
110110   %
110110   SUBR XSBPRTAB
110110   INTEGER CSWITCH
110111   XSBPRTAB: A=:CSWITCH; *IOF
110113          A=:CSWITCH; T=:D; X:=ASBPRTAB
110116          DO WHILE X<<AEBPRTAB
110121             T:=MBSPRTAB; *LDATX TXCBP
110123             IF A=D THEN                                          % ENTRY FOUND (CBPTERM=ACTUAL TERMINAL)
110125                *LDATX TXBPR
110126                T:=CSWITCH GOSW L0,L1,L2
110133   L3:          A BZERO BPRTMOUT                                 % DISABLE TIMEOUT
110134   FELLS:       T:=MBSPRTAB; *STATX TXBPR
110136                GO SKPR
110137             FI; X+BPRTSIZE
110140          OD
110141          IF CSWITCH=0 THEN                                       % SET-PERMANENT-CONNECTION
110143             X:=ASBPRTAB
110144             DO WHILE X<<AEBPRTAB
110147                T:=MBSPRTAB; *LDATX TXCBP
110151                IF A=0 THEN                                      % FREE ENTRY IS FOUND
110152                   A:=D; *STATX TXCBP; LDATX TXBPR               % SET CBPTERM=ACTUAL TERMINAL
110155                   A BONE BPCFIXED; *STATX TXBPR                 % SET PERMANENT CONNECTION
110157                   GO SKPR
110160                FI; X+BPRTSIZE
110161             OD; *ION; EXIT                                      % ERROR
110164          FI
110164   SKPR:  *ION; EXIT AD1
110166   L0:    A BONE BPCFIXED; GO FELLS                               % SET-PERMANENT-CONNECTION
110170   L1:    A BZERO BPCFIXED; GO FELLS                              % RESET-PERMANENT-CONNECTION
110172   L2:    A BONE BPRTMOUT; GO FELLS                               % ENABLE-TIMOUT
110174   RBUS
110177
110177   %=========================================================================
110177   %             S R T O N
```

```
110177   %
110177   % SUBROUTINE TO CLEAR RTOFF FOR AN RT-PROGRAM
110177   %
110177   % ENTRY:      X=RT-DESCRIPTION ADDR.
110177   %
110177   % EXIT:       OK
110177   %
110177   SUBR SRTON
110177   SRTON: *IOF
110200          X.ACTPRI BZERO 5RTOFF=:X.ACTPRI
110203          *ION; EXIT
110205   RBUS
110205
110205
110205
110205   %=========================================================================
110205   %       S M E N D V T A B
110205   @ICR
110205   INTEGER ARRAY SMENDVTAB:=(EN000,0,EN200,EN300,EN400,EN500,EN600,EN700,
110215          E1000,E1100,E1200,E1300,E1400,E1500,E1600,E1700,
110225          E2000,E2100,E2200,E2300,E2400);
110232   @CR;
110232
110232   INTEGER SMSTROFI:='S OUTPUT FILE: '
110242
110242   %=========================================================================
110242   %
110242   %   ALL THE FOLLOWING ROUTINES ARE DOUPLICATED FROM THE OPCOM SEGMENT (3)
110242   %   AND THE NAMES HAVE BE CHANGED BY ADDING 'SM' (SERVICE/MAIL) IN FRONT
110242   %   OF THE OLD ROUTINE NAMES:
110242   %
110242   %       ENTER, LEAVE, 3LEV2, LEAVX, LEAV3
110242   %       CHTERM
110242   %       YESNO
110242   %       ABLOOK
110242   %       ABL1
110242   %       GCOM
110242   %       EDIT
110242   %       SRCHINT
110242   %       GPAR, AGPAR, SGPAR, SAGPAR
110242   %       CREAD, OREAD, CWRITE, OWRITE, BACKSPACE, SCAB
110242   %       OUTTEXT
110242   %       OCTU, DECU, DTDEC
110242   %       CRLF, TCO, TCI, TCO2, WINB
110242   %
110242   %=========================================================================
110242
110242
110242   %=========================================================================
110242   % 28.1        S M E N T E R   S M L E A V E   S M 2 L E V
110242   %             S M X L E A V   S M 3 L E A V
110242   %
110242   %  SUBROUTINES FOR SUBROUTINE SMENTER AND SMLEAVE
110242
110242   SUBR SMENTER,SMLEAVE,SM2LEAV,SMXLEAV,SM3LEAV
110242   INTEGER STKERR:='STACK ERROR'
110250   DISP 0; DOUBLE ADSV; REAL V0SV,V1SV,V2SV; PSID
110250
110250   SMENTER: TAD=:SAVTAD; X=:INDEX
110252          IF STPNT-"STEND">=0 GO ERR
```

```
110255          X=:A; AD=:STPNT.ADSV
110260          FV0=:X.V0SV; FV3=:X.V1SV; FV6=:X.V2SV
110266          X+STDELTA=:STPNT; SAVTAD; X:=INDEX; EXIT
110273
110273  SM3LEAV: TAD=:SAVTAD; X=:INDEX; STPNT-STDELTA=:STPNT=:X
110301          IF A-"STBEG"<0 GO ERR
110303          MIN X.S1; MIN X.S1; GO LEV
110306  SMXLEAV: TAD=:SAVTAD; T:=X; STPNT-STDELTA=:STPNT=:X
110314          IF A-"STBEG"<0 GO ERR
110316          MIN X.S1; T=:X.S0; GO LEV
110321  SM2LEAV: TAD=:SAVTAD; STPNT-STDELTA=:STPNT=:X
110326          IF A-"STBEG"<0 GO ERR; MIN X.S1; GO LEV
110332  SMLEAVE: TAD=:SAVTAD; STPNT-STDELTA=:STPNT=:X
110337          IF A-"STBEG"<0 GO ERR
110341  LEV:    X.V0SV=:FV0; X.V1SV=:FV3; X.V2SV=:FV6
110347          X.ADSV; A=:X; D=:L; SAVTAD; EXIT
110354  ERR:    "STBEG"=:STPNT; "STKERR"; CALL SMOUTTEXT;
110360          "OPCOR"=:RETUAD; T:="6P3RET"; *MON 2MCAL
110364  RBUS
110373
110373
110373  %==========================================================================
110373  % 28.2        C H T E R M
110373
110373  % SUBROUTINE TO CHECK LOG. NO. OF A TERMINAL
110373  % ENTRY:      A=DATAFIELD
110373  % RETURN:     ERROR
110373  % SKIPRETURN:OK
110373  SUBR SMCHTERM
110373  SMCHTERM: L=:D; CALL SMENTER
110375          IF A><0 THEN X:="BACKTAB"; A=:D
110400                DO WHILE X.S0><-1
110404                 IF A=D GO SM2LEAV; X+1    %OK
110407                OD; D=:A
110411          FI; GO SMLEAVE       %NO TERMINAL
110412  RBUS
110416  %==========================================================================
110416  % 28.3        Y E S N O
110416  %
110416  % SUBROUTINE TO CHECK FOR YES AND NO
110416  % A POINTS TO NAME ON START; RETURN WITH A=0:NO,A=1:YES,A=-1:ERROR
110416  SUBR SMYESNO
110416  INTEGER YNTAB:=(YES,0,1,0,    NO,0,0,0,    -1)
110427  INTEGER YES:='YES',NO:='NO'
110433  SMYESNO: L=:D; CALL SMENTER; X:="YNTAB"; CALL SMABLOOK
110437          IF ><0 THEN -1 ELSE T.CMAND FI; GO SMLEAVE
110445  RBUS
110451
110451  %==========================================================================
110451  % 28.4        S M A B L O O K
110451
110451  %SUBROUTINE FOR ABBREVIATION LOOKUP, ENTRY: X=TABLE TOP
110451  %RETURN IN A-REG:0=OK,-1=NOT FOUND,-2=AMBIGUOUS; T=TABLE ELEMENT ADDRESS
110451  %D=0 IF EXACT MATCH
110451
110451  SUBR SMABLOOK
110451  DISP -200; INTEGER RESULT,FNDADR,ACPNT; PSID
110451  SMABLOOK: L=:D; CALL SMENTER; -1=:RESULT; CPNT=:ACPNT
110457          DO IF X.CNAME=-1 THEN RESULT; X:=FNDADR; GO OUT FI
110466                T:=ACPNT=:CPNT; CALL SM1ABL; IF =0 THEN D:=0; GO OUTD FI
```

```
110474                  IF =1 THEN       % SUBSET
110477                    IF RESULT=0 THEN -2=:RESULT
110503                    ELSE IF =-1 THEN 0=:RESULT; X=:FNDADR
110511                    FI FI
110511                  FI
110511                  X+4
110512            OD
110513   OUT:     D:=-1                    % NOT EXACT MATCH
110514   OUTD:    X=:T; GO SMLEAVE
110516   RBUS
110521
110521   %==========================================================================
110521   % 28.5        S M 1 A B L
110521   %
110521   %ABBREVIATION CHECK ROUTINE, COMPARING THE COMMAND STING TO A STRING
110521   %ENTER WITH A=STRING; EXIT WITH A: 0=OK, 1=SUBSET, 2=NO MATCH
110521
110521   SUBR SM1ABL
110521   DISP -200; INTEGER PCPNT, CHAR,OCHAR,SUBSFLAG; PSID
110521   %AUXILIARY SUBROUTINE RETURNS A=1 IF LETTER OR DIGIT, ELSE 0
110521   ALPHANUM:IF >=##A AND <=##Z GO ALP; IF =##* GO ALP
110532           IF >=##0 AND <=##9 THEN
110540   ALP:    1 ELSE A:=0 FI; EXIT
110544
110544   SM1ABL:  L=:D; CALL SMENTER; A=:OSTRING; 0=:OPNT=:SUBSFLAG
110551           DO     CPNT=:PCPNT      %FOR EACH PART
110553                  DO               %FOR EACH CHARACTER
110553                    X:=0; CALL SMCREAD; A=:CHAR; CALL ALPHANUM; X+A; X+A
110553                    CALL SMOREAD; A=:OCHAR; CALL ALPHANUM; X+A
110561                    X GOSW NCNO,NCYO,YCNO,YCYO
110565
110572   YCYO:            IF CHAR><OCHAR AND ><##* GO YCNO2
110601           OD
110602   YCNO:          GO YCNO2      % NOT FOUND
110603   NCYO:          DO CALL SMOREAD; A=:OCHAR; CALL ALPHANUM WHILE ><0; OD
110610                  MIN SUBSFLAG
110611   NCNO:          X:=0; IF CHAR=##- THEN X+2 FI; IF OCHAR=T THEN X+1 FI
110623                  X GOSW NCNO2,NCYO2,YCNO2,YCYO2
110630   YCYO2: OD
110631   NCYO2: X:=1 GO OUT
110633   NCNO2: IF SUBSFLAG><0 THEN X:=1 FI; GO OUT
110637   YCNO2: X:=2; CALL SMCREAD; IF =##- GO YCNO2; CALL ALPHANUM; IF ><0 GO YCNO2
110646   OUT:   CALL SMBACKSPACE; X=:A; GO SMLEAVE
110651   RBUS
110656
110656   %==========================================================================
110656   % 28.6        S M G C O M
110656
110656   %SUBROUTINE TO PUT TT LINE INTO COMMAND BUFFER
110656   SUBR SMGCOM
110656   DISP -200; INTEGER CHAR; PSID
110656   SMGCOM:  L=:D; CALL SMENTER
110660           IF BCHFLAG=0 THEN CALL SMEDIT
110663           ELSE  T:="COMSTRING"=:CSTRING; 0=:CPNT
110667                 IF ><2 THEN TTIFIELD.BCHNUM SHZ 2-3; FI
110676                 IF BATAB(A)<0 THEN "JAB2"=:RETUAD; GO OUT; FI
110704                 DO CALL SMTCI; WHILE A=15 OD
110711                 IF A><##@ THEN "BILCMND"=:RETUAD; GO OUT; FI
110717                 DO CALL SMTCI IF A=11 THEN A:=## ; FI CALL SMCWRITE WHILE ><15 OD
110731           FI; 0=:CPNT; GO SMLEAVE
110733   OUT:    T:="6P3RET"; *MON 2MCAL
```

```
110735   RBUS
110747
110747   %==========================================================================
110747   % 28.7        S M E D I T
110747
110747   % SUBROUTINE TO DO QED-COMPATIBLE LINE-EDITING
110747   SUBR SMEDIT
110747   INTEGER ARRAY TABTA:=(10,16,36,50,62,74,106,120,0,0,0)  %TAB TABLE
110762   @ICR
110762   INTEGER ARRAY CCH:=(3NEXTI,CTA,CTS,CTC,CTD,CTE,CTF,BELL,CTH,CTI,3NEXTI,CTQ,
110776   CTL,CTM,BELL,CTO,CTP,CTQ,CTR,CTS,CTT,CTU,CTV,CTW,CTX,CTY,CTZ,BELL,BELL,BELL);@CR;
111020   DISP -200; INTEGER SEMOD,TCHAR; INTEGER POINTER NEXTI; PSID
111020   SMEDIT:  L=:D; CALL SMENTER; 0=:SEMOD; "3NEXTI"=:"NEXTI"
111025            "WORKA"=:OSTRING; "COMSTRING"=:CSTRING; 1; *MON 2BRKM; MON 2ECHO
111034   NEDIT:  0=:CPNT=:OPNT; DO CALL SMCREAD; CALL SMOWRITE WHILE ><15 OD
111044            0=:CPNT=:OPNT; GO NEXTI
111047   BELL:   7; CALL SMTCO
111051            DO
111051   3NEXTI:        CALL SMTCI; A=:TCHAR
111053                  IF =177 THEN 1 FI                        %DEL AS CTA
111057                  IF <36 AND X:=CCH(A)><0 THEN X=:P FI %CTRL CHAR.
111066   STORE:         CALL SMCWRITE; IF SEMOD=0 THEN CALL GETSOLD FI
111072            OD
111073   OUT:    GO SMLEAVE   % RETURN, EDITING FINISHED
111074   *)FILL
111110
111110   % PROCESSING CONTROL CHARACTERS:
111110   CTA:    IF CPNT=0 GO BELL; CALL SMBACKSP;
111113           CALL VDUSTTY; IF ><0 THEN CALL VDUSBS
111116           ELSE ##^; CALL SMTCO FI; GO NEXTI
111122   CTC:    CALL GETSOLD; IF =15 GO BELL; CALL SMCWRITE; CALL SMTCO; GO NEXTI
111131   CTD:    DO CALL SMOREAD; CALL SMCWRITE WHILE ><15; CALL SMTCO OD; CALL SMCRLF; GO OUT
111142   CTE:    IF SEMOD=0 THEN 1=:SEMOD; ##< ELSE 0=:SEMOD; ##> FI; CALL SMTCO; GO NEXTI
111154   CTF:    DO CALL SMOREAD; CALL SMCWRITE WHILE ><15 OD; GO OUT
111163   CTH:    X:=OPNT; T:=OSTRING; *LBYT
111166           IF =15 GO BELL
111171           DO CALL GETSOLD WHILE><15; CALL SMCWRITE; CALL SMTCO OD; GO NEXTI
111201   CTI:    FOR X:=0 TO 12 DO IF TABTA(X)=0 GO BELL WHILE A-1-CPNT<=0 OD
111215           FOR X:=A- DO CALL GETSOLD; 40; CALL SMCWRITE; CALL SMTCO; OD; GO NEXTI
111225   CTL:    14; CALL SMCWRITE; CALL SMCRLF
111230   CTM:    15; CALL SMCWRITE; GO OUT
111233   CTO:    CALL SMTCI; A=:TCHAR; CALL SRCSHOLD
111236           DO CALL SMOREAD WHILE ><TCHAR; CALL SMCWRITE; CALL SMTCO OD
111245           OPNT-1=:OPNT; GO NEXTI; *)FILL
111264   CTP:    CALL SMTCI; A=:TCHAR; CALL SRCSHOLD
111267           DO CALL SMOREAD WHILE ><TCHAR; ##%; CALL SMTCO OD;OPNT-1=:OPNT;GO NEXTI
111302   CTQ:    0=:OPNT; IF CPNT=0 GO FAR BELL;
111306           CALL VDUSTTY; IF ><0 THEN
111310                 DO CALL VDUSBS; CPNT-1=:CPNT; WHILE ><0 OD
111316           ELSE 0=:CPNT; ##_; CALL SMTCO; CALL SMCRLF FI
111323           GO  NEXTI
111324   CTR:    12; CALL SMTCO
111326   CTRO:   FOR X:=OPNT TO 177 DO T:=OSTRING; *LBYT
111334                 WHILE ><15; CALL SMTCO
111340           OD; CALL SMCRLF
111343           FOR X:=0 TO CPNT-1 DO T:=CSTRING; *LBYT
111352                 CALL SMTCO; OD; GO NEXTI
111356   CTS:    CALL GETSOLD; IF =15 GO FAR BELL; ##%; CALL SMTCO; GO NEXTI
111365   CTT:    CALL SMCRLF; FOR X:=0 TO CPNT-1 DO 40; CALL SMTCO OD; GO CTRO
111400   CTU:    FOR X:=0 TO 12 DO IF TABTA(X)=0 GO FAR BELL WHILE A-1-CPNT<=0 OD
```

```
111415          FOR X:=A- DO CALL GETSOLD; IF =15 GO FAR BELL; CALL SMCWRITE;CALL SMTCO OD
111426          GO NEXTI; *)FILL
111442  CTV:    CALL SMTCI; IF A=:TCHAR>=40 THEN CALL SMTCO
111450          ELSE ##&; CALL SMTCO; TCHAR+100; CALL SMTCO;TCHAR
111457          FI; GO FAR STORE
111460  CTW:    DO IF CPNT=0 GO FAR BELL; CALL SMBACKSP; CALL SMCREAD WHILE =40
111470                CALL SMBACKSP; CALL VDUSTTY; IF ><0 THEN CALL VDUSBS FI; OD
111475          DO WHILE CPNT><0; CALL SMBACKSP; CALL SMCREAD WHILE ><40;
111504                CALL SMBACKSP; CALL VDUSTTY; IF ><0 THEN CALL VDUSBS FI; OD
111511          CALL VDUSTTY; IF =0 THEN ##\; CALL SMTCO FI; GO NEXTI
111516  CTX:    CALL SMTCI; A=:TCHAR; CALL SRCSHOLD
111521          DO CALL SMOREAD; A=:X; ##%; CALL SMTCO WHILE X><TCHAR OD; GO NEXTI
111532  CTY:    DO CALL GETSOLD; CALL SMCWRITE WHILE ><15;.OD; CALL SMCRLF; D=:SEMOD
111542          GO FAR NEDIT
111543  CTZ:    CALL SMTCI; A=:TCHAR; CALL SRCSHOLD
111546          DO CALL SMOREAD; A=:X; CALL SMCWRITE; CALL  SMTCO WHILE X><TCHAR OD
111556          GO NEXTI
111557  *)FILL
111575
111575  % SUBROUTINE TO TEST FOR VDU TERMINAL WITH BACKSPACE
111575  VDUSTTY:IF TTIFIELD.CTTYP BIT 15 AND A BIT 16 THEN 1 %VDU WITH BS
111604          ELSE "0" FI; EXIT                                %HARD COPY
111607
111607  % SUBROUTINE TO OUTPUT BS SPACE BS TO THE VDU
111607  VDUSBS: L=:X; 10; CALL SMTCO; 40; CALL SMTCO; 10; CALL SMTCO; X=:P
111617
111617  % SUBROUTINE TO GET A BYTE FROM OLDSTRING, AND BACKSPACE IF CR
111617  % RETURN: A=CHARACTER
111617  GETSOLD: L=:D; CALL SMENTER; CALL SMOREAD
111622          IF =15 THEN T:=OPNT-1=:OPNT FI; GO SMLEAVE
111631
111631  % SUBROUTINE TO SEARCH IN OLDSTRING FOR A CHARACTER
111631  % A=CHARACTER; RETURN ONLY IF CHARACTER IS FOUND
111631  SRCSHOLD: FOR X=:D:=OPNT TO 177 DO T:=OSTRING; *LBYT
111640                WHILE ><TCHAR; IF =15 GO FAR BELL
111646          OD; D=:X; EXIT
111652  RBUS
111657
111657
111657  %==========================================================================
111657  % 28.8          S R C H I N T
111657
111657  % SUBROUTINE TO SEARCH FOR LEGAL INTERVAL
111657  %CALL+1 POINTS TO LIMIT TABLE, TERMINATED BY -1; A=VALUE
111657  %EXIT: A=INTERVAL NUMBER
111657
111657  SUBR SMSRCHINT
111657  DISP -200; INTEGER ARRAY POINTER ARR; PSID
111657  SMSRCHINT: L=:D; CALL SMENTER; X:=D.SO=:"ARR"; X:=-1; A=:T
111666          DO     X+1; IF ARR(X)<0 GO OUT
111671                 AD SHZ -10; D SHZ -10
111673                 IF T>=A AND T<=D GO OUT
111677          OD
111700  OUT:    X=:A; GO SM2LEV
111702  RBUS
111704
111704  %==========================================================================
111704  %              S M T I M U S E D
111704  %
111704  SUBR SMTMTUSED
```

```
111704
111704   INTEGER TIUIS:='$ TIME USED IS',OUTO:=' OUT OF'
111720   DISP -200; INTEGER 1TIMX,2TIMX,PRG,SEG; PSID
111720
111720   SMTMTUSED :L=:D; CALL SMENTER
111722          A=:PRG; T=:SEG
111724          X:="1TIMON"; CALL GET1L; CALL ERRFATAL; A=:1TIMX; T:=SEG
111731          X:="2TIMON"; CALL GET1L; CALL ERRFATAL; A=:2TIMX
111735   TIMCOM: "TIUIS"; CALL SMOUTTEXT; *MON 2TIME
111740          A:=:D-2TIMX; *RDCR ADC DD
111743          A:=:D-1TIMX; AD=:TIOOF; AD:=PRG.DTINT; CALL SMTIMOUT
111751          "OUTO"; CALL SMOUTTEXT; AD:=TIOOF; CALL SMTIMOUT
111755          GO SMLEAVE
111756   RBUS
111770
111770
111770   %==========================================================================
111770   % 18.23       S M T I M O U T
111770   %
111770   %SUBROUTINE TO WRITE MINUTES AND SECONDS; AD=TIME IN BASIC UNITS
111770   SUBR SMTIMOUT
111770   INTEGER XMIN:=' MINS',XSEC:=' SECS',XBSEC:=' BASIC TIME UNIT'
112007   DISP -200; INTEGER REM; PSID
112007   SMTIMOUT: D=:T:=L; CALL SMENTER
112012          T=:D
112013          IF A=0 AND D<<62 THEN
112017             A:=D=:X; CALL SMDTDEC; "XBSEC"; CALL SMOUTTEXT
112024             IF X><1 THEN ##S; CALL SMTCO FI
112031          ELSE
112032             T:=9TIM2; *RDIV ST
112034             T:=D=:REM; IF A><0 THEN CALL SMDTDEC; "XMIN"; CALL SMOUTTEXT FI
112042             REM=:D:=0; T:=9TIM1; *RDIV ST
112047             CALL SMDTDEC; "XSEC"; CALL SMOUTTEXT
112052          FI; GO SMLEAVE
112053   RBUS
112065
112065
112065   %==========================================================================
112065   % 28.9        S M G P A R   S M A G P A R   S M S G P A R   S A G P A R
112065   %
112065   %SUBROUTINE TO GET PARAMETER
112065   % IF SMGPAR:
112065   %       ENTRY: A=DESCRIPTOR STRING
112065   %       EXIT:  A=INTEGER,FILE STRING,RT-PROG OR NAME STRING
112065   %              OR AD = DOUBLE INTEGER
112065   %
112065   % IF SMAGPAR:
112065   %       ENTRY: A=DESCRIPTOR STRING
112065   %       EXIT:  NO PARAMETER FOUND (USE DEFAULT VALUE)
112065   %       EXIT AD1: A=INTEGER,FILE STRING,RT-PROG OR NAME STRING
112065   %                 OR AD = DOUBLE INTEGER
112065   %
112065   % IF SMSGPAR
112065   %       ENTRY:    A=DESCRIPTOR STRING
112065   %       EXIT:     ILLEGAL PARAMETER
112065   %       EXIT AD1: A=PARAMETER
112065   %
112065   % IF SAGPAR
112065   %       ENTRY:    A=DESCRIPTOR STRING
112065   %       EXIT:     ILLEGAL PARAMETER
```

```
112065   %       EXIT AD1: NO PARAMETER FOUND (USE DEFAULT VALUE)
112065   %       EXIT AD1+1: A=PARAMETER
112065   % SAGPAR: SAME AS SMAGPAR BUT CALLED FROM SERVICE PROGRAM ONLY
112065   %
112065   %
112065   %
112065   %
112065
112065   SUBR SMGPAR,SMAGPAR,SMSGPAR,SAGPAR,SSSLEA
112065   INTEGER DELIM:=(#  ,#,,,15\15,-1),DESCR:=(#II,#SS,#RR,#NN,#GG,#DD,-1)
112100   INTEGER ALPN:=(#AZ,#09,-1),XSALPN:=(#AZ,#09,#--,#__,-1)
112110   DISP -200
112110          INTEGER OVAL,DVAL,MINFLAG,CHAR,TYPCHAR,RTN1,RTN2,RTN3,NIFLAG,AGPFLG
112110          INTEGER SERVFLG
112110          DOUBLE DRTN1=RTN1,DRTN2=RTN2; REAL RTNAME=RTN1
112110          INTEGER PCPNT=DVAL,TYP2CHAR=MINFLAG,NNCHAR=NIFLAG
112110          INTEGER O1VAL=OVAL,O2VAL=DVAL
112110          INTEGER D1VAL=RTN1,D2VAL=RTN2
112110          DOUBLE DOVAL =O1VAL
112110          DOUBLE DDVAL =D1VAL
112110   PSID
112110
112110   SAGPAR: L=:D; CALL SMENTER; T:=1=:SERVFLG; GO AGPF
112115   SMSGPAR: L=:D; CALL SMENTER; T:=1=:SERVFLG; GO GPF
112122
112122   SMAGPAR: L=:D; CALL SMENTER; O=:SERVFLG
112125   AGPF:   T:=1=:AGPFLG; GO FELLS
112130   SMGPAR:L=:D; CALL SMENTER; O=:SERVFLG
112133   GPF:    O=:AGPFLG
112134   FELLS: A=:OSTRING; O=:OPNT=:NIFLAG
112137          CALL SMCREAD; CALL SMSRCHINT(DELIM); GOSW DEL,DEL,CRET,ALNUM
112147   ALNUM: O=:CPNT; GO DEL
112151   CRET:  GO CCRET
112152          IF BCHFLAG><0 THEN O=:CPNT; T:=CSTRING; X:=0; A:=15; *SBYT
112161          ELSE
112162   CCRET:     OSTRING=:X+1; CALL SMOUTTEXT; CALL SMGCOM; X=:OSTRING; O=:OPNT
112171          FI
112171   DEL:   CALL SMSCAB
112172          IF T:=AGPFLG><0 AND A=15 OR A=##, THEN
112203                IF SERVFLG=0 GO SMLEAVE; GO SM2LEAV
112207          FI
112207          CALL SMOREAD; CALL SMSRCHINT(DESCR)
112212          GOSW INT,FAR SYMFIL,RTPROG,FAR NAME,FAR GSGNAM,FAR DBINT,FAR ERR; *)FILL
112240   INT:   CALL SMOREAD
112241   INTRT: A=:TYPCHAR; O=:OVAL=:DVAL=:NIFLAG
112245          CALL SMCREAD; IF =##- THEN 1 ELSE CALL SMBACKSP;"O" FI; A=:MINFLAG
112256          DO     CALL SMCREAD; A=:CHAR; A-60; IF <0 OR >11 GO OUT
112265                 IF >7 THEN MIN NIFLAG FI
112271                 A=:T:=10*OVAL+T=:OVAL:=12*DVAL+T=:DVAL
112302          OD
112303   OUT:   IF CHAR=##D OR =##B THEN A=:TYPCHAR ELSE CALL SMBACKSP FI
112315          IF TYPCHAR=##D THEN DVAL ELSE IF NIFLAG><0 GO OERR; OVAL FI
112326          IF X:=MINFLAG><0 THEN A- FI; GO RETURN; *)FILL.
112335
112335   PACKCHAR: DRTN1 SH 6; A=:RTN1; T:=CHAR BZERO 6
112342             DRTN2 SH 6; D+T; AD=:DRTN2
112346             EXIT
112347
112347   RTPROG: O=:RTN1=:RTN2=:RTN3; CPNT=:PCPNT
112354          DO CALL SMCREAD; CALL SMSRCHINT(ALPN) WHILE =1 OD; T:=PCPNT=:CPNT
```

```
112365          IF =2 THEN ##0; GO INTRT FI % NUMBER, NOT NAME
112372          DO     CALL SMCREAD; A=:CHAR; CALL SMSRCHINT(ALPN); IF =2 GO OUT2
112401                 CALL PACKCHAR
112402          OD
112403   OUT2:  CALL SMBACKSP; RTNAME; CALL OPSYS(NAMSR); GO ERR; GO RETURN
112411
112411   RETURN: A=:OVAL; CALL SMCREAD; CALL SMSRCHINT(DELIM); IF =3 GO ERR
112420          CALL SMBACKSP; OVAL
112422   RETU1: IF T:=AGPFLG=0 THEN IF T:=SERVFLG=0 GO SMLEAVE; GO SM2LEAV FI
112431          IF T:=SERVFLG=0 GO SM2LEAV; GO SM3LEAV
112435
112435   *)FILL
112444   SYMFIL: "WORKA"=:OSTRING=:OVAL; 0=:OPNT
112450          DO CALL SMCREAD; A=:CHAR; CALL SMSRCHINT(DELIM); IF <3 GO OUT3; CHAR; CALL SMOWRITE OD
112462   OUT3:  CALL SMBACKSP; ##'; CALL SMOWRITE; 0=:OPNT; OVAL; GO RETURN
112470
112470   NAME:  CPNT; GO RETU1
112472   OERR:  ER208; GO ERRR
112474   ERR:   ER109
112475   ERRR:  *MON 64
112476          IF "F1204".RTRES><RTREF THEN
112503             X:=STPNT-STDELTA; "SMERR"=:X.S1; GO SMLEAVE
112510   SMERR:    GO SMLEAVE
112511          FI; IF SERVFLG=0 GO SSSLEA; GO SMLEAVE
112515   *)FILL
112530
112530   GSGNAM: CALL SMOREAD; A=:TYP2CHAR
112532           "WORKA"=:OSTRING; 0=:NNCHAR; CPNT=:PCPNT
112537           DO CALL SMCREAD; A=:CHAR; WHILE A-60<<=11; MIN NNCHAR; OD
112547           IF NNCHAR><0 AND CHAR=##D OR A=##B THEN CALL SMCREAD; A=:CHAR FI
112562           PCPNT=:CPNT; CHAR; CALL SMSRCHINT(DELIM)
112567           IF A<<3 THEN
112572              IF TYP2CHAR=##G THEN ##0; GO FAR INTRT FI
112600           FI; 0=:RTN1=:RTN2=:RTN3; 0=:OPNT=:NNCHAR
112605           DO
112605                CALL SMCREAD; A=:CHAR; CALL SMSRCHINT(XSALPN)
112611           WHILE A><4
112614                IF A=1 AND NNCHAR=0 GO ERRRS
112621                MIN NNCHAR; CHAR; CALL SMOWRITE; CALL FAR PACKCHAR
112625                IF NNCHAR>7 GO ERRRS
112631          OD; CHAR; CALL SMSRCHINT(DELIM); IF A=3 GO ERRRS
112640          CALL SMBACKSP; ##'; CALL SMOWRITE
112643          IF TYP2CHAR=##S THEN
112647             IF NNCHAR=0 THEN A:=-1 ELSE A:=OSTRING FI; GO FAR RETURN
112655          FI; RTNAME; CALL OPSYS(SNAMSR); GO FAR ERRRS; GO FAR RETURN
112662
112662   INTEGER XEILS:='$ILLEGAL SEGMENT NAME'
112675   ERRRS: "XEILS"; CALL SMOUTTEXT; GO FAR ERRR
112700   *)FILL
112716
112716   DBINT: CALL SMOREAD
112717          A=: TYPCHAR;0=:D1VAL=:D2VAL=:D1VAL=:D2VAL=:MINFLAG
112725          CALL SMCREAD;IF = ##- THEN 1 ELSE CALL SMBACKSP;"0" FI; A=: MINFLAG
112736          DO     CALL SMCREAD; A=: CHAR
112740                 IF <##0 OR >##9 GO DBOUT
112746                 IF>##7 THEN MIN NIFLAG FI
112752            % DECIMAL:
112752                 A-##0 =:T; AD:=DDVAL SHZ 3; A=:L; D=:X
112760                 AD:=DDVAL SHZ 1; D+X; A:=A+C+L; D+T; A:=A+C
112767                 AD=:DDVAL
```

```
112770            %OCTAL:
112770                 A:=CHAR
112771                 A-##0 =:T; AD:=DOVAL SHZ 3; D+T; AD=: DOVAL
112777          CD
113000          *)FILL
113003   DBOUT: IF CHAR =##D OR =##B THEN A=:TYPCHAR  ELSE CALL SMBACKSP FI
113015          IF TYPCHAR =##D THEN AD:=DDVAL ELSE IF NIFLAG><0 GO FAR OERR;AD:=DOVAL FI
113027          IF X:=MINFLAG><0 THEN  %  AD:= -AD
113031               D=:T; A=:L; 0=:A=:D; D-T; A:=A+C-1-L
113041          FI
113041
113041          AD=: DOVAL
113042          CALL SMCREAD;CALL SMSRCHINT(DELIM); IF = 3·GO FAR ERR
113050          CALL SMBACKSP;AD:= DOVAL;GO FAR RETU1
113053   RBUS
113061
113061
113061   %=========================================================================
113061   %              S M K G P A R
113061   %
113061   % ROUTINE TO GET PARAMETER IF THERE IS ONE
113061   % IF IT IS, SKIPRETURN, ELSE NO SKIP
113061   % PARAMETERS AS GPAR
113061   SUBR SMKGPAR
113061   DISP -200; INTEGER OST; PSID
113061   SMKGPAR: L=:D; CALL SMENTER; A=:OST; X:=0
113065   LOOP:  CALL SMCREAD; IF A=40 THEN X:=1; GO LOOP FI
113073          IF A=15 GO SMLEAVE; IF X><0 THEN CALL SMBACK FI
113100          CALL SMBACK; A:=OST; CALL SMGPAR; GO SM2LEAV
113104   RBUS
113112
113112
113112
113112   %=========================================================================
113112   %
113112   %              S T R I N G   I / O   R O U T I N E S
113112   %
113112   %=========================================================================
113112   % 28.10         S M C R E A D   S M O R E A D   S M C W R I T E   S M O W R I T E
113112   %               S M B A C K S P   S M S C A B
113112   %
113112
113112   SUBR SMCREAD,SMOREAD,SMCWRITE,SMOWRITE,SMBACKSP,SMSCAB,SMCRLF
113112   INTEGER ILLST:='ILL. STRING'
113120
113120   SMCREAD: X=:D:=CPNT; MIN CPNT; T:=CSTRING; GO LBY
113125   SMOREAD: X=:D:=OPNT; MIN OPNT; T:=OSTRING
113131   LBY:    *LBYT
113132           GO OUT
113133
113133   SMCWRITE: X=:D:=CPNT; MIN CPNT; T:=CSTRING; GO SBY
113140   SMOWRITE: X=:D:=OPNT; MIN OPNT; T:=OSTRING
113144   SBY:    *SBYT
113145   OUT:    IF X>=5CBUFSIZE THEN "ILLST"; CALL SMOUTTEXT; X:=0; T:=CSTRING; 15; *SBYT
113156                 T:=-1; *MON 2CLOS; MON 2ERMS
113161                 "EROPCOM"=:RETUAD; T:="6P3RET"; *MON 2MCAL
113165           FI
113165           D=:X; EXIT
113167
113167   SMBACKSP: IF CPNT><0 THEN A-1=:CPNT FI; EXIT
```

```
113174
113174   SMSCAB: L=:D; CALL SMENTER
113176   LOOP:  CALL SMCREAD; IF =40 GO LOOP
113202          T:=CPNT-1=:CPNT; GO SMLEAVE
113206   RBUS
113216
113216   %==========================================================================
113216   %              S M C C L E A R
113216   %
113216   %    SUBROUTINE TO CLEAR COMMAND STRING BUFFER
113216   %    CALLED FROM SUBCOMMANDS ON PARAMETER ERRORS, IN ORDER TO
113216   %    STAY INSIDE THE SUBCOMMAND
113216   %
113216   SUBR SMCCLEAR
113216   SMCCLEAR: L=: D;CALL SMENTER
113220             0=:CPNT; T:=CSTRING
113222             X:=0 ;   A:=15
113224             *SBYT
113225             GO SMLEAVE
113226   RBUS
113230   %==========================================================================
113230   % 28.11       S M O U T T E X T
113230
113230   %SUBROUTINE TO OUTPUT TEXT ON TELETYPE
113230   %A-REG. POINTS TO STRING TERMINATED WITH'
113230   %THE VARIABLE "TEXTADR" IS MOVED TO THE LOCATION AFTER THE STRING
113230
113230   SUBR SMOUTTEXT
113230   DISP -200; INTEGER PNT,XREG; PSID
113230   SMOUTTEXT: L=:D; CALL SMENTER; A=:PNT; X=:XREG
113234          FOR X:=0 TO 1000 DO
113240                T:=PNT; *LBYT
113242                WHILE ><##' AND A NBIT 7
113247                IF A=##$ THEN CALL SMCRLF ELSE CALL SMTCO FI
113255          OD; A:=X SHZ -1+1+PNT=:TEXTADR
113264          X:=XREG; GO SMLEAVE
113266   RBUS
113273
113273   %==========================================================================
113273   %              S M D D E C U T
113273   %
113273   % SUBROUTINE TO OUTPUT A DECIMAL NUMBER IN AD REGISTERS
113273   %
113273   SUBR SMDDECUT
113273
113273   DISP -200; INTEGER FLAGR,CNT; DOUBLE NUMBR; PSID
113273
113273   @ICR;
113273   DOUBLE ARRAY DPT10:=(35632,145000,2765,160400,230,113200,17,41100,
113303                       1,103240,0,23420,0,1750,0,144,0,12,0,1);
113317   @CR;
113317
113317   SMDDECUT: T:=D; L=:D; CALL SMENTR
113322          D:=T; AD=:NUMBR; 0=:FLAGR
113325          FOR X:=0 STEP 2 TO 22 DO
113331             0=:CNT; AD:=DPT10(X); A=:L; D=:T
113335             AD:=NUMBR
113336   LOOP:     D-T; A:=A+C-1-L
113342             IF A>=0 THEN MIN CNT; GO LOOP FI
113345             D+T; A:=A+C+L; AD=:NUMBR
```

```
113351              IF CNT><0 OR FLAGR><0 THEN
113355                 CNT+##0; CALL SMTCO; 1=:FLAGR
113362              ELSE
113363                 IF X=22 THEN 60 ELSE 40 FI; CALL SMTCO
113372              FI
113372           OD; GO SMLEAVE
113375   RBUS
113401
113401   %=========================================================================
113401   %              S M 2 D E C U T
113401   %
113401   SUBR SM2DECUT
113401   SM2DECUT: L=:D; CALL SMENTER
113403           A=:D:=0; T:=12; *RDIV ST
113407           D=:X; A+60; CALL SMTCO; 60+X; CALL SMTCO
113415           GO SMLEAVE
113416   RBUS
113421
113421
113421   %=========================================================================
113421   % 28.12       S M O C T U   S M D E C U   S M D T D E C   S M 3 O C T U T
113421
113421   %SUBROUTINES FOR INTEGER OUTPUT; VALUE IN A-REG.
113421   SUBR SMOCTU,SMDECU,SMDTDEC,SM3OCTUT,SMDOCTU
113421   @DEC
113421   DATA(10000,1000,100,10,1); INTEGER ARRAY CONST(0)
113426   @OCT
113426   DISP -200; DOUBLE DVAL; INTEGER VALUE,NULLFLAG; PSID
113426   CHBLANK: IF =0 AND X><-1 AND T:=NULLFLAG=0 THEN 40
113436           ELSE A+60; MIN NULLFLAG FI; EXIT
113442
113442   SMDOCTU: T:=D; L=:D; CALL SMENTER
113445           D:=T; AD=:DVAL; 40; CALL SMTCO; 0=:NULLFLAG
113452           DVAL; A SHZ -16/\3; CALL CHBLANK; CALL SMTCO
113457           DVAL SHR 22=:DVAL
113462           FOR X:=-12 DO
113463              DVAL SHR 3=:DVAL; A/\7; CALL CHBLANK; CALL SMTCO
113471           OD; GO SMLEAVE
113473
113473   SMOCTU: L=:D; CALL SMENTER; AD SHZ -17=:DVAL; 40; CALL SMTCO; 0=:NULLFLAG
113502           FOR X:=-6 DO DVAL; CALL CHBLANK; CALL SMTCO
113506                 DVAL SH 20 SHZ -15=:DVAL
113512           OD; GO SMLEAVE
113514
113514   SM3OCTUT: L=:D; CALL SMENTR
113516           AD SHZ -11=:DVAL; 0=:NULLFLAG
113521           40; CALL SMTCO
113523           FOR X:=-3 DO; DVAL SH 3=:DVAL; A/\7; CALL CHBLANK; CALL SMTCO OD
113533           GO SMLEAVE
113534
113534   SMDTDEC: L=:D; CALL SMENTER; A=:VALUE; GO DOUT
113540   SMDECU: L=:D; CALL SMENTER; A=:X; 40;CALL SMTCO
113545           IF X<0 THEN X-; ##- ELSE 40 FI; CALL SMTCO; X=:VALUE
113555   DOUT:   0=:NULLFLAG
113556           FOR X:=-5 DO VALUE=:D; A:=0; T:=CONST(X); *RDIV ST
113564                 T:=D=:VALUE; CALL CHBLANK; CALL SMTCO
113570           OD; GO SMLEAVE
113572   RBUS
113600
113600   %=========================================================================
```

```
113600   %              S M P E R C E N T
113600   %
113600   % SUBROUTINE TO COMPUTE AND PRINT A PERCENTAGE
113600   % AD=VALUE; X=POINTER TO TOTAL (DOUBLE)
113600   SUBR SMPERCENT
113600   DISP -200; DOUBLE ADVAL,DDO=SO; PSID
113600   SMPERCENT: T:=D; L=:D; CALL SMENTER; D:=T; AD=:ADVAL
113605          AD:=X.DDO; X:=0
113607          DO WHILE A><0; AD SHZ -1; X+1 OD; IF D<0 THEN AD SHZ -1; X+1 FI
113617          IF D BIT 16 THEN D SHZ -1; X+1 FI
113623          D=:T; ADVAL SH 2; FOR X-, DO AD SHZ -1 OD; 144; *RMPY SA DD; RDIV ST
113634          IF A+1 SHZ-1>=144 THEN ##1; CALL SMTCO; #00; CALL SM2TCO
113645          ELSE A=:X; 40; CALL SMTCO; X=:A; CALL SM2DEC
113653          FI; GO SMLEAVE
113654   RBUS
113662
113662
113662
113662   %=========================================================================
113662   %
113662   %              T E R M I N A L   I / O
113662   %
113662   %=========================================================================
113662   % 28.13       S M C R L F   S M T C O   S M T C I   S M T C O 2   S M W I N B
113662   %
113662
113662   SUBR SMCRLF,SMTCO,SMTCI,SM2TCO,SMWINB,SMSPAC
113662   INTEGER POINTER CCUSER:=CUSER
113663
113663   % SUBROUTINE TO OUTPUT A NUMBER OF SPACES
113663   % ENTRY: A=NUMBER OF SPACES TO OUTPUT
113663   SMSPAC: X=:D:=A-; T:=TDVN
113667           FOR X DO
113667              240; *MON 2OUTB; JMP *+1
113672           OD; X:=D; EXIT
113675
113675   SMCRLF: L=:D; CALL SMENTER; 15; CALL SMTCO; 12; CALL SMTCO; GO SMLEAVE
113704   SM2TCO: L=:D; CALL SMENTER; A=:X SHZ -10; CALL SMTCO
113711          377/\X; CALL SMTCO; GO SMLEAVE
113715   SMTCO:  L=:D; T:=TDVN; CALL SETPARITY; *MON 2OUTB; 0; COPY SD DP
113723   SMTCI:  L=:D; CALL SMENTER
113725          IF X:=TTIFIELD.FLAGB NBIT 5ESCON AND CCUSER>=0 THEN
113733            CALL ESCON
113734          FI  T:=1
113735   SMWINB:  *MON 2INBT; MON 2ERMS
113737          IF X NBIT 5ESCON THEN CALL ESCOFF FI; A/\177
113743          A=:T; A:=TTIFIELD.CESCP/\377; A:=:T
113750          IF A BIT 5 AND BIT 6 AND A><T AND A><177 AND A><X.CESCP SHZ -10 THEN
113765             A BZERO 5 FI
113766          IF =15 THEN CALL SMCRLF; 15 FI
113773          GO SMLEAVE
113774   RBUS
114012
114012   %=========================================================================
114012   %              S M D Y N A L L O C
114012   %
114012   % TEST IF A TERMINAL (TAD ETC) IS TEMPORARLY OR FIXED CONNECTED
114012   % TO A BACKGROUND PROGRAM
114012   %
114012   % ENTRY:      X=DATAFIELD
```

```
114012   %
114012   % EXIT:       NOT TEMPORARLY CONNECTION
114012   %
114012   % EXIT+1:     A=INDEX IN BACKGROUND TABLE
114012   %
114012   SUBR SMDYNALLOC
114012   SMDYNALLOC: L=:D; CALL SMENTER
114014           IF "BPTMP"=0 GO SMLEAVE
114017           CALL GBTINDX; GO SMLEAVE; A=:D
114022           IF X.TYPRING NBIT 5TERM AND A NBIT 5BAD THEN GO SMLEAVE FI
114030           A:=D; GO SM2LEAV
114032   RBUS
114037
114037   %=====================================================================
114037   %             S M T A C T I V E
114037   %
114037   % SUBROUTINE TO CHECK IF A TERMINAL (OR TAD) WITH TEMPORARLY CONNECTION
114037   % TO A BACKGROUND PROGRAM IS ACTIVE OR NOT
114037   %
114037   % ENTRY:      X=DATAFIELD
114037   %             A=DBPROG
114037   %
114037   % EXIT:       NOT ACTIVE
114037   %
114037   % EXIT+1:     ACTIVE
114037   %
114037   SUBR SMTACTIVE
114037   SMTACTIVE: A=:T
114040           IF "BPTMP"=0 THEN EXITA FI
114043           IF X.RTRES=T THEN EXITA FI
114047           EXIT
114050   RBUS
114051
114051   INTEGER ARRAY SMFIBUF(5WCBUFSIZE+1)     % USED INSTEAD OF FIBUF ON THE
114136                                           % SYSTEM SEGMENT SO ALL COMMANDS
114136                                           % CAN BE EXECUTED IN A BATCH JOB
114136
114136
114136   %=====================================================================
114136   % 28.14       C O M M A N D   T A B L E
114136   %
114136
114136   INTEGER ARRAY SSBUF=?
114136   INTEGER PTERMNO:='IDTERMINAL NUMBER: '
114150   INTEGER 6PINOUT:='N INPUT OR OUTPUT? ',6BFTYP:='BPUN',6DFTYP:='DATA'
114170   INTEGER 6PRTNAM:='R RT NAME: ',    6TILBOUNDS:='$ILL. ADDRESS BOUNDS'
114211   INTEGER 6PAREA:='N MEMORY, IMAGE, SAVE-AREA OR SEGMENT? '
114235   INTEGER 6PSEGNO:='GGSEGMENT (NAME OR NUMBER(OCT)): ',     6PLOGU:='IOLOG.UNIT NO.: '
114267   INTEGER 6PADDR:='IOADDRESS: ',        6PSPNUMB:='IOSPOOLING INDEX: '
114307   INTEGER 6FFHE:='N FORM-FEED BEFORE? ', 6FFTR:='N FORM-FEED AFTER? '
114334   INTEGER 6TILLINE:='$TOO LONG STRING'
114345   INTEGER ARRAY 6P3RET(0)
114345   INTEGER RETUAD,RETSG:=377\5OPSEG         % RETURN ADDRESS ON OPCOM SEGMENT (3)
114347   INTEGER 6PRTFIL:=("5NASEM",NULL,NULL)
114352   INTEGER 6RTLRES:=("5RTLSEM",NULL,"1"); *)FILL
114360   INTEGER 6CTABVALUE,6TINCR,6BLCKI,6BLOCK,6CFILNO,6FINAM,NWORDS:=100
114367   INTEGER 6RPAR:=(6CFILNO,NULL,SSBUF,6BLOCK,NWORDS)
114374   INTEGER 6WPAR:=(6CFILNO,NULL,SSBUF,6BLOCK,NWORDS)
114401   INTEGER ARRAY 6IDMAX:=(MAX10,MAX11,MAX12,MAX13)
114405   INTEGER ARRAY 6IDTAB:=(ITB10,ITB11,ITB12,ITB13)
```

```
114411   INTEGER ARRAY 6EIDTAB:=(ITE10,ITE11,ITE12,ITE13)
114415   INTEGER 6SINTR:='SINTRAN:DATA',6MACM:='MACM-AREA:DATA'
114434   INTEGER 6TERRP:='$ILLEGAL PARAMETER'
114446   INTEGER 6ILLA:='ILLEGAL ADDRESS'
114456   INTEGER 6STRNUM:='IONUMBER:'
114463   INTEGER 6ADRMOD:=-9SMRE       % ADDRESS DISP. FOR POF ON MACM-AREA
114464   INTEGER 6XALT
114465   INTEGER SMCILLP=6TERRP
114465   INTEGER 6CFUPAR:='N $FUNCTION: '
114474   INTEGER SMFAMB:='$AMBIGUOUS FUNCTION'
114506   INTEGER SMTXFILL:='$ILLEGAL FUNCTION'
114517   INTEGER SMCAMBIGP:='$AMBIGUOUS PARAMETER'
114532
114532   @ICR
114532   INTEGER SCOMTAB:=(
114532   COM01,0,CDAFI,0,                COM02,0,ILUTAB,0,
114542   COM03,0,RLUTAB,0,               COM04,0,ITIMTAB,0,
114552   COM05,0,RTIMTAB,0,              COM06,0,IIDTAB,0,
114562   COM07,0,RIDTAB,0,               COM08,0,IEIDTAB,0,
114572   COM09,0,REIDTAB,0,              COM10,0,CRBIN,0,
114602   COM11,0,CDRTC,0,                COM12,0,DRTDES,0,
114612   COM13,0,DSTENTRY,0,             COM14,0,DEFSGFI,0,
114622   COM15,0,ODUMP,0,                COM16,0,ADUMP,0,
114632   COM17,0,DVSTRING,0,             COM18,0,DUSSUBR,0,
114642   COM19,0,DURPRO,0,               COM20,0,DURSUB,0,
114652                                   COM22,0,IBACKT,0,
114656   COM23,0,RBACKT,0,               COM24,0,DTSLICE,0,
114666   COM25,0,DUMCALL,0,              COM26,0,CHVARI,0,
114676   COM27,0,SSDNUMB,0,              COM28,0,ISPHEAD,0,
114706   COM29,0,DBSPROG,0,              COM30,0,SWDENT,0,
114716   COM31,0,SCHPRI,0,               COM32,0,LIADDR,0,
114726   COM33,0,IIIOX,0,                COM34,0,RFIOX,0,
114736   COM35,0,LISCOM,0,               COM36,0,SSLEAV,0,
114746   COM37,0,RSPHEAD,0,              COM38,0,DLSGFI,0,
114756   COM39,0,LISCOM,0,               COM40,0,ISYSEG,0,
114766   COM41,0,ITSLIS,0,               COM42,0,XRTSLIS,0,
114776   COM43,0,SCPROT,0,               COM44,0,CHIOBS,0,
115006   COM45,0,SCSFS,0,                COM46,0,XINST,0,
115016   COM47,0,XMREM,0,                COM50,0,REMCMM,0,
115026   COM51,0,CCCOM,0,                COM52,0,SENCT,0,
115036   COM53,0,CENCT,0,                COM54,0,DFPRMT,0,
115046   COM55,0,SGWPE,0,                COM56,0,SGWPR,0,
115056   COM57,0,PTLAM,2,                COM58,0,PFLAM,2,
115066   COM59,0,CRLAM,2,                COM60,0,DELAM,2,
115076   COM61,0,LAREA,2,                COM62,0,LAINF,2,
115106   COM63,0,LLAMC,2,                COM64,0,SLAMC,2,
115116   COM65,0,DFHDL,0,                COM66,0,LIHDL,0,
115126   COM67,0,CMMCLG,0,               COM68,0,CMCPULG,0,
115136   COM69,0,SWPLO,0,                COM70,0,LOGDISC,0,
115146   COM71,0,CMCHTABLE,0,            COM72,0,CMLTSL,0,
115156   COM73,0,CPTSLCLASS,0,           GPSTA,0,SGPIB,0,
115166   GPSTO,0,KGPIB,0,                GPBSI,0,SGPBS,0,
115176   COM74,0,FCPUL,0,                COM75,0,LAPRO,2,
115206   COM76,0,BAUTIL,0,               COM77,0,LTSPR,0,
115216   COM78,0,SSCLDMO,0,              COM79,0,RSCLDMO,0,
115226   0,0,0,0,                        0,0,0,0,
115236   0,0,0,0,                        0,0,0,0,
115246   0,0,0,0,                        0,0,0,0,
115256   -1);
115257   @CR;
```

```
115257
115257
115257   %=========================================================================
115257   % 28.15      S E R V S I N T   6 S S M O N
115257   %
115257   SUBR SINSERV,SSMON,SSLEAV,SSSLEA
115257
115257   DISP 2; INTEGER ROUTADR; PSID
115257   DISP 6; INTEGER OFTP; PSID
115257   DISP -200; INTEGER POINTER 6LREG; PSID
115257
115257   % COMMAND NAMES
115257
115257   INTEGER COM01:='CHANGE-DATAFIELD'
115270   INTEGER COM02:='INSERT-IN-LOGICAL-UNIT-TABLE'
115307   INTEGER COM03:='REMOVE-FROM-LOGICAL-UNIT-TABLE'
115327   INTEGER COM04:='INSERT-IN-TIMER-TABLE'
115342   INTEGER COM05:='REMOVE-FROM-TIMER-TABLE'
115356   INTEGER COM06:='INSERT-IN-IDENT-TABLE'
115371   INTEGER COM07:='REMOVE-FROM-IDENT-TABLE'
115405   INTEGER COM08:='INSERT-IN-EXTENDED-IDENT-TABLE'
115425   INTEGER COM09:='REMOVE-FROM-EXTENDED-IDENT-TABLE'
115446   INTEGER COM10:='READ-BINARY'
115454   INTEGER COM11:='DEFINE-RTCOMMON-SIZE'
115467   INTEGER COM12:='DUMP-RT-DESCRIPTION'
115501   INTEGER COM13:='DUMP-SEGMENT-TABLE-ENTRY'
115516   INTEGER COM14:='DEFINE-SEGMENT-FILE'
115530   INTEGER COM15:='OCTAL-DUMP'
115536   INTEGER COM16:='ASCII-DUMP'
115544   INTEGER COM17:='DEFINE-TITLE'
115553   INTEGER COM18:='DEFINE-USER-START-SUBROUTINE'
115572   INTEGER COM19:='DEFINE-USER-RESTART-PROGRAM'
115610   INTEGER COM20:='DEFINE-USER-RESTART-SUBROUTINE'
115630   INTEGER COM22:='INSERT-IN-BACKGROUND-TABLE'
115646   INTEGER COM23:='REMOVE-FROM-BACKGROUND-TABLE'
115665   INTEGER COM24:='DEFINE-TIME-SLICE'
115676   INTEGER COM25:='DEFINE-USER-MONITOR-CALL'
115713   INTEGER COM26:='CHANGE-VARIABLE'
115723   INTEGER COM27:='SET-SPOOLING-DEVICE-NUMBER'
115741   INTEGER COM28:='INSERT-SPOOLING-HEADER'
115755   INTEGER COM29:='DEFINE-BATCH-SUPERVISOR'
115771   INTEGER COM30:='SWAP-DIRECTORY-ENTRIES'
116005   INTEGER COM31:='SET-CHANNEL-PRIORITY'
116020   INTEGER COM32:='LIST-ADDRESSES'
116030   INTEGER COM33:='INSERT-IN-IOX-TABLE'
116042   INTEGER COM34:='REMOVE-FROM-IOX-TABLE'
116055   INTEGER COM35:='LIST-SERVICE-COMMANDS'
116070   INTEGER COM36:='EXIT'
116073   INTEGER COM37:='REMOVE-SPOOLING-HEADER'
116107   INTEGER COM38:='DELETE-SEGMENT-FILE'
116121   INTEGER COM39:='HELP'
116124   INTEGER COM40:='INITIALIZE-SYSTEM-SEGMENT'
116141   INTEGER COM41:='INSERT-IN-TIME-SLICE'
116154   INTEGER COM42:='REMOVE-FROM-TIME-SLICE'
116170   INTEGER COM43:='SET-COMMAND-PROTECTION'
116204   INTEGER COM44:='CHANGE-BUFFER-SIZE'
116216   INTEGER COM45:='SET-CLOSED-SCRATCH-FILE-SIZE'
1  235   INTEGER COM46:='START-XMSG'
   43    INTEGER COM47:='STOP-XMSG'
116250   INTEGER COM50:='REMOVE-SINTRAN-COMMAND'
```

```
116264   INTEGER COM51:='CC'
116266   INTEGER COM52:='SET-MAX-ENTER-COUNT'
116300   INTEGER COM53:='CLEAR-ENTER-COUNT'
116311   INTEGER COM54:='DEFINE-PROMPT-STRING'
116324   INTEGER COM55:='SEGMENT-WRITE-PERMIT'
116337   INTEGER COM56:='SEGMENT-WRITE-PROTECT'
116352   INTEGER COM57:='PAGES-TO-LAMU'
116361   INTEGER COM58:='PAGES-FROM-LAMU'
116371   INTEGER COM59:='CREATE-LAMU'
116377   INTEGER COM60:='DELETE-LAMU'
116405   INTEGER COM61:='LAMU-AREAS'
116413   INTEGER COM62:='LAMU-INFORMATION'
116424   INTEGER COM63:='LIST-LAMU-CONSTANTS'
116436   INTEGER COM64:='SET-LAMU-CONSTANTS'
116450   INTEGER COM65:='DEFINE-HDLC-BUFFER'
116462   INTEGER COM66:='LIST-HDLC-BUFFER'
116473   INTEGER COM67:='MONCALL-LOG'
116501   INTEGER COM68:='CPU-LOG'
116505   INTEGER COM69:='SWAPPING-LOG'
116514   INTEGER COM70:='DISC-ACCESS-LOG'
116524   INTEGER COM71:='CHANGE-TABLE'
116533   INTEGER COM72:='LIST-TIME-SLICE-PARAMETERS'
116551   INTEGER COM73:='LIST-TIME-SLICE-CLASS'
116564   INTEGER GPSTA:='START-GPIB'
116572   INTEGER GPSTO:='STOP-GPIB'
116577   INTEGER GPBSI:='CHANGE-GPIB-BUFFERSIZE'
116613   INTEGER COM74:='FIND-CPULOOPTIME'
116624   INTEGER COM75:='PROTECT-LAMU'
116633   INTEGER COM76:='BACKGROUND-ALLOCATION-UTILITIES'
116653   INTEGER COM77:='LIST-TIME-SLICED-PROGRAMS'
116670   INTEGER COM78:='SET-COLDSTART-MODE-FILE'
116704   INTEGER COM79:='RESET-COLDSTART-MODE-FILE'
116721
116721   INTEGER TNLGBATCH:='$COMMAND NOT LEGAL IN BATCH$'
116740   INTEGER TNSCOM:='$NO SUCH COMMAND',TAMBIG:='$AMBIGUOUS COMMAND'
116763   INTEGER TNIMPL:='$NOT IMPLEMENTED',6TILFI:='$ILLEGAL FILE'
117003   INTEGER RESPAR:=("SEMSERV",NULL,"1")
117006   INTEGER 6S3BCK:="5OP2SEG"; *)FILL
117012
117012   INTEGER SLREG=?,STREG=?
117012
117012
117012   CHLFILE: T:=L=:"6LREG"
117014          X:=A; "6DFTYP"; T:=2; *MON 2NOPE
117020          GO ERR2
117021          CALL LOGPH; IF A=0 THEN CALL ERRFATAL FI
117024          IF A.OFTP NBIT OCBIT AND A NBIT OABIT GO ERR2
117032          GO 6LREG
117033   ERR2:  "6TILFI"; CALL SMOUTTEXT; GO SSLEAV
117036   SINSERV: T=:STREG; A:=L=:SLREG
117041          GO SSMON
117042
117042   INTEGER SLREG,STREG
117044
117044   ERR:    "6TERRP"; CALL SMOUTTEXT; GO SSLEAV; *)FILL
117057   SSMON:  "STBEG"=:STPNT; 1=:TDVN; CALL SMCRLF; ###; CALL SMTCO
117066           "6RTLRES"; *MON 2RELE
117070           "6PRTFIL"; *MON 2RELE
117072           CALL WIMBACK
117073           -1=:6BLCKI; T:=-1; *MON 2CLOS; JMP *+1
```

```
117100          CALL SMGCOM; CALL SMCREAD; IF A=15 GO SSMON
117105          0=:CPNT; X:="SCOMTAB"; CALL SMABLOOK; T=:X
117111          IF A><0 THEN
117112              IF A=-1 THEN "TNSCOM" ELSE "TAMBIG" FI
117120              CALL SMOUTTEXT; GO SSMON
117122          FI; IF X.ROUTADR=0 GO ENIMPLEMENT
117124          IF BCHFLAG=1 AND X.S3=1 THEN "TNLGBAT"; CALL SMOUTTEXT; GO SSMON FI
117137          IF X.S3=2 AND GNLAMU=0 GO ENIMPLEMENT
117145          X:=X.ROUTADR; *JPL ,X                % GO TO COMMAND ROUTINE
117147          GO SSMON
117150
117150   ENIMPLEMENT: "TNIMPL"; CALL SMOUTTEXT; GO SSMON
117153
117153   %  EXIT TO OPCOR
117153   SSSLEA: MIN SLREG
117154
117154   %  EXIT TO OPCOM
117154   SSLEAV:MIN SLREG
117155
117155   %  EXIT TO EROPCOM
117155   ERLEAV:T:=STREG; SLREG=:L; "RESPAR"; *MON 2RELE; MON 2MEXI
117163
117163   RBUS
117207
117207
117207   %=============================================================================
117207   % 28.16       6 E R R P A R   6 E R R O P E R
117207   %
117207   % SUBROUTINE TO WRITE ERROR MESSAGES
117207   %
117207   SUBR 6ERRPAR,6ERROPER
117207   INTEGER TERRO:='$ERROR IN OPERATION'
117221   6ERRPAR: "6TERRP"; GO ERRF
117223   6ERROPER: "TERRO"
117224   ERRF:  T:=1=:TDVN; CALL SMOUTTEXT; GO SSMON
117230   RBUS
117235
117235
117235   %=============================================================================
117235   % 28.17       F I L E R R
117235   %
117235   % SUBROUTINE TO WRITE A FILE-SYSTEM ERROR MESSAGE
117235   SUBR SMFILERR
117235   SMFILERR: *MON 64
117236          GO SSMON
117237   RBUS
117240
117240
117240   %=============================================================================
117240   % 28.18       6 0 0 P E N
117240   %
117240   % SUBROUTINE TO READ A "OUTPUT FILE:" NAME, AND OPENS THE FILES
117240   %
117240   SUBR 6OOPEN
117240   6OOPEN: L=:D; CALL SMENTER
117242           "SMSTROFI"; CALL SAGPAR; GO SSMON; 1
117246          IF A><1 THEN
117251                A=:X:="TYPS"; T:=0; *MON 2NOPE
117255                GO CFIERR
117256          FI; GO SM2LEAVE
```

```
117257   CFIERR: *MON 64
117260           GO SMLEAVE
117261   RBUS
117270
117270
117270
117270   %=========================================================================
117270   % 28.19       D R T D E S
117270   %
117270   % DUMP-RT-DESCRIPTION <RT-PROGRAM> (<OUTPUT FILE>)
117270   SUBR DRTDES
117270   DISP -200; INTEGER CRTDADR,CDEV,CAREA,CINDEX,EINDEX,TXINDEX,CADDR; PSID
117270
117270   @ICR;
117270   INTEGER ARRAY TRTDTEXT:=(
117270           XRT0,XRT1,XRT2,XRT3,XRT4,XRT5,XRT6,XRT7,XRT10,
117270           XRT11,XRT12,XRT13,XRT14,XRT15,XRT16);
117301   INTEGER ARRAY XTRTDTEXT:=(
117307           YRT0,YRT1,YRT2,YRT3,YRT4,YRT5,YRT6,YRT7,YRT10,
117307           YRT11,YRT12,YRT13,YRT14,YRT15,YRT16,YRT17);
117320
117327   INTEGER XRT0:='TLINK:    ',XRT1:='STATUS:   ',XRT2:='DTIM1:    ',XRT3:='DTIM2:    ',
117327   XRT4:='DTIN1:    ',XRT5:='DTIN2:    ',XRT6:='STADR:    ',XRT7:='SEGM:     ',
117353   YRT0:='DPREG:    ',
117377   YRT1:='DXREG:    ',YRT2:='DTREG:    ',YRT3:='DAREG:    ',YRT4:='DDREG:    ',
117404   YRT5:='DLREG:    ',YRT6:='DSREG:    ',YRT7:='DBREG:    ',XRT10:='WLINK:    ',
117430   XRT11:='ACTSEG:  ',XRT12:='ACTPRI:  ',XRT13:='BRESLINK:',
117454   XRT14:='RSEGM:   ',XRT15:='WINDOW:  ',XRT16:='RTDLGADD:',
117473   YRT10:='BITMAP:  ',YRT11:='BITM1:   ',YRT12:='BITM2:    ',YRT13:='BITM3:    ',
117512   YRT14:='BITM4:   ',YRT15:='BITM5:   ',YRT16:='BITM6:    ',YRT17:='BITM7:    ';
117536   @CR;
117562
117562   DRTDES: L=:D; CALL SMENTER
117562           "6PRTNAM"; CALL SMSGPAR; GO SSMON; IF A=0 THEN RTREF FI
117564           A=:CRTDADR; CALL CHRTDES; GO 6ERRPAR
117571           CALL ASKAREA; GO 6ERRPAR; A=:CAREA
117574           CALL 6OOPEN; GO SMLEAVE; A=:CDEV
117577           T:=5RTSIZE; X:=CRTDADR; CAREA GOSW CRMEMO,CRIMAGE,CRSAVE
117602           GO 6ERRPAR
117611   CRMEMO: CALL RMEMO; GO DRTD
117612   CRIMAGE: CALL RIMAGE; GO DRTD
117614   CRSAVE: CALL 6SOPEN; CALL RSAVE
117616           CRTDADDR/\37=:CINDEX+"5RTSIZE-1"=:EINDEX; GO DRTD1
117620   DRTD:   0=:CINDEX; "5RTSIZE-1"=:EINDEX
117626   DRTD1:  CDEV=:TDVN; 0=:TXINDEX; CALL ESCON
117631           FOR CINDEX TO "RTDLGADDR" DO
117635                CALL SMCRLF; TRTDTEXT(TXINDEX); CALL SMOUTTEXT
117641                SSBUF(CINDEX); CALL SMOCTU; MIN TXINDEX
117645           OD; CRTDADDR.RTDLGADDR=:CADDR; 0=:CINDEX
117651           FOR CINDEX TO "BITM7" DO
117661              CALL SMCRLF; XTRTDTEXT(CINDEX); CALL SMOUTTEXT
117665              IF CAREA=0 THEN                                  % MEMORY
117671                 T:=0; CADDR+CINDEX=:X; *LDATX
117673              ELSE
117700                 A:=0
117701              FI; CALL SMOCTU
117702           OD; CALL ESCOFF; GO SMLEAVE
117703   RBUS
117711
117743
117743
```

```
117743   %-=========================================================================
117743   % 28.20       D S T E N T R Y
117743   %
117743   % DUMP-SEGMENT-TABLE-ENTRY <SEGMENT NUMBER> (<OUTPUT FILE>)
117743   SUBR DSTENTRY
117743   DISP -200; INTEGER CSEGNO,CSETADR,CAREA,CDEV,CINDEX,ENDINDEX,TXINDEX; PSID
117743
117743   INTEGER ARRAY TSEGTTEXT:=(XST0,XST1,XST2,XST3,XST4)
117750   INTEGER XST0:='SEGLINK:  ',XST1:='BPAGELINK:',XST2:='LOGADR:    '
117772   INTEGER XST3:='MADR:     ',XST4:='FLAG:     '
120006   DSTENTRY: L=:D; CALL SMENTER
120010           "6PSEGNO"; CALL SMSGPAR; GO SSMON; A=:CSEGNO; CALL LEGSCHECK; GO 6ERRPAR
120016           X=:CSETADR; CALL ASKAREA; GO 6ERRPAR
120021           A=:CAREA; CALL 6OOPEN; GO SMLEAVE; A=:CDEV
120025           IF A><1 THEN
120030                 A=:X:="TYPS"; T:=0; *MON 2NOPE
120034                 GO SMFILERR
120035           FI; A=:CDEV
120036           T:=5SEGSIZE; X:=CSETADR; CAREA GOSW CRMEMO,CRIMAGE,CRSAVE
120045           GO 6ERRPAR
120046   CRMEMO: CALL RMEMO; GO DSTE
120050   CRIMAGE: CALL RIMAGE; GO DSTE
120052   CRSAVE: CALL 6SOPEN; CALL RSAVE
120054           CSETADR/\37=:CINDEX+"5SEGSIZE-1"=:ENDINDEX; GO DSTE1
120062   DSTE:   0=:CINDEX; "5SEGSIZE-1"=:ENDINDEX
120065   DSTE1:  CDEV=:TDVN; 0=:TXINDEX; CALL ESCON
120071           FOR CINDEX TO ENDINDEX DO
120075                 CALL SMCRLF; TSEGTTEXT(TXINDEX); CALL SMOUTTEXT
120101                 SSBUF(CINDEX); CALL SMOCTU; MIN TXINDEX
120105           OD; CALL ESCON; GO SMLEAVE
120113   RBUS
120142
120142
120142   %=========================================================================
120142   % 28.21       R I M A G E   R M E M O   R S A V E
120142   %
120142   % ROUTINE TO READ A NUMBER OF WORDS TO THE BUFFER SSBUF
120142   % FROM MEMORY, CORE IMAGE, SINTRAN-AREA OR MACM-AREA
120142   %
120142   % ENTRY: T=NO. OF WORDS TO READ
120142   %        X=ADDRESS OF FIRST WORD
120142
120142   SUBR RIMAGE,RMEMO,RSAVE
120142   DISP -200; INTEGER NOELEM,CINDEX; PSID
120142   RIMAGE: L=:D; CALL SMENTER
120144           T-=:NOELEM; 0=:CINDEX
120147           FOR NOELEM DO
120147                 T:=2; CALL GET1L; GO ERR; X=:D; A=:SSBUF(CINDEX)
120155                 MIN CINDEX; X:=D+1
120157           OD; GO SMLEAVE
120162   RMEMO:  L=:D; CALL SMENTER
120164           T-=:NOELEM; 0=:CINDEX
120167           FOR NOELEM DO
120167                 T:=0; CALL GET1L; GO ERR
120172                  X=:D; A=:SSBUF(CINDEX); MIN CINDEX; X:=D+1
120177           OD; GO SMLEAVE
120202   RSAVE:  L=:D; CALL SMENTER
120204           CALL R1SAVE; GO SMLEAVE
120206   ERR:    GO 6ERROPER
120207   RBUS
```

```
120215
120215
120215   %=========================================================================
120215   % 28.22       S E L A R E A
120215   %
120215   % SUBROUTINE TO ASK WHICH AREA SHOULD BE ACCESSED
120215   %
120215   % ENTRY: A=LEGAL AREA; A BIT 0=1: MEMORY
120215   %                      A BIT 1=1: CORE-IMAGE
120215   %                      A BIT 2=1: SAVE-AREA
120215   %
120215   % RETURN: AREA TO ACCESS SPECIFIED IN A REGISTER (BIT 0-2)
120215   %
120215   SUBR SELAREA
120215   DISP -200; INTEGER RESULT,LEGAREA; PSID
120215
120215   INTEGER TMEMO:='N MEMORY? ',TIMAGE:='N IMAGE? '
120230   INTEGER TSAVE:='N SAVE-AREA? '
120237
120237   SELAREA: L=:D; CALL SMENTER
120241           A=:LEGAREA; 0=:RESULT
120243           IF LEGAREA BIT BMEMO THEN
120246               "TMEMO"; CALL SMSGPAR; GO SSMON; CALL SMYESNO
120252               IF A=1 THEN RESULT BONE BMEMO=:RESULT FI
120260           FI
120260           IF LEGAREA BIT BIML THEN
120263               "TIMAGE"; CALL SMSGPAR; GO SSMON; CALL SMYESNO
120267               IF A=1 THEN RESULT BONE BIML=:RESULT FI
120275           FI
120275           IF LEGAREA BIT BSAVE THEN
120300               "TSAVE"; CALL SMSGPAR; GO SSMON; CALL SMYESNO
120304               IF A=1 THEN RESULT BONE BSAVE=:RESULT FI
120312           FI; RESULT; GO SMLEAVE
120314   RBUS
120324
120324
120324   %=========================================================================
120324   % 28.23       A S K A R E A
120324   %
120324   % SUBROUTINE TO SPECIFIE ONE AREA TO ACCESS
120324   %
120324   % RETURN: ILLEGAL AREA
120324   % SKIP RETURN: A=0: MEMORY
120324   %               A=1: CORE-IMAGE
120324   %               A=2: SAVE-AREA
120324   %               A=3: SEGMENT
120324   %
120324   SUBR ASKAREA
120324   INTEGER ARRAY TSELAR:=(TMEMO,0,0,0, TIMAGE,1,0,0, TSAVE,2,0,0, TSEGM,3,0,0, -1)
120345   INTEGER TMEMO:='MEMORY',TIMAGE:='IMAGE',TSAVE:='SAVE-AREA'
120361   INTEGER TSEGM:='SEGMENT'
120365
120365   ASKAREA: L=:D; CALL SMENTER
120367           "GPAREA"; CALL SMSGPAR; GO SSMON; X:="TSELAR"
120373           CALL SMABLOOK; IF A><0 GO SMLEAVE
120376           T.S1; GO SM2LEAV
120401   RBUS
120411
120411
120411   %=========================================================================
```

```
120411   % 28.24      L E G S C H E C K
120411   %
120411   % SUBROUTINE TO CHECK FOR LEGAL SEGMENT NUMBER
120411   %
120411   % ENTRY: A=SEGMENT NO.
120411   %
120411   % RETURN: ILLEGAL SEGMENT NUMBER
120411   % SKIP RETURN: SEGMENT NO. OK, X=ADDRESS OF SEGMENT TABLE ENTRY
120411
120411   SUBR LEGSCHECK
120411   LEGSCHECK: IF A<2 OR A>377 THEN EXIT FI
120420          A*5SEGSIZE+SEGSTART
120422          IF A<ESGTABLE THEN A=:X; EXITA FI
120427          EXIT
120430   RBUS
120434
120434
120434   %=========================================================================
120434   % 28.25      C H R T D E S
120434   %
120434   % SUBROUTINE TO CHECK FOR LEGAL RT-DESCRIPTION ADDRESS
120434   %
120434   % ENTRY: A=RT-DESCRIPTION ADDRESS
120434   %
120434   % RETURN: ILLEGAL RT-DESCRIPTION ADDRESS
120434   % SKIP RETURN: RT-DESCRIPTION ADDRESS OK
120434   %
120434
120434   SUBR CHRTDES
120434   CHRTDES: IF A>=SEGSTART OR A<RTSTART THEN EXIT FI
120443          A:=A-RTSTART=:D:=0; T:=5RTSIZE; *RDIV ST
120450          IF D><0 THEN EXIT FI
120453          EXITA
120454   RBUS
120456
120456
120456
120456   %=========================================================================
120456   % 28.26      I M T A B   I S T A B   I I T A B
120456   %           F F M T A B   F F I T A B   F F S T A B
120456   %
120456   % SUBROUTINE TO SEARCH FOR A WANTED ENTRY IN ONE OF THE
120456   % SYSTEM TABLES
120456   %
120456   % ENTRY: T=END OF TABLE MARK VALUE
120456   %        A=VALUE TO SEARCH FOR
120456   %        X=START OF TABLE
120456   %        6CTABVALUE=VALUE TO INSERT IN TABLE ENTRY
120456   %
120456   % RETURN: NO ENTRY FOUND
120456   % SKIP RETURN: VALUE IS SET IN TABLE ELEMENT
120456
120456   SUBR IMTAB,IITAB,ISTAB,FFMTAB,FFITAB,FFSTAB
120456   DISP -200; INTEGER CVALUE,ENDVALUE,ROUFLG; PSID
120456   IMTAB: L=:D; CALL SMENTER; 0=:ROUFLG; A=:CVALUE
120462   IMT1:  T=:ENDVALUE
120463          DO CALL R1MEMO WHILE A><ENDVALUE
120467                IF A=CVALUE THEN
120472                   IF ROUFLG=0 THEN 6CTABVAL; CALL W1MEMO; FI
120476                   GO OUT
```

```
120477                  FI
120477                  6TINCR+X=:X
120502          OD; GO OUT1
120504   IITAB: L=:D; CALL SMENTER; 0=:ROUFLG; A=:CVALUE
120510   IIT1:  T=:ENDVALUE
120511          DO CALL R1IMAGE WHILE A><ENDVALUE
120515                  IF A=CVALUE THEN
120520                     IF ROUFLG=0 THEN 6CTABVAL; CALL W1IMAGE; CALL WIMBACK FI
120525                     GO OUT
120526                  FI
120526                  6TINCR+X=:X
120531          OD; GO OUT1
120533   ISTAB: L=:D; CALL SMENTER; 0=:ROUFLG; A=:CVALUE
120537   IST1:  T=:ENDVALUE
120540          DO CALL R1SAVE WHILE A><ENDVALUE
120544                  IF A=CVALUE THEN
120547                     IF ROUFLG=0 THEN 6CTABVAL; CALL W1SAVE; CALL WXSAVE FI
120554                     GO OUT
120555                  FI
120555                  6TINCR+X=:X
120560          OD
120561   OUT1:  A:=X; GO SMLEAVE
120563   OUT:   A:=X; GO SM2LEAV
120565   FFMTAB: L=:D; CALL SMENTER
120567          A=:CVALUE; 1=:ROUFLG; GO IMT1
120573   FFITAB: L=:D; CALL SMENTER
120575          A=:CVALUE; 1=:ROUFLG; GO IIT1
120601   FFSTAB: L=:D; CALL SMENTER
120603          A=:CVALUE; 1=:ROUFLG; GO IST1
120607
120607   RBUS
120624
120624
120624   %==============================================================================
120624   % 28.27        I T I M T A B   R T I M T A B
120624   %
120624   % INSERT-IN-TIMER-TABLE <LOG. UNIT> <INPUT/OUTPUT>
120624   % REMOVE-FROM-TIMER-TABLE <LOG.UNIT> <INPUT/OUTPUT>
120624   %
120624   SUBR ITIMTAB,RTIMTAB
120624   DISP -200; INTEGER ROUFLG,CLOGU,INOUT,CDATAF,CVALUE,LEGARE; PSID
120624
120624   ITIMTAB: L=:D; CALL SMENTER
120626          0=:ROUFLG; GO L1
120630   RTIMTAB: L=:D; CALL SMENTER
120632          1=:ROUFLG
120634   L1:    "6PLOGU"; CALL SMSGPAR; GO SSMON; A=:CLOGU
120640          CALL INOROUT; GO ERR; A=:INOUT; CLOGU; CALL LOGPH; A=:X
120646          IF INOUT=0 THEN
120650                IF X=0 GO ERR; X=:CLOGU
120652          ELSE
120653                IF A:=D=0 GO ERR; A=:CLOGU
120656          FI; IF ROUFLG><0 THEN CLOGU FI; A=:CVALUE
120662          7; CALL SELAREA; A=:LEGAREA
120665          IF A><0 THEN
120666                1=:6TINCR
120670                IF CVALU><0 THEN A:=0 ELSE CLOGU FI
120675                A=:6CTABVALUE
120676                IF LEGAREA BIT BMEMO THEN
120701                    X:="TMRTA"
```

```
120702  LLMEM:            T:=-1; CVALUE; CALL IMTAB; CALL CERR1; GO LIMAG; GO LLMEM
120710                 FI
120712  LIMAG:         IF LEGAREA BIT BIML THEN
120713                    X:="TMRTA"
120714  LLIMAG:           T:=-1; CVALUE; CALL IITAB; CALL CERR1; GO LSAVE; GO LLIMAG
120722                 FI
120722  LSAVE:         IF LEGAREA BIT BSAVE THEN
120725                     CALL 6SOPEN; X:="TMRTA"
120727  LLSAV:             T:=-1; CVALUE; CALL ISTAB; CALL CERR1; GO FINO; GO LLSAV
120735  FINO:   FI; FI; GO SMLEAVE
120736  ERR:    GO 6ERRPAR
120737  CERR1:  IF A><"ETMRT" THEN X:=A+1; EXITA FI
120744  ERRO:   GO 6ERROPER
120745  RBUS
120772
120772
120772  %=========================================================================
120772  % 28.28        6 S O P E N   6 M O P E N   6 C L O S E
120772  %
120772  % SUBROUTINE TO OPEN AND CLOSE THE MACM-AREA OR THE SINTRAN FILE
120772  % EXIT:  FILE NUMBER IN 6CFILNO
120772  %
120772  SUBR 6SOPEN,6MOPEN,6CLOSE
120772  6SOPEN: L=:D; CALL SMENTER
120774          X:="6SINTR"; GO L1
120776  6MOPEN: L=:D; CALL SMENTER
121000          X:="6MACM"
121001  L1:     T:=2; "6DFTYP"; *MON 2NOPE; JMP ERR
121005          A=:6CFILNO=:T; 40; *MON 76; JMP ERR
121012          -1=:6BLCKI; GO SMLEAVE
121015  6CLOSE: T:=6CFILNO; *MON 2CLOS; JMP ERR
121020          -1=:6BLCKI; EXIT
121023  ERR:    *MON 64
121024          GO SSMON
121025  RBUS
121035
121035
121035  %=========================================================================
121035  % 28.29        I N O R O U T
121035  %
121035  % SUBROUTINE TO ASK FOR THE PARAMETER "INPUT OR OUTPUT?"
121035  %
121035  % RETURN: ILLEGAL PARAMETER
121035  % SKIP RETURN: A=0; INPUT
121035  %               A=1; OUTPUT
121035  %
121035  SUBR INOROUT
121035  INTEGER TXINOUT:=(TXI,0,0,0, TXO,1,0,0, -1)
121046  INTEGER TXI:='INPUT',TXO:='OUTPUT'
121055  INOROUT: L=:D; CALL SMENTER
121057          "6PINOUT"; CALL SMSGPAR; GO SSMON
121062          X:="TXINOUT"; CALL SMABLOOK; IF A><0 GO SMLEAVE
121066          T.S1; GO SM2LEAV
121071  RBUS
121101
121101
121101
121101  %=========================================================================
121101  % 28.30        R 1 I M A G E   R 1 S A V E   WR 1 M E M O
121101  %             W 1 I M A G E   W 1 S A V E   W 1 M E M O
```

```
121101   %              W X S A V E   W I M B A C K
121101   %
121101   % SUBROUTINES TO READ OR WRITE ONE WORD FROM/TO THE SAVE-AREA
121101   % OR THE CORE-IMAGE AREA
121101   % ENTRY IF WRITE: A=VALUE
121101   %                  X=ADDRESS
121101   %
121101   % ENTRY IF READ: X=ADDRESS
121101   %
121101   % RETURN IF READ: A=VALUE
121101   %
121101   SUBR R1IMAGE,R1SAVE,W1IMAGE,W1SAVE,WXSAVE,WIMBACK,R1MEMO,W1MEMO
121101   DISP -200; INTEGER CADDR,CVALUE; PSID
121101   INTEGER IMPAR:="5CIMSEG"
121102   R1IMAGE: L=:D; CALL SMENTER; T:=5CIMSEG
121105   L1:     CALL GET1L; GO ERR; GO SMLEAVE
121110   R1MEMO: L=:D; CALL SMENTER; T:=0; GO L1
121114   W1IMAGE: L=:D; CALL SMENTER; T:=5CIMSEG
121117   L2:      CALL PUT1L; GO ERR; GO SMLEAVE
121122   W1MEMO: L=:D; CALL SMENTER; T:=0; GO L2
121126   R1SAVE: L=:D; CALL SMENTER
121130           X=:CADDR
121131           IF CADDR SHZ -5><6BLCKI THEN
121136                 IF 6BLCKI><-1 THEN CALL WXSAVE FI
121143                 CADDR SHZ -5=:6BLOCK=:6BLCKI; T:=6CFILNO; "6RPAR"; *MON 2RFIL
121152                 IF A><0 GO 6ERROPER
121154           FI; CADDR/\37; SSBUF(A); GO SMLEAVE
121161   W1SAVE: L=:D; CALL SMENTER
121163           X=:CADDR; A=:CVALUE
121165           IF CADDR SHZ -5><6BLCKI THEN CALL R1SAVE FI
121173           CADDR/\37; T:=CVALUE=:SSBUF(A); GO SMLEAVE
121201   WXSAVE: 6BLCKI=:6BLOCK; "6WPAR"; T:=6CFILNO; *MON 2WFIL
121206           IF A><0 GO 6ERROPER
121210           -1=:6BLCKI; EXIT
121213   ERR:    GO 6ERROPER
121214   WIMBACK: "IMPAR"; *MON 2WSEG
121216           EXIT
121217   RBUS
121237
121237
121237   %==========================================================================
121237   % 28.31        I I D T A B   R I D T A B   I E I D T A B   R E I D T A B
121237   %
121237   % INSERT-IN-IDENT-TABLE <LEVEL> <LOG.UNIT> <INPUT/OUTPUT> <IDENT>
121237   % REMOVE-FROM-IDENT-TABLE <LEVEL> <LOG.UNIT> <INPUT/OUTPUT> <IDENT>
121237   % INSERT-IN-EXTENDED-IDENT-TABLE <LEVEL> <LOG.UNIT> <I/O> <IDENT>
121237   % REMOVE-FROM-EXTENDED-IDENT-TABLE <LEVEL> <LOG.UNIT> <I/O> <IDENT>
121237
121237   SUBR IIDTAB,RIDTAB,IEIDTAB,REIDTAB
121237
121237   DISP -200
121237   INTEGER ROUFLG,CLEVEL,INOUT,CIDENT,LEGAREA,CVALUE,CLOGU,CTABADDR
121237   INTEGER VAL1,VAL2,IVAL1,IVAL2
121237   PSID
121237
121237   INTEGER 6PLEVEL:='IOLEVEL: ',6PIDENT:='IOIDENT: '
121251   IIDTAB: A:=0; GO L1
121253   RIDTAB: A:=1; GO L1
121255   IEIDTAB: A:=2; GO L1
121257   REIDTAB: A:=3
```

```
121260   L1:     L=:D; CALL SMENTER
121262           A=:ROUFLG; "6PLEVEL"; CALL SMSGPAR; GO SSMON; A=:CLEVEL
121267           "6PLOGU"; CALL SMSGPAR; GO SSMON; A=:CLOGU
121273           CALL INOROUT; GO ERR; A=:INOUT
121276           "6PIDENT"; CALL SMSGPAR; GO SSMON; A=:CIDENT
121302           IF CLEVEL-12<0 OR A>3 GO ERR; A=:CLEVEL
121311           CLOGU; CALL LOGPH; A=:X
121314           IF INOUT=0 THEN
121316                 IF X=0 GO ERR; X=:CLOGU
121320           ELSE
121321                 IF D=0 GO ERR; A:=D=:CLOGU
121325           FI; 7; CALL SELAREA; A=:LEGAREA
121330           IF A=0 GO SMLEAVE
121332           ROUFLG GOSW IIRI,IIRI,IEIREI,IEIREI
121340   *)FILL
121352   ERR:    GO 6ERRPAR
121353   IIRI:   IF CIDENT<0 OR A>6IDMAX(CLEVEL) GO ERR
121361           IF ROUFLG=0 THEN CLOGU ELSE A:=0 FI; A=:CVALUE
121367           6IDTAB(X)+CIDENT-1=:X
121373           IF LEGAREA BIT BMEMO THEN CVALUE; CALL W1MEMO FI
121400           IF LEGAREA BIT BIML THEN CVALUE; CALL W1IMAGE; CALL WIMBACK FI
121406           IF LEGAREA BIT BSAVE THEN
121411              CALL 6MOPEN; CVALUE; CALL W1SAVE; CALL WXSAVE
121415           FI; GO SMLEAVE
121416   *)FILL
121430   IEIREI: 2=:6TINCR
121432           IF ROUFLG=2 THEN
121436                 0=:VAL1=:VAL2; CIDENT=:IVAL1; CLOGU=:IVAL2
121444           ELSE
121445                 CIDENT=:VAL1; CLOGU=:VAL2; 0=:IVAL1=:IVAL2
121453           FI; X:=6EIDTAB(CLEVEL)=:CTABADDR; VAL1=:6CTABVALUE
121460           IF LEGAREA BIT BMEMO THEN
121463   LL1:          T:=-1; VAL1; CALL IMTAB; GO ERR1; X:=A+1; CALL R1MEMO
121471                 IF A=VAL2 THEN
121474                    IVAL2; CALL W1MEMO; X-1; IVAL1; CALL W1MEMO
121501                 ELSE X+1; GO LL1
121504           FI; FI
121504           IF LEGAREA BIT BIML THEN
121507              X:=CTABADDR
121510   LL2:       T:=-1; VAL1; CALL IITAB; GO ERR1; X:=A+1; CALL R1IMAG
121516              IF A=VAL2 THEN
121521                 IVAL2; CALL W1IMAGE; X-1; IVAL1; CALL W1IMAGE; CALL WIMBACK
121527              ELSE X+1; GO LL2
121532           FI; FI
121532           IF LEGAREA BIT BSAVE THEN
121535                 CALL 6SOPEN; X:=CTABADDR
121537   LL3:          T:=-1; VAL1; CALL ISTAB; GO ERR1; X:=A+1; CALL R1SAVE
121545                 IF A=VAL2 THEN
121550                    IVAL2; CALL W1SAVE; X-1; IVAL1; CALL W1SAVE; CALL WXSAVE
121556                 ELSE X+1; GO LL3
121561           FI; FI; GO SMLEAVE
121562   ERR1:   GO 6ERROPER
121563   RBUS
121604
121604
121604
121604   %=========================================================================
121604   % 28.32        A D U M P   O D U M P
121604   %
121604   % ASCII-DUMP <AREA> (<SEG. NO.>) <LOWER ADDR> <UPPER ADDR> (<OUTPUT FILE>)
```

```
121604   % OCTAL-DUMP <AREA> (<SEG. NO.>) <LOWER ADDR> <UPPER ADDR> (<OUTPUT FILE>)
121604
121604   SUBR ADUMP,ODUMP
121604
121604   DISP -200
121604   INTEGER ROUFLG,CAREA,LLIM,ULIM,CLLIM,CULIM,CSETADR,CSEGN,COUNT
121604   INTEGER CDEV=COUNT,CWORD=ULIM
121604   PSID
121604   INTEGER PLIM:='IOLOWER ADDRESS: ',PULIM:='IOUPPER ADDRESS: '
121626
121626   ADUMP: A:=0; GO L1
121630   ODUMP: A:=1
121631   L1:    L=:D; CALL SMENTER
121633          A=:ROUFLG
121634          CALL ASKAREA; GO 6ERRPAR; A=:CAREA
121637          IF A=3 THEN
121642                "6PSEGNO"; CALL SMSGPAR; GO SSMON; A=:CSEGN
121646                CALL LEGSCHECK; GO 6ERRPAR
121650                X.LOGADR/\77 SH 12=:LLIM; X.LOGADR SHZ -10 SH 12+LLIM=:ULIM
121661                IF X.LOGADR SHZ -10><0 THEN ULIM-1=:ULIM FI
121667                X=:CSETADR
121670          FI; "PLIM"; CALL SMSGPAR; GO SSMON; A=:CLLIM
121674          "PULIM"; CALL SMSGPAR; GO SSMON; A=:CULIM
121700          CALL 6OOPEN; GO SMLEAVE; A=:CDEV
121703          IF CAREA=0 THEN 0=:LLIM; -1=:ULIM
121710          ELSE IF A=1 THEN 0=:LLIM; 175777=:ULIM
121717          ELSE IF A=2 THEN 0=:LLIM; 175777=:ULIM
121726          FI; FI; FI
121726          IF CLLIM-LLIM<<0 OR ULIM-CULIM<<0 GO ERR3
121736          IF CULIM-CLLIM<<0 GO ERR3
121742          CDEV=:TDVN; GO L2; *)FILL
121763   ERR3:  "6TILBOUNDS"; CALL SMOUTTEXT; GO SSMON
121766   ERR4:  GO 6ERROPER
121767
121767   L2:    IF CAREA=2 THEN CALL 6SOPEN FI; CALL ESCON
121775          IF ROUFLG><0 GO LODUMP
121777          DO
121777                CALL SMCRLF; CLLIM; CALL SMOCTU; #: ; CALL SM2TCO; -40=:COUNT
122006                FOR COUNT DO
122006                     X:=CLLIM
122007                     IF CAREA=0 THEN CALL R1MEMO
122012                     ELSE IF A=1 THEN CALL R1IMAGE
122017                     ELSE IF A=2 THEN CALL R1SAVE
122024                     ELSE T:=CSEGN; CALL GET1L; GO 6ERROPER
122030                     FI; FI; FI; A=:CWORD; A SHZ -10; CALL SMTCO
122033                     IF CWORD SHZ -10/\177=15 THEN
122041                          CALL SMCRLF; CLLIM; CALL SMOCTU; #: ; CALL SM2TCO
122046                          -40=:COUNT
122050                     FI; CWORD/\377; CALL SMTCO
122053                     IF CLLIM=CULIM GO OUT; MIN CLLIM
122060                     IF CWORD/\177=15 THEN
122065                          CALL SMCRLF; CLLIM; CALL SMOCTU; #: ; CALL SM2TCO
122072                          -40=:COUNT
122074                     FI
122074                OD
122076          OD
122077   LODUMP: DO
122077                CALL SMCRLF; CLLIM; CALL SMOCTU; #: ; CALL SM2TCO; -10=:COUNT
122106                FOR COUNT DO
122106                     X:=CLLIM
```

```
122107                    IF CAREA=0 THEN CALL R1MEMO
122112                    ELSE IF A=1 THEN CALL R1IMAGE
122117                    ELSE IF A=2 THEN CALL R1SAVE
122124                    ELSE T:=CSEGN; CALL GET1L; GO 6ERROPER
122130                    FI; FI; FI; CALL SMOCTU
122131                    IF CLLIM=CULIM GO OUT; MIN CLLIM
122136                 OD
122140           OD
122141   OUT:    CALL ESCOFF; GO SMLEAVE
122143   RBUS
122166
122166
122166   %=================================================================
122166   % 28.33       I B A C K   R B A C K
122166   %
122166   % INSERT-IN-BACKGROUND-TABLE <LOG. UNIT> <INPUT/OUTPUT>'
122166   % REMOVE-FROM-BACKGROUND-TABLE <LOG. UNIT> <INPUT/OUTPUT>'
122166
122166   SUBR IBACKT,RBACKT
122166
122166   DISP -200; INTEGER ROUFLG,CVALUE,CLOGU,INOUT,LEGAREA,CADDR,DYNAL; PSID
122166
122166   INTEGER PROACTIVE:='$PROGRAM IS ACTIVE'
122200   IBACKT: A:=0; GO L1
122202   RBACKT: A:=1
122203   L1:     L=:D; CALL SMENTER
122205           A=:ROUFLG
122206           "6PLOGU"; CALL SMSGPAR; GO SSMON; A=:CLOGU
122212           CALL INOROUT; GO FAR ERR; A=:INOUT; CLOGU; CALL LOGPH
122217           IF T:=INOUT><0 THEN A:=D FI; IF A=0 GO FAR ERR; A=:CLOGU
122226           IF ROUFLG><0 THEN CLOGU FI; A=:CVALUE
122232           7; CALL SELAREA; A=:LEGAREA
122235           IF A=0 GO SMLEAVE
122237                 1=:6TINCR
122241                 IF LEGAREA BIT BMEMO THEN
122244                    X:="BACKT"; T:=-1; CVALUE; CALL FFMTAB; GO FAR ERR; A=:CADDR
122252                    IF ROUFLG><0 THEN
122254                       X:=CADDR.SO; 0=:DYNAL; CALL SMDYNALLOC; GO NODY1; MIN DYNAL
122262   NODY1:              T:="DBPROG"; CALL XGTDFADDR
122264                       IF A><0 THEN
122265                          IF T:=DYNAL><0 THEN CALL SMTACTIVE; GO NOAC1; GO AC1 FI
122273                          IF A.WLINK><0 THEN
122276   AC1:                      "PROACTIVE"; CALL SMOUTTEXT; GO SMLEAVE; *)FILL
122323                          FI
122323                       ELSE A=:X FI
122325                    ELSE
122326                       X:=CLOGU; T:="DBPROG"; CALL XGTDFADDR; A=:X
122332                    FI; IF X=0 GO FAR ERR; X.ACTPRI
122335                    IF T:=ROUFLG><0 THEN A BONE 5RTOF ELSE A BZERO 5RTOF FI
122343                    A=:X.ACTPRI
122344   NOAC1:           IF ROUFLG=0 THEN CLOGU ELSE A:=0 FI; A=:CADDR.SO
122353                 FI; GO FILL1; *)FILL
122357   FILL1:        IF LEGAREA BIT BIML THEN
122362                    T:=-1; CVALUE; X:="BACKT"; CALL FFITAB; GO ERR; A=:CADDR
122370                    X:=CLOGU; CALL SMDYNALLOC; GO NODY2; GO DYNA2
122374   NODY2:           IF ROUFLG><0 THEN
122376                       X:=CADDR; CALL R1IMAGE
122400                    ELSE CLOGU
122402                    FI
122402
```

```
122402   @LIB CXCPU-,
122402   @LIB CXCPU
                      IF T:=CLOGU.TYPRING BIT 5TERM THEN A+"ZDBPROG" ELSE A+"DBPROG" FI
122402                A=:X; CALL R1IMAGE
122411
122413   @ELIB
                      A+"ACTPRI"=:X; CALL R1IMAGE
122413                IF T:=ROUFLG><0 THEN A BONE 5RTOF ELSE A BZERO 5RTOF FI
122416                CALL W1IMAGE
122424
122425   DYNA2:       X:=CADDR
122426                IF ROUFLG=0 THEN CLOGU ELSE A:=0 FI; CALL W1IMAGE
122434                CALL WIMBACK
122435             FI
122435             IF LEGARE BIT BSAVE THEN
122440                CALL 6SOPEN
122441                T:=-1; CVALUE; X:="BACKT"; CALL FFSTAB; GO ERR; A=:CADDR
122447                X:=CLOGU; CALL SMDYNALLOC; GO NODY3; GO DYNA3
122453   NODY3:       IF ROUFLG><0 THEN
122455                   X:=CADDR; CALL R1SAVE
122457                ELSE CLOGU
122461                FI
122461   @LIB CXCPU-,
122461   @LIB CXCPU
122461                IF T:=CLOGU.TYPRING BIT 5TERM THEN A+"ZDBPROG" ELSE A+"DBPROG" FI
122470                A=:X; CALL R1SAVE
122472   @ELIB
122472                A+"ACTPRI"=:X; CALL R1SAVE
122475                IF T:=ROUFLG><0 THEN A BONE 5RTOF ELSE A BZERO 5RTOF FI
122503                CALL W1SAVE
122504   DYNA3:       X:=CADDR
122505                IF ROUFLG=0 THEN CLOGU ELSE A:=0 FI; CALL W1SAVE; CALL WXSAVE
122514             FI
122514          GO SMLEAVE
122515   ERR:   GO 6ERRPAR
122516   RBUS
122536
122536   %=================================================================
122536   % 28.34        D U S S U B R
122536   %
122536   % DEFINE-USER-START-SUBROUTINE <ADDRESS>
122536
122536   SUBR DUSSUBR
122536   DISP -200; INTEGER CADDR,LEGAREA; PSID
122536   DUSSUBR: L=:D; CALL SMENTER
122540          "6PADDR"; CALL SMSGPAR; GO SSMON; A=:CADDR
122544          6; CALL SELAREA; A=:LEGAREA
122547          IF A><0 THEN
122550                IF A BIT BIML THEN
122552                   X:="USTART"; CADDR; CALL W1IMAGE; CALL WIMBACK
122556                FI
122556                IF LEGAREA BIT BSAVE THEN
122561                     CALL 6SOPEN; X:="USTART"; CADDR; CALL W1SAVE; CALL WXSAVE
122566                FI
122566          FI; GO SMLEAVE
122567   RBUS
122603
122603   %=================================================================
122603   % 28.35        D U R S U B R
122603   %
```

```
122603   % DEFINE-USER-RESTART-SUBROUTINE <ADDRESS>
122603
122603   SUBR DURSUBR
122603   DISP -200; INTEGER CADDR,LEGAREA; PSID
122603   DURSUBR: L=:D; CALL SMENTER
122605           "6PADDR"; CALL SMSGPAR; GO SSMON; A=:CADDR; 7; CALL SELAREA; A=:LEGAREA
122614           IF A><0 THEN
122615                 IF A BIT BMEMO THEN CADDR=:URESTART FI
122621                 IF LEGAREA BIT BIML THEN
122624                    X:="URESTART"; CADDR; CALL W1IMAGE; CALL WIMBACK
122630                 FI
122630                 IF LEGAREA BIT BSAVE THEN
122633                      CALL 6SOPEN; X:="URESTART"; CADDR; CALL W1SAVE; CALL WXSAVE
122640                 FI
122640           FI; GO SMLEAVE
122641   RBUS
122655
122655
122655   %==========================================================================
122655   % 28.36       D U R P R O
122655   %
122655   % DEFINE-USER-RESTART-PROGRAM <PROGRAM>
122655
122655   SUBR DURPRO
122655   DISP -200; INTEGER CADDR,LEGAREA; PSID
122655   DURPRO: L=:D; CALL SMENTER
122657           "6PRTNAM"; CALL SMSGPAR; GO SSMON; A=:CADDR
122663           IF A><0 THEN CALL CHRTDES; GO 6ERRPAR FI
122666           7; CALL SELAREA; A=:LEGAREA
122671           IF A><0 THEN
122672                 IF A BIT BMEMO THEN CADDR=:URPRO FI
122676                 IF LEGAREA BIT BIML THEN
122701                    X:="URPRO"; CADDR; CALL W1IMAGE; CALL WIMBACK
122705                 FI
122705                 IF LEGAREA BIT BSAVE THEN
122710                    CALL 6SOPEN; X:="URPRO"; CADDR; CALL W1SAVE; CALL WXSAVE
122715                 FI
122715           FI; GO SMLEAVE
122716   RBUS
122734
122734   @LIB OLD
122734
122734   %==========================================================================
122734   % 28.37B      D F P R M T
122734   %
122734   % DEFINE-PROMPT-STRING <TEXT STRING>
122734   % DEFINE-TITLE <TEXT STRING>
122734
122734   SUBR DFPRMT,DVSTRING
122734
122734   DISP -200; INTEGER CDBYTP,LEGAREA,CCHAR,INLOCAL,ROUTSWITCH,CADDR; PSID
122734
122734   INTEGER TDVTITLE:='TITLE: '
122740   INTEGER TDVSTRING:='STRING: '
122745   INTEGER CQUEST:='N PROMPT STRING IN LOCAL MODE? '
122765   INTEGER PWSCB:="50PSEG"
122766
122766   STCHAR: A=:D
122767           IF CDBYTP=41 THEN "6TILLINE"; CALL SMOUTTEXT; GO SSMON FI
122776           T:="SMFIBUF"; X:=CDBYTP; A:=D; *SBYT
```

```
123002          MIN CDBYTP; EXIT; *)FILL
123011
123011   DVSTRING: L=:D; CALL SMENTER
123013          0=:ROUTSWITCH; A:="USTXT"; GO FELLS
123016   DFPRMT: L=:D; CALL SMENTER
123020          1=:ROUTSWITCH; A:="USTX4"
123023   FELLS: A=:CADDR; 0=:CDBYTP=:INLOCAL; CALL SMCREAD
123027          IF A=15 THEN
123032              IF ROUTSWITCH><0 THEN "TDVSTRING" ELSE "TDVTITLE" FI
123037              CALL SMOUTTEXT
123040          ELSE
123041              DO CALL SMCREAD WHILE A><15 AND A><##'; CALL STCHAR OD
123052              IF A=##' THEN CALL STCHAR; GO L2 FI
123057          FI
123057          DO
123057                CALL SMGCOM
123060                DO CALL SMCREAD WHILE A><15 AND A><##'; CALL STCHAR OD
123071                IF A=##' THEN CALL STCHAR; GO L2 FI
123076          OD
123077   L2:    IF ROUTSWITCH><0 THEN
123101             "CQUEST"; CALL SMSGPAR; GO SSMON; CALL SMYESNO; A=:INLOCAL
123106          FI; 5; CALL SELAREA; A=:LEGAREA
123111          IF A><0 THEN
123112                IF A BIT BMEMO THEN 0=:CDBYTP
123115                   DO
123115                        T:="SMFIBUF"; X:=CDBYTP; *LBYT
123120                        A=:CCHAR; CDBYTP SHZ -1+CADDR=:X
123125                        T:=50PSEG; CALL GET1L; GO L3
123130                        IF T:=CDBYTP BIT "0" THEN A/\177400\/CCHAR
123135                        ELSE A/\377; T:=CCHAR SH 10; A\/T
123142                        FI;  T:=50PSEG; CALL PUT1L; GO L3
123145                        IF CCHAR=##' THEN GO L3 FI; MIN CDBYTP
123153                   OD
123154   L3:             IF ROUTSWITCH><0 THEN
123156                      IF INLOCAL=1 THEN "USTX4" ELSE "USTX5" FI
123165                      X:="LUSTX"; T:=50PSEG; CALL PUT1L; GO L31
123171                   FI; "PWSCB"; *MON 2WSEG
123173                FI; GO L31; *)FILL
123221   L31:         IF LEGAREA BIT BSAVE THEN
123224                   CALL 6MOPEN; 0=:CDBYTP            % MACM-AREA
123226                   DO
123226                      X:=CDBYTP; T:="SMFIBUF"; *LBYT
123231                      A=:CCHAR; CDBYTP SHZ -1+CADDR-BCSTA=:X; CALL R1SAVE
123240                      IF T:=CDBYTP BIT "0" THEN A/\177400\/CCHAR
123245                      ELSE A/\377; T:=CCHAR SH 10; A\/T
123252                      FI; CALL W1SAVE; IF CCHAR=##' GO L4
123257                      MIN CDBYTP
123260                  OD
123261   L4:            IF ROUTSWITCH><0 THEN
123263                     IF INLOCAL=1 THEN X:="USTX4" ELSE X:="USTX5" FI
123272                     "LUSTX"-BCSTA; X:=:A; CALL W1SAVE
123276                  FI; CALL WXSAVE
123277              FI
123277          FI
123277   OUT:   GO SMLEAVE
123300   RBUS
123314
123314
123314   %=========================================================================
123314   %              S M G F I N F O
```

```
123314   %
123314   %  SUBROUTINE TO COLLECT INFORMATION ABOUT A FILE
123314   %
123314   % ENTRY:      A=FILE TYPE POINTER
123314   %             X=FILE NAME POINTER
123314   %             T=ACCESS CODE USED IN OPEN
123314   %
123314   % EXIT:       ERROR
123314   %
123314   % EXIT+1:     THE INFORMATION ABOUT THE FILE IS SAVED IN THE GLOBAL ARRAY 9FINFO
123314   %
123314
123314   INTEGER ARRAY 9FINFO(0)
123314   INTEGER 9CABLPAGE                        % NUMBER OF SECTORS PER PAGE
123315   INTEGER 9CLOGU                           % LOGICAL DEVICE NUMBER OF FILE DIRECTORY DEVICE
123316   INTEGER 9CUNIT                           % LOGICAL DEVICE UNIT (DRIVE NUMBER)
123317   INTEGER 1CFSTART,2CFSTART
123321           DOUBLE 9CFSTART=1CFSTART         % START ADDR OF FILE IN DISC ADDRESS
123321   INTEGER 1CFSIZE,2CFSIZE
123323           DOUBLE 9CFSIZE=1CFSIZE           % SIZE OF FILE IN PAGES
123323   INTEGER 9CCDFELT
123324   INTEGER 9CFNO
123325
123325   SUBR SMGFINFO
123325
123325   DISP -200
123325   INTEGER CLUNIT,CSUBDIRS,1CDBASE,2CDBASE,CCNAMTA,NEGFLG
123325   PSID
123325
123325   64ERR: *MON 64
123326          GO LEAVE
123327
123327   SMGFINFO: L=:D; CALL SMENTER
123331          *MON 2NOPE
123332          GO 64ERR; A=:9CFNO
123334          CALL LOGPH; IF A=0 THEN CALL ERRFATAL FI
123337          IF A.OFFTP NBIT OABIT AND A NBIT OCBIT THEN
123345             ER209; *MON 64
123347             GO SMLEAVE
123350          FI; AD:=X.OFFP=:9CFSTART; AD:=X.OFPAG=:9CFSIZE
123354          X.OFDIR*DTLEN+"DIRTA"+"DUNIT"=:X                    % GET DUNIT FROM DIRECTORY ENTRY.
123361          T:=5FILSEG; CALL GET1L; CALL ERRFATAL
123364          A=:D/\7777=:9CLOGU; A:=D SHZ -14=:9CUNIT; A:=X+"LUNIT"-"DUNIT"=:X
123376          CALL GET1L; CALL ERRFATAL; A=:CLUNIT
123401          A/\377*NTLEN+"NAMTA"=:CCNAMTA+"SECTO"=:X
123407          CALL GET1L; CALL ERRFATAL; A=:T; A:=2000=:D:=0; *RDIV ST
123416          IF D><0 THEN CALL ERRFATAL FI; A=:9CABLPAGE
123422          A:=X+"NFLAG"-"SECTO"=:X; T:=5FILSEG; CALL GET1L; CALL ERRFATAL
123431          A=:NEGFLG; 9CLOGU; CALL LOGPH; IF A=0 GO 6ERRPAR; A=:9CCDFELT
123437          IF A.TYPRING BIT 5FLOP GO L2
123443          IF NEGFLG NBIT CTBIT AND CLUNIT SHZ -11/\7><0 THEN
123452             A=:CSUBDIRS
123453             CCNAMTA+"PAVAI"; T:=5FILSEG; CALL GET1L; CALL ERRFATAL; A=:1CDBASE
123461             X+1; CALL GET1L; CALL ERRFATAL; A=:2CDBASE=:D:=1CDBASE; T:=CSUBDIRS
123470             DO WHILE T-1><0
123473                A:=:D+2CDBASE; A:=:D; A:=A+C+1CDBASE
123500             OD; A:=:D+2CFSTART; A:=:D; A:=A+C+1CFSTART; AD=:9CFSTART
123507          FI; GO L2; *)FILL
123544
123544   L2:    IF NEGFLG BIT 5PHOENIX THEN CLUNIT/\7000 SHZ -6+9CUNIT=:9CUNIT FI
```

```
123554          IF 9CABLPAGE=1 THEN AD:=9CFSTART
123561          ELSE IF A=2 THEN AD:=9CFSTART SH 1
123567          ELSE IF A=4 THEN AD:=9CFSTART SH 2
123575          ELSE IF A=10 THEN AD:=9CFSTART SH 3
123603          ELSE IF A=20 THEN AD:=9CFSTART SH 4
123611          ELSE CALL ERRFATAL
123613          FI; FI; FI; FI; FI
123613          IF T:=NEGFLG BIT CTBIT AND T:=CLUNIT BIT 10 THEN D BONE 17 FI
123622          AD=:9CFSTART
123623          GO SM2LEAV
123624   RBUS
123632
123632
123632   %=================================================================
123632   % 28.38      D E F S G F I   D L S G F I
123632   %
123632   % DEFINE-SEGMENT-FILE <AREA> <SEGMENT FILE NO.> <SEGMENT FILE NAME>
123632   % DELETE-SEGMENT-FILE <AREA> <SEGMENT FILE NO.>
123632   %
123632
123632   INTEGER ARRAY 5BUFR=?
123632
123632   SUBR DEFSGFI,DLSGFI
123632
123632   DISP -200
123632   INTEGER CUNIT
123632   INTEGER CSGFINO                          % SEGMENT FILE NUMBER.
123632   INTEGER CABLPAGE                         % NUMBER OF SECTORS PER 1K PAGE ON THE DISK.
123632   INTEGER ROUFLG                           % ROUTINE FLAG: =0, DEFINE-SEG-FILE; =1, DELETE-SEG-FILE
123632   INTEGER CAREA
123632   INTEGER CLOGU                            % LOGICAL DEVICE NUMBER OF DISK ON WHICH SEGMENT FILE IS DEFINED.
123632   INTEGER CFISTART                         % FIRST DISK PAGE IN THE SEGMENT FILE
123632   INTEGER CFISIZE                          % NUMBER OF PAGES IN THE SEGMENT FILE
123632   INTEGER CINDEX
123632   INTEGER CLUNIT                           % LOGICAL SUB-UNIT AND NAME INDEX OF DIRECTORY.
123632   INTEGER NEGFLG
123632   INTEGER FINAM
123632   INTEGER CADDR=ROUFLG
123632   INTEGER CADMOD=FINAM
123632   PSID
123632
123632   INTEGER SGFINO:='IOSEGMENT FILE NO: '
123644   INTEGER SGFNAM:='S SEGMENT FILE NAME: '
123657   INTEGER SGFADEF:='$SEGMENT FILE ALREADY DEFINED '
123677   INTEGER XMEM:='IN MEMORY$',XSAV:='IN SAVE-AREA$'
123714   INTEGER REDEF:='N REDEFINE SEGMENT FILE? '
123731   INTEGER ILLFILE:='$NOT CONTIGUOUS FILE'
123744   INTEGER RTLINU:='$RT-LOADER IS IN USE$'
123757   INTEGER PWRTLB:="5RT2SG",PWRT2LB:="5IRTSG"
123761   INTEGER CET1:='$ILLEGAL SEGMENT FILE NUMBER (LEGAL RANGE 0-3)$'
124011   +)FILL
124013
124013   ASKRDF: A:=L=:CINDEX                                                    % ASK : REDEFINE
124015          "SGFADEF"; CALL SMOUTTEXT; IF CAREA BIT BMEMO THEN "XMEM" ELSE "XSAV" FI
124025          CALL SMOUTTEXT; "SGFNAM+1"; CALL SMOUTTEXT
124030          "SSBUF"+40; CALL SMOUTTEXT; CALL SMCRLF
124034          IF BCHFLAG=0 THEN
124036             "REDEF"; CALL SMSGPAR; GO SSMON
124041             CALL SMYESNO; IF A><1 GO SMLEAVE
124045          FI; CINDEX=:L; EXIT
```

```
124050   *)FILL
124064
124064   INTEGER TWARN:='$FILESIZE IS GREATHER THAN 16383 PAGES,$ONLY THE FIRST 16383 PAGES CAN BE USED FOR SEGMENT
124143   DLSGFI: A:=1; GO FELLS
124145   DEFSGFI: A:=0
124146   FELLS: L=:D; CALL SMENTER; A=:ROUFLG
124151          5; CALL SELAREA; A=:CAREA; IF A=0 GO SMLEAVE
124156          "SGFINO"; CALL SMSGPAR; GO SSMON; A=:CSGFINO          % SEGMENT FILE NUMBER.
124162          IF ROUFLG><0 GO L1
124164          "SGFNAM"; CALL SMSGPAR; GO SSMON; A=:FINAM            % SEGMENT FILE NAME.
124170          FOR X:=0 TO 100 DO
124174                T:=FINAM;    *LBYT
124176                T:="SSBUF"; *SBYT
124200                IF A/\177=##' GO L1
124204          OD
124206          GO 6ERRPAR
124207   *)FILL
124221   CFILERR: *MON 64
124222          GO SMLEAVE
124223
124223   L1:    IF CSGFINO<0 OR A>3 GO FAR ERR
124230          IF ROUFLG><0 GO FAR DLSG1
124233          X:="SSBUF"; "6DFTYP"; T:=1; CALL SMGFINFO; GO SSMON
124240   @LIB OLD
124240          AD:=9CFSIZE
124241          IF A><0 OR D>>37777 THEN T=:2CFSIZE; "TWARN"; CALL SMOUTTEXT FI
124250          GO BYPAS; *)FILL
124265   BYPAS: IF CAREA BIT BMEMO THEN
124270                XSGFNAM+CSGFINO=:X; T:=5RT2SG; CALL GET1L; CALL ERRFATAL
124276                IF A><0 THEN
124277                   X+4; CALL GET1L; CALL ERRFATAL; A=:X; 40=:CINDEX
124305                   DO WHILE CINDEX<100
124311                      T:=5RT2SG; CALL GET1L; CALL ERRFATAL; X=:D; A=:SSBUF(CINDEX)
124317                      MIN CINDEX; D+1=:X
124322                   OD; CALL FAR ASKRDF
124324                FI
124324                "6RTLRES"; *MON 2RESR
124326                IF A><0 THEN "RTLINUSE"; CALL SMOUTTEXT; GO SSMON FI
124332                CSGFINO SH 1+"BLST"=:X; 1CFSTART=:X.SO; CALL W1IMAGE
124341                X+1; 2CFSTART=:X.SO; CALL W1IMAGE
124345                "ABLPAGE"+CSGFINO=:X; 9CABLPAGE=;X.SO; CALL W1IMAGE
124353                "MASSNO"+CSGFINO=:X; 9CLOGU=;X.SO; CALL W1IMAGE
124361                "MASSUNIT"+CSGFINO=:X; 9CUNIT=;X.SO; CALL W1IMAGE
124367                9CCDFELT+"TYPRING"=;X; X.SO/\177774+3=:X.SO
124376                CALL R1IMAGE; A/\177774+3; CALL W1IMAGE
124402                "SIZF"+CSGFINO=:X; 2CFSIZE=:X.SO; CALL W1IMAGE; CALL WIMBACK
124411                0=:CINDEX; XSGFNAM+CSGFINO+4=:X
124416                T:=5RT2SG; CALL GET1L; CALL ERRFATAL; A=:X
124422                DO WHILE CINDEX<40
124426                   X=:D; SSBUF(CINDEX); X:=D; T:=5RT2SG; CALL PUT1L; CALL ERRFATAL
124435                   X+1; MIN CINDEX
124437                OD; XSGFNAM+CSGFINO=:X; 1; T:=5RT2SG; CALL PUT1L; CALL ERRFATAL
124447          FI; "PWRTLB"; *MON 2WSEG
124451          GO OUT1
124452   *)FILL
124506   ERR:   "CET1"; CALL SMOUTTEXT; GO SSMON
124511   ERR2:  "ILLFILE"; CALL SMOUTTEXT; GO SSMON
124514   OUT1:  IF CAREA NBIT BSAVE THEN GO SMLEAVE FI
124520          A BZERO BMEMO=:CAREA
124522          "6PRTFIL"; *MON 2RESR
```

```
124524          FOR X:=0 TO 40 DO SSBUF(X)=:5BUFR(X); OD
124534          IF 6XALT><0 GO FAR DEFALT
124537          5IRTSG*5SEGSIZE+SEGSTART
124542          A.LOGADR/\77 SH 12=:CADMOD+CSGFINO+XSGFNAM=:CADDR=:X
124553          T:="5IRTSG"; CALL GET1L; CALL ERRFATAL
124556          IF A><0 THEN
124557                X+4; T:="5IRTSG"; CALL GET1L; CALL ERRFATAL
124563                A+CADMOD=:X; 40=:CINDEX
124567                DO WHILE CINDEX<100
124573                   T:="5IRTSG"; CALL GET1L; CALL ERRFATAL
124576                   X=:D; A=:5BUFR(CINDEX); MIN CINDEX; D+1=:X
124604                OD
124605                FOR X:=40 TO 77 DO 5BUFR(X)=:SSBUF(X); OD
124615                CALL FAR ASKRDF
124616          FI
124616          X:=CADDR+4=:X; T:="5IRTSG"; CALL GET1L; CALL ERRFATAL
124624          A+CADMOD=:X; 0=:CINDEX
124627          DO WHILE CINDEX<40
124633                X=:D; 5BUFR(CINDEX); X:=D; T:="5IRTSG"; CALL PUT1L; CALL ERRFATAL
124642                X+1; MIN CINDEX
124644          OD
124645          X:=CADDR; 1; T:="5IRTSG"; CALL PUT1L; CALL ERRFATAL
124652          "PWRT2LB"; *MON 2WSEG
124654          GO L3; *)FILL
124700
124700   L3:    CALL 6SOPEN
124701          CSGFINO SH 1+"BLST"=:X; 1CFSTART; CALL W1SAVE
124707          X+1; 2CFSTART; CALL W1SAVE
124712          "SIZF"+CSGFINO=:X; 2CFSIZE; CALL W1SAVE
124717          "ABLPAGE"+CSGFINO=:X; 9CABLPAGE; CALL W1SAVE
124724          "MASSNO"+CSGFINO=:X; 9CLOGU; CALL W1SAVE
124731          9CCDFELT+"TYPRING"=:X; CALL R1SAVE; A/\177774+3; CALL W1SAVE
124740          "MASSUNIT"+CSGFINO=:X; 9CUNIT; CALL W1SAVE; CALL WXSAVE
124746          GO SMLEAVE
124747   *)FILL
124772
124772   DEFALT: CALL ERRFATAL
124773
124773   DLSG1: CSGFINO+XSGFNAM=:X=:CADDR
124777          IF CAREA BIT BMEMO THEN
125002                "6RTLRES"; *MON 2RESR
125004                IF A><0 THEN "RTLINUSE"; CALL SMOUTTEXT; GO SSMON FI
125010                A:=0; T:=5RT2SG; CALL PUT1L; CALL ERRFATAL
125014                "PWRTLB"; *MON 2WSEG
125016                IF CSGFINO><0 THEN
125020                   A SH 1+"BLST"=:X; 0=:X.S0; A:=0; CALL W1IMAGE
125026                   X+1; 0=:X.S0; A:=0; CALL W1IMAGE
125032                   CSGFINO+"SIZF"=:X
125035                   A:=0; A=:X.S0; CALL W1IMAGE
125040                   "MASSNO"+CSGFINO=:X; A:=0; CALL W1IMAGE; T:=X.S0; 0=:X.S0
125047                   IF "MASSNO".S0><T AND X.S1><T AND X.S2><T AND X.S3><T THEN
125064                      A:=T; CALL LOGPH
125066                      IF A><0 THEN
125067                         A+"TYPRING"=:X; X.S0 BZERO "0"=:X.S0
125074                         CALL R1IMAGE; A BZERO "0"; CALL W1IMAGE
125077                      FI
125077                   FI; CALL WIMBACK
125100                FI
125100          FI; GO DLSG2; *)FILL
125121
```

```
125121   INTEGER CCLOGU
125122   DLSG2: IF CAREA BIT BSAVE THEN
125125             5IRTSG*5SEGSIZE+SEGSTART
125130             A.LOGADR/\77 SH 12+CADDR=:X
125136             A:=0; T:="5IRTSG"; CALL PUT1L; CALL ERRFATAL
125142             "PWRT2LB"; *MON 2WSEG
125144             IF CSGFINO><0 THEN
125146                CALL 6SOPEN                          % OPEN SINTRAN:DATA
125147                A SH 1+"BLST"=:X; A:=0; CALL W1SAVE
125154                X+1; A:=0; CALL W1SAVE
125157                CSGFINO+"SIZF"=:X; A:=0; CALL W1SAVE
125164                "MASSNO"+CSGFINO=:X; CALL R1SAVE; A=:CCLOGU
125171                A:=0; CALL W1SAVE
125173                X:="MASSNO"; CALL R1SAVE; IF A=CCLOGU GO DLSG3
125200                X+1; CALL R1SAVE; IF A=CCLOGU GO DLSG3
125205                X+1; CALL R1SAVE; IF A=CCLOGU GO DLSG3
125212                X+1; CALL R1SAVE; IF A=CCLOGU GO DLSG3
125217                CCLOGU; CALL LOGPH
125221                IF A><0 THEN
125222                   A+"TYPRING"=:X; CALL R1SAVE; A/\177776; CALL W1SAVE
125227   DLSG3:       FI; CALL WXSAVE
125230            FI
125230         FI; GO SMLEAVE
125231   RBUS
125252
125252
125252   %=============================================================================
125252   % 28.39       C R B I N
125252   %
125252   % READ-BINARY <AREA> <FILE>
125252
125252   SUBR CRBIN
125252   DISP -200; INTEGER CAREA,CLLIM,CULIM,CDEV,WORDN,CHSUM,CSEGN,LLIM,ULIM; PSID
125252   INTEGER PINFIL:='S INPUT FILE: '
125262   INTEGER TCHERR:='$CHECKSUM ERROR'
125272
125272   RCHAR: T:=CDEV; *MON 2INBT
125274          GO CFILERR; EXIT
125276   RWORD: T:=CDEV; *MON 2INBT
125300          GO CFILERR; A SH 10=:D; *MON 2INBT
125304          GO CFILERR; A\/D; EXIT
125307   ADCHSUM: A=:D+CHSUM=:CHSUM:=D; EXIT
125314
125314   CFILERR: *MON 64
125315          GO SMLEAVE
125316
125316   CRBIN: L=:D; CALL SMENTER
125320          CALL ASKAREA; GO 6ERRPAR; A=:CAREA
125323          IF A=3 THEN
125326                "6PSEGNO"; CALL SMSGPAR; GO SSMON; A=:CSEGN
125332          FI
125332          "PINFIL"; CALL SMSGPAR; GO SSMON; A=:X:="6BFTYP"; T:=1; *MON 2NOPE
125341          GO CFILERR; A=:CDEV
125343          DO CALL RCHAR WHILE A><##! OD
125350          CALL RWORD; A=:CLLIM; CALL RWORD; A=:T
125354          A-1+CLLIM; IF C GO ERR2; A=:CULIM; T-=:WORDN; X:=CLLIM; 0=:CHSUM
125365          CAREA GOSW CRMEMO,CRIMAGE,CRSAVE,CRSEG
125373   ERR2:  "6TILBOUNDS"; CALL SMOUTTEXT; GO SSMON
125376   *)FILL
125411
```

```
125411   CRMEMO: FOR WORDN DO
125411                CALL RWORD; CALL ADCHSUM; CALL W1MEMO; X+1
125415           OD
125417   CHCHSUM: CALL RWORD; IF A><CHSUM THEN "TCHERR"; CALL SMOUTTEXT; GO SSMON FI
125426           IF CAREA=2 THEN CALL WXSAVE
125433           ELSE IF A=1 THEN CALL WIMBACK
125440                ELSE IF A=3 THEN "CSEGN"+B=:LLIM; A+1; *MON 2WSGB
125451                     FI
125451                FI
125451           FI; GO SMLEAVE
125452   CRIMAGE: X:="ENDCORE"; CALL R1IMAGE; IF A-CULIM<<0 GO ERR2
125457           X:=CLLIM
125460           FOR WORDN DO
125460                CALL RWORD; CALL ADCHSUM; CALL W1IMAGE; X+1
125464           OD; GO CHCHSUM
125467   CRSAVE: CALL 6SOPEN; X:="ENDCORE"; CALL R1SAVE; IF A-CULIM<<0 GO ERR2
125475           X:=CLLIM
125476           FOR WORDN DO
125476                CALL RWORD; CALL ADCHSUM; CALL W1SAVE; X+1
125502           OD; GO CHCHSUM
125505   CRSEG: CSEGN; CALL LEGSCHECK; GO 6ERRPAR
125510           IF X.LOGADR SHZ -10 SH 12=0 GO ERR2; A=:ULIM
125515           X.LOGADR/\77 SH 12=:LLIM+ULIM-1=:ULIM
125524           IF A-CULIM<<0 OR CLLIM-LLIM<<0 GO ERR2
125533           X:=CLLIM
125534           FOR WORDN DO
125534                CALL FAR RWORD; CALL  FAR ADCHSUM; T:=CSEGN; CALL PUT1L; GO ERR3
125541                X+1
125542           OD; GO CHCHSUM
125545   ERR3:   GO 6ERROPER
125546   RBUS
125573
125573
125573   %==========================================================================
125573   % 28.40        I I I O X   R F I O X
125573   %
125573   % INSERT-IN-IOX-TABLE <VALUE>
125573   % REMOVE-FROM-IOX-TABLE <VALUE>
125573
125573   SUBR IIIOX,RFIOX
125573
125573   DISP -200; INTEGER ROUFLG,LEGAREA,CIOXVALUE,CVALUE; PSID
125573   INTEGER IOXVALUE:='IODEVICE NUMBER: '
125604
125604   IIIOX: A:=0; GO L1
125606   RFIOX: A:=1
125607   L1:    L=:D; CALL SMENTER; A=:ROUFLG
125612          "IOXVALUE"; CALL SMSGPAR; GO SSMON; A=:CIOXVALUE
125616          7; CALL SELAREA; A=:LEGAREA
125621          IF A><0 THEN
125622                IF ROUFLG=0 THEN CIOXVALUE=:6CTABVALUE; A:=0
125627                ELSE 0=:6CTABVALUE; A:=CIOXVALUE
125632                FI; A=:CVALUE; 1=:6TINCR; X:="IOXTA"
125636                IF LEGAREA BIT BMEMO THEN
125641                   T:=-1; CVALUE; CALL IMTAB; GO ERR
125645                FI
125645                IF LEGAREA BIT BIML THEN
125650                   T:=-1; X:="IOXTA"; CVALUE; CALL IITAB; GO ERR
125655                FI
125655                IF LEGAREA BIT BSAVE THEN
```

```
125660                    CALL 6SOPEN; X:="IOXTA"; T:=-1; CVALUE; CALL ISTAB; GO ERR
125666            FI; FI; GO SMLEAVE
125667    ERR:    GO 6ERRPAR
125670    RBUS
125706
125706
125706    %==========================================================================
125706    % 28.41        L I S C O M
125706    %
125706    % LIST-SERVICE-COMMANDS (<OUTPUT FILE>)
125706
125706    SUBR LISCOM
125706    INTEGER CTXCOM:='S COMMAND: '
125714    LISCOM: L=:D; CALL SMENTER
125716            CALL 6OOPEN; GO SMLEAVE; A=:TDVN
125721           "CTXCOM"; CALL SMKGPAR; A:="TXAPPS"; A=:CSTRING
125725            X:="SCOMTAB"
125726           DO WHILE X.S0><-1
125732              IF A><0 THEN
125733                 0=:CPNT; CALL SM1ABL
125735                 IF A><2 THEN CALL SMCRLF; X.S0; CALL SMOUTTEXT FI
125743              FI; X+4
125744           OD; CALL SMCRLF
125746           T:=TDVN; *MON 2CLOS; JMP * 1
125751           GO SMLEAVE
125752    RBUS
125765
125765
125765    %==========================================================================
125765    % 28.42        C H V A R I
125765    %
125765    % CHANGE-VARIABLE <VARIABLE> (<INDEX>) (<VALUE>)
125765    %
125765    SUBR CHVARI
125765
125765    DISP -200; INTEGER CTABADDR,CINDEX,PFLAG,CVALUE,CLEGAREA; PSID
125765    DISP 0; INTEGER VNAME,VALADDR,LEGAREA,TINDEX; PSID
125765
125765    @ICR
125765    INTEGER ARRAY VARITAB:=(
125765            XV1,MAXP,7,0,            XV2,FIXMAX,7,0,
125775            XV3,ENDCOR,6,0,          XV4,LOADI,7,0,
126005            XV5,BGFPAGE,6,0,         XV6,BGLPAGE,6,0,
126015            XV7,RTFPAGE,6,0,         XV8,RTLPAGE,6,0,
126025            XV9,CCFPAGE,6,0,         XV10,CCLPAGE,6,0,
126035            XV11,CNVRT,7,17,
126041            XV13,9TMCT,7,100,       XV14,IMASK,6,0,
126051            XV15,USEGADR,7,0,        XV16,IDNTS,7,4,
126061            XV17,EXTDS,7,3,          XV18,USLOGOUT,7,0,
126071            XV19,TABLES,7,3,         XV20,UNAFLAG,7,0,
126101            XV21,CACHLIM,7,0,        XV22,LCACHLIM,7,0,
126111            XV23,UCACHLIM,7,0,       XV24,EXSECUR,7,0,
126121            XV25,BUFERASE,7,0,       XV26,SWPFLAG,6,0,
126131            XV27,CERNENABLE,7,0,     XV28,5DBFLAG,7,0,
126141            XV29,CPULOOPTIME,7,0,    XV30,NMATP,7,0,
126151            XV31,BYPINITC,6,0,
126155            -1);
126156    @CR;
126156    INTEGER XV1:='MAXP',XV2:='FIXMAX',XV3:='ENDCOR',XV4:='LOADI',XV5:='BGFPAGE'
126200    INTEGER XV6:='BGLPAGE',XV7:='RTFPAGE',XV8:='RTLPAGE',XV9:='CCFPAGE'
```

```
126220   INTEGER XV10:='CCLPAGE',XV11:='CNVRT',XV12:='SAFILNO',XV13:='TMCTAB'
126237   INTEGER XV14:='IMASK',XV15:='USEGADR',XV16:='IDNTS',XV17:='EXTDS'
126254   INTEGER XV18:='USLOGOUT',XV19:='TABLES',XV20:='UNAFLAG'
126271   INTEGER XV21:='CACHLIM',XV22:='LCACHLIM',XV23:='UCACHLIM'
126307   INTEGER XV24:='EXSECURITY',XV25:='BUFERASE',XV26:='SWPFLAG'
126326   INTEGER XV27:='CERNENABLE',XV28:='5DBFLAG',XV29:='CPULOOPTIME',XV30:='NMATP'
126351   INTEGER XV31:='BYPINITC'
126356
126356   INTEGER PVARIABLE:='N VARIABLE: ',PINDEX:='IOINDEX: ',PVALUE:='IOVALUE: '
126377   INTEGER TMEMO:='$RESIDENT:',TIML:='$IMAGE:',TSAVE:='$SAVE AREA:'
126417
126417   CHVARI: L=:D; CALL SMENTER
126421          "PVARIABLE"; CALL SMSGPAR; GO SSMON; X:="VARITAB"; CALL SMABLOOK
126426          IF A><0 OR T.VALADDR=0 GO 6ERRPAR; T=:CTABADDR; 0=:CINDEX
126435          IF T.TINDEX><0 THEN
126440                "PINDEX"; CALL SAGPAR; GO SSMON; A:=0; A=:CINDEX
126445                IF A<0 OR A>X.TINDEX GO 6ERRPAR
126451          FI
126451          0=:PFLAG; "PVALUE"; CALL SAGPAR; GO SSMON; MIN PFLAG; A=:CVALUE
126457          CTABADDR.LEGAREA; CALL SELAREA; IF A=0 GO SMLEAVE; A=:CLEGAREA
126465          X.VALADDR+CINDEX=:X
126470          IF CLEGAREA BIT BMEMO THEN
126473                "TMEMO"; CALL SMOUTTEXT; CALL R1MEMO; CALL SMOCTU
126477                IF PFLAG=0 THEN CVALUE; CALL W1MEMO FI
126503          FI
126503          IF CLEGAREA BIT BIML THEN
126506                "TIML"; CALL SMOUTTEXT; CALL R1IMAGE; CALL SMOCTU
126512                IF PFLAG=0 THEN CVALUE; CALL W1IMAGE; CALL WIMBACK FI
126517          FI
126517          IF CLEGAREA BIT BSAVE THEN
126522                CALL 6SOPEN; "TSAVE"; CALL SMOUTTEXT; CALL R1SAVE; CALL SMOCTU
126527                IF PFLAG=0 THEN CVALUE; CALL W1SAVE; CALL WXSAVE FI
126534          FI; GO SMLEAVE
126535   RBUS
126567
126567   %=========================================================================
126567   % 28.44        S W D E N T
126567   %
126567   % SWAP-DIRECTORY-ENTRIES <DIR.INDEX 1> <DIR.INDEX 2>
126567   %
126567   SUBR SWDENT
126567   DISP -200
126567   INTEGER C1INDEX,C2INDEX,ENDINDEX,LEGAREA,CWORD
126567   PSID
126567   INTEGER PD1INDEX:='IODIR. INDEX 1: ',PD2INDEX:='IODIR. INDEX 2: '
126611   SWDENT: L=:D; CALL SMENTER
126613          "PD1INDEX"; CALL SMSGPAR; GO SSMON; A=:C1INDEX
126617          "PD2INDEX"; CALL SMSGPAR; GO SSMON; A=:C2INDEX
126623          4; CALL SELAREA; IF A=0 GO SMLEAVE; A=:LEGAREA
126630          IF C1INDEX=C2INDEX GO SMLEAVE; A*DTLEN+"DIRTA"+2000=:C1INDEX
126640          C2INDEX*DTLEN+"DIRTA"+2000=:C2INDEX
126645          IF A>>"ENDDT+2000" OR C1INDEX>>T GO 6ERRPAR
126653          C1INDEX+"DTLEN-1"=:ENDINDEX; CALL 6MOPEN
126657          FOR C1INDEX TO ENDINDEX DO
126663                X:=C1INDEX; CALL R1SAVE; A=:CWORD
126666                X:=C2INDEX; CALL R1SAVE; X:=C1INDEX; CALL W1SAVE
126672                X:=C2INDEX; CWORD; CALL W1SAVE; MIN C2INDEX
126676          OD; CALL WXSAVE; GO SMLEAVE
126704   RBUS
126724.
```

```
126724
126724   %==========================================================================
126724   % 28.45      I L U T A B   R L U T A B
126724   %
126724   % INSERT-IN-LOGICAL-UNIT-TABLE <LOG.UNIT> <INPUT/OUTPUT> <DATAF. ADDR>
126724   % REMOVE-FROM-LOGICAL-UNIT-TABLE <LOG. UNIT> <INPUT/OUTPUT>
126724   %
126724   SUBR ILUTAB,RLUTAB
126724   DISP -200; INTEGER ROUFLG,CLOGU,INOUT,CDFADDR,LEGAREA,CVALUE; PSID
126724   INTEGER PDFADDR:='IODATAFIELD ADDRESS: '
126737
126737   ILUTAB: A:=0; GO L1
126741   RLUTAB: A:=1
126742   L1:     L=:D; CALL SMENTER; A=:ROUFLG
126745           "6PLOGU"; CALL SMSGPAR; GO SSMON; A=:CLOGU
126751           CALL INOROUT; GO 6ERRPAR; A=:INOUT
126754           IF ROUFLG=0 THEN
126756                 "PDFADDR"; CALL SMSGPAR; GO SSMON; A=:CDFADDR
126762           FI; IF CLOGU SHZ -6=1 OR A>24 GO 6ERRPAR
126772           A=:X; CLOGU/\77+A+2+CNVRT(X); IF A-SMENDVTAB(X)<0 GO OK; GO 6ERRPAR
127003   OK:     7; CALL SELAREA; IF A=0 GO 6ERRPAR; A=:LEGAREA
127010           CLOGU SHZ -6=:X; CLOGU/\77+A+1+INOUT+CNVRT(X)=:X
127021           IF ROUFLG=0 THEN CDFADDR ELSE A:=0 FI; A=:CVALUE
127027           IF T:=LEGAREA BIT BMEMO THEN CALL W1MEMO FI
127033           IF LEGAREA BIT BIML THEN CVALUE; CALL W1IMAGE; CALL WIMBACK FI
127041           IF LEGAREA BIT BSAVE THEN
127044                 CALL 6SOPEN; CVALUE; CALL W1SAVE; CALL WXSAVE
127050           FI; GO SMLEAVE
127051   RBUS
127073
127073
127073   %==========================================================================
127073   % 28.46      D U M C A L L
127073   %
127073   % DEFINE-USER-MONITOR-CALL <MON.CALL NO.> <ADDRESS> (<TYPE>)
127073   %
127073   SUBR DUMCALL
127073   DISP -200; INTEGER CTYPE,CMONCALL,CADDRESS,LEGAREA,CADR1,CADR2; PSID
127073   INTEGER PMCALL:='IOMON.CALL NO.: ',PMTYPE:='IOMON.TYPE: '
127113   DUMCALL: L=:D; CALL SMENTER
127115           "PMCALL"; CALL SMSGPAR; GO SSMON; A=:CMONCALL
127121           "6PADDR"; CALL SMSGPAR; GO SSMON; A=:CADDRESS
127125           "PMTYPE"; CALL SAGPAR; GO SSMON; 1; A=:CTYPE
127132           7; CALL SELAREA; IF A=0 GO SMLEAVE; A=:LEGAREA
127137           IF CMONCALL<0 OR A>337 GO ERR
127144           IF CTYPE<0 OR A>34 GO ERR
127151           CMONCALL SHZ -1+"9TMCTAB"=:CADR2; "9MCTAB"+CMONCALL=:CADR1
127160           IF LEGAREA BIT BMEMO THEN
127163                 CADDRESS=:CADR1.S0; T:="9TMCTAB"; X:=CMONCALL; CTYPE; *SBYT
127172           FI
127172           IF LEGAREA BIT BIML THEN
127175                 X:=CADR1; CADDRESS; CALL W1IMAGE; X:=CADR2; CALL R1IMAGE
127202                 IF T:=CMONCALL BIT "0" THEN A/\177400\/CTYPE
127207                 ELSE A/\377; T:=CTYPE SH 10; A\/T
127214                 FI; CALL W1IMAGE; CALL WIMBACK
127216           FI
127216           IF LEGAREA BIT BSAVE THEN
127221                 CALL 6SOPEN; X:=CADR1; CADDRESS; CALL W1SAVE
127225                 X:=CADR2; CALL R1SAVE
127227                 IF T:=CMONCALL BIT "0" THEN A/\177400\/CTYPE
```

```
127234                  ELSE A/\377; T:=CTYPE SH 10; A\/T
127241                  FI; CALL W1SAVE; CALL WXSAVE
127243            FI; GO SMLEAVE
127244    ERR:    GO 6ERRPAR
127245    RBUS
127273
127273
127273    %=========================================================================
127273    % 28.47        S S D N U M B
127273    %
127273    % SET-SPOOLING-DEVICE-NUMBER <SPOOLING INDEX> <LOG.DEVICE NO.>
127273    %
127273    SUBR SSDNUMB
127273    DISP -200; INTEGER CSPDEV,CLOGU,LEGAREA; PSID
127273    INTEGER PLOGD:='IOLOG. DEVICE NO.: '
127305    INTEGER CETXT:='$NOT ALLOWED WHEN SPOOLING IS STARTED$'
127331    SSDNUMB: L=:D; CALL SMENTER
127333            "6PSPNUMB"; CALL SMSGPAR; GO SSMON; A-1=:CSPDEV
127340            "PLOGD"; CALL SMSGPAR; GO SSMON; A=:CLOGU
127344            7; CALL SELAREA; IF A=0 GO SMLEAVE; A=:LEGAREA
127351            CSPDEV*SPLEN+"SPTAB"; IF Q GO 6ERRPAR
127356            IF A<<"SPTAB" OR A>>="ENDSP" GO 6ERRPAR; A=:D
127365            IF A.SPROG.WLINK><0 THEN "CETXT"; CALL SMOUTTEXT; GO SMLEAVE FI
127374            A:=D+"SPERI"=:X
127377            IF LEGAREA BIT BMEMO THEN CLOGU=:X,SO FI
127404            IF LEGAREA BIT BIML THEN CLOGU; CALL W1IMAGE; CALL WIMBACK FI
127412            IF LEGAREA BIT BSAVE THEN
127415                  CALL 6SOPEN; CLOGU; CALL W1SAVE; CALL WXSAVE
127421            FI; GO SMLEAVE
127422    RBUS
127445
127445
127445    %=========================================================================
127445    % 28.48        D B S P R O G
127445    %
127445    % DEFINE-BATCH-SUPERVISOR <RT-PROGRAM>
127445    %
127445    SUBR DBSPROG
127445    DISP -200; INTEGER CRTDADDR,LEGAREA; PSID
127445    DBSPROG: L=:D; CALL SMENTER
127447            "6PRTNAM"; CALL SMSGPAR; GO SSMON; A=:CRTDADDR
127453            IF A><0 THEN CALL CHRTDES; GO 6ERRPAR FI
127456            7; CALL SELAREA; IF A=0 GO SMLEAVE; A=:LEGAREA; X:="USRTB"
127464            IF LEGAREA BIT BMEMO THEN CRTDADDR=:X.SO FI
127471            IF LEGAREA BIT BIML THEN CRTDADDR; CALL W1IMAGE; CALL WIMBACK FI
127477            IF LEGAREA BIT BSAVE THEN
127502                  CALL 6SOPEN; CRTDADDR; CALL W1SAVE; CALL WXSAVE
127506            FI; GO SMLEAVE
127507    RBUS
127525
127525
127525
127525    %=========================================================================
127525    % 28.49        L I A D D R
127525    %
127525    % LIST-ADDRESSES (<OUTPUT FILE>)
127525    %
127525    SUBR LIADDR
127525    INTEGER ARRAY LADDRTAB:=(XT1,XT2,XT3,XT4,XT5,XT6,XT7,XT8,-1)
127536    INTEGER XT1:='RET',XT2:='GET0',XT3:='GET1',XT4:='GET2',XT5:='GET3'
```

"8 COSP

331

127434

SSDNU + 103 / COSDA

```
127554   INTEGER XT6:='GET4',XT7:='GET5',XT8:='CCTAB'
127565   INTEGER ARRAY AADRTAB:=(RET,GET0,GET1,GET2,GET3,GET4,GET5,CCTAB)
127575   _IADDR: L=:D; CALL SMENTER
127577           CALL 6OOPEN; GO SMLEAVE; A=:TDVN; X:=0
127603           DO CALL SMCRLF; WHILE LADDRTAB(X)><-1
127610                 CALL SMOUTTEXT; ##:; CALL SMTCO; AADRTAB(X); CALL SMOCTU; X+1
127616           OD; GO SMLEAVE
127620   RBUS
127632
127632
127632
127632   %=========================================================================
127632   % 28.51       I S P H E A D   R S P H E A D        .
127632   %
127632   % REMOVE-SPOOLING-HEADER <SPOOLING INDEX>
127632   % INSERT-SPOOLING-HEADER <SPOOLING INDEX>
127632   %
127632   SUBR ISPHEAD,RSPHEAD
127632
127632   DISP -200
127632   INTEGER ROUFLG                  % ROUTINE FLAG; = 0, INSERT; = 1 REMOVE
127632   INTEGER CSPDEV
127632   INTEGER LEGAREA
127632   INTEGER FFATR                   % FORM-FEED AFTER TRAILER FLAG.
127632   INTEGER FFAHD                   % FORM-FEED AFTER HEADER FLAG.
127632   INTEGER CSEG
127632   PSID
127632
127632   INTEGER AHEADR:=HEAPR           % ADDRESS OF HEADER ROUTINE.
127633   INTEGER ATRAPR:=TRAPR           % ADDRESS OF TRAILER ROUTINE.
127634   SYMBOL SPHEROUT=11              % DISPLACEMENT OF HEADER IN SPOOLING BASEFIELD.
127634   INTEGER PWSEG:=BFIELD+CSEG
127635
127635   ISPHEAD: A:=0; GO L1
127637   RSPHEAD: A:=1
127640   L1:      L=:D; CALL SMENTER; A=:ROUFLG
127643            "6PSPNUMB"; CALL SMSGPAR; GO SSMON; IF A = 0 GO 6ERRPAR
127650            A-1=:CSPDEV
127652            IF ROUFLG >< 0 THEN
127654               "6FFHE"; CALL SMSGPAR; GO SSMON; CALL SMYESNO     % FORM FEED AFTER HEADER
127660               IF A=-1 GO SSMON; A=:FFAHD
127664               "6FFTR"; CALL SMSGPAR; GO SSMON; CALL SMYESNO     % FORM-FEED AFTER TRAILER.
127670               IF A=-1 GO SSMON; A=:FFATR
127674            FI
127674            5; CALL SELAREA; IF A=0 GO SMLEAVE; A=:LEGAREA
127701            CSPDEV*SPLEN+"SPTAB"; IF Q GO 6ERRPAR
127706            IF A<<"SPTAB" OR A>>="ENDSP" GO  6ERRPAR
127714            CSPDEV*SSPLEN+"SSPTAB"+SPHEROUT=:X                    % ADDRESS ON SPOOLING SEGMENT
127721            IF ROUFLG >< 0 THEN
127723               IF LEGAREA BIT BMEMO THEN
127726                  A:=FFAHD; T:=SPSGM; CALL PUT1L; GO 6ERROPER
127732                  X+1; A:=FFATR;  CALL PUT1L; GO 6ERROPER; X-1
127737               FI
127737   %           IF LEGAREA BIT BIML THEN     % DUMMY IN NEW SPOOLING
127737   %           FI
127737               IF LEGAREA BIT BSAVE THEN
127742                  T:=5ISPS; A:=FFAHD; CALL PUT1L; GO 6ERROPER
127746                  X+1; A:=FFATR; CALL PUT1L; GO 6ERROPER;
127752               FI
127752            ELSE
```

```
127753              IF LEGAREA BIT BMEMO THEN
127756                 AHEADR; T:=SPSGM; CALL PUT1L; GO 6ERROPER;
127762                 X+1; ATRAPR; CALL PUT1L; GO 6ERROPER; X-1
127767              FI
127767   %       IF LEGAREA BIT BIML THEN        % DUMMY
127767   %       FI
127767              IF LEGAREA BIT BSAVE THEN
127772                 T:=5ISPS; AHEADR; CALL PUT1L; GO 6ERROPER
127776                 X+1; ATRAPR; CALL PUT1L; GO 6ERROPER
130002              FI
130002          FI; 5ISPS=:CSEG; "PWSEG"; *MON 2WSEG
130006          SPSGM=:CSEG; "PWSEG"; *MON 2WSEG
130012          GO SMLEAVE.
130013
130013   RBUS
130035
130035   %=========================================================================
130035   % 28.52       S C H P R I
130035   %
130035   % SET-CHANNEL-PRIORITY <CHANNEL NAME> <PRIORITY>
130035   %
130035   SUBR SCHPRI
130035   DISP -200; INTEGER CPRIO,CFINO,LEGAREA; PSID
130035   DISP -4; INTEGER PRICH; PSID
130035   INTEGER PCHPRI:='IOCHANNEL PRIORITY: ',6CHNAME:='S CHANNEL NAME: '
130061   SCHPRI: L=:D; CALL SMENTER
130063          "6CHNAM"; CALL SMSGPAR; GO SSMON; A=:X; T:=1; *MON 2NOPE
130071          GO CFILERR; A=:CFINO; "PCHPRI"; CALL SMSGPAR; GO SSMON
130076          IF A<0 OR A>377 GO 6ERRPAR; A=:CPRIO
130103          7; CALL SELAREA; IF A=0 GO SMLEAVE; A=:LEGAREA
130110          CFINO; CALL LOGPH; IF A=0 THEN CALL ERRFATAL FI
130114          IF D=0 GO 6ERRPAR; A:=D+"PRICH"=:X
130121          IF LEGAREA BIT BMEMO THEN CPRIO=:X.SO FI
130126          IF LEGAREA BIT BIML THEN CPRIO; CALL W1IMAGE; CALL WIMBACK FI
130134          IF LEGAREA BIT BSAVE THEN
130137                CALL 6SOPEN; CPRIO; CALL W1SAVE; CALL WXSAVE
130143          FI; GO SMLEAVE
130144   CFILERR: *MON 64
130145          GO SMLEAVE
130146   RBUS
130167
130167
130167   %=========================================================================
130167   % 28.53       C D A F I
130167   %
130167   % CHANGE-DATAFIELD <LOG.UNIT> <INPUT/OUTPUT>
130167   %
130167   SUBR CDAFI
130167   DISP -200
130167   INTEGER CLOGU,INOUT,CAREA,CADR,DIGIT,NDIGIT,CDFADDR,LLIM,ULIM,CCHAR,TERMX,CTERM
130167   INTEGER POINTER CLINK
130167   INTEGER NEGFLG=INOUT,CALPH=CLOGU
130167   PSID
130167   DISP 1; INTEGER DISPVAL,DISPTYPE,ZDISPVAL; PSID
130167
130167   % DISPTYPE:  =0: NOT EXISTING IN IMAGE/SAVE FOR TERMINALS IN VSX
130167   %            =1: ANOTHER DISPLACEMENT IN RESIDENT THAN IN IMAGE/SAVE.
130167   %            =2: SAME DISPLACEMENT IN RESIDENT AS IN IMAGE/SAVE
130167   %            =3: NOT EXISTING IN VSX VERSION
130167
```

```
130167   @ICR
130167   INTEGER ARRAY DFDISPTAB:=(
130167          XDI0,RESLINK,2,0,       XDI1,RTRES,2,0,
130177          XDI2,BWLINK,2,0,        XDI3,TYPRING,2,0,
130207          XDI4,ISTATE,2,0,        XDI5,MLINK,2,0,
130217          XDI6,MFUNC,2,0,         XDI7,TRLREG,0,0,
130227          XDI10,HSTAT,0,0,         XDI11,MTRANS,0,0,
130237          XDI12,DFOPP,0,0,        XDI13,DERROR,0,0,
130247          XDI14,BUFST,0,0,        XDI15,MAX,0,0,
130257          XDI16,BHOLD,0,0,        XDI17,HENTE,0,0,
130267          XDI20,CFREE,0,0,        XDI21,FYLLE,0,0,
130277          XDI22,MINBHOLD,0,0,     XDI23,MAXBHOLD,0,0,
130307          XDI24,IMAXBHOLD,0,0,    XDI26,TERM,0,0,
130317          XDI27,CHARI,0,0,        XM32,TRG,0,0,
130327          XM31,ARG,0,0,           XM30,DRG,0,0,
130337          XM27,XRG,0,0,           XM26,CTRG,0,0,
130347          XM25,CARG,0,0,          XM24,CXRG,0,0,
130357          XM23,ERCNT,0,0,         XM22,SERRB,0,0,
130367          XM21,WERRB,0,0,         XM20,AERRB,0,0,
130377          XM17,TACNS,0,0,         XM16,TACOUNT,0,0,
130407          XM15,COMFL,0,0,         XM14,BLSZ,0,0,
130417          XX20,IOLOG,0,0,         XX21,WFLAG,0,0,
130427          XX22,ICORAD,0,0,        XX23,IBLOAD,0,0,
130437          XX24,IMAXW,0,0,         XX25,IFUNC,0,0,
130447          XX26,IRETW,0,0,         XX27,MRSTA,0,0,
130457          XX30,SSREF,0,0,         XX31,STRSEG,0,0,
130467          XX32,DRT,0,0,           XY22,BSTATE,0,0,
130477          XY23,DBPROG,1,ZDBPROG,XY24,DBADR,0,0,
130507          YM14,TSPEED,1,ZTSPEED,YM13,CNTREG,1,ZCNTREG,
130517          YM12,DFLAG,1,ZDFLAG,    YM7,LAST,0,0,
130527          YM5,TMR,0,0,            YM4,TTMR,0,0,
130537          YM3,HDEV,0,0,           XP7,IOTRANS,0,0,
130547          XP10,STDEV,0,0,         XP11,SETDV,0,0,
130557          XP23,TSTATE,3,0,        XP42,FLAGB,0,0,
130567          XP16,-43,0,0,           XMM17,CTTYP,1,ZCTTYP,
130577          XMM11,ECHOTAB,0,0,      XMM10,BRKTAB,0,0,
130607          XMM6,TMSUB,0,0,         XMM2,STDRIV,2,0,
130617          XMM1,DRIVER,2,0,        XVSX1,ROUSPEC,1,ZROUSPEC,
130627          XVSX2,CESCP,1,ZCESCP,   XM45,TINFO,0,0,
130637          -1);
130640
130640   INTEGER XDI0:='RESLINK',XDI1:='RTRES',XDI2:='BWLINK',XDI3:='TYPRING',
130657   XDI4:='ISTATE',XDI5:='MLINK',XDI6:='MFUNC',XDI7:='TRLREG',XDI10:='HSTAT',
130700   XDI11:='MTRANS',XDI12:='DFOPP',XDI13:='DERROR',XDI14:='BUFST',XDI15:='MAX',
130720   XDI16:='BHOLD',XDI17:='HENTE',XDI20:='CFREE',XDI21:='FYLLE',XDI22:='MINBHOLD',
130741   XDI23:='MAXBHOLD',XDI24:='IMAXBHOLD',XDI26:='TERM',XDI27:='CHARI',
130761   XM32:='TRG',XM31:='ARG',XM30:='DRG',XM27:='XRG',XM26:='CTRG',XM25:='CARG',
130777   XM24:='CXRG',XM23:='ERCNT',XM22:='SERRB',XM21:='WERRB',XM20:='AERRB',
131016   XM17:='TACNS',XM16:='TACOUNT',XM15:='COMFL',XM14:='BLSZ',XX20:='IOLOG',
131036   XX21:='WFLAG',XX22:='ICORAD',XX23:='IBLOAD',XX24:='IMAXW',XX25:='IFUNC',
131057   XX26:='IRETW',XX27:='MRSTA',XX30:='SSREF',XX31:='STRSEG',XX32:='DRT',
131076   XY22:='BSTATE',XY23:='DBPROG',XY24:='DBADR',
131111   YM14:='TSPEED',YM13:='CNTREG',YM12:='DFLAG',YM7:='LAST',YM5:='TMR',
131131   YM4:='TTMR',YM3:='HDEV',XP7:='IOTRANS',XP10:='STDEV',XP11:='SETDV',
131151   XP23:='TSTATE',XP42:='FLAGB',XP16:='CONVTAB',XMM17:='CTTYP',XMM11:='ECHOTAB',
131173   XMM10:='BRKTAB',XMM6:='TMSUB',XMM2:='STDRIV',XMM1:='DRIVER',
131212   TMEMO:=' MEMORY  ',TIML:=' IMAGE  ',TSAVE:='SAVE-AREA',
131231   XVSX1:='ROUSPEC',XVSX2:='CESCP',XM45:='TINFO';
131243   @CR;
131243   INTEGER TILADDR:='$ILL. ADDRESS',WRFL=?,CTBADDR=?,VSXDISP=?,CDFTYPE=?,SYDISP=?
```

```
131252   INTEGER POINTER PCDFTYPE=?, PWRFL=?
131252   INTEGER TNXSIST:='$NOT EXISTING IN IMAGE/SAVE-AREA$'
131273   INTEGER TNVSX:='$NOT EXISTING IN VSX VERSION$'
131312   INTEGER VSXMSYMB:='$ONLY SYMBOLIC DISPLACEMENTS ARE ALLOWED FOR TERMINALS IN IMAGE/SAVE AREA$'
131360   INTEGER POINTER PPWRFL:=WRFL
131361
131361   SYMBOL BIRES=1                % LEGAL DISPLACEMENT FOR TERMINALS IN IMAGE/SAVE
131361   SYMBOL BEXDF=2                % ACCESS DATAFIELD OUTSIDE RESIDENT WHEN TERMINALS
131361
131361   CDAFI: L=:D; CALL SMENTER
131363          "6PLOGU"; CALL SMSGPAR; GO SSMON; A=:CLOGU
131367          CALL INOROUT; GO 6ERRPAR; A=:INOUT; CLOGU; CALL LOGPH
131374          IF T:=INOUT=0 THEN IF A=0 GO 6ERRPAR
131401          ELSE IF A:=D=0 GO 6ERRPAR
131405          FI; A=:CDFADDR+100=:ULIM-300=:LLIM
131412          7; CALL SELAREA; IF A=0 GO SMLEAVE; A=:CAREA
131417          IF A BIT BSAVE THEN CALL 6SOPEN FI
131422   @LIB CXCPU
131422          IF CDFADDR.TYPRING BIT 5TERM THEN 1=:PCDFTYPE ELSE 0=:PCDFTYPE FI
131432   @ELIB
131432          2; *MON 2BRKM; MON 2ECHO
131435          CALL SMCRLF
131436          IF CAREA BIT BMEMO THEN "TMEMO"; CALL SMOUTTEXT FI
131443          IF CAREA BIT BIML THEN "TIML"; CALL SMOUTTEXT FI
131450          IF CAREA BIT BSAVE THEN "TSAVE"; CALL SMOUTTEXT FI
131455          CALL SMCRLF; CALL SMCRLF; 0=:PPWRFL
131460   L1:    CALL DISPREAD; GO L2; IF A=##/ GO L3
131465   L2:    IF A=15 AND NDIGIT=0 GO L1
131472   L22:   ##?; CALL SMTCO; CALL SMCRLF
131475          IF CTERM><15 AND A><##/ THEN
131504                DO CALL SMTCI WHILE A><15 AND A><##/ OD
131514          FI; GO L1; *)FILL
131540
131540   INTEGER CTBADDR,VSXDISP,CISADR
131543   INTEGER POINTER PCDFTYPE:=CDFTYPE
131544
131544   L3:    IF CALPH=0 THEN -1=:CTBADDR FI
131550          DIGIT=:VSXDISP+CDFADDR=:CADR=:CISADR; GO PCONTENT
131556   LOOP:
131556   L30:          CALL DISPREAD; GO FAR L31; IF A=##/ GO L3
131563                 IF CALPH><0 GO FAR L22
131566   @LIB CXCPU-,
131566   @LIB CXCPU
131566                 IF CAREA BIT BMEMO AND CDFTYPE><0 THEN
131573                    IF VSXDISP="TYPRING" THEN DIGIT=:CADR.S0 FI
131602                    IF CDFTYPE BIT BEXDF THEN
131605                       DIGIT; X:=CDFADDR; T:=VSXDISP; CALL XSTDFADDR
131611                    ELSE DIGIT=:CADR.S0
131615                    FI
131615                 ELSE IF CAREA BIT BMEMO THEN DIGIT=:CADR.S0 FI
131624                 FI
131624                 IF CDFTYPE><0 AND A NBIT BIRES GO NIMSAV
131630                 X:=CISADR
131631   @ELIB
131631                 IF CAREA BIT BIML THEN DIGIT; CALL W1IMAGE FI
131636                 IF CAREA BIT BSAVE THEN 1=:PWRFL; DIGIT; CALL W1SAVE FI
131645   NIMSAV:       MIN CADR; MIN CISADR; -1=:CTBADDR; MIN VSXDISP; 0/\0
131653   @LIB CXCPU
131653                 IF CDFTYPE><0 AND CAREA BIT BIML OR A BIT BSAVE OR SYDISP><0 GO FAR L1
131665   @ELIB
```

```
131665   PCONTENT:     IF CADR-LLIM<<0 OR ULIM-CADR<<0 THEN
131675                     "TILADDR"; CALL SMOUTTEXT; GO FAR L1; *)FILL
131712   INTEGER SYDISP
131713   INTEGER CDFTYPE
131714   INTEGER POINTE PVSXDISP:=VSXDISP,PWRFL:=WRFL,PCISADR:=CISADR
131717                  FI
131717   @LIB CXCPU-,
131717   @LIB CXCPU
131717                  CALPH=:SYDISP                                          % REMEMBER IF SYMBOLIC DISP
131721                  IF CDFTYPE><0 THEN
131723                     1=:CDFTYPE
131725                     IF CTBADDR><-1 THEN
131731                        IF A.DISPTYPE=3 THEN "TNVSX"; CALL SMOUTTEXT; GO FAR L1 FI
131741                        IF A=1 THEN
131744                           T:=CDFTYPE BONE BIRES=:CDFTYPE
131747                           X.ZDISPVAL+CDFADDR=:PCISADR
131752                        ELSE IF A=2 AND CAREA BIT BMEMO GO GRESID
131761                        FI
131761                     ELSE
131762                        IF CAREA BIT BIML OR A BIT BSAVE THEN
131767                           "VSXMSYMB"; CALL SMOUTTEXT; GO FAR L1
131772                        FI
131772                     FI
131772                  FI
131772                  IF CAREA BIT BMEMO THEN
131775                     IF CDFTYPE><0 THEN
131777                        A BONE BEXDF=:CDFTYPE
132001                        T:=PVSXDISP; X:=CDFADDR; CALL XGTDFADDR
132004                     ELSE
132005   GRESID:              CADR.SO
132007                     FI; CALL SMOCTU; 40; CALL SMTCO
132012                  FI
132012                  IF CDFTYPE><0 AND A NBIT BIRES THEN
132016                     IF CAREA BIT BIML OR A BIT BSAVE THEN
132023                        "TNXSIST"; CALL SMOUTTEXT; GO FAR L1
132026                     FI
132026                  FI
132026   @ELIB
132026                  X:=PCISADR
132026                  IF CAREA BIT BIML THEN CALL R1IMAGE; CALL SMOCTU; 40; CALL SMTCO FI
132027                  IF CAREA BIT BSAVE THEN 1=:WRFL; CALL R1SAVE; CALL SMOCTU; 40; CALL SMTCO FI
132036                  40; CALL SMTCO
132047
132051   EDO:    GO FAR LOOP; *)FILL
132065
132065   INTEGER WRFL
132066
132066   L31:    IF A=15 AND NDIGIT=0 GO FAR NIMSAV
132074           GO FAR L22
132075   ERR:    GO 6ERRPAR
132076   DISPREAD: 0=:DIGIT=:NDIGIT; A:=L=:"CLINK"; X:=0
132103           DO WHILE X<100
132106                 CALL SMTCI; A/\177=:CTERM; T:="SMFIBUF"; *SBYT
132113                 IF A=15 OR A=##/ GO LD1
132121                 IF A=##. OR A=##@ GO OUT
132127                 X+1
132130           OD; "6TILLINE"; CALL SMOUTTEXT
132133   OUT:    1; *MON 2BRKM; MON 2ECHO
132136           IF CAREA BIT BSAVE AND WRFL=1 THEN CALL WXSAVE FI
132146           GO SSMON; *)FILL
132161   LD1:    X=:TERMX:=0; 0=:NEGFLG=:CALPH
```

```
132165          DO
132165                T:="SMFIBUF"; *LBYT
132167          WHILE X><TERMX
132172                A=:CCHAR; IF A=40 AND NDIGIT=0 GO NEXTD
132200                IF CCHAR<##0 OR A>##7 THEN
132207                   IF A=##- AND NDIGIT=0 AND NEGFLG=0 THEN MIN NEGFLG; GO NEXTD FI
132220                   CCHAR; GO CHALPH
132222                FI; CCHAR-60; T:=DIGIT SH 3+A=:DIGIT; MIN NDIGIT
132231   NEXTD:       X+1
132232          OD; IF NEGFLG><0 THEN DIGIT-=:DIGIT FI
132240          IF T:=NDIGIT><0 THEN MIN "CLINK" FI; CTERM; GO CLINK
132246
132246   INTEGER POINTER PCTABADDR:=CTBADDR
132247   CHALPH: X:=TERMX; T:="SMFIBUF"; ##'; *SBYT
132253          X:="DFDISPTAB"; "SMFIBUF"=:CSTRING; 0=:CPNT; CALL SMABLOOK
132260          IF A=0 THEN
132261             MIN "CLINK"; T.DISPVAL=:DIGIT; MIN CALPH; CTERM; X=:PCTABADDR
132270          ELSE A:=0
132272          FI; GO CLINK
132273   RBUS
132276
132276
132276   %==========================================================================
132276   % 28.54       C D R T C
132276   %
132276   % DEFINE-RTCOMMON-SIZE <COMMON SIZE> <FIRST PHYSICAL PAGE>
132276   %
132276   SUBR CDRTC
132276   DISP -200; INTEGER LEGAREA,CSIZE,COUNT,FPAGE,DEFAULT; PSID
132276   INTEGER PSIZE:='IORTCOMMON SIZE: '
132307   INTEGER FRSTP:='IOFIRST PHYSICAL PAGE(OCT): '
132326   CDRTC: L=:D; CALL SMENTER
132330          "PSIZE"; CALL SMSGPAR; GO 6ERRPAR; A=:CSIZE
132334          0=:DEFAULT; "FRSTP"; CALL SAGPAR; GO 6ERRPAR; MIN DEFAULT
132341   CDRT2: A=:FPAGE
132342          IF CSIZE=0 THEN 0=:FPAGE; GO L1 FI
132346          IF A>>100 GO 6ERRPAR
132351          IF DEFAULT><0 THEN 0=:FPAGE; GO L1 FI
132355          IF FPAGE+CSIZE>>ENDPAGE+1 GO 6ERRPAR
132363          IF FPAGE<=LRESP GO 6ERRPAR
132367          IF FPAGE>=FPOFP AND A<=LPOFP GO 6ERRPAR
132376          GO L1; *)FILL
132411   L1:    6; CALL SELAREA; IF A=0 GO SMLEAVE; A=:LEGAREA
132416          IF A BIT BIML THEN
132420                X:="CCSTART"; CALL R1IMAGE; FPAGE; CALL W1IMAGE
132424                X:="CCTAB"; 0=:COUNT
132426                DO CALL R1IMAGE WHILE A=0; MIN COUNT; X+1; OD
132433                IF CSIZE<0 OR A>COUNT SHZ -1 GO 6ERRPAR
132441                X:="CCNOX"; CSIZE; CALL W1IMAGE; X:="CCFPAGE"
132445                100-CSIZE; CALL W1IMAGE; X:="RTLPAGE"
132451                77-CSIZE; CALL W1IMAGE; CALL WIMBACK
132455          FI
132455          IF LEGAREA BIT BSAVE THEN
132460                CALL 6SOPEN; X:="CCSTART"
132462                CALL R1SAVE; FPAGE; CALL W1SAVE
132465                0=:COUNT; X:="CCTAB"
132467                DO CALL R1SAVE WHILE A=0; MIN COUNT; X+1; OD
132474                IF CSIZE<0 OR A>COUNT SHZ -1 GO 6ERRPAR
132502                X:="CCNOX"; CSIZE; CALL W1SAVE
132505                X:="CCFPAGE"; 100-CSIZE; CALL W1SAVE
```

```
132511                  X:="RTLPAGE"; 77-CSIZE; CALL W1SAVE; CALL WXSAVE
132516            FI; GO SMLEAVE
132517    RBUS
132536
132536
132536    %=========================================================================
132536    % 28.55       I S Y S E G
132536    %
132536    % INITIALIZE-SYSTEM-SEGMENT <SEGMENT NUMBER>
132536    %
132536    %  THE ROUTINE IS LOCATED ON SEGMENT 3 (OPCOM SEGMENT)
132536
132536    SUBR ISYSEG
132536
132536    INTEGER POINTER LREG
132537    ISYSEG: A:=L=:"LREG"; CALL OPSYS(OISYSEG); GO LREG
132544    RBUS
132545
132545
132545
132545    %=========================================================================
132545    % 28.57       S C P R O T
132545    %
132545    % SET-COMMAND-PROTECTION <COMMAND> <PROTECTION>
132545    %
132545    % A PART OF THE ROUTINE IS LOCATED ON SEGMENT 3 (OPCOM SEGMENT)
132545    %
132545    SUBR SCPROT
132545
132545    DISP -200; INTEGER CCOMADR,XCPROT,LEGAREA; PSID
132545    INTEGER P6PROT:='N PROTECTION (SYSTEM, RT OR PUBLIC): '
132570    INTEGER ARRAY TXPROT:=(TUSYS,2,0,0, TURT,1,0,0, TUPUB,0,0,0,-1)
132605    INTEGER TUSYS:='SYSTEM', TURT:='RT', TUPUB:='PUBLIC'
132617    INTEGER PWSCB:="50PSEG"
132620
132620    SCPROT: L=:D; CALL SMENTER; CALL OPSYS(OSCPROT); GO SMLEAVE
132625            T=:CCOMADR; "P6PROT"; CALL SMSGPAR; GO SMLEAVE
132631            X:="TXPROT"; CALL SMABLOOK
132633            IF A><0 GO 6ERRPAR
132635            T.S1=:XCPROT; 5; CALL SELAREA; A=:LEGAREA
132643            IF A BIT BMEMO THEN
132645                  CCOMADR+"CPROTECT"=:X; T:=50PSEG; XCPROT; CALL PUT1L; GO 6ERROPER
132654                  "PWSCB"; *MON 2WSEG
132656            FI
132656            IF LEGAREA BIT BSAVE THEN
132661                  CALL 6MOPEN                % MACM-AREA
132662                  A:=CCOMADR+"CPROTECT"-BCSTA=:X
132666                  XCPROT; CALL W1SAVE; CALL WXSAVE
132671            FI; GO SMLEAVE
132672    RBUS
132713
132713    %=========================================================================
132713    % 28.58       S C S F S
132713    %
132713    % COMMAND: SET-CLOSED-SCRATCH-FILE-SIZE  <TERMINAL NUMBER> <SIZE>
132713    %
132713    SUBR SCSFS
132713    DISP -200; INTEGER CTRMNO,CFLSIZE,LEGAREA,CSEGN,CDFAD; PSID
132713    INTEGER TPASIZE:='IOSCRATCH FILE SIZE IN PAGES: '
132713    INTEGER TXEDY:='$THERE IS NO SPECIFIC SCRATCH FILE FOR THIS LOGICAL UNIT$'
132733
```

```
132770
132770   SCSFS: L=:D; CALL SMENTER
132772          "PTERMNO"; CALL SAGPAR; GO SSMON; TTNO; A=:CTRMNO
132777          "TPASIZE"; CALL SAGPAR; GO SSMON; 100; A=:CFLSIZE
133004          5; CALL SELAREA; IF A=0 GO SMLEAVE; A=:LEGAREA
133011          CTRMNO; CALL LOGPH; IF A=0 GO 6ERRPAR; A=:CDFAD
133016          CALL SMCHTERM; GO 6ERRPAR
133020          X:=CDFAD; CALL SMDYNALLOC; GO L1; GO ERRDYN
133024   L1:    T:="DBPROG"; CALL XGTDFADDR; IF A=0 GO 6ERRPAR
133030          A.SEGM SHZ -10=:CSEGN
133034          IF LEGAREA BIT BMEMO THEN
133037                X:="SCRSIZE"; T:=CSEGN; CFLSIZE; CALL PUT1L; GO 6ERROPER
133044          FI
133044          IF LEGAREA BIT BSAVE THEN
133047                CALL 6SOPEN
133050                X:="SCRSIZE"; CFLSIZE; CALL W1SAVE; CALL WXSAVE
133054          FI; GO SMLEAVE
133055   ERRDYN: "TXEDY"; CALL SMOUTTEXT; GO SMLEAVE
133060   RBUS
133105
133105   %=========================================================================
133105   % 28.59       C H I O B S
133105   %
133105   % COMMAND: CHANGE-BUFFER-SIZE <LOG. UNIT> <IN/OUT>
133105   %
133105   SUBR CHIOBS
133105
133105   DISP -200; INTEGER CLOGU,INOUT,CIOBSIZE,LEGAREA,CVALUE,CDFADR,WDEV,BSUM; PSID
133105   DISP -200; INTEGER SVIND=WDEV; PSID
133105   INTEGER TPAIBS:='IOBUFFER SIZE IN WORDS: '
133122   INTEGER POINTER LREG=?,R1RUT=?,W1RUT=?,WCLOS=?
133122   @LIB CXCPU
133122   INTEGER TD1BA:='NOT ENOUGH SPACE IN TERMINAL BUFFER POOL!'
133147   INTEGER TBTLA:='TOTAL BUFFER TO LARGE! (INPUT+OUTPUT)'
133172   @ELIB
133172
133172   % ROUTINE TO SET BUFFER SIZES IN IOBUT TABLE
133172   SBSZ:   A:=L=:"LREG"; X:="IOBUT"
133175           DO CALL R1RUT WHILE A><-1
133201              IF A/\107777=CLOGU THEN
133205                 X+1; CALL R1RUT
133207                 IF A BIT 17 THEN
133211                    1=:WDEV; CIOBSIZE BONE 17
133215                 ELSE
133216                    0=:WDEV; CIOBSIZE
133220                 FI; CALL W1RUT
133221                 CIOBSIZE; IF T:=WDEV=0 THEN A SH 1 FI; A=:CVALUE
133227                 CDFADR+"MAX"=:X; CVALUE; CALL W1RUT
133234                 CDFADDR+"CFREE"=:X; CVALUE; CALL W1RUT
133241                 CALL WCLOS; MIN "LREG"; GO LREG
133244              FI; X+2
133245           OD; GO 6ERRPAR
133247   *)FILL
133254
133254   INTEGER POINTER LREG,R1RUT,W1RUT,WCLOS
133260   @LIB CXCPU
133260   INTEGER POINTER PTD1BA:=TD1BA, PTBTLA:=TBTLA
133262
133262   % ROUTINE TO SET BUFFER SIZES FOR TERMINALS IN TIOBU (VSX)
133262   STBSZ: A:=L=:"LREG"
```

```
133264          X:="TIOBU"; CLOGU BZERO 17=:CLOGU
133270          DO CALL R1RUT WHILE A><-1
133274             IF A=CLOGU THEN
133277                X+1; CALL R1RUT; A=:BSUM; X+1; CALL R1RUT
133304                IF T:=INOUT=0 THEN
133307                   T:=BSUM=:CVALUE; A=:BSUM; X-1=:SVIND
133314                ELSE
133315                   A=:CVALUE; X=:SVIND
133317                FI
133317                IF CIOBSIZE+BSUM+TDISIZ+TDOSIZ>2000 THEN
133326                   "PTBTLA"; CALL SMOUTTEXT; GO LREG
133331                FI
133331                CIOBSIZE; CALL W1RUT
133333                X:="TIOBU"; 0=:BSUM
133335                DO CALL R1RUT WHILE A><-1
133341                   A+TDISIZ+BSUM=:BSUM; IF C GO BTLAR; X+1; CALL R1RUT
133350                   A+TDOSIZ+BSUM=:BSUM; IF C GO BTLAR; X+2
133356                OD
133357                CALL WCLOS; MIN "LREG"; GO LREG
133362   BTLAR:       X:=SVIND; A:=CVALUE; CALL W1RUT; CALL WCLOS
133366                "PTD1BA"; CALL SMOUTTEXT; GO LREG
133371             FI; X+3
133372          OD; GO 6ERRPAR; *)FILL
133400   @ELIB
133400
133400   INTEGER POINTER CHBSIZ
133401   INTEGER POINTER PR1RUT:=R1RUT,PW1RUT:=W1RUT,PWCLOS:=WCLOS
133404
133404   CHIOBS: L=:D; CALL SMENTER
133406          "6PLOGU"; CALL SMSGPAR; GO SSMON; A=:CLOGU
133412          CALL INOROUT; GO 6ERRPAR; A=:INOUT; IF A=1 THEN CLOGU BONE 17=:CLOGU FI
133423          "TPAIBS"; CALL SMSGPAR; GO 6ERRPAR; A=:CIOBSIZE
133427          IF A<=0 GO 6ERRPAR
133431          6; CALL SELAREA; IF A=0 GO SMLEAVE; A=:LEGAREA
133436          CLOGU /\37777; CALL LOGPH; IF T:=INOUT><0 THEN A:=D FI
133445          IF A=0 GO 6ERRPAR; A=:CDFADR
133450   @LIB CXCPU
133450          IF A.TYPRING BIT 5TERM THEN "STBSZ"=:"CHBSIZ" ELSE "SBSZ"=:"CHBSIZ" FI
133461   @ELIB
133461   @LIB CXCPU-,
133461          IF LEGAREA BIT BIML THEN                    % IMAGE
133464             "R1IMAGE"=:PR1RUT; "W1IMAGE"=:PW1RUT; "WIMBACK"=:PWCLOS
133472                CALL CHBSIZ; GO SMLEAVE
133474          FI
133474          IF LEGAREA BIT BSAVE THEN
133477             CALL 6SOPEN
133500             "R1SAVE"=:PR1RUT; "W1SAVE"=:PW1RUT; "WXSAVE"=:PWCLOS
133506             CALL CHBSIZ; GO SMLEAVE
133510          FI
133510          GO SMLEAVE
133511   RBUS
133535
133535   %=================================================================
133535   % 28.60       X I N S T     X M R E M
133535   %
133535   % COMMAND: START-XMSG
133535   %           STOP-XMSG
133535   %
133535
133535   SUBR XINST,XMREM
```

```
133535   INTEGER NLTXT:='$ERROR: XMSG has not been loaded'
133556   INTEGER NGTXT:='$ERROR: This Sintran has not been generated for XMSG'
133611   INTEGER CZTXT:='$ERROR: Command timeout - no reaction from XMSG'
133641   INTEGER ARTXT:='$ERROR: XMSG is already running: Use STOP-XMSG to stop it!'
133677   INTEGER STTXT:='$ERROR: XMSG startup failed: see error device for cause'
133733   INTEGER OK1TX:='$OK: XMSG started.'
133745   INTEGER OK2TX:='$OK: XMSG terminated.'
133760   INTEGER XPAR1:=5XSG1,XPAR2,PPAR1:=XPAR1,PPAR2:=XPAR2 % PARAMS FOR FIXC/UNFIX
133764   INTEGER YPAR1,YPARP:=YPAR1,HPS:=("31","1"),CNT   % PARAMS 4 RT AND ABORT
133771
133771   XINST: L=:D; CALL SMENTER; CALL XMCHK;            % CHECK LOADED.
133774          IF XMSGU(4)=-3 THEN "ARTXT"; GO XMUT; FI  % ALREADY RUNNING?
134003          -1=:XMSGU(4); -12=:CNT; "YPARP"; *MON 2RT % set flag and START XROUT
134012          DO CALL DUMF; WHILE T<=0; IF XMSGU(4)>0 GO STER; CALL WAIF; OD % Wait til started
134023          -3=:XMSGU(4); "OK1TX"; GO XMUT            % FLAG RUNNING
134030   STER:  "STTXT"; GO XMUT
134032
134032   XMREM: L=:D; CALL SMENTER; CALL XMCHK;            % CHECK LOAD. SAVE FLAG
134035          IF XMSGU(4) < -2 THEN                     % IS XMSG RUNNING AT ALL?
134042             -2=:XMSGU(4); "YPARP"; *MON 2ABOR      % FLAG "TERMINATE"; ABORT XROUT
134047             -12=:CNT; "YPARP"; *MON 2RT            % Kick him
134053             DO CALL DUMF; WHILE T>0; CALL WAIF OD  % Wait til stopped
134060          FI; "OK2TX"; GO XMUT
134062
134062   % local routine to execute a dummy call to XMSG
134062   DUMF:  T:=0; *MON 2XMSG; EXIT
134065   % Local routine to check timeout count and wait
134065   WAIF:  IF CNT+1=0 GO CZXMS; A=:CNT; "HPS"; *MON 2HOLD; EXIT
134074   % LOCAL ROUTINE TO CHECK XMSG LOADED AND RETURN X=SEG. ADDR, D=FLAG
134074   % ALSO SETS UP ADDRESS OF XROUT IN PARAMETER LIST FOR RT/ABORT
134074   INTEGER XROUN:='XROUT',XROUP:=XROUN
134100   XMCHK: X:="XMSGU";
134101          IF X.S2=0 GO NGXMS; IF X.S4=0 GO NLXMS    % CHECK GENERATED AND LOADED
134105          "XROUP"; *MON 2GRTD                       % GET RT ADDRESS OF XROUT
134107          IF <0 GO NLXMS; A=:YPAR1                  % SAVE IF VALID
134111          5XSG1*5SEGSIZE+SEGSTART=:X;               % ADDR OF SEGMENT DESCRIPTOR
134115          IF X.FLAG=:D BIT 5DEMAND GO NLXMS         % MUST BE NON-DEMAND
134121          EXIT
134122
134122   CZXMS: "CZTXT"; GO CERR
134124   NGXMS: "NGTXT"; GO CERR
134126   NLXMS: "NLTXT"
134127   CERR:
134127   XMUT:  CALL SMOUTTEXT; GO SMLEAVE
134131   RBUS
134161
134161   %==========================================================================
134161   % 28.61       R E M C O M
134161   %
134161   % COMMAND:  REMOVE-SINTRAN-COMMAND <COMMAND>
134161   %
134161   % A PART OF THE ROUTINE IS LOCATED ON SEGMENT 3 (OPCOM SEGMENT)
134161   %
134161   SUBR REMCMM
134161
134161   DISP -200; INTEGER CCOMADR,LEGAREA; PSID
134161   INTEGER PWSCB:="5OPSEG"
134162
134162   REMCMM: L=:D; CALL SMENTER; CALL OPSYS(OSCPROT); GO SMLEAVE
134167           T=:CCOMADR; 5; CALL SELAREA; A=:LEGAREA
```

```
134173          IF A BIT BMEMO THEN
134175                CCOMADR=:X; T:=5OPSEG; CALL GET1L; GO 6ERROPER
134202                X:=A; #**; CALL PUT1L; GO 6ERROPER
134206                X:=CCOMADR+2; A:="CCDUM"; CALL PUT1L; GO 6ERROPER
134213                X:=CCOMADR+1; A:=0;       CALL PUT1L; GO 6ERROPER  % CCDUM IS ON COM.SEG
134220                "PWSCB"; *MON 2WSEG
134222          FI
134222          IF LEGAREA BIT BSAVE THEN
134225                CALL 6MOPEN                % MACM-AREA
134226                CCOMADR-BCSTA=:X; CALL R1SAVE; A-BCSTA=:X; #**; CALL W1SAVE
134236                CCOMADR-BCSTA+2=:X; A:="CCDUM"; CALL W1SAVE
134244                CCOMADR-BCSTA+1=:X; A:=0; CALL W1SAVE; CALL WXSAVE
134253          FI; GO SMLEAVE
134254   RBUS
134274
134274   %=========================================================================
134274   % 28.62       C C C O M
134274   %
134274   % COMMAND:  CC
134274   %
134274   % COMMENT COMMAND (NO OPERATION)
134274   %
134274   SUBR CCCOM
134274   CCCOM: L=:D; CALL SMENTER; GO SMLEAVE
134277   RBUS
134301
134301   %=========================================================================
134301   % 28.63       S E N C T   C E N C T
134301   %
134301   % COMMAND: SET-MAX-ENTER-COUNT <TERMINAL NUMBER> <VALUE>
134301   %          CLEAR-ENTER-COUNT <TERMINAL NUMBER>
134301   %
134301   SUBR SENCT,CENCT
134301
134301   DISP -200
134301   INTEGER CTRMNO,LEGAREA,CSEGN,CDFAD,ENFLG,CENCNT,DYNAL,CINDX
134301   PSID
134301   INTEGER QWANT:='N DO YOU WANT TO USE THE ENTER COUNT FEATURE? '
134331   INTEGER QMAXE:='IDMAX ENTER COUNT: '
134343   INTEGER TXENACT:='$THIS LOGICAL UNIT IS NOT CONNECTED TO A BACKGROUND PROGRAM$'
134402   INTEGER PWCSGB:=BFIELD+CSEGN
134403
134403   SENCT: A:=0; GO L1
134405   CENCT: A:=1
134406   L1:    L=:D; CALL SMENTER; A=:ENFLG
134411          "PTERMNO"; CALL SAGPAR; GO SSMON; TTNO; A=:CTRMNO
134416          IF ENFLG=0 THEN
134420   L2:       "QWANT"; CALL SMSGPAR; GO SSMON; CALL SMYESNO
134424             IF A=-1 GO 6ERRPAR
134427             IF A=0 THEN -1=:CENCNT
134432                    ELSE "QMAXE"; CALL SAGPAR; GO SSMON; A:=12; A=:CENCNT
134440             FI
134440          FI
134440          IF ENFLG=0 THEN 5 ELSE 1; FI
134445          CALL SELAREA; IF A=0 GO SMLEAVE; A=:LEGAREA
134451          CTRMNO; CALL LOGPH; IF A=0 GO 6ERRPAR; A=:CDFAD
134456          CALL SMCHTERM; GO 6ERRPAR
134460          X:=CDFAD; 0=:DYNAL; CALL SMDYNALLOC; GO NODYN; A=:CINDX; MIN DYNAL
134466          IF ENFLG=0 THEN
134470             CINDX*5PRVT+"PRVTTABLE"+"PRVMAXCT"=:X; 5PT3S=:CSEGN; GO FELLS
```

```
134500          FI;
134500   NODYN: X:=CDFAD; T:="DBPROG"; CALL XGTDFADDR; IF A=0 GO 6ERRPAR
134505          A.SEGM SHZ -10=:CSEGN
134511          IF ENFLG=0 THEN
134513             X:="MAXCT"
134514          ELSE
134515             A:=X; X:=CDFAD
134517             CALL SMTACTIVE; GO ENACT
134521             X:="ENTCT"; 0=:CENCNT
134523          FI
134523   FELLS: IF LEGAREA BIT BMEMO THEN
134526             T:=CSEGN; CENCNT; CALL PUT1L; GO 6ERROPER
134532          FI
134532          IF LEGAREA BIT BSAVE THEN
134535             IF DYNAL><0 THEN
134537                5IPT3S=:CSEGN; CENCNT; CALL PUT1L; GO 6ERROPER
134544             ELSE
134545                CALL 6SOPEN
134546                CENCNT; CALL W1SAVE; CALL WXSAVE
134551             FI
134551          FI; "PWCSGB"; *MON 2WSEG
134553          IF DYNAL><0 THEN
134555             IF CSEGN=5IPT3S THEN 5PT3S ELSE A:=T FI
134564             A=:CSEGN; "PWCSGB"; *MON 2WSEG
134567          FI; GO SMLEAVE
134570   ENACT: "TXENACT"; CALL SMOUTTEXT; GO SMLEAVE
134573   RBUS
134631   %==========================================================================
134631   %       SEGMENT-WRITE-PROTECT    <SEGMENT NUMBER>
134631   %       SEGMENT-WRITE-PERMITT    <SEGMENT NUMBER>
134631   %
134631   SUBR SGWPR,SGWPE
134631
134631   DISP -200; INTEGER CFLAG; PSID
134631
134631   INTEGER BADSEG:='BAD SEGMENT'
134637   INTEGER TXILSG:='NOT ALLOWED WHEN SEGMENT IS IN USE BY YOU'
134664
134664   SGWPR: L=:D; CALL SMENTER; A:=0; GO FELLS
134670   SGWPE: L=:D; CALL SMENTER; A:=1
134673   FELLS: A=:CFLAG; "6PSEGNO"; CALL SMSGPAR; GO SSMON
134677          IF A<2 OR A>SGMAX GO ERR
134705          A=:D*5SEGSIZE+SEGSTART=:X
134711          IF X.FLAG=0 OR A BIT 5INHB GO ERR
134715          IF CFLAG=0 THEN CALL SGAND; GO ERR1; ELSE CALL SGOR; GO ERR1 FI     % IN RESIDENT
134724          GO SMLEAVE
134725   ERR:   "BADSEG"; CALL SMOUTTEXT; GO SMLEAVE
134730   ERR1:  "TXILSG"; CALL SMOUTTEXT; GO SMLEAVE
134733   RBUS
134750
134750   %==========================================================================
134750   % 28.64       D F H D L   L I H D L
134750   %
134750   % COMMAND: DEFINE-HDLC-BUFFER <LDN> <# OF PAGES>
134750   %          LIST-HDLC BUFFER <LDN> <# OF PAGES>
134750   %
134750   SUBR DFHDL,LIHDL
134750
134750   DISP -200
134750   INTEGER ROUFLG,CLOGU,LEGAREA
```

```
134750   PSID
134750   INTEGER INUSE:='$LDN RESERVED'
134757   INTEGER NOSPA:='  $SPACE NOT AVAILABLE'
134773   INTEGER BF1:='$ BANK NO (OCT): '
135004   INTEGER BF2:=' ADDRESS (OCT): '
135015   INTEGER BSIZE:=' BUFFER SIZE (BYTES DEC): '
135033   INTEGER CSIZE:='IDBUFFER SIZE (PAGES DEC): '
135051   INTEGER HLDN,HIOF,HIRET:=1
135054   INTEGER HHLDN:=HLDN,HHIOF:=HIOF,HHIRET:=HIRET
135057   INTEGER PAR1,PAR2,PAR3,PAR4,PAR5,PAR22
135065   INTEGER PPAR1:=PAR1,PPAR2:=PAR2,PPAR3:=PAR3,PPAR4:=PAR4,PPAR5:=PAR5
135072   DFHDL: L=:D; CALL SMENTER
135074          0=:ROUFLG; GO L1
135076   LIHDL: L=:D; CALL SMENTER; 1=:ROUFLG
135102   L1:    "6PLOGU"; CALL SMSGPAR; GO SSMON; A=:X=:HLDN
135107          A/\1=:HIOF:=X; CALL LOGPH; A=:X
135114          IF HIOF = 0 THEN
135116            IF X = 0 GO FAR ERR1; X=:CLOGU
135121          ELSE
135122            IF A:=D=0 GO FAR ERR1; A=:CLOGU
135126          FI
135126          IF CLOGU.TYPRING><0 OR X.HXCC><HXCOD GO FAR ERR1     % NOT HDLC DEVICE
135135          IF ROUFLG><0 GO FAR EFI                              % MUST RESERVE IF DEFINE
135140             "HHLDN"; * MON 122
135142             IF A >< 0 GO FAR ERR2
135144             1; CALL SELAREA; A=:LEGAREA
135147             IF A  = 0 GO FAR HDLEAV
135151             "CSIZE"; CALL SMSGPAR; GO FAR HDLEAV; A=:PAR22
135155             IF A >= 40 GO FAR HDLEAV
135160             0=:PAR1; HLDN=:PAR2; 6=:PAR4=:PAR5
135166             0=:PAR3                                          % THIS IS THE DCB (FUNCTION 0 IS SUPER MASTER
135167             "PPAR1"; *MON 201                                % DO A SUPER MASTER CLEAR
135171             IF CLOGU.SWBUF >< 0 THEN                         % BUFFERS FIXED BEFORE ?
135174                6=:PAR1; X.FIXID=:PAR2; X.LWPHY=:PAR3         % YES, UNFIX THEM
135202                "PPAR1"; * MON 61; JMP *+1
135205             FI
135205             0=:X.SWBUF; 0=:X.MAX; 5=:PAR1
135211             IF PAR22=:PAR2 = 0 GO FAR HDLEAV                  % ASKED FOR 0 PAGES
135215             0=:PAR3
135216             GO L2; *)FILL
135236
135236   INTEGER POINTER POPA1:=PAR1,POPA2:=PAR2,POPA3:=PAR3,POPA4:=PAR4
135242   INTEGER CMXP
135243   INTEGER POINTER LREG
135244
135244   % SUBROUTINE TO CHECK WHER IN PHYSICAL MEMORY THE HDLC-BUFFERS MAY BE PLACED
135244   CSUBR:     A;=L=:"LREG"; X:=0
135247              DO WHILE X<<"CUMSIZE*2"
135252                 X=:L
135253                 A:="CUMTABLE"; X+A; T:=0; *LDDTX
135257                 IF A<<=POPA3 AND D+1>>T THEN T:=D=:POPA3 FI
135267                 IF X:=POPA3>> POPA4 GO LREG
135273                 IF A<<=POPA4 AND A>>=POPA3 THEN A-1=:POPA4 FI
135303                 X:=L+2
135305              OD; MIN "LREG"; GO LREG
135310
135310   L2:        T:=CORMBANK; X:=ECORMAP; *LDATX DPAGP
135313              A=:CMXP
135314              DO
135314                 POPA3/\177700+77=:POPA4                       % DO NOT CROSS MEM.BANKS
```

```
135320                  CALL CSUBR; GO MBERR; POPA3/\177700=:T
135325                  IF POPA4>>T+77 THEN T=:POPA4 FI
135332                  "PPAR1"; * MON 61
135334                  GO MBERR; GO OK
135336   MBERR:         IF POPA3>>CMXP GO ERR4                          % NO BUFFER SPACE AVAILABLE
135342                  POPA3+1=:POPA3                                  % TRY NEXT PAGE
135345               OD
135346   OK:         T=:CLOGU.FIXID; A=:X.LWPHY; 1=:X.SWBUF; POPA2 SHZ 13 =: X.MAX   % UPDATE DATAFIELD  MAX IN BYTES
135356               X.LWPHY /\ 77 SHZ 12 =: X.BUFST                   % START WITHIN BANK
135362               X.LWPHY /\ 177700 SHZ -6 =: X.MASTB               % BANK NUMBER
135366               0=:X.HINIF                                        % BUFFERPOOL WILL BE INITIATED
135367               GO HDLEAV; *)FILL
135376   EFI:
135376            "BF1"; CALL SMOUTTEXT; CLOGU.MASTB; CALL SMOCTU
135403            "BF2"; CALL SMOUTTEXT; CLOGU.BUFST; CALL SMOCTU
135410            "BSIZE"; CALL SMOUTTEXT
135412            CLOGU.MAX; CALL SMDTDEC; GO HDLEAV
135416   ERR1:    GO 6ERRPAR
135417   ERR2:    "INUSE"; CALL SMOUTTEXT; GO HDLEAV
135422   ERR4:    "NOSPACE"; CALL SMOUTTEXT
135424   HDLEAV:IF ROUFLG=0 THEN                                       % RELEASE HDLC
135426               "HHLDN"; *MON 123
135430            FI; GO SMLEAVE
135431   RBUS
135444
135444   @DEV 1
135444   @DEV (S-S-J)SINE-3
```

```
135444
135444  %==========================================================================
135444  %
135444  %   29.0      M A I L   S Y S T E M
135444  %
135444  %==========================================================================
135444  % 29.1        G L O B A L   D A T A
135444  %
135444
135444  INTEGER STMSYS:=0                % START/STOP FLAG
135445  INTEGER FNO                      % OPEN FILE NUMBER FOR MAILBOX
135446  % ALSO FOUND ON BLOCK 0:
135446  INTEGER MESMAX(3)                % MAX.NUMBER OF MESAGES
135451  INTEGER ARRAY ORBITTAB(400)      % A BIT FOR EACH USER HAVING MAIL
136051
136051  % MESSAGE TRANSFER BUFFER:
136051  INTEGER FRUSER                   % FROM USER INDEX (-1 IF FREE ENTRY)
136052  DOUBLE DATIM                     % PACKED DATE AND TIME
136054  INTEGER ARRAY BITTAB(400)        % A BIT FOR EACH USER WAITING FOR THIS MESSAGE
136454  INTEGER ARRAY MESBUF(375)        % THE MESSAGE
137051
137051  INTEGER MLCTAB:=MLCOMTAB         % START OF COMMAND TABLE
137052
137052
137052  %==========================================================================
137052  % 29.2        M A I L
137052  %
137052  %  (SINTRAN)COMMAND:  MAIL <OUTPUT FILE>
137052
137052  SUBR SMMAIL
137052  INTEGER FILNAME:='(SYSTEM)MAILBOX:DATA',FTYPE:='DATA'
137070  INTEGER STROUT:='S OUTPUT FILE: '
137100  INTEGER XNOINI:='NOT INITIALIZED',XNOTST:='NOT STARTED'
137116  INTEGER XNOMAIL:='NO MAILBOX FILE: ',XPROT:='PROTECTED COMMAND'
137140  INTEGER XERNOTF:='NO SUCH COMMAND',XERAMB:='AMBIGUOUS'
137155  INTEGER MLSEM:=("5MLSEM",NULL,"1"), PWSBC;="5OP2SEG",STREG=?,SLREG=?
137161  *)FILL
137164  INTEGER POINTER CCUSER:=CUSER
137165  DISP -200
137165          INTEGER SAVECUSER,INDXX
137165          INTEGER POINTER FUNC
137165  PSID
137165
137165  SMMAIL: T=:STREG; A:=L=:SLREG
137170          CCUSER=:SAVECUSER                          % MOD. 26/6/80 FOR
137172          CALL FILSYS(GSYSI); GO FILERR; T=:CCUSER        % 4096 USERS
137176          X:="FILNAME"; "FTYPE"; T:=2; *MON 2NOPE
137202          GO NOMAIL; A=:FNO; SAVECUSER=:CCUSER
137206          "0"; CALL RBUFF; GO NOINI; IF FRUSER<=0 GO NOINI
137214          IF STMSYS=0 GO NOTSTART
137216  % PRINT WAITING MAIL:
137216          X:=CURUSER; CALL BTLOAD
137220          IF ><0 THEN
137221                "STROUT"; CALL SAGPAR; GO FAR XEXI; 1
137225                IF ><1 THEN A=:X:="TYPS"; T:=5; *MON 2NOPE; MON 2QERM
137235                FI; A=:TDVN; 1=:INDXX
137240                X:=CURUSER; CALL BTCLEAR; X:="BITTAB"; "ORBITTAB"
137244                T:=400; CALL COPYB
137246                "0"; CALL WBUFF; GO SMFILERR; 1=:INDXX
137253                FOR INDXX TO MESMAX DO CALL RBUFF; GO OUT
```

```
137261                    IF FRUSER><-1 THEN X:=CURUSER; CALL BTLOAD
137267                      IF ><0 THEN CALL BTCLEAR; CALL WMESS; A:=0
137273                        FOR X:=0 TO 377 DO A\/BITTAB(X) OD
137302                        IF =0 THEN -1=:FRUSER FI; INDXX; CALL WBUFF; GO SMFILERR
137310                    FI FI
137310                  OD
137314   OUT:           T:=TDVN; *MON 2CLOS; 0
137317          FI; GO COMN
137320
137320   INTEGER STREG,SLREG
137322   *)FILL
137351   %ERROR EXITS:
137351   NOMAIL: SAVECUSER=:CCUSER; "XNOMAIL"; CALL SMOUTTEXT; "FILNAME"
137356   ERR:    CALL SMOUTTEXT; GO EXI
137360
137360   NOINI: "XNOINI"; GO ERTEST
137362   NOTSTART: "XNOTSTART"
137363   ERTEST: CALL SMOUTTEXT; IF PASSTYPE=2 GO COMN; GO EXI
137371
137371   % COMMAND PROCESSING:
137371   COMN:  DO      CALL ESCOFF; 1=:TDVN; CALL SMCRLF; ###; CALL SMTCO; CALL SMGCOM
137400                  X:="MLCOMTAB"; CALL SMABLOOK; T=:X
137403                  IF =-1 GO ERNOTF; IF =-2 GO ERAMB
137411                  IF X.CPROTECT>PASSTYPE GO ERRC
137415                  X.CMAND=:"FUNC"; IF =0 GO EXI; CALL FUNC
137421          OD
137422   EXI:   T:=FNO; *MON 2CLOS; JMP *1
137425          "PWSBC"; *MON 2WSEG
137427          "MLSEM"; *MON 2RELE
137431   XEXI:  T:=STREG; A:=SLREG=:L; *MON 2MEXI
137435
137435   % ERROR EXITS:
137435   ERRC:   "XPROT"; GO ERCO
137437   ERNOTF: "XERNOTF"; GO ERCO
137441   ERAMB: "XERAMB"
137442   ERCO:  CALL SMOUTTEXT; GO COMN
137444   *)FILL
137466
137466
137466   %=========================================================================
137466   % 29.3        C O M M A N D   T A B L E
137466   %
137466
137466   @ICR
137466   INTEGER ARRAY MLCOMTAB:=(
137466   ML0,0,MLHELP,0,        ML1,0,0,0,
137476   ML2,0,0,0,             ML3,0,MLINIT,2,
137506   ML4,0,MLBROAD,2,       ML5,0,MLDBROAD,2,
137516   ML6,0,MLSEND,0,        ML7,0,MLDSEND,0,
137526   ML8,0,MLLMESS,0,       ML9,0,MLLBROAD,2,
137536   ML10,0,MLDLMESS,0,     ML11,0,MLDLBROAD,2,
137546   ML12,0,MLSTOP,2,       ML13,0,MLRUN,2,-1),
137557
137557   ML0:='HELP',ML1:='EXIT',
137565   ML2:='FINISH',ML3:='INITIALIZE',
137577   ML4:='BROADCAST',ML5:='DIRECT-BROADCAST',
137615   ML6:='SEND-MESSAGE',ML7:='SEND-DIRECT-MESSAGE',
137636   ML8:='LIST-MESSAGE',ML9:='LIST-BROADCASTS',
137655   ML10:='DELETE-MESSAGE',ML11:='DELETE-BROADCAST',
137676   ML12:='STOP-MAIL-SYSTEM',ML13:='RUN-MAIL-SYSTEM';
```

```
137717   ©CR;
137717   RBUS
137717
137717   %=========================================================================
137717   % 29.4        Y O U H A V E M A I L
137717   %
137717   %
137717   % SUBROUTINE CALLED FROM LOGIN AND LOGOUT ON OPSEG (3)
137717   % RETURN:  NO MAIL
137717   % SKIPRETURN:  MAIL WAITING
137717
137717   SUBR YOUHAVEMAIL
137717
137717   INTEGER YOUMA:=(7\40,'$ *** YOU HAVE MAIL ***$')
137735   YOUHAVEMAIL: L=:D; CALL SMENTER
137737          IF STMSYS=0 GO SMLEAVE; X:=CURUSER; CALL XBTLOAD; GO SMLEAVE
137745          IF ><0 THEN
137746             T:=1; A:=16; *MON 2SYCN; JMP * 1        % (BAD)
137752             "YOUMA"; CALL SMOUTTEXT
137754             T:=1; A:=17; *MON 2SYCN; JMP * 1        % (BAD)
137760             GO SM2LEAV
137761          FI
137761          GO SMLEAVE
137762   RBUS
137771
137771   %=========================================================================
137771   %
137771   %              M A I L   C O M M A N D S
137771   %
137771   %=========================================================================
137771   % 29.5        M L H E L P
137771   %
137771   %  COMMAND: HELP
137771
137771   SUBR MLHELP
137771   MLHELP: L=:D; CALL SMENTER; CALL ESCON; X:=MLCTAB
137775          DO WHILE X.CNAME><-1; CALL SMOUTTEXT; CALL SMCRLF; X+4 OD
140005          GO SMLEAVE
140006   RBUS
140014
140014   %=========================================================================
140014   % 29.6        M L I N I T
140014   %
140014   %  COMMAND: INITIALIZE <MAILBOX>
140014
140014   SUBR MLINIT
140014   INTEGER STRBOX:='IDMAX. NO. OF MESSAGES (512 WORDS EACH ): '
140042   DISP -200; INTEGER INDXX; PSID
140042   MLINIT: L=:D; CALL SMENTER; "STRBOX"; CALL SMSGPAR; GO SMLEAVE
140047          A=:MESMAX=:FRUSER
140051          FOR X:=0 TO 377 DO 0=:BITTAB(X) OD
140060          "0"; CALL WBUFF; GO SMFILERR; 1=:INDXX
140065          FOR INDXX TO MESMAX DO CALL RBUFF; GO OUT; -1=:FRUSER
140075                INDXX; CALL WBUFF; GO SMFILERR
140100          OD
140104   OUT:   0=:STMSYS; GO SMLEAVE
140106   RBUS
140122
140122   %=========================================================================
140122   % 29.7        M L S T O P
```

```
140122   %
140122   % COMMAND: STOP-MAIL-SYSTEM
140122
140122   SUBR MLSTOP
140122   MLSTOP: 0=:STMSYS; EXIT
140124   RBUS
140125
140125   %==========================================================================
140125   % 29.8        M L R U N
140125   %
140125   % COMMAND: RUN-MAIL-SYSTEM
140125
140125   SUBR MLRUN
140125   MLRUN: L=:D; CALL SMENTER; "0"; CALL RBUFF; GO SMFILERR
140132          X:="FRUSER"; "MESMAX"; T:=403; CALL COPYB
140136          1=:STMSYS; GO SMLEAVE
140141   RBUS
140152
140152   %==========================================================================
140152   % 29.9        M L S E N D   M L B R O A D
140152   %
140152   % COMMAND: SEND-MESSAGE <TO USER>
140152   %          BROADCAST
140152
140152   SUBR MLSEND,MLBROAD
140152   INTEGER NOROM:='MAILBOX FULL',TOUSER:='S TO USER: '
140167   INTEGER XMES:='$MAIL INDEX: '
140176   DISP -200; INTEGER MESNO,BROAD,INDXX,DIRIN; PSID
140176
140176   MLSEND: "0"; GO MLC
140200   MLBROAD: 1
140201   MLC:    L=:D; CALL SMENTER; A=:BROAD; 1=:MESNO
140206           FOR MESNO TO MESMAX DO A=:X; CALL RBUFF; CALL EXPAND
140215                IF FRUSER=-1 GO FOUND    % FREE ENTRY
140221           OD; "NOROM"; GO ERR
140227   FOUND:  FOR X:=0 TO 377 DO 0=:BITTAB(X) OD
140236           IF BROAD><0 THEN -1 =: DIRIN % BROADCAST
140242              DO                          % MOD. 25/6/80 FOR 4096 USERS
140242                 T := DIRIN; CALL FILSYS(GNEXM); GO ENDL
140246                 T=:DIRIN; 0=:INDXX
140250                 FOR INDXX TO 377 DO
140254                    T:=DIRIN; X:="WORKA"
140256                    CALL FILSYS(GUSEN); GO NOUS
140261                    T SHZ 10; X:=INDXX \/T; CALL BTSTORE        % X = DIR \ USER
140265   NOUS:         OD
140271              OD
140272   ENDL:      40000=:FRUSER
140274           ELSE                    % SINGLE MESSAGE
140275                 "TOUSER"; CALL SMSGPAR; GO SMLEAVE; A=:X
140301                 CALL FILSYS(GMUSI); GO ERNOUS
140304                 T =: X; CALL BTSTORE
140306                 CURUSER=:FRUSER
140310           FI
140310           CALL READMESS; CALL FILSYS(GDATE); AD=:DATIM
140314           MESNO; CALL WBUFF; GO SMFILERR
140317           FOR X:=0 TO 377 DO ORBITTAB(X)\/BITTAB(X)=:ORBITTAB(X) OD
140330           "0"; CALL RBUFF; GO SMFILERR; X:="ORBITTAB"; "BITTAB"; T:=400; CALL COPYB
140337           "0"; CALL WBUFF; GO SMFILERR
140342           "XMES"; CALL SMOUTTEXT; MESNO; CALL SMDECU
140346           GO SMLEAVE
```

```
140347   ERR:    CALL SMOUTTEXT; GO SMLEAVE
140351   ERNOUS: *MON 2ERMS
140352           GO SMLEAVE
140353   RBUS
140404
140404
140404   %==========================================================================
140404   % 29.10        E X P A N D
140404   %
140404   % SUBROUTINE TO EXPAND THE MAILBOX FILE
140404   % A=ERROR NUMBER
140404   % X=MAIL INDEX
140404
140404   SUBR EXPAND
140404   DISP -200; INTEGER INDXX; PSID
140404   EXPAND: L=:D; CALL SMENTER; IF ><22 GO SMFILERR  % NO SUCH PAGE
140411           X=:INDXX; -1=:FRUSER
140414           FOR X TO INDXX+3 DO X=:A; CALL WBUFF; GO SMFILERR OD
140425           GO SMLEAVE
140426   RBUS
140433
140433   %==========================================================================
140433   % 29.11        M L D S E N D
140433   %
140433   % COMMAND: SEND-DIRECT-MESSAGE <TERMINAL>
140433
140433   SUBR MLDSEND
140433
140433   INTEGER PARTERM:='IDTO TERMINAL NUMBER: '
140447   INTEGER XNOTERM:='NO TERMINAL'
140455
140455   DISP -200; INTEGER TERMNO,TERMDFIELD,SAVTYP,SVSCREEN,ITYPR,CODF; PSID
140455
140455   MLDSEND: L=:D; CALL SMENTER
140457           "PARTERM"; CALL SMSGPAR                    % GET DESTINATION DEVICE
140461           GO SMLEAVE
140462           IF A=1 THEN A:=1206 FI
140466           A=:TERMNO
140467           CALL LOGPH; T:=D=:CODF; CALL SMCHTERM
140473           GO NOTERM; A=:TERMDFIELD                   % STORE INPUT DATAFIELD OR LEAVE
140475           IF A.TYPRING NBIT 5TERM AND A NBIT 5BAD GO NOTERM  % NOT TERMINAL OR BAD
140503           A=:ITYPR
140504           IF A BIT 5TERM THEN
140506              IF X:=CODF=0 GO NOTERM
140510              T:="TTMR"; -2; CALL XSTDFADDR
140513              "MLTTOMR"; T:="KTMSUB"; CALL XSTDFADDR
140516              RTREF=:CURMAIL
140520           ELSE
140521              IF X:=X.DFOPP=0 GO NOTERM
140523           FI
140523           CURUSER=:FRUSER; CALL READMESS             % GET MESSAGE
140526           CALL FILSYS(GDATE); AD=:DATIM            % PACK TIME
140531           X.TYPRING=:SAVTYP                        % SAVE OLD TYPRING
140533           A BONE 5NORESRV=:X.TYPRING               % AVOID RESERVING DEVICE
140535           T:="SCREEN"; CALL XGTDFADDR; A=:SVSCREEN
140540           T:="SCREEN"; A:=0; CALL XSTDFADDR
140543           TERMNO=:TDVN=:T
140546           A:=16; *MON 2SYCN; JMP * 1               % (BAD)
140551           7:CALL SMTCO                             % WRITE BELL
140553           CALL WMESS                               % WRITE MESSAGE
```

```
140554          T:=TDVN; A:=17; *MON 2SYCN; JMP * 1     % (BAD)
140560          SAVTYP =: X.TYPRING                     % RESTORE
140562          IF A BIT 5TERM THEN
140564             T:="TTMR"; -10; CALL XSTDFADDR
140567             T:="KTMSUB"; "TTOMR"; CALL XSTDFADDR
140572          FI; T:="SCREEN"; SVSCREEN; CALL XSTDFADDR
140575          1=:TDVN
140577          GO SMLEAVE
140600
140600   NOTERM: "XNOTERM"; CALL SMOUTTEXT; GO SMLEAVE
140603
140603   RBUS
140634
140634   %=============================================================================
140634   % 29.12       M L D B R O A
140634   %
140634   % COMMAND: DIRECT-BROADCAST
140634
140634   SUBR MLDBROA
140634
140634   DISP -200
140634   INTEGER POINTER TPNT,TPN2; INTEGER SAVTYP,SVSCREEN,ITYPR,CODF,IDF
140634   PSID
140634
140634   MLDBROA: L=:D; CALL SMENTER; CALL READMESS; 40000=:FRUSER
140641          CALL FILSYS(GDATE); AD=:DATIM; "BACKTAB"=:"TPNT"
140646   LOOP:  IF X:=TPNT=-1 GO FAR ELOOP
140652             IF X=0 GO FAR NEXT
140654                IF X.TYPRING NBIT 5TERM AND A NBIT 5BAD GO FAR NEXT
140661                A=:ITYPR; X=:IDF
140663                FOR X:=0 TO 2477 DO     % FIND LOG. NO.
140667                   X=:A; CALL LOGPH; IF =TPNT GO OUT
140674                OD; CALL ERRFATAL
140677   OUT:         IF X=1 THEN X:=1206 FI; X=:TDVN
140704                A:=X; CALL LOGPH; A:=D=:CODF; IF A=0 GO FAR NEXT
140712                X:="TMRTE"-2
140714                DO CALL LDPIOF WHILE A><-1; IF =D GO OUT2; X+1; OD
140724                GO FAR NEXT; *)FILL                     % NOT IN TIMER TABLE
140745   OUT2:        IDF.TYPRING; X:=CODF
140750                IF A BIT 5TERM THEN
140752                   T:="TTMR"; -2; CALL XSTDFADDR
140755                   T:="KTMSUB"; "MLTTOMR"; CALL XSTDFADDR
140760                   RTREF=:CURMAIL
140762                   X.TYPRING=:SAVTYP BONE 5NORESERV=:X.TYPRING
140766                   T:="SCREEN"; CALL XGTDFADDR; A=:SVSCREEN
140771                   T:="SCREEN"; A:=0; CALL XSTDFADDR
140774                   7; CALL SMTCO; CALL WMESS
140777                   T:="TTMR"; -10; CALL XSTDFADDR
141002                   T:="KTMSUB"; "TTOMR"; CALL XSTDFADDR
141005                   T:="SCREEN"; SVSCREEN; CALL XSTDFADDR
141010                ELSE
141011                   X.TYPRING=:SAVTYP BONE 5NORESERV=:X.TYPRING
141015                   X.SCREEN=:SVSCREEN; 0=:X.SCREEN
141020                   T:=TDVN; 16; *MON 2SYCN; JMP * 1
141024                   7; CALL SMTCO; CALL WMESS
141027                   T:=TDVN; 17; *MON 2SYCN; JMP * 1
141033                   SVSCREEN=:X.SCREEN
141035                FI
141035                1=:TDVN; SAVTYP=:X.TYPRING
141041   NEXT:      MIN "TPNT"
```

```
141042          GO FAR LOOP
141043   ELOOP: GO SMLEAVE
141044   RBUS
141062
141062   %==========================================================================
141062   % 29.13       M L L M E S S   M L L B R O A D
141062   %
141062   % COMMAND: LIST-MESSAGES <OUTPUT FILE>
141062   %          LIST-BROADCASTS <OUTPUT FILE>
141062
141062   SUBR MLLMESS,MLLBROAD
141062   INTEGER STROUT:='S OUTPUT FILE: ',IND:='$MAIL INDEX:'
141101   INTEGER TOU:='$TO USER  ',BADM:='$BAD MAIL ENTRY$'.
141120   DISP -200; INTEGER BROAD,INDXX,USCOUNT,XUSNO; PSID
141120
141120   MLLMESS: "0"; GO MLLIST
141122   MLLBROA: 1
141123   MLLIST: L=:D; CALL SMENTER; A=:BROAD; "STROUT"; CALL SAGPAR; GO SMLEAVE; 1
141132          IF ><1 THEN A=:X;="TYPS"; T:=0; *MON 2NOPE
141141                GO ERRNOTF
141142          FI; A=:TDVN; 1=:INDXX
141145          FOR INDXX TO MESMAX DO CALL RBUFF; GO OUT
141153                IF FRUSER><-1 THEN                          % MOD. 25/6/80
141157                   IF A><CURUSER THEN CALL FILSYS(TUSSY); GO NOTSY; FI
141165                   0=:USCOUNT=:XUSNO                        % FOR 4096 USERS
141167                  FOR X:=0 TO 7777  DO CALL BTLOAD
141174                    IF ><0 THEN MIN USCOUNT; X=:XUSNO FI
141177                  OD
141201                  "IND"; CALL SMOUTTEXT; INDXX; CALL SMDECU
141205                  IF BROAD=0 THEN
141207                   IF FRUSER<40000 THEN
141213                    IF >=0 AND USCOUNT=1 THEN
141220                      "TOU"; CALL SMOUTTEXT; A:=XUSNO/\377; T:=XUSNO SHZ -10
141226                      X:="WORKA"; CALL FILSYS(GUSEN); GO OUT2
141232                      "WORKA"; CALL SMOUTTEXT
141234   OUT2:              CALL WMESS
141235                    ELSE "BADM"; CALL SMOUTTEXT
141240                    FI
141240                   FI
141240                  ELSE          % POSSIBLY  BROADCAST
141241                    IF FRUSER=40000 THEN CALL WMESS FI
141246                FI FI
141246   NOTSY: OD
141252   OUT:   T:=TDVN; *MON 2CLOS; 0
141255          GO SMLEAVE
141256   ERRNOF: *MON 2ERMS
141257          GO SMLEAVE
141260   RBUS
141305
141305   %==========================================================================
141305   % 29.14       M L D L M   M L D L B R O A
141305   %
141305   % COMMAND: DELETE-MESSAGE <INDEX>
141305   %          DELETE-BROADCAST <INDEX>
141305
141305   SUBR MLDLM,MLDLBROA
141305   INTEGER STRNO:='IDMAIL INDEX: ',NOMAIL:='NO MAIL'
141321   INTEGER NOTMES:='NOT MESSAGE',NOTBROA:='NOT BROADCAST'
141336   INTEGER NOTYOU:='NOT SENT BY YOU'
141346   DISP -200; INTEGER BROAD,MESNO,INDXX; PSID
```

```
141346
141346   MLDLM: "0"; GO DELM
141350   MLDLBROA: 1
141351   DELM:   L=:D; CALL SMENTER; A=:BROAD
141354           "STRNO"; CALL SMSGPAR; GO SMLEAVE; A=:MESNO; IF <0 OR >MESMAX GO ERRN
141364           CALL RBUFF; GO ERRN; IF FRUSER=-1 GO ERRN
141372           IF BROAD=0 AND FRUSER>=40000 THEN "NOTMES"; GO ERR FI
141402           IF BROAD=1 AND FRUSER><40000 THEN "NOTBROA"; GO ERR FI
141414           IF CURUSER><FRUSER THEN CALL FILSYS(TUSSY) GO NOTY; FI
141423           -1=:FRUSER; MESNO; CALL WBUFF; GO SMFILERR
141430   % UPDATE ORBITTAB:
141430           FOR X:=0 TO 377 DO 0=:ORBITTAB(X) OD; 1=:INDXX
141441           FOR INDXX TO MESMAX DO CALL RBUFF; GO OUT
141447                 IF FRUSER><-1 THEN
141453                   FOR X:=0 TO 377 DO ORBITTAB(X)\/BITTAB(X)=:ORBITTAB(X) OD
141464                 FI
141464           OD
141470   OUT:    "0"; CALL RBUFF; GO SMFILERR; X:="ORBITTAB"; "BITTAB"; T:=400; CALL COPYB
141477           "0"; CALL WBUFF; GO SMFILERR
141502           GO SMLEAVE
141503   ERRN:   "NOMAIL"
141504   ERR:    CALL SMOUTTEXT; GO SMLEAVE
141506   NOTY:   "NOTYOU"; GO ERR
141510   RBUS
141535
141535   %==========================================================================
141535   %
141535   %              S U B R O U T I N E S
141535   %
141535   %==========================================================================
141535   % 29.15        R E A D M E S S
141535   %
141535
141535   % SUBROUTINE TO READ MESSAGE FROM TERMINAL AND PUT IT
141535   % IN MESSAGE BUFFER, TERMINATED BY CTRL L
141535
141535   SUBR READMESS
141535   INTEGER TYPMESS:='TYPE YOUR MESSAGE, TERMINATED BY CONTROL L:$'
141564   INTEGER BFULL:='MESSAGE BUFFER FULL'
141576
141576   READMESS: L=:D; CALL SMENTER; "TYPMESS"; CALL SMOUTTEXT
141602           FOR X:=0 TO 770 DO T:=0; *MON 2INBT; MON 2QERM
141611                 A BZERO 7; IF =15 THEN ##$ FI; T:="MESBUF"; *SBYT
141620                 IF =14 GO OUT   % CTRL L
141623           OD; "BFULL"; CALL SMOUTTEXT
141627   OUT:    ##'; T:="MESBUF"; *SBYT
141632           GO SMLEAVE
141633   RBUS
141642
141642   %==========================================================================
141642   % 29.16        W M E S S
141642   %
141642
141642   % SUBROUTINE TO OUTPUT A MESSAGE OR BROADCAST.
141642   % THE MESSAGE IS FETCHED FROM THE MESSAGE BUFFER
141642
141642   SUBR WMESS
141642   INTEGER BRCAST:='$ B R O A D C A S T
141656   INTEGER FRUS:='$ M A I L   FROM USER
141672
```

```
141672   WMESS: L=:D; CALL SMENTER
141674          IF FRUSER=40000 THEN  % BROADCAST
141700                "BRCAST"; CALL SMOUTTEXT
141702          ELSE  "FRUS"; CALL SMOUTTEXT; FRUSER/\377; T:=FRUSER SHZ -10
141711                X:="WORKA"; CALL FILSYS(GUSEN); GO OUT
141715                "WORKA"; CALL SMOUTTEXT; "40\40"; CALL SM2TCO
141721   OUT:   FI; DATIM; CALL FILSYS(LDATE); CALL SMCRLF; CALL SMCRLF
141726          CALL ESCON; "MESBUF"; CALL SMOUTTEXT; CALL SMCRLF; CALL ESCOFF
141733          GO SMLEAVE
141734   RBUS
141755
141755   %==========================================================================
141755   % 29.17       R B U F F   W B U F F
141755   %
141755
141755   % SUBROUTINES TO READ OR WRITE A MESSAGE BLOCK (512 WORDS)
141755   % ENTRY:      A=BLOCK NO
141755   % RETURN:     ERROR - A=ERROR NO.
141755   %SKIPRETURN: OK
141755
141755   SUBR RBUFF,WBUFF
141755   INTEGER PAR:=(FNO,NULL,FRUSER,BLNO,"1000"), BLNO
141763   RBUFF: A SHZ 1=:BLNO; "PAR"; *MON 2RFIL
141767          GO OUT
141770   WBUFF: A SHZ 1=:BLNO; "PAR"; *MON 2WFIL
141774   OUT:   IF =0 THEN EXITA FI; EXIT
141777   RBUS
142001
142001   %==========================================================================
142001   %            B I T   O P E R A T I O N S
142001   %
142001   % 29.18       B S T O R E   X B T L O A D   B T L O A D   B T C L E A R
142001   %
142001
142001   % ENTRY:      X=BITNO (I.E. DIR. AND USER NUMBER)
142001   %RETURN:     A><0 IF BIT SET (FOR XBTLOAD AND BTLOAD)
142001
142001   SUBR BTSTORE,XBTLOAD,BTLOAD,BTCLEAR
142001   INTEGER ILLUSER:='ILL. USER NO.'
142010   INTEGER SHAINSTR(0); *SHA
142011
142011   % SET BIT IN BITTAB:
142011   BTSTORE: L=:D; CALL SMENTER; "BITTAB":=:X; AD SHZ -4
142016           IF A>377 GO ERR
142021          X+A; SHAINSTR; D SHZ -14+A; 1; *EXR SD
142027          A\/X.S0=:X.S0; GO SMLEAVE
142032
142032   % CLEAR BIT IN BITTAB:
142032   BTCLEAR: L=:D; CALL SMENTER; "BITTAB":=:X; AD SHZ -4
142037           IF A>377 GO ERR
142042          X+A; SHAINSTR; D SHZ -14+A; 1; *EXR SD
142050          A-,/\X.S0=:X.S0; GO SMLEAVE
142054
142054   % TEST BIT IN ORBITTAB:
142054   XBTLOAD: "ORBITTAB"; L=:D; CALL SMENTER
142057          A:=:X; AD SHZ -4
142061          IF A>>377 GO SMLEAVE
142064          X+A; SHAINSTR; D SHZ -14+A; 1; *EXR SD
142072          A/\X.S0; GO SM2LEAV
142074
```

```
142074   % TEST BIT IN BITTAB:
142074   BTLOAD: "BITTAB"
142075   BTL:    L=:D; CALL SMENTER; A:=:X; AD SHZ -4
142101           IF A>377 GO ERR
142104           X+A; SHAINSTR; D SHZ -14+A; 1;*EXR SD
142112           A/\X.SO; GO SMLEAVE
142114   ERR:    "ILLUSER"; CALL SMOUTTEXT; "OPCOM"=:RETUAD; T:="6P3RET"; *MON 2MCAL
142122   RBUS
142134
142134
142134   %=================================================================
142134   % 29.19      5 B U F R   S S B U F
142134   %
142134   % BUFFERS FOR THE RTFIL ROUTINES AND THE SERVICE PROGRAM
142134   INTEGER ARRAY 5BUFR(100),SSBUF(100)
```

```
142334
142334   %=========================================================================%
142334   %                                                                         %
142334   %              L A M U     S E R V I C E     R O U T I N E S              %
142334   %                                                                         %
142334   %=========================================================================%
142334
142334   % DISP 0        % LAMU DESCRIPTOR DEFINITION
142334   %         INTEGER LAMPP          % PHYSICAL PAGE
142334   %         INTEGER LAMNP          % NUMBER OF PAGES IN LAMU
142334   %         INTEGER LAMPR          % LAMU PROTECTION
142334   % PSID
142334
142334   %              PARAMETERS WHEN CALLING MON 2LAMU
142334   INTEGER LAMFU, PAR1, PAR2, PAR3
142340   INTEGER LAPLIST:=LAMFU, PARA1:=PAR1, PARA2:=PAR2, PARA3:=PAR3
142344
142344
142344   INTEGER NSLAR:='NO SUCH LAMU AREA'
142355   INTEGER LWIAR:='LAMU WITHIN AREA'
142366   INTEGER NFLAD:='NO FREE LAMU AREA DESCRIPTOR'
142405   INTEGER TPPAG:='IOFIRST PHYSICAL PAGE: '
142421   INTEGER TNPAG:='IONUMBER OF PAGES: '
142433   INTEGER TLID :='IOLAMU ID: '
142441   INTEGER TRFP :='$RETURNED FIRST PHYSICAL PAGE: '
142461
142461   %=========================================================================
142461   %              P T L A M
142461   %       PAGES TO LAMU    <FIRST PHYSICAL PAGE>, <NUMBER OF PAGES>
142461   SUBR PTLAM
142461   DISP -200
142461          INTEGER ARIX           % LAMU AREA INDEX
142461          INTEGER FPP            % FIRST PHYSICAL PAGE
142461          INTEGER LPP            % LAST PHYSICAL PAGE
142461          INTEGER LEGAREA
142461          INTEGER FOPP           % FIRST OLD PHYSICAL PAGE
142461          INTEGER LOPP           % LAST OLD PHYSICAL PAGE
142461          INTEGER FREEADR
142461          INTEGER TABADR
142461   PSID
142461
142461   INTEGER AROC     :='$AREA OCCUPIED'
142471   INTEGER OUTPHM   :='$OUTSIDE PHYSICAL MEMORY'
142506   INTEGER INTERERR:='$INTERNAL ERROR'
142516   INTEGER INMEM    :=' IN MEMORY'
142524   INTEGER ONIMA    :=' ON IMAGE '
142532
142532   PTLAM: L=:D; CALL SMENTER
142534          "TPPAG"; CALL SMSGPAR; GO SSMON; A=:FPP
142540          IF A < 100 AND A><0 THEN "AROC"; GO FAR PTER2; FI
142546          "TNPAG"; CALL SMSGPAR; GO SSMON; A=:PAR1
142552          IF FPP = 0 THEN                          % SELECT FIRST AVABLE PAGES
142554             100=:PAR2; 37777=:PAR3
142560          ELSE IF A < 100 THEN "AROC";GO FAR PTER2
142566          ELSE
142567             A=:PAR2; A+PAR1-1=:PAR3               % USER HAS SELECTED PHYS. PAGES
142573          FI;FI
142573          3; CALL SELAREA; A=:LEGAREA
142576          IF LEGAREA BIT BMEMO THEN
142601                0=:ARIX
```

```
142602                  FOR ARIX TO "NINSZ-1" DO
142606                     A SH 1 + "LAMAR"=:X; T:=0; *LDATX
142613                     IF A = 0 THEN
142614                        5=:LAMFU; "LAPLIST"; *MON 61
142620                        GO FAR M61ER; X:="FXCTAB"+T;
142623                        IF A >< FPP AND T >< 0 THEN CALL ERRFATAL FI
142631                        IF T=0 THEN A=;PAR2 FI        % UPDATE RETURNED PHYS PAGE
142634                        T:=0; A:=0; *STATX            % CLEAR ENTRY IN FXCTAB
142637                        ARIX SH 1 + "LAMAR"=:X; T:=0; A:=PAR2+PAR1-1=:D:=PAR2; *STDTX
142652                        IF FPP=0 THEN "TRFP"; CALL SMOUTTEXT;"INMEM";CALL SMOUTTEXT
142660                          PAR2; CALL SMOCTU
142662                        FI; GO L1
142663                     FI
142663                  OD
142667                 "NFLAD"; GO FAR PTER2; *)FILL
142720              FI
142720     L1:      IF LEGAREA NBIT BIML GO SMLEAVE
142723              IF FPP = 0 THEN PAR2+PAR1-1=:PAR3; FI
142731                    0=:ARIX; FOR ARIX TO "NNSWSZ-1" DO          % CHECK UP NSWPAGE
142736                       A SH 1+"NSWPAGE"=:X=:TABADR; CALL R1IMAGE
142743                       IF A >< 0 THEN
142744                           A=:FOPP; X:=TABADR+1; CALL R1IMAGE; A=:LOPP
142751                           IF PAR2 <<= LOPP AND PAR2 >>= FOPP THEN
142761                              IF FPP = 0 AND PAR3+1 << 37777 THEN
142770                                 PAR2+1=:PAR2; GO L1
142774                              ELSE "AROC"; GO FAR PTER3; *)FILL
143010                              FI
143010                           FI
143010                       FI
143010                  OD
143014                  0=:ARIX; -1=:FREEADR                        % CHECK UP NINITPAGE
143017                  FOR ARIX TO "NINSZ-1" DO
143023                     A SH 1+"NINITAB"=:X=:TABADR; CALL R1IMAGE
143030                     IF A = 0 THEN
143031                        IF FREEADR = -1 THEN TABADR=:FREEADR FI
143037                     ELSE
143040                        A=:FOPP; X:=TABADR+1; CALL R1IMAGE; A=:LOPP
143045                        IF PAR2 <<= LOPP AND PAR3 >>= FOPP THEN
143055                           IF FPP = 0 AND PAR3+1 << 37777 THEN
143064                              PAR2+1=:PAR2; GO L1
143070                           ELSE "AROC"; GO FAR PTER3; *)FILL
143102                           FI
143102                        FI
143102                     FI
143102                  OD
143106                  IF FREEADR >< -1 THEN                       % UPDATE NINITPAGE
143112                     X:=FREEADR;  PAR2; CALL W1IMAGE
143115                     X:=FREEADR+1;PAR3; CALL W1IMAGE
143121                     CALL WIMBACK;
143122                     IF FPP=0 THEN "TRFP"; CALL SMOUTTEXT;"ONIMA"; CALL SMOUTTEXT
143130                     PAR2; CALL SMOCTU
143132                     FI
143132                  ELSE
143133                     "NFLAD"; GO PTER3; *)FILL
143146                  FI
143146              GO SMLEAVE
143147
143147     M61ER:   IF A = 0 THEN
143150                  A:="AROC"
143151              ELSE IF A = 1 THEN
```

```
143155                  "OUTPHM"
143156          ELSE
143157                  "INTERERR"
143160          FI; FI
143160  PTER2: K:="0"; GO L3
143162  PTER3: K:= 1
143163  L3:    CALL SMOUTTEXT; IF K THEN "ONIMA"; ELSE "INMEM" FI
143171         CALL SMOUTTEXT; GO SSMON
143173  RBUS
143203
143203
143203  %=========================================================================
143203  %              P F L A M
143203  %       PAGES-FROM-LAMU        <FIRST PAGE>
143203
143203  SUBR PFLAM
143203  DISP -200
143203          INTEGER LAMID                      % LAMU IDENTIFICATION
143203          INTEGER CP                         % CURRENT PAGE
143203          INTEGER FPP                        % FIRST PHYSICAL PAGE
143203          INTEGER LPP                        % LAST PHYSICAL PAGE
143203          INTEGER ARIX                       % AREA INDEX
143203          INTEGER LEGAREA
143203          INTEGER TABADR
143203  PSID
143203
143203  PFLAM: L=:D; CALL SMENTER
143205         0=:ARIX; "TPPAG"; CALL SMSGPAR; GO SSMON; A=:FPP
143212         3; CALL SELAREA; A=:LEGAREA
143215         IF LEGAREA BIT BMEMO THEN
143220                 0=:ARIX
143221                 FOR ARIX TO NINSZ-1 DO
143226                    A SH 1 + "LAMAR"=:X; T:=0; *LDDTX
143233                    IF A = FPP THEN
143236                       A:=D=:LPP; CALL ESCOFF % ENTERS A CRITICAL SEQUENCE
143241                       A:=0; T:=0; *STATX     % NOBODY CAN NOW CREATE A LAMU
143244                       GO AROK                % WITHIN THIS AREA
143245                    FI
143245                 OD
143251                 "NSLAR"; CALL SMOUTTEXT; GO SSMON
143254  AROK:          0=:LAMID
143255                 FOR LAMID TO GNLAMU-1 DO
143262                    A*LDTSZ+LAMDT=:X; T:=LAMBANK; *LDATX LMPP
143267                    IF A >< 0 THEN            % LAMU IS DEFINED
143270                       IF A >>= FPP AND A <<= LPP THEN
143276                          ARIX SH 1 + "LAMAR"=:X; T:=0; FPP; *STATX
143305                          "LWIAR"; CALL SMOUTTEXT; CALL ESCON; GO SSMON;
143311                       FI
143311                    FI
143311                 OD
143315                 %     FPP-LPP  OVER TO SIII SWAPPING
143315                 FPP=:CP; FOR CP TO LPP DO CALL PTSINTRAN OD
143330                 CALL ESCON
143331          FI
143331          IF LEGAREA BIT BIML THEN
143334                 0=:ARIX
143335                 FOR ARIX TO "NINSZ-1" DO
143341                    A SH 1 + "NINITAB"=:X=:TABADR; CALL R1IMAGE
143346                    IF A = FPP THEN
143351                       X:=TABADR; A:=0; CALL W1IMAGE
```

```
143354                      X:=TABADR+1; A:=0; CALL W1IMAGE
143360                      CALL WIMBACK; GO SMLEAVE
143362                   FI
143362                OD
143366                "NSLAR"; CALL SMOUTTEXT; GO SSMON
143371            FI
143371            GO SMLEAVE
143372    RBUS
143417
143417    %==========================================================================
143417    %              C R L A M
143417    %       CREATE-LAMU              <LAMU ID>,<SIZE>,<LOGICAL START PAGE>,
143417    %                                <PHYSICAL START PAGE>
143417
143417    SUBR CRLAM
143417    DISP -200
143417            INTEGER LAMID
143417            INTEGER FPP
143417    PSID
143417
143417    INTEGER TLSZ:='IOSIZE: '
143424    INTEGER TRID:='$RETURNED LAMU ID: '
143436    INTEGER TPPNO:='IOPHYSICAL START PAGE: '
143452
143452    CRLAM: L=:D; CALL SMENTER
143454           "TLID"; CALL SAGPAR; GO SSMON; A:=0; A=:PAR1=:LAMID
143462           "TLSZ"; CALL SAGPAR; GO SSMON; A:=1; A=:PAR2
143467           "TPPNO";CALL SAGPAR; GO SSMON; A:=0; A=:PAR3=:FPP
143475           1=:LAMFU; "LAPLIST"; *MON 2LAMU
143501           GO CFILERR
143502           IF LAMID = 0 THEN "TRID"; CALL SMOUTTEXT; PAR1; CALL SMOCTU FI
143510           IF FPP   = 0 THEN "TRFP"; CALL SMOUTTEXT; PAR3; CALL SMOCTU FI
143516           CALL SMLEAVE
143517    CFILERR: *MON 64
143520           GO SMLEAVE
143521    RBUS
143541
143541    %==========================================================================
143541    %              D E L A M
143541    %       DELETE-LAMU              <LAMU ID>
143541
143541    SUBR DELAM
143541    DELAM: L=:D; CALL SMENTER
143543           "TLID"; CALL SAGPAR; GO SSMON; A:=0; A=:PAR1
143550           2=:LAMFU; "LAPLIST"; *MON 2LAMU
143554           GO CFILERR; CALL SMLEAVE
143556    CFILERR: *MON 64
143557           GO SMLEAVE
143560    RBUS
143570
143570    %==========================================================================
143570    %              PRLAM
143570    %              PROTECT LAMU  <LAMU ID>,<RING>,<PROTECTION BITS (RFW)>
143570    SUBR LAPRO
143570    INTEGER TLRI:='IORING: '
143575    INTEGER TLPR:='N PROTECTION BITS (RFW): '
143612    DISP -200
143612      INTEGER CCAR
143612    PSID
143612    SYMBOL FP=15,RP,WP
```

```
143612
143612
143612   LAPRO: L=:D; CALL SMENTER
143614          "TLID"; CALL SAGPAR;GO SSMON; A:=0; A=:PAR1      % LAMU ID
143621          "TLRI"; CALL SAGPAR;GO SSMON; A:=0;              % LAMU RING
143625          IF A > 2 OR A < 0 GO ILLPAR
143632          A SHZ 11 =:PAR2
143634
143634          "TLPR"; CALL SAGPAR;GO SSMON; GO L1;             % LAMU PROT
143640          FOR X := -3 DO
143641            CALL SMCREAD; A=:CCAR
143643            IF A=##F AND PAR2 NBIT FP THEN A BONE FP=:PAR2
143653               ELSE IF CCAR=T THEN GO ILLPAR
143660                  ELSE IF A=##R AND PAR2 NBIT RP THEN A BONE RP=:PAR2
143671                     ELSE IF CCAR=T THEN GO ILLPAR
143676                        ELSE IF A=##W AND PAR2 NBIT WP THEN A BONE WP=:PAR2
143707                           ELSE IF A=##  OR = 15 THEN GO L2
143717                              ELSE GO ILLPAR
143721            FI; FI; FI; FI; FI; FI
143721          OD; GO L2; *)FILL
143734   L1:    PAR2 BONE RP; A BONE WP =:PAR2                   % DEFAULT PROTECTION
143740   L2:    7=:LAMFU; "LAPLIST"; *MON 2LAMU
143744          GO CFILERR
143745          GO   SMLEAVE
143746   ILLPAR:"6TERRP"; CALL SMOUTTEXT; GO SMLEAVE
143751   CFILERR: *MON 64
143752          GO SMLEAVE
143753   RBUS
143761   %=========================================================================
143761   %             L A R E A
143761   %      LAMU-AREAS            <OUTPUT FILE>
143761
143761   SUBR LAREA
143761   DISP -200
143761          INTEGER HEADWR                  % HEADER WRITTEN FLAG
143761          INTEGER ARIX                    % AREA INDEX
143761          INTEGER FPP                     % FIRST PHYSICAL PAGE
143761          INTEGER LPP                     % LAST PHYSICAL PAGE
143761          DOUBLE DPP=FPP
143761   PSID
143761   INTEGER TLARH:='$AREA NO      FIRST PH. PAGE    LAST PH. PAGE'
144010
144010   LAREA: L=:D; CALL SMENTER; CALL 6OOPEN; GO SMLEAVE; A=:TDVN; 0=:ARIX; -1=:HEADWR
144020          FOR ARIX TO NINSZ-1 DO
144025                A SH 1 + "LAMAR"=:X; T:=0; *LDDTX
144032                IF A >< 0 THEN
144033                   AD=:DPP; MIN HEADWR; GO L1; "TLARH"; CALL SMOUTTEXT; -22=:HEADWR
144042   L1:             CALL SMCRLF; % A:=5; CALL SMSPACE
144043                   ARIX; CALL SMOCTU; A:=15; CALL SMSPACE
144047                   A:=FPP; CALL SMOCTU; A:=12; CALL SMSPACE; A:=LPP; CALL SMOCTU
144055                FI
144055          OD; GO SMLEAVE
144062   RBUS
144074
144074   %=====================================================================
144074   %             L A I N F
144074   %      LAMU-INFORMATION      <LAMU ID>,<OUTPUT FILE>
144074
144074   SUBR LAINF
144074
```

```
144074   SYMBOL LAORA=174                % LAMU ID OUTSIDE LEGAL RANGE
144074   DISP -200
144074          INTEGER LAMID
144074          INTEGER MAXLAMU
144074          INTEGER XLAMID
144074          INTEGER HEADWR          % HEADER WRITTEN
144074          INTEGER CLAMAR          % LAMU AREA IN MEMORY
144074          INTEGER CPP             % CURRENT PHYSICAL PAGE
144074          INTEGER CNP             % CURRENT NUMBER OF PAGES
144074          INTEGER CPR             % CURRENT RING AND PROTECTION
144074
144074   PSID
144074   SYMBOL FP=15,RP,WP
144074   INTEGER THEAD:='$    LAMU ID        SIZE     PH.PAGE     RING     PROT.     LAMU AREA'
144135   INTEGER TLNDF:='$LAMU NOT DEFINED'
144146   INTEGER INERR:='$INTERNAL ERROR'
144156
144156   LAINF: L=:D; CALL SMENTER
144160          "TLID"; CALL SAGPAR; GO SSMON; A:=-1; A=:LAMID=:XLAMID
144166          CALL 6OOPEN; GO SMLEAVE; A=:TDVN
144171          IF LAMID = -1 THEN
144175                0=:LAMID; GNLAMU-1=:MAXLAMU
144201          ELSE
144202                IF A=:MAXLAMU >>= GNLAMU THEN LAORA; GO FILERR; *)FILL
144220                FI
144220          FI
144220         -1=:HEADWR
144222          FOR LAMID TO MAXLAMU DO
144226                A*LDTSZ+LAMDT=:X; T:=LAMBANK; *LDDTX LMPP; LDXTX LMPR
144234                IF A >< 0 THEN
144235                   A=:CPP:=D=:CNP; X=:CPR
144241                   MIN HEADWR; GO L1; "THEAD"; CALL SMOUTTEXT;-22=:HEADWR
144247   L1:             CALL SMCRLF; A:=4; CALL SMSPACE; LAMID; CALL SMOCTU; A:=4; CALL SMSPACE
144256                   CNP; CALL SMOCTU; A:=4; CALL SMSPACE; CPP; CALL SMOCTU
144264                   A:=1; CALL SMSPACE
144266                   CPR SHZ -11 /\ 3; CALL SMOCTU; A:=5; CALL SMSPACE        % RING
144274                   IF CPR BIT RP THEN ##R; CALL SMTCO; ELSE ## ; CALL SMTCO; FI
144304                   IF CPR BIT FP THEN ##F; CALL SMTCO; ELSE ## ; CALL SMTCO; FI
144314                   IF CPR BIT WP THEN ##W; CALL SMTCO; ELSE ## ; CALL SMTCO; FI
144324                   0=:CLAMAR; A:=7; CALL SMSPACE
144327                   FOR CLAMAR TO NINSZ-1 DO
144334                      A SH 1+"LAMAR"=:X; T:=0; *LDDTX
144341                      WHILE T:=CPP << A OR T >> D
144346                   OD; IF CLAMAR = NINSZ THEN "INERR"; CALL SMOUTTEXT ELSE CALL SMOCTU FI
144362                ELSE
144363                   IF XLAMID >< -1 THEN "TLNDF"; CALL SMOUTTEXT; FI
144371                FI
144371          OD
144375          GO SMLEAVE
144376   RBUS
144413
144413   %=================================================================
144413   %              L L A M C
144413   %       LIST-LAMU-CONSTANTS
144413
144413   SUBR LLAMC
144413   INTEGER HLAMC:='$          NO. OF LAMUS PR RT PROG    TOTAL NO. OF LAMUS$MEMORY: '
144454   INTEGER TIMAG:='$IMAGE:   '
144461   LLAMC: L=:D; CALL SMENTER
144463          "HLAMC"; CALL SMOUTTEXT; A:=12; CALL SMSPACE; A:=GNLPRT; CALL SMOCTU
```

```
144471          A:=24; CALL SMSPACE; A:=GNLAMU; CALL SMOCTU; "TIMAG"; CALL SMOUTTEXT
144477          A:=12; CALL SMSPACE; X:="GNLPRT"; CALL R1IMAGE; CALL SMOCTU
144504          A:=24; CALL SMSPACE; X:="GNLAMU"; CALL R1IMAGE; CALL SMOCTU
144511          GO SMLEAVE
144512   RBUS
144524
144524   %=========================================================================
144524   %             S L A M C
144524   %       SET-LAMU-CONSTANTS
144524
144524   SUBR SLAMC
144524   INTEGER TNLPRT:='IONO. OF LAMUS PR. RT PROGRAM: '
144544   INTEGER TNLAMU:='IOTOTAL NO. OF LAMUS: '
144560   SLAMC: L=:D; CALL SMENTER
144562          "TNLPRT"; CALL SAGPAR; GO SSMON; A:=GNLPRT; X:="GNLPRT"; CALL W1IMAGE
144570          "TNLAMU"; CALL SAGPAR; GO SSMON; A:=GNLAMU; X:="GNLAMU"; CALL W1IMAGE
144576          CALL WIMBACK; GO SMLEAVE
144600   RBUS
```

```
144612
144612  %=================================================================
144612  %
144612  %          @START-GPIB-CONTROLLER       S G P I B
144612  %          @STOP-GPIB-CONTROLLER        K G P I B
144612  %          @CHANGE GPIB BUFFERSIZE      S G P B S
144612  %
144612  %=================================================================
144612
144612  SUBR SGPIB,KGPIB,SGPBS,GEPNE
144612  DISP -200
144612    INTEGER CIDN,GPNUM,GPDFI,GPITE,SIZE,AREA,LOCNT
144612  PSID
144612  INTEGER TPAR:=("31","1")
144614
144614
144614  SGPIB: D:=L;CALL SMENT                                          % IS GPIB IMPLEMENTED
144616         CALL GPIIM; GO ERR08
144620         T:=0;*MON 2XMSG                                          % CHECK IF XMSG IS RUNNING
144622         IF T<= 0 GO ERR07
144624         CALL GEPNE;T=:GPNUM                                      % REJECT COMMAND
144626         T:=-1;IF A = T GO ERR01                                  % SAVE DATAFIELD ADDRESS
144631         A=:GPDFI
144632  %=================================================================
144632  %      START GPIB DRIVER
144632  %=================================================================
144632         A:=GPDFI+"GPRUN"=:X;CALL R1MEMO                          % READ RUNFLAG
144632                                                                  % GPIB ALLREADY RUNNING
144636         IF A+1 = 0 GO ERR05                                      % T= ENTRY POINT FOR XMSG TRANSFER R0
144640         A:=GPDFI=:X; T:=X."GPXTR"
144643         CALL STAGPIB
144644         -10=:LOCNT
144646
144646         DO
144646            A:=LOCNT
144647            WHILE >< 0
144650            A+1=:LOCNT;A:="TPAR";* MON 104
144654            A:=GPDFI+"GPRUN"=:X;CALL R1MEMO                       % READ RUNFLAG
144660            IF >< 0 GO FTST
144661         OD
144662         GO ERR02
144663  FTST:  IF = -1 GO OUT01
144666         GO ERR03
144667
144667  OUT01: "GPIOK";CALL SMOUTTEXT;GO FAR OUT00                      % OK EXIT
144672  ERR01: "GPIER";CALL SMOUTTEXT;GO FAR OUT00                      % ERROR EXIT
144675  ERR02: "GPICS";CALL SMOUTTEXT;GO FAR OUT00                      % ERROR EXIT
144700  ERR03: X:=A;"GPISE";CALL SMOUTTEXT;A:=X;CALL SMOCTUT            % ERROR EXIT
144705         CALL  SMCRLF;GO FAR OUT00
144707  INTEGER TNIMPL:='$GPIB NOT IMPLEMENTED$'
144723  ERR05: "GPAST";CALL SMOUTTEXT;GO FAR OUT00                      % ERROR EXIT
144726  ERR06: "GPAAB";CALL SMOUTTEXT;GO FAR OUT00                      % ERROR EXIT
144731  ERR07: "GPXNR";CALL SMOUTTEXT;GO FAR OUT00                      % ERROR EXIT
144734  ERR08:"TNIMPL";CALL SMOUTTEXT;GO FAR OUT00                      % ERROR EXIT
144737
144737  *)FILL
144764  INTEGER GPIER:='**** ILLEGAL CONTROLLER ****$'
145003  INTEGER GPIOK:='OK, CONTROLLER STARTED$'
145017  INTEGER GPICS:='GPIB STARTUP ERROR. CONTROLLER NOT STARTED$'
145045  INTEGER GPISE:='GPIB STARTUP ERROR. ERRORCODE :'
145065  INTEGER GPAST:='GPIB ALLREADY STARTED. USE STOP-GPIB TO STOP IT$'
```

```
145116   INTEGER GPAAB:='GPIB NOT RUNNING$'
145127   INTEGER GPXNR:='XMSG NOT RUNNING. CANNOT START GPIB DRIVER$'
145155   INTEGER GPCNO:='IOCONTROLLER NO.:'
145166   INTEGER MEMTX:='MEMORY  '
145173   INTEGER IMATX:='IMAGE   '
145200   INTEGER SAVTX:='SAVE-AREA'
145205   INTEGER BLNK:= '
145223   INTEGER GPUSB:='USER BUFFER SIZE (OCT)  '
145240   INTEGER GPDMB:='DMA BUFFER SIZE (OCT)   '
145255   INTEGER DESC:= 'IO    '
145261
145261   %=================================================================
145261   %       STOP GPIB
145261   %=================================================================
145261   KGPIB: D:=L;CALL SMENTER;
145263          CALL GPIIM; GO FAR ERR08                                  % IS GPIB IMPLEMENTED
145265          CALL GEPNE;T:=-1
145267          IF = T GO FAR ERR01;A=:GPDFI                              % REJECT COMMAND
145272          A:=GPDFI+"GPRUN"=:X;CALL R1MEMO                           % READ RUNFLAG
145276          IF  = 0 GO FAR ERR06                                      % GPIB NOT RUNNING
145300          A:=GPDFI; T:="GPIBH"; CALL STOGPIB
145303          GO OUT00
145304
145304
145304   %=================================================================
145304   %       CHANGE GPIB BUFFERSIZE
145304   %=================================================================
145304   SGPBS: D:=L;CALL SMENTER;
145306          CALL GPIIM; GO FAR ERR08                                  % IS GPIB IMPLEMENTED
145310          CALL GEPNE;T:=-1
145312          IF = T GO FAR ERR01                                       % REJECT COMMAND
145314          A=:GPDFI:=7;CALL SELAR;IF = 0 GO OUT00;A=:AREA            % SAVE DATAFIELD ADDRESS AND CHECK AR
145321          X:=A;"BLNK";CALL SMOUTTEXT
145324          IF X BIT 0 THEN "MEMTX";CALL SMOUTTEXT FI
145330          IF X BIT 1 THEN "IMATX";CALL SMOUTTEXT FI
145334          IF X BIT 2 THEN "SAVTX";CALL SMOUTTEXT FI
145340          CALL SMCRLF;CALL SMCRLF
145342          "GPUSB";CALL SMOUTTEXT
145344          A:=GPDFI+"USIZE"=:X;CALL GVAL
145350          "GPDMB";CALL SMOUTTEXT
145352          A:=GPDFI+"DSIZE"=:X;CALL GVAL
145356          GO OUT00
145357
145357   *)FILL
145405
145405   %%% GET VALUE AND WRITE IN SELECTED AREA
145405
145405
145405   GVAL:  D:=L;CALL SMENTER
145407          IF AREA BIT 0 THEN
145412             CALL R1MEMO;CALL SMOCTUT
145414          FI
145414          IF AREA BIT 1 THEN
145417             CALL R1IMAG;CALL SMOCTUT
145421          FI
145421          IF AREA BIT 2 THEN
145424             CALL R1SAVE;CALL SMOCTUT
145426          FI
145426          "DESC";CALL SMAGPAR;GO NCHAN                              % GET INPUT AND IF EMTY PARAMETER USE
145431          T:=AREA;IF T BIT 0 THEN CALL W1MEMO FI                    % WRITE TO MEMORY
```

```
145435          T:=AREA;IF T BIT 1 THEN CALL W1IMAG FI                  % WRITE TO IMAGE
145441          T:=AREA;IF T BIT 2 THEN CALL W1SAVE FI                  % WRITE TO SAVE
145445   NCHAN: GO SMLEAVE
145446
145446
145446   % LOCAL SUBROUTINEI
145446   % GET CONTROLLER NUMBER AND DATAFIELD ADDRESS
145446   INTEGER ARRAY GPARR:=(2120,2121,2122,2123,2124,2125,2126,2127)
145456   GEPNE: D:=L;CALL SMENTER;"GPCNO";CALL SMAGPAR;A:=0;A=:GPNUM
145464          IF A < 0 OR A>7 GO ERR00
145470          GPARR(A);CALL LOGPH
145473          IF A><0 GO OUT00
145474   ERR00: A:=-1
145475   OUT00: T:=GPNUM;GO SMLEAVE
145477
145477
145477   % LOCAL SUBROUTINE , IS GPIB-DRIVER IMPLEMENTED ?
145477   GPIIM: IF "GPIBT"=0 THEN EXIT FI; EXITA
145503
145503
145503
145503   RBUS
145522
145522
145522   %=========================================================================
145522   % 28.18        S  M O O P E N
145522   %
145522   % SUBROUTINE TO READ A "OUTPUT FILE:" NAME, AND OPENS THE FILES
145522   %
145522   SUBR SMOOPEN
145522   SMOOPEN: L=:D; CALL SMENTER
145524            "SMSTROFI"; CALL SAGPAR; GO SMLEAVE; 1
145530           IF A><1 THEN
145533                 A=:X:="TYPS"; T:=0; *MON 2NOPE
145537                 GO ERR
145540           FI; GO SM2LEAVE
145541   ERR:    *MON 64
145542           GO SMLEAVE
145543   RBUS
145551
145551
145551   %=========================================================================
145551   %
145551   %    24.0      D E V I C E   H A N D L I N G
145551   %
145551   %=========================================================================
145551   % 24.1         T A P F U
145551   %
145551   % COMMAND: DEVICE-FUNCTION <FILE NAME> <FUNCTION NAME> <OPT. PAR 1> <OPT. PAR 2>
145551   %
145551
145551   %        FUNCTION TABLE:        (A) ENTRY POINT (LABEL) FOR THIS FUNCTION.
145551   %                               (B) FUNCTION CODE OR FUNCTION.
145551   %                               (C) DESCRIPTIVE TEXT;   1 = NO. OF OPERATIONS
145551   %                                                       2 = BUFFER ADDRESS
145551   %                                                       3 = VALUE
145551   %                                                       4 = UNIT NO.
145551   %                                                       5 = FORMAT
145551   %                                                       6 = DENSITY (BPI)
145551   %
145551   @ICR
```

```
145551   INTEGER ARRAY FUNCTAB:=(
145551           TF0, 0,2,0,            TF1, 1,2,0,
145561           TF4, 4,2,0,
145565           TF5, 5,0,0,            TF6, 6,0,0,
145575           TF7, 7,0,0,           TF10,10,1,0,
145605          TF11,11,1,0,           TF12,12,0,0,
145615          TF13,13,0,0,           TF14,14,0,0,
145625          TF15,15,1,0,           TF16,16,1,0,
145635          TF17,17,0,0,           TF20,20,0,0,
145645          TF21,21,0,0,           TF23,23,3,0,
145655          TF24,24,0,0,           TF30,30,0,0,
145665          TF31,31,0,0,           TF32,32,1,0,
145675          TF33,33,4,0,          TFX23,23,6,0,
145705          TF40,40,5,0,           TF41,41,0,0,
145715          TF42,42,0,0,           TF43,43,2,0,
145725          TF44,44,2,0,           TF45,45,0,0,
145735          TF25,25,0,0,           TF2,  2,2,0,
145745          TF26,26,2,0,           TF27,27,2,0,
145755          TF46,46,0,0,           TF47,47,7,0,
145765          -1);
145766   @CR;
145766
145766   SUBR TAPEFU
145766
145766   SYMBOL 5CAS=11
145766   %            DISPLACEMENTS ON THE SYSTEM SEGMENT.
145766   DISP -200
145766   INTEGER TFLOG,TFFUN,TFNOT,TFNWO,TFSTAT,TFADR,TFBYT,TLADR,TBIFI
145766   INTEGER BUFHEADER,LOSBUF,CDFADDR
145766   INTEGER POINTER CLINK
145766   INTEGER SAVED=LOSBUF
145766   INTEGER ARRAY TFXADDR=BUFHEADER
145766   INTEGER TBPFI=TFNOT,CCHAR=TFSTAT,COUNT=TFFUN,CHSUM=TFNWO
145766   INTEGER MCOUNT=TFADR,LCHSUM=TLADR
145766   PSID
145766   DISP 25; INTEGER CASUN; PSID
145766   %            DISPLACEMENTS WITHIN FUNCTAB
145766   DISP 1; INTEGER TFCOD,FDESC; PSID
145766
145766   INTEGER TCDTY:=''              % DEFAULT FILE TYPE
145767   INTEGER TCFUNC:='N FUNCTION: '
145776   INTEGER TNOTM:='IDNO. OF OPERATIONS (DECIMAL): '
146016   INTEGER TADDR:='IOBUFFER ADDRESS: ',      TVALUE:='IOVALUE: ',          TSTAE:='STATUS: '
146042   INTEGER TUNIT:='IOUNIT NO.:',              TILLE:='$ILLEGAL CODE$',   SVALUE:='IDDENSITY(BPI): '
146071   INTEGER TNOWO:='IONO. OF WORDS (OCTAL): ',  SIDA:=' SIDE A',            SIDB:=' SIDE B'
146116   INTEGER TYPBP:='BPUN',                     TBFIL:='S BINARY FILE: '
146131   INTEGER TFERR:='$FLOPPY DISC TRANSFER ERROR$'
146150   INTEGER TNBIF:='$NO BIT FILE ON DISKETTE$'
146165   INTEGER ILCFILE:='$ILLEGAL CONTENT ON BINARY FILE$'
146206   INTEGER TILLD:='$ILLEGAL DEVICE$'
146217   INTEGER PROT:='$PROTECTED FUNCTION$'
146232   INTEGER TRWORDS:='NO. OF WORDS READ: '
146244   INTEGER TRBYTES:='NO. OF BYTES READ: '
146256   INTEGER TXF46:='DISC ADDRESS: '
146266   INTEGER TNOBYT:='IONO. OF BYTES (OCTAL): '
146303   INTEGER TFORMAT:='IOFORMAT: '
146311   INTEGER TFX0:='READ-ERROR STATUS:  ',       TFX1:='NO. OF READ-ERRORS: '
146337   INTEGER TFX2:='WRITE-ERROR STATUS: ',       TFX3:='NO. OF WRITE-ERRORS:'
146365   INTEGER TCDAD:='IODISC ADDRESS: '
146376   INTEGER ARRAY TXADDR:=(TFX0,TFX1,TFX2,TFX3)
```

```
146402   @ICR
146402   INTEGER M144P:=(
146402          BFIELD+TFFUN,BFIELD+TFXADR,BFIELD+TFLOG,BFIELD+TFNWO,BFIELD+TFNWO);
146407   %
146407   %       FUNCTION TITLES
146407   %
146407   INTEGER   TF0:='READ-RECORD',          TF1:='WRITE-RECORD',          TF5:='UNLOCK-AND-STOP',
146434             TF6:='LOCK-CASSETTE',        TF7:='ERASE-TAPE',           TF10:='ADVANCE-TO-EOF',
146461            TF11:='REVERSE-TO-EOF',
146471            TF12:='WRITE-EOF',           TF13:='REWIND',               TF14:='WRITE-ERASE-GAP',
146512            TF15:='BACKSPACE-RECORDS',   TF16:='ADVANCE-RECORDS',
146533            TF17:='UNLOAD',
146537            TF20:='READ-STATUS',         TF21:='CLEAR-DEVICE',         TF23:='SELECT-PARITY-AND-DENSITY',
146571           TFX23:='SELECT-DENSITY',
146601            TF24:='READ-LAST-STATUS',    TF30:='SET-ALPHANUMERIC-MODE',
146625            TF31:='SET-GRAPHIC-MODE',    TF32:='GIVE-FORM-FEED',       TF33:='CLEAR-SELECTED-UNIT',
146660            TF40:='SET-FLOPPY-FORMAT',   TF41:='FORMAT-FLOPPY',        TF42:='READ-FORMAT',
146706            TF43:='READ-DELETED-RECORD',TF44:='WRITE-DELETED-RECORD',
146733            TF25:='READ-TAPE-STATUS',     TF2:='READ-ODD-NUMBER-OF-BYTES',
146761            TF26:='READ-BYTE-RECORD',    TF27:='WRITE-BYTE-RECORD',
147003            TF46:='GET-CURRENT-DISC-ADDRESS',
147020            TF45:='DUMP-BOOTSTRAP',TF47:='SET-CURRENT-DISC-ADDRESS',
147045            TF4:='READ-BACKWARDS';
147055   @CR;
147055
147055   @DEC
147055   INTEGER D6250:=6250              % DENSITY FOR STC-CONTROLLER.
147056   INTEGER DS1600:=1600
147057   INTEGER DS800:=800
147060   INTEGER CSTRFILE:='S FILE NAME:'
147067   @OCT
147067   TAPEFU: L=:D; CALL SMENTER
147071          "CSTRFILE"; CALL SMGPAR; A=:X; A:="TCDTY"; T:=2; *MON 2NOPE; MON 65
147100          A=:TFLOG; "TCFUNC"; CALL SMGPAR; X:="FUNCTAB"; CALL SMABLOOK      % T-REG : TABLE ELEMENT
147105          IF A><0 THEN
147106                IF A=-1 THEN "SMTXFILL" ELSE "SMFAMBIG" FI
147114                CALL SMOUTTEXT; GO FAR OUT
147116          FI
147116          T.TFCOD=:TFFUN; IF A=45 GO FAR FUN42                % TFFUN - FUCTION
147124          IF X.FDESC><0 THEN
147126                IF A=1 THEN "TNOTM"                  % NO. OF OPERATIONS
147132                ELSE IF A=2 THEN "TADDR"             % BUFFER ADDRESS
147137                     ELSE IF A=4 THEN "TUNIT"
147144                          ELSE IF A=6 THEN
147150                                  "SVALUE"; CALL SMGPAR
147152                                  IF A=DS1600 THEN A:=0
147156                                  ELSE IF A=D6250 THEN A:=1
147163                                       ELSE IF A=DS800 THEN A:=2
147170                                       ELSE GO FAR ERR23
147172                                  FI;  FI   FI
147172                               ELSE IF A=3 THEN "TVALUE"; CALL SMGPAR
147200                                       IF A<0 OR A>6 GO FAR ERR23
147204                                    ELSE IF A=7 THEN "TCDAD"; CALL SMGPAR
147212                                         ELSE "TFORMAT"; CALL SMGPAR
147215                                         FI
147215                                     FI
147215                               FI
147215                               A=:TFNWO; A:=1; GO TAPF1
147220                FI;  FI;  FI;  CALL SMGPAR
147221          ELSE A:=1
```

```
147223          FI; A=:D
147224          GO TAPF1
147225   *)FILL
147251   TAPF1: A-=:TFNOT
147253          IF X.FDESC=2 THEN                % READ OR WRITE
147257             A:=D=:TFADR
147261             IF X.TFCOD=26 OR A=27 THEN "TNOBYT" ELSE "TNOWO" FI
147273             CALL SMGPAR; A=:TFNWO; TFADR
147276             CALL ESCOFF; CALL CM144; CALL ESCON
147301             IF A><0 GO ERR
147302             IF TFFUN=0 OR A=2 OR A=26 OR A=4 THEN
147315                IF A=26 THEN "TRBYTES" ELSE "TRWORDS" FI
147323                CALL SMOUTTEXT; TFNWO; CALL SMOCTU
147326             FI
147326          ELSE
147327             IF TFNOT=0 GO OUT
147331             FOR TFNOT DO
147331                CALL ESCON; CALL ESCOF
147333                IF TFFUN=25 THEN 4=:TFNWO FI
147341                "M144P"; *MON 2MAGT
147343                A=:TFSTAT
147344                IF TFFUN=20 OR A=24 THEN      % READ STATUS
147353                   "TSTAE"; CALL SMOUTTEXT; TFSTAT; CALL SMOCTU
147357                   TFLOG; CALL LOGPH; IF A=0 THEN A:=D FI
147363                   IF A.TYPRING BIT 5CAS THEN
147367                      IF TFSTAT BIT 6 THEN "SIDA" ELSE "SIDB" FI
147375                      CALL SMOUTTEXT
147376                   FI; GO OUT
147377                ELSE
147400                   IF TFSTAT><0 GO ERR
147402                FI
147402             OD
147404             IF TFFUN=46 THEN "TXF46"; CALL SMOUTTEXT; TFNWO; CALL SMOCTU FI
147414             IF TFFUN=42 THEN "TFORMAT+1"; CALL SMOUTTEXT; TFNWO; CALL SMOCTU FI
147424             IF TFFUN=25 THEN
147430                FOR X:=0 TO 3 DO
147434                   CALL SMCRLF; TXADDR(X); CALL SMOUTTEXT; TFXADDR(X); CALL SMOCTU
147441                OD
147443          FI; FI
147443   OUT:   T:=TFLOG; *MON 2CLOS; JMP *+1
147446   RETU:  GO SMLEAVE
147447   ERR:   *MON 64
147450          GO OUT
147451   *)FILL
147475
147475   ERRBU: 172; *MON 64                          % NO DEVICE BUFFER AVAILABLE
147477   ERRF1: T:=TBPFI; *MON 2CLOS; JMP *+1
147502          GO OUT
147503   ERR2:  "TFERR"; CALL SMOUTTEXT                 % TRANSFER ERROR ON FLOPPY DISK
147505   ERRF2: T:=TFLOG; CALL R3BUF; CALL ERRFATAL   % RELEASE BUFFER
147510          GO ERRF1
147511   ERR3:  *MON 64
147512          GO ERRF2
147513   ERR4:  "TNBIF"; CALL SMOUTTEXT; GO ERRF2       % NO BIT FILE ON DISKETTE
147516   ERR1:  "TILLD"; CALL SMOUTTEXT; GO ERRF2       % ILLEGAL DEVICE
147521   ERR23: "TILLE"; CALL SMOUTTEXT; GO OUT         % ILLEGAL DENSITY-CODE
147524
147524
147524   FUN42: GO NOPRO              % TO MAKE IT IMPOSSIBLE FOR USERS OTHER THAN SYSTEM TO EXECUTE THIS COMMAND P
147525          IF PASSTYPE><2 THEN "PROT"; CALL SMOUTTEXT;GO RETU FI  % LOCATION TO 124001
```

```
147534   NOPRO: "TBFIL"; CALL SMGPAR; A=:X:="TYPBP"; T:=1; *MON 2NOPE; JMP ERR
147543          A=:TBPFI; T:=TFLOG; 2000; CALL G3BUF; GO ERRBU; T=:BUFHEADER
147551          D=:L; A=:T; RTREF.ACTPRI=:D; L=:X; CALL DBTRANS; X=:LOSBUF
147561          TFLOG; CALL LOGPH; IF A=0 GO ERR1          % ILLEGAL DEVICE
147564          IF A.TYPRING NBIT M144B GO ERR1; X=:CDFADDR
147571          X.CLOGDV; CALL LOGPH; IF A=0 OR A.TYPRING NBIT 5FLOP GO ERR1
147600          A:=0; CALL RCALABSTR
147602          LOSBUF+"2000-3"=:X; IF X.S0 =0 GO ERR4       % NO BIT FILE ON DISKETTE
147607          A=:TBIFI; GO L2        % ADDRESS OF BIT FILE
147611   *)FILL
147633
147633   L2:    0=:LCHSUM=:TFBYT
147635          DO CALL CINBT; WHILE A><##!; OD
147642          ##0; CALL STWORD; ##/; CALL STWORD; ##2; CALL STWORD
147650          15; CALL STWORD; 12; CALL STWORD; ##2; CALL STWORD; ##!; CALL STWORD
147660          CALL CINBT; IF A><0 GO FAR ERR5
147663          CALL CINBT; IF A><0 GO FAR ERR5
147666          CALL CINBT; A SH 10=:D; CALL CINBT; A\/D; A+A+4-=:MCOUNT
147677          A:=0; CALL STWORD; CALL STWORD; CALL STWORD; 100; CALL STWORD
147705          CALL CCCINBT; GO FAR ERR5; CALL STWORD; A SH 10=:D
147712          CALL CCCINBT; GO ERR5; CALL STWORD; A\/D; CALL ACHSUM
147717          A+A-1-=:COUNT
147723          -77=:TFNWO
147725          FOR TFNWO DO
147725                CALL CCCINBT; GO ERR5; CALL STWORD; A SH 10=:D
147732                CALL CCCINBT; GO ERR5; CALL STWORD; A\/D; CALL ACHSUM
147737          OD; GO FILSK
147742
147742   ACHSUM: A=:D+LCHSUM=:LCHSUM:=D; EXIT
147747   STWORD: A=:X
147750   L3:     LOSBUF+TFBYT; A:=:X; A=:X.S0; MIN TFBYT; EXIT
147756   CCCINBT: MIN MCOUNT; GO CCINBT; GO FAR L6
147761   CCINBT:  MIN COUNT;  GO L4; EXIT
147764   L4:     L+1
147765   CINBT:  T:=TBPFI; *MON 2INBT
147767           GO FAR ERR3; A=:CCHAR; A:=L=:"CLINK"; A:=D=:SAVED; RTREF.ACTPRI=:D
150000           T:=BUFHEADER.DBUFB; X:=X.BUFFER;
150003           CALL DBTRANS; A:=SAVED=:D; X=:LOSBUF; CCHAR; GO CLINK
150011   *)FILL
150050
150050   FILSK: LCHSUM SHZ -10; CALL STWORD; LCHSUM/\377; CALL STWORD
150056           A:=0; CALL STWORD
150060           240=:TFBYT
150062           DO
150062                 CALL CCCINBT; GO ERR5; A SH 10=:D
150066                 CALL CCCINBT; GO ERR5; A\/D
150071              WHILE A=0
150072           OD; CALL STWORD
150074           DO
150074                 CALL CCCINBT; GO L5; A SH 10=:D
150100                 CALL CCCINBT; GO ERR5; A\/D; CALL STWORD
150104           OD
150105
150105   *)FILL
150111
150111
150111   ERR5:  "ILCFILE"; CALL SMOUTTEXT; GO FAR ERRF2
150114
150114   % RCALABSTR IS CALLED WHEN THE MASTER BLOCK IS READ, AND WHEN THE BIT FILE
150114   % IS READ. THE A REG WILL THEREFORE BE 0 OR BIT FILE ADDR.
```

```
150114   % AS IT STANDS NOW, RCALABSTR WILL WORK ONLY FOR TWO FORMATS, 0 AND D17.
150114
150114   RCALABSTR: A=:BUFHEADER.DBLO1; 4=:X.DBLOCK
150120              CDFADDR.CLOGDV; L=:X; CALL LOGPH; X=:L; IF A=0 GO FAR ERR1
150127              T:=CDFADDR.CASUN
150131              A+T+"FDIFORM"=:X
150134              IF X.S0 >< 0 THEN 2=:BUFHEADER.DBLOCK; X.DBLO1 SHZ -1 =:X.DBLO1 FI
150144              A:=T SH 6; GO CALABSTR
150147   WCALABSTR: CDFADDR.CASUN SH 6+1
150153   CALABSTR:  A=:BUFHEADER.DKFUN; T:=CDFADDR.CLOGDV; BUFHEADER+"DPNTO"; *MON 2ABST
150162          IF A<0 GO FAR ERR2
150164          EXIT
150165
150165   L5:    CALL WCALABSTR; TBIFI SHZ 2; CALL RCALABSTR
150171          100=:TFBYT
150173          DO
150173                CALL FAR CCCINBT; GO ERR5; A SH 10=:D; CALL FAR CCCINBT; GO ERR5
150201                A\/D; CALL FAR STWORD
150203          OD
150204   L6:    CALL WCALABSTR
150205          T:=TFLOG; CALL R3BUF; CALL ERRFATAL  % RELEASE BUFFER
150210          T:=TBPFI; *MON 2CLOS; JMP *+1   % CLOSE BINARY FILE
150213          GO FAR OUT
150214
150214   RBUS
150232
150232   %==========================================================================
150232   % 24.2        L D V F C O M
150232   %
150232   % COMMAND: LIST-DEVICE-FUNCTION <OUTPUT FILE>
150232   %
150232   SUBR LDVFCOM
150232   INTEGER CTXCOM:='S COMMAND: '
150240   LDVFCOM: L=:D; CALL SMENTER
150242          CALL SMOOPEN; GO SMLEAVE; A=:TDVN
150245          "CTXCOM"; CALL SMKGPAR; A:="TXAPPS"; A=:CSTRING
150251          X:="FUNCTAB"; CALL ESCON
150253          DO WHILE X.S0><-1
150257             IF A><0 THEN
150260                0=:CPNT; CALL SM1ABL
150262                IF A><2 THEN CALL SMCRLF; X.S0; CALL SMOUTTEXT FI
150270             FI; X+4
150271          OD; CALL SMCRLF
150273          T:=TDVN; *MON 2CLOS; JMP *+1
150276          GO SMLEAVE
150277   RBUS
150313
150313   %==========================================================================
150313   % 18.13       L O O K A T
150313
150313   % COMMAND: LOOK-AT AREA,ADDRESS
150313   %
150313   %COMMAND TO EXAMINE AND MODIFY LOCATIONS AND REGISTERS
150313   SUBR LOOKAT
150313   SYMBOL T1=10000,T2=20000,T3=30000
150313   @ICR
150313   INTEGER ARRAY LOOKTAB:=(
150313          ST0,0,GPCORE,0,
150317          ST1,0,GPCORE,0,
150323          ST2,0,GPSG,1,
```

```
150327          ST3,0,GPIM,2,
150333          ST4,0,RESIDENT,2,
150337          ST5,0,CCOM,1,
150343          ST6,0,GPCORE,0,
150347          ST7,0,GPREGS,1,
150353          ST8,0,AGPCOR,0,
150357          ST9,0,GPDSG,1,
150363          ST10,0,GPPHY,2,
150367          -1),
150370   ST0:='CORE',ST1:='',ST2:='SEGMENT',ST3:='IMAGE',
150403   ST4:='RESIDENT',ST5:='RTCOMMON',ST6:='MEMORY',ST7:='REGISTERS',
150426   ST8:='ALT-MEMORY',ST9:='ALT-SEGMENT',ST10:='PHYSICAL-MEMORY',
150452   STRTYPE:='N MEMORY,ALT-MEMORY,SEGMENT,IMAGE,$RESIDENT,RTCOMMON,REGISTERS,ALT-SEGMENT,PHYSICAL-MEMORY: ',
150531   CHTAB:=(#PP,#XX,#TT,#AA,#DD,#LL,#SS,#BB,#07,#--,#//,15\15,-1),
150546   INPT:=(1,1,1,1,1,1,1,1,0,0,2,3,-1),
150563   RREGNO:=(2,7,6,5,1,4,0,3),
150573   PROT:='PROTECTED',READY:='READY:$',ENDL:='$-END',
150607   PWSGB:=GPSEGM,
150610   TILSG:='$ILLEGAL TO LOOK AT CURRENT REENTRANT SEGMENT$',
150640   TXILSG:='$ILLEGAL SEGMENT NUMBER$',
150655   HUFF=?;
150655   @CR;
150655   DISP -200
150655          INTEGER CPRO,ADR1,ADR,STATE,CHAR,BCHAR,VAL1,VALUE,REGNO
150655          INTEGER POINTER LINK1,LINK2
150655          INTEGER MINFLAG=CPRO,CLEVL=STATE
150655          DOUBLE DDADR=ADR1,DVALUE=VAL1
150655   PSID
150655
150655   % AUXILIARY SUBROUTINES:
150655   BBACKSP:CHAR=:BCHAR; EXIT
150660   BTCI:  T:=L=:"LINK1"; IF BCHAR><0 THEN 0=:BCHAR ELSE CALL SMTCI FI;A=:CHAR;GO LINK1
150671   % ASEMBLE OCTAL NUMBER, RESULT IN VALUE AND A-REG.
150671   ASSOCT: T:=L=:"LINK2"; 0=:VALUE=:VAL1=:MINFLAG; CALL BBACKSP
150677          CALL BTCI; IF =##- THEN MIN MINFLAG ELSE CALL BBACKSP FI
150706          DO
150706             CALL BTCI; IF A-60<0 OR>7 GO OUT
150714             T:=A; AD:=DVALUE SH 3; D+T; AD=:DVALUE
150721          OD
150722   OUT:   CALL BBACKSP; VALUE
150724          IF T:=MINFLAG><0 THEN
150727             AD:=DVALUE; *COPY CM2 SD DD; COPY CM1 ADC SA DA
150732             AD=:DVALUE; A:=VALUE
150734          FI; GO LINK2
150735   *)FILL
150736
150736   %ENTRYPOINT:
150736   LOOKAT: L=:D; CALL SMENTER;  "STRTYPE"; CALL SMGPAR
150742          X:="LOOKTAB"; CALL SMABLOOK; T=:X
150745          IF ><0 THEN "6TERRP"; CALL SMOUTTEXT; GO SMLEAVE FI
150751          X.CPROTECT=:CPRO; X.CMAND=:P
150755   GPREGS: -1=:GPSEGM; GO OUT1
150760   GPCORE: BCSEG/\377=:GPSEGM; GO OUT1
150764   AGPCOR: BCSEG/\377 BONE 17=:GPSEGM; GO OUT1
150771   GPIM:  5CIMSEG=:GPSEGM; GO OUT1
150774   RESIDENT: 0=:GPSEG; GO OUT1
150776   CCOM:  1=:GPSEGM; GO OUT1
151001   GPPHY: -2=:GPSEGM; GO OUT1
151004   GPDSG: "6PSEGNO"; CALL SMSGPAR; GO SMLEAVE
151007          A BONE 17=:GPSEGM BZERO 17; GO SGFEL
```

```
151013   GPSG:   "6PSEGNO"; CALL SMSGPAR; GO SMLEAVE; A=:GPSEGM
151017   SGFEL:  IF <USEGM THEN MIN CPRO FI
151023           IF A=0 OR A>>SGMAX THEN "TXILSG"; CALL SMOUTTEXT; GO SMLEAVE FI
151032           IF A=RTREF.RSEGM THEN "TILSG"; CALL SMOUTTEXT; GO SMLEAVE FI
151041           GO OUT1; *)FILL
151062   OUT1:   IF CPRO>PASSTYPE THEN "PROT"; CALL SMOUTTEXT; GO SMLEAVE FI
151071           "READY"; CALL SMOUTTEXT
151073
151073           IF BCHFLAG=0 THEN 2; *MON 2ECHO; MON 2BRKM
151100           FI; 0=:ADR=:ADR1
151102   INI:    0=:STATE=:BCHAR
151104           IF GPSEGM=-1 GO FAR LOOREGS
151110   LOOP:   CALL FAR BTCI; IF =40 GO LOOP; IF =##. OR =##@ GO EXI
151122           IF =##* OR =##: THEN
151130              CALL SMCRLF
151131              IF GPSEGM=-2 THEN DDADR; CALL SMDOCTU ELSE ADR; CALL SMOCTU FI
151142              CALL SMCRLF; GO LOOP
151144           FI; CALL SMSRCHINT(CHTAB); A=:X; IF <=7 THEN A=:REGNO FI; IF >13 GO ERR
151156           4*STATE+INPT(X):=HUFF(A)
151163           T:=A SHZ -14=:STATE; A SH 4 SH-4+"BB"=:P
151172   *)FILL
151207
151207   BB:
151207   DIGO:   CALL FAR ASSOCT; GO LOOP
151211   LETTO:  CALL FAR BTCI; IF =40 GO LETTO; IF ><##/ GO ERR; ESCBLOCK(REGNO);GO LISTVAL
151223   SLSH1:  DVALUE=:DDADR; GO GETVALUE
151226   CRET1:  IF GPSEGM=-2 THEN
151232              X:=ADR; T:=ADR1; VALUE; *STATX
151236           ELSE
151237              VALUE; X:=ADR; T:=GPSEGM; CALL PUT1L; GO ERR2
151244           FI
151244   CRETO:  DDADR; D+1; A:=A+C; AD=:DDADR
151250   GETVA:  IF GPSEGM=-2 THEN
151254              X:=ADR; T:=ADR1; *LDATX
151257           ELSE
151260              X:=ADR; T:=GPSEGM; CALL GET1L; GO ERR2
151264           FI
151264   LISTVAL: CALL SMOCTU; 40; CALL SMTCO; GO LOOP
151270   CRET3:  VALUE=:ESCBLOCK(REGNO)
151273   RETURN: GO LOOP
151274   ERR2:   "6ILLA"; CALL SMOUTTEXT; GO FAR ERRF
151277
151277   EXI:    IF GPSEGM BZERO 17<377 AND A>1 THEN
151307              A=:GPSEGM; "PWSGB"; *MON 2WSEG
151312           FI
151312   RETU:   "ENDL"; CALL SMOUTTEXT; GO SMLEAVE
151315   ERR:    DO
151315           CALL FAR BTCI; CALL MBTERM
151317           WHILE A/\177><15
151323           OD
151324           CALL SMCRLF; ##?; CALL SMTCO; CALL SMCRLF
151330   ERRF:   IF BCHFLAG=0 GO FAR INI
151333           X:="BILPAR"; GO TOOPCSEG                    % TO ROUTINE ON OPCOM SEG.
151335   *)FILL
151361
151361   INTEGER CIRW(0); *IRW
151362   INTEGER CIRR(0); *IRR
151363   LOOREGS: 0=:VALUE; CALL FAR BTCI; CALL MBTERM; IF A=40 GO LOOREGS
151371           CALL SMSRCHINT(CHTAB); IF A=13 GO LOOREGS
151376           IF A>10 GO FAR ERR
```

```
151401          IF A=T THEN
151403                CALL FAR ASSOCT; IF MINFLAG><0 GO FAR ERR
151407                CALL FAR BTCI; CALL MBTERM
151411                CALL SMSRCHINT(CHTAB); IF A>7 GO FAR ERR
151416          FI
151416   LX:    RREGNO(A)=:REGNO; VALUE=:CLEVL
151423          IF A<0 OR A>17 GO FAR ERR
151427          CALL FAR BTCI; CALL MBTERM; IF A><##/ GO FAR ERR
151434          CLEVL SH 3+CIRR+REGNO=:T; *EXR ST
151442          CALL SMOCTU; 40; CALL SMTCO
151445   L1:    CALL FAR BTCI; CALL MBTERM; IF A=40 GO L1
151452          CALL SMSRCHINT(CHTAB); IF A>13 GO FAR ERR
151457          IF A=T GO LOOREGS
151461          IF A<10 THEN 0=:VALUE; GO LX FI
151466          IF A><T GO FAR ERR
151470          CALL FAR ASSOCT; CALL FAR BTCI; CALL MBTERM; CALL SMSRCHINT(CHTAB)
151475          IF A=13 THEN
151500                IF CLEVL<5 AND A><2 GO FAR ERR
151507                IF A>11 AND A><17 GO FAR ERR
151515                A SH 3+CIRW+REGNO=:T; VALUE; *EXR ST
151523                GO LOOREGS
151524          FI
151524          IF A<10 GO LX
151527          GO FAR ERR
151530   MBTERM: IF A=##. OR A=##@ GO FAR RETU
151536          EXIT
151537   *)FILL
151554
151554   @ICR
151554   INTEGER ARRAY HUFF:=(
151554          DIGO-BB+T1,LETTO-BB+T2,ERR-BB,CRETO-BB,
151560          ERR-BB,ERR-BB,SLSH1-BB,CRET1-BB,
151564          DIGO-BB+T3,LETTO-BB+T2,ERR-BB,RETURN-BB,
151570          ERR-BB,ERR-BB,SLSH1-BB,CRET3-BB);
151574   @CR;
151574   RBUS
151574
151574
151574   %=============================================================================
151574   % 18.26       T E R M S T A T
151574
151574   % COMMAND: TERMINAL-STATUS <LOG.UNIT> <INTERVAL>
151574   %
151574   SUBR TERMSTAT
151574   DISP -200
151574          INTEGER LOGN,ITIN,SYSG,CURP,WORD,COUNT,PRG,TPFIELD
151574          INTEGER POINTER TPNT
151574          INTEGER CDBPROG=WORD,CBSTATE=COUNT,CCBPN=WORD
151574   PSID
151574   INTEGER STRTIME:='IDINTERVAL(SEC.): '
151606   INTEGER USERMODE:=' USER    ',COMODE:=' COMMAND'
151620   INTEGER SRTWT:=' RTWT    ',SHOLD:=' HOLD    '
151632   @ICR
151632   INTEGER HEADING:=
151632   ' LOG.NO  USER              MODE     CPU-MIN OUT OF  LAST COMMAND$';
151672   @CR;
151672   INTEGER TPNTS:=TPFIELD+BFIELD
151673   INTEGER PARHOLD:=(ITIN+BFIELD,"2")
151675
151675   TERMSTAT: L=:D; CALL SMENTR
```

```
151677          "PTERMNO"; CALL SMGPAR; A=:LOGN
151702          "STRTIME"; CALL SMGPAR; A=:ITIN
151705   NEXT:  CALL SMCRLF; "HEADING"; CALL SMOUTTEXT
151710          IF LOGN><0 THEN        % ONE TERMINAL ONLY
151712                CALL LOGPH; CALL SMCHTERM; GO FAR BYP
151715                A=:TPFIELD; TPNTS=:"TPNT"
151720          ELSE "BACKTAB"=:"TPNT"
151723          FI; GO BYPA; *)FILL
151740   BYPA:  IF X:=TPNT=-1 GO FAR BYP
151744            IF X=0 GO FAR NXT
151746                T:="BSTATE"; CALL XGTDFADDR; A=:CBSTATE
151751                T:="DBPROG"; CALL XGTDFADDR; A=:CDBPROG
151754                CALL SMTACTIVE; GO FAR NXT
151756                IF CBSTATE=0 OR A=5CFILTRA OR CDBPROG=0 GO FAR NXT
151766                T:=CDBPROG.SEGM SHZ -10=:SYSG; GO LABL1; *)FILL
152001   LABL1:       X:="CUSER"; CALL GET1L; GO FAR ERR            % MOD. 24/6 FOR
152004                IF A>=0 THEN                                  % 4096 USERS
152005                  A=:CURP
152006                  X:="TTNO"; CALL GET1L; GO FAR ERR; CALL SMDECU; 40; CALL SMTCO
152014                  T:=CURP SHZ -10; A:=CURP/\377               % 4096 USERS
152020                  X:="WORKA"; CALL FILSYS(GUSEN); GO FILERR
152024                  FOR X:=0 TO 17 DO T:="WORKA"; *LBYT
152032                    WHILE ><##'; CALL SMTCO
152036                  OD; GO L1; *)FILL
152052   L1:            FOR X TO 17 DO 40; CALL SMTCO OD
152061                  IF CDBPROG.WLINK=0 THEN
152064                    IF X.TLINK=0 THEN "SRTWT" ELSE "SHOLD" FI
152071                  ELSE
152072                    IF CBSTATE=2 THEN "USERMODE" ELSE "COMODE" FI
152101                  FI; CALL SMOUTTEXT
152102                  CDBPROG.DTINT; T:=9TIM2; *RDIV ST
152106                  CALL SMDECU; 0=:TDVN; X=:A; T:=SYSG; CALL SMTMTUSED
152113                  1=:TDVN; TIOOF; T:=9TIM2; *RDIV ST
152120                  CALL SMDECU; "40\40"; CALL SMTCO2; "COMSTRING"=:CURP
152125                  IF EXSECUR NBIT 7NCMLINE GO NOTCENT
152130                  X:="CURUSER"; T:=SYSG; CALL GET1L; GO ERR
152134                  IF A><CURUSER AND PASSTYPE><2 GO NXT1
152143   NOTCENT:       -17=:COUNT
152145                  FOR COUNT DO
152145                    T:=SYSG; X:=CURP; CALL GET1L; GO ERR
152151                    A=:WORD SHZ -10 WHILE ><15; CALL CCTCO
152157                    WORD/\177 WHILE ><15; CALL CCTCO; MIN CURP
152166                  OD
152170   NXT1:          CALL SMCRLF
152171                FI
152171   %%         FI
152171   NXT:       IF LOGN><0 GO BYP
152173              CALL ESCON; CALL ESCOFF
152175              MIN "TPNT"
152176           GO FAR BYPA; *)FILL
152224   BYP:    IF ITIN><0 THEN
152226                CALL ESCON; "PARHOLD"; *MON 2HOLD
152231                CALL ESCOFF; GO FAR NEXT
152233           FI
152233   OUT:    GO SMLEAVE
152234   ERR:    "ILLA"; CALL SMOUTTEXT; GO SMLEAVE
152237   CCTCO:  IF A<40  THEN ##& FI; GO SMTCO
152244
152244   RBUS
152254
```

```
152254
152254   %=========================================================================
152254   % 18.22       L I S T R T
152254
152254   %COMMAND: LIST-RT-DESCRIPTION <RT NAME>
152254   %
152254   SUBR LISTRT
152254   @ICR
152254   INTEGER STRRT:='R RT NAME: ',XTQ:='IN TIME QUEUE$',
152272           XS17:='I/O-WAIT ',XS16:='REPEAT ',XS15:='RTWT ',XS14:='INTV ',
152311           XS13:='ABSET ',XR:=' RING:',XP:=' PRIORITY:',
152327           XIP:='$INITIAL PIT: ',XNP:=' NORMAL PIT: ',XAP:=' ALTERNATIVE PIT: ',
152360           XTL:='$TIME LEFT:',XLS:='$LAST STARTED:',XINT:='$INTERVAL:',
152404           XST:='$START ADDRESS:',XSEG:=', SEGMENTS:',XWT:='$WAITING FOR:',
152431           XACT:='$ACTUAL SEGM.:',XBG:=' BACKGROUND',XOF:=' RTOFF',
152453           REGNAME:='PXTADLSB',XPASS:='$PASSIVE',XREA:='$READY',
152471           XRSG:='$REENTRANT SEGMENT:',TIDX:='INPUT',TODX:='OUTPUT',
152512           LAMTX:='$LAMUS CONNECTED   LOGICAL PAGE', TXLGU:='  LOGICAL UNIT',
152542           RESSO:='$RESERVED DATAFIELDS  LOGICAL UNIT';
152564   @CR;
152564   INTEGER ARRAY SBITS:=(XS17,XS16,XS15,XS14,XS13)
152571   DISP -200
152571           INTEGER WORD,WTDEVICE; INTEGER ARRAY POINTER REGS,RTADR
152571           INTEGER DAC11,DAC22; DOUBLE DACC2=DAC11
152571           INTEGER POINTER LREG
152571           INTEGER RTNO,CINDX,LTXW=WORD, XMAX=RTNO, LAMNO=WTDEVICE
152571           INTEGER CDATF=WORD,CLOGU=WTDEVICE,LGGROUP=DAC11
152571   PSID
152571
152571   %LOCAL SUBROUTINE FOR DOUBLE PRECISION SUBTRACTION
152571   DSUB:   A:=:L=:"LREG"; D=:T; DACC2; D-; *COPY CM1 ADC SA DA
152577           D+T; A+L+C; GO LREG
152602
152602
152602   LISTRT: L=:D; CALL SMENTER
152604            "STRRT"; CALL SMGPAR; IF =0 THEN RTREF FI
152610           A=:X; IF >=SEGSTART OR A-RTSTART<0 GO SMLEAVE
152617           A=:D:=0; T:=5RTSIZE; *RDIV ST
152623           A=:RTNO; IF D><0 GO SMLEAVE; CALL RTDCOPY; X=:WTDEVICE
152630           IF "WORKA".TLINK><0 THEN "XTQ"; CALL SMOUTTEXT FI
152635           X.STATUS=:WORD
152637           FOR X:=0 TO 4 DO
152643                 IF WORD<0 THEN SBITS(X); CALL SMOUTTEXT FI; WORD SH 1=:WORD
152652           OD
152654           GO BYP; *)FILL
152671   BYP:    "XR"; CALL SMOUTTEXT; "WORKA".STATUS SH -10/\3+60; CALL SMTCO
152701           "XP"; CALL SMOUTTEXT; X.STATUS/\377; CALL SMDECU
152706           "XIP"; CALL SMOUTTEXT; X.ACTPRI SHZ -13/\3+60; CALL SMTCO
152715           "XNP"; CALL SMOUTTEXT; X.ACTPRI SHZ -11/\3+60; CALL SMTCO
152724           "XAP"; CALL SMOUTTEXT; X.ACTPRI SHZ -7/\3+60; CALL SMTCO
152733           IF X.DTIM1\/X.DTIM2><0 THEN
152736              IF X.TLINK><0 THEN
152740                 "XTL"; CALL SMOUTTEXT; MTIME=:DACC2; X.DTIME; CALL DSUB
152746              ELSE  "XLS"; CALL SMOUTTEXT; X.DTIME=:DACC2; MTIME; CALL DSUB
152755              FI; CALL SMTIMOUT
152756           FI
152756           IF X.STATUS BIT 5INT THEN
152761                 "XINT"; CALL SMOUTTEXT; X.DTINT; CALL SMTIMOUT
152765           FI
152765           "XST"; CALL SMOUTTEXT; X.STADR; CALL SMOCTU; "XSEG"; CALL SMOUTTEXT
```

```
152773          X.SEGM SHZ -10; CALL SMOCTU; X.SEGM/\377; CALL SMOCTU
153001          "WORKA+5RTSIZE"=:"REGS"
153003          FOR X:=0 TO 7 DO  CALL SMCRLF; T:="REGNAME"; *LBYT
153012                CALL SMTCO; ##=; CALL SMTCO; REGS(X); CALL SMOCTU
153017          OD; GO BYP2; *)FILL
153050   BYP2:  IF WTDEVICE=0 THEN "XPASS"; CALL SMOUTTEXT
153054          ELSE IF <<="BEXQU" THEN "XREA"; CALL SMOUTTEXT
153062          ELSE "XWT"; CALL SMOUTTEXT; WTDEVICE; CALL SMOCTU
153067               WTDEVICE; CALL SMFFLOGDV; GO LABL1; A=:CLOGU
153073               "TXLGU"; CALL SMOUTTEXT
153075               CLOGU BZERO 17; CALL SMOCTU
153100               4; CALL SMSPAC
153102               IF CLOGU BIT 17 THEN "TODX" ELSE "TIDX" FI; CALL SMOUTTEXT
153111          FI FI
153111   LABL1: "XACT"; CALL SMOUTTEXT
153113          "WORKA".ACTSEG SHZ -10; CALL SMOCTU; X.ACTSEG/\377; CALL SMOCTU
153122          IF X.ACTPRI BIT 17 THEN "XBG"; CALL SMOUTTEXT FI
153127          IF X.ACTPRI BIT 16 THEN "XOF"; CALL SMOUTTEXT FI
153134          IF X.RSEGM><0 THEN "XRSG"; CALL SMOUTTEXT; X.RSEGM; CALL SMOCTU FI
153142          GO BYP3; *)FILL
153164   BYP3:  IF WORKA("5RTSIZE+10")><-1 THEN
153171             "RESSO"; CALL SMOUTTEXT
153173             FOR X TO "5WORKSIZE-1" DO WHILE WORKA(X)><-1
153202                A=:CDATF; X=:CINDX; CALL SMCRLF; 10; CALL SMSPACE
153207                A:=CDATF; CALL SMOCTU; CDATF; CALL SMFFLOGDV; GO INDOO; A=:CLOGU
153215                10; CALL SMSPACE; CLOGU BZERO 17; CALL SMOCTU; 4; CALL SMSPACE
153224                IF CLOGU BIT 17 THEN "TODX" ELSE "TIDX" FI; CALL SMOUTTEXT
153233   INDOO:       X:=CINDX
153234             OD
153236          FI; GO BYP4; *)FILL
153250
153250   BYP4:  RTNO*GNLPRT*ALMSZ+LAMACT=:X; GNLPRT*ALMSZ-ALMSZ+X=:XMAX; 0=:LTXW
153263          FOR X STEP ALMSZ TO XMAX DO
153266                T:=LAMBANK; *LDATX LMCN
153270                IF A >< 0 THEN
153271                   IF LTXW = 0 THEN 1=:LTXW; "LAMTX"; CALL SMOUTTEXT FI; CALL SMCRLF
153300                   T:=LAMBANK; *LDATX LMCN
153302                   CALL SMOCTU; X=:CINDX; 13; CALL SMSPACE
153306                   X:=CINDX; T:=LAMBANK; *LDATX LMLP
153311                   CALL SMOCTU
153312                FI
153312          OD; CALL SMCRLF; GO SMLEAVE
153316   RBUS
153330
153330   %=============================================================================
153330   %       S M F F L O G D V
153330   %
153330   % SUBROUTINE TO FIND LOGICAL DEVICE NUMBER FROM DATAFIELD ADDRESS
153330   %
153330   % ENTRY:      A=DATAFIELD ADDRESS
153330   %
153330   % EXIT:       NO LOGICAL DEVICE NUBER FOUND
153330   %
153330   % EXIT+1:     A=LOGICAL DEVICE NUMBER
153330   %             A BIT 17=1 THEN OUTPUT ELSE INPUT
153330   %
153330   SUBR SMFFLOGDV
153330
153330   DISP -200; INTEGER LGGROUP,CLOGU,CDATF; PSID
153330
```

```
153330   SMFFLOGDV: L=:D; CALL SMENTR
153332          A=:CDATF; 0=:LGGROUP
153334          DO WHILE LGGROUP<25
153340             IF LGGROUP><1 THEN
153344                X:=CNVRT(A); A SHZ 6=:CLOGU; T:=0; *LDATX
153352                A+A+X=:L; X+1
153356                DO WHILE X<<L
153360                   T:=0; *LDDTX
153362                   IF A=CDATF GO FOUND
153365                   IF D=T AND CLOGU><FIXEDEV THEN CLOGU BONE 17=:CLOGU; GO FOUND FI
153377                   X+2; MIN CLOGU
153401                OD
153402             FI; MIN LGGROUP
153403          OD; GO SMLEAVE
153405   FOUND: A:=CLOGU; GO SM2LEAV
153407   RBUS
153413
153413   %=================================================================================
153413   % 18.23       U E L G N
153413   %
153413   %       COMMAND: UE-AUTOMATIC-LOGIN  <ALL TERMINALS?> <ENABLE/DISABLE FLAG> <LDN>
153413   %
153413   %       ALL TERMINALS?:        YES/NO
153413   %
153413   %       ENABLE/DISABLE FLAG:   1=ENABLE,0=DISABLE
153413   %
153413   %       LDN:                   TERMINAL NUMBER (DEFAULT=OWN)
153413   %
153413   %
153413   %       THE ROUTINE "UELGN" READS THE WORD UEFLG (FLAG WORD UTILIZED BY USER-ENVIRONMENT) FOR THE TERMINAL(
153413   %       IN QUESTION AND SETS/RESETS BIT 13 (5UEAL). WHEN SET, THE END USER WILL BE LOGGED IN AS USER
153413   %       "USER-ENVIRONMENT" WHEN PRESSING THE "ESCAPE" AND THE UE-LOGIN PROGRAM WILL BE STARTED.
153413   %
153413   %=================================================================================
153413
153413   SUBR UELGN
153413   DISP -200
153413          INTEGER LOGN,TPFIELD,CDBPROG,SYSG,WORD,CEDFL,DYNAL
153413          INTEGER POINTER TPNT,REXAX
153413   PSID
153413   INTEGER UEHDR:='IDENABLE=1/DISABLE=0: '
153427   INTEGER UEALL:='N ALL TERMINALS ? '
153441   INTEGER UETRN:='IDTERMINAL NUMBER: '
153453   INTEGER PWSCG:=GPSEGM          % GPSEGM IS ON SYSTEM SEGMENT (USED ALSO IN LOOK-AT)
153454   UELGN: L=:D; CALL SMENTER
153456          0=:DYNAL
153457          "UEALL"; CALL SMGPAR; CALL SMYESNO
153462          A=:WORD                % YES=1, NO=0
153463          IF A=-1 GO ERR1                                  % ILLEGAL PARAMETER
153466          "UEHDR"; CALL SMGPAR; A=:CEDFL
153471          IF A<0 OR >1 THEN  GO ERR1 FI                    % ILLEGAL PARAMETER
153476          IF WORD=0 THEN                                   % NOT ALL TERMINALS
153500            "UETRN"; CALL SMGPAR; A=:LOGN
153503             IF A=0 THEN TTNO=:LOGN FI                     % ONE TERMINAL ONLY
153506             CALL LOGPH; CALL SMCHTERM; GO ERR2            % GET DF.ADDR AND CHECK IF TERMMINAL
153511             A=:X; CALL STFLG
153513          ELSE
153514             "BACKTAB"=:"TPNT";                            % INITIATE POINTER TO DATAFIELDS
153516             DO WHILE X:=TPNT><-1
153522                CALL STFLG
```

```
153523                 MIN "TPNT"; 0/\0;
153525              OD
153526           FI
153526   UEOUT:  IF DYNAL><0 THEN
153530              5PT3S=:GPSEGM; "PWSCG"; *MON 2WSEG
153534           FI; GO SMLEAV
153535   ERR1:   174; GO ERR
153537   ERR2:   311; GO ERR
153541   ERR3:   153; GO ERR
153543   ERR:    *MON 64                                    % ILLEGAL ADDRESS REF. IN MONITOR CALL
153544           GO UEOUT
153545   *)FILL
153562   % LOCAL SUBROUTINE TO ENABLE/DISABLE AUTOMATIC LOGIN:
153562   % ENTRY: X-REG: POINTER TO DATAFIELD
153562   STFLG: A:=L=:"REXAX"
153564           IF X><0 THEN
153565              CALL SMDYNALLOC; GO NODYN; MIN DYNAL
153570              A*5PRVT+"PRVTTABLE"+"PRVUEFLG"=:X
153574              T:=5PT3S; CALL GET1L; GO ERR3
153577              IF T:=CEDFL>0 THEN A BONE 5UEAL ELSE A BZERO 5UEAL FI
153605              T:=5PT3S; CALL PUT1L; GO ERR3
153610              GO REXAX
153611
153611   NODYN:     T:="DBPROG"; CALL XGTDFADDR; A=:CDBPROG    % GET RT-DESCR. ADDR.
153614              T:=CDBPROG.SEGM SHZ -10=:SYSG               % AND SYSTEM SEGMENT
153620              X:="BCHFLG"; CALL GET1L; GO ERR3
153623              IF A=1 OR =-1 THEN GO REXAX; FI
153632              T:=SYSG; X:="UEFLG"; CALL GET1L; GO ERR3
153636              T:=CEDFL
153637              IF T>0 THEN A BONE 5UEAL ELSE A BZERO 5UEAL FI
153644              T:=SYSG; X:="UEFLG"; CALL PUT1L; GO ERR3 % SAVE NEW VALUE ON SYSTEM SEGMENT
153650              SYSG/\377=:GPSEGM; "PWSCG"; *MON 2WSEG
153655           FI
153655           GO REXAX
153656   RBUS
153672
153672   %==========================================================================================%
153672   %                                                                                          %
153672   %       S M T R S                                                                          %
153672   %                   MONITOR CALL TO PERFORM SIMILAR FUNCTION AS THE COMMAND                %
153672   %                              @TERMINAL-STATUS, EXCEPT THAT                               %
153672   %                   IT WILL GET THE STATUS FOR ONE LOGICAL DEVICE NO. ONLY                 %
153672   %                                                                                          %
153672   %       INPUT PARAMETERS:      T-REGISTER = LDN                                            %
153672   %                              A-REGISTER = ADDRESS OF RECEIVING PROGRAM ARRAY             %
153672   %                                                                                          %
153672   %       OUTPUT FROM TRTER TO PROGRAM'S ARRAY:                                              %
153672   %                                                                                          %
153672   %                              WORD  0 - 7: USER NAME (IF LESS THAN 16 CHR.,TERMINATED WITH A " ")  %
153672   %                              WORD     10: MODE ( 1 = COMMAND, 2 = USER, 3 = RTWT, 4 = HOLD )  %
153672   %                              WORD     11: STATE ( -1= PASSIVE,0 = IDLE(BATCH ONLY),1=ACTIVE   %
153672   %                              WORD     12: CPU MINUTES USED                                  %
153672   %                              WORD     13: TIME THE TERMINAL HAS BEEN LOGGED ON IN MINUTES   %
153672   %                              WORD 14 -25: LAST COMMAND EXECUTED                             %
153672   %                                                                                          %
153672   %       EXIT:                  SKIP RETURN = OK                                            %
153672   %                              RETURN      = ERROR                                         %
153672   %                                                                                          %
153672   %==========================================================================================%
153672   SUBR SMTRS
```

```
153672   DISP -200
153672          INTEGER BREG,LREG,LOGN,SYSG,WPNT,TSEGM
153672          INTEGER INDXX=TSEGM,INDYY=LOGN,T2SEG,APTAB,NPTAB
153672          INTEGER CDBPROG=APTAB,CBSTATE=NPTAB,CIDLE,WORD=CIDLE
153672          INTEGER 1TIMX=APTAB,2TIMX=NPTAB
153672   PSID
153672   % ENTERS THIS ROUTINE WITH X=BREG AND A=LREG:
153672   SMTRS: X=:BREG; A=:LREG
153674          X.ZTREG=:LOGN
153676          IF A=0 THEN A:=174; GO FAR TRERR FI    % ILLEGAL PARAMETER
153701          X.STRSEG=:TSEGM; X.ZAREG=:WPNT;         % PREPARATIONS NECESSARY BEFORE
153705          X.OLDPAG SHZ -7/\3=:APTAB; X.OLDPAG SHZ -11/\3=:NPTAB % PUT1L MAY BE USED TO
153715          TSEGM/\377=:T2SEG*5SEGSIZE+SEGSTART=:X
153723          X.LOGADR SHZ -10              %    DELIVER REQUESTED INFORMATION
153725          IF A=200 AND APTAB><NPTAB THEN T2SEG BONE 17=:T2SEG FI % TO THE CALLING PROGRAM.
153737          LOGN; CALL LOGPH                    % GET ADDRESS OF TMNL DATA FIELD
153741          CALL SMCHTERM; GO FAR TRNOT; A=:X
153744          T:="DBPROG"; CALL XGTDFADDR; A=:CDBPROG   % AND SAVE THEM IN WORK-FIELDS
153747          CALL SMTACTIVE; GO LABL1
153751          T:="BSTATE"; CALL XGTDFADDR; A=:CBSTATE   % "ISTATE","BSTATE" AND "DBPROG"
153754          IF A=0 OR =5CFILTRA OR CDBPROG=0 THEN
153762   LABL1:    A:=-1;X:=WPNT+11; T:=T2SEG; CALL PUT1L; GO FAR ERR; GO FAR TROUT % BCHXX/BAKXX IS PASSIVE
153771          FI
153771          GO L0; *)FILL
154012   L0:    IF X.TYPRING NBIT 5TERM THEN        % IF IT WAS A TERMINAL
154015             T:="IDLE"; CALL XGTDFADDR;
154017          ELSE
154020             A:=1
154021          FI; A=:CIDLE
154022          T:=CDBPROG.SEGM SHZ -10=:SYSG; X:="CUSER"; CALL GET1L; GO FAR ERR % GET CURRENT USER INDEX
154031          IF A<0 THEN GO FAR TRNIL FI                % NO USER LOGGED ON
154033          A=:T; T SHZ -10; A/\377; X:="WORKA" % GET USER NAME AND SAVE IT IN "WORKA"
154037          CALL FILSYS(GUSEN); GO FILERR; D=:X % ON SYSTEM SEGMENT
154043          FOR X TO 7 DO
154046             A:=WORKA(X); X=:D
154050             X:=WPNT; T:=T2SEG; CALL PUT1L; GO FAR ERR % DELIVER LOGGED IN USER
154054             MIN WPNT; 0/\0; D=:X
154057          OD
154061          IF CDBPROG.WLINK=0 THEN             % IF BACKGR. PROG  NOT IN EXEC. QUEUE
154064             IF X.TLINK=0 THEN 3 ELSE 4 FI  % IS IT IN "RTWT"(3) OR "HOLD"(4)????
154071          ELSE
154072             IF CBSTATE><2 THEN 1 FI         % 2=USER MODE,1=COMMAND MODE
154077          FI
154077          A=:CBSTATE; X:=WPNT; T:=T2SEG; CALL PUT1L; GO FAR ERR; MIN WPNT; 0/\0 % DELIVER MODE
154106          GO L2; *)FILL
154121   L2:    IF CIDLE>1 THEN 1 FI      % BATCH: IF ACTIVE IDLE MAY VARY
154126          T:=T2SEG; X:=WPNT; CALL PUT1L; GO FAR ERR; MIN WPNT; 0/\0 % DELIVER STATE
154134          CDBPROG.DTINT; T:=9TIM2; *RDIV ST                          % GET CPU MINUTES USED
154140          T:=T2SEG; X:=WPNT; CALL PUT1L; GO FAR ERR; MIN WPNT; 0/\0
154146          T:=SYSG; X:="1TIMON"
154150          CALL GET1L; GO FAR ERR; A=:1TIMX   % GET LOG ON TIME FROM SYSTEM SEGMENT
154153          X+1; CALL GET1L; GO FAR ERR; A=:2TIMX; AD:=ATIME % & LEVEL 13 TIME (ATIME)
154160          A:=:D-2TIMX; *RDCR ADC DD      % AND CALCULATE HOW MANY CPU MINUTES
154163          A:=:D-1TIMX; T:=9TIM2; *RDIV ST                % THE TERMINAL HAS BEEN LOGGED ON
154167          T:=T2SEG; X:=WPNT; CALL PUT1L; GO FAR ERR; MIN WPNT; 0/\0 % DELIVER TO PROGRAM
154175          "COMSTRING"=:INDXX
154177          IF EXSECUR NBIT 7NCMLINE GO NOTCENT
154202          X:="CURUSER"; T:=SYSG; CALL GET1L; GO FAR ERR
154206          IF A><CURUSER AND PASSTYPE><2 GO SLBYT
154215          GO NOTCENT; *)FILL
```

```
154227   NOTCENT: -12=:INDYY
154231           FOR INDYY DO
154231              T:=SYSG; X:=INDXX; CALL GET1L; GO ERR; MIN INDXX; 0/\0      % GET LAST COMMAND EXECUTED
154237              A=:WORD SHZ -10 % FROM COMMAND STRING BUFFER
154241              IF A=15 THEN
154244   SLBYT:        -1=:INDYY; A:=23400=:WORD; ELSE
154251              IF WORD/\177=15 THEN -1=:INDYY; WORD/\177400\/47; ELSE WORD; FI; FI
154265              T:=T2SEG; X:=WPNT; CALL PUT1L; GO ERR;
154271              MIN WPNT; 0/\0
154273           OD
154275   TROUT: A:=0; MIN LREG
154277   TRERR: A=:BREG.ZAREG;
154301          LREG=:L:=BREG=:B; TAD:=ZTADREG; X:=ZXREG; EXIT
154310   TRNIL: A:=-1; GO TRERR      % NO USER LOGGED ON
154312   TRNOT: A:=311; GO TRERR     % LDN IS NOT A TERMINAL
154314   ERR:   A:=153; GO TRERR     % ERROR EXIT FROM GET1L/PUT1L
154316   RBUS
154325
154325
154325   %=====================================================================
154325   %              C O S M O S   C O M M A N D S
154325   %
154325   % 22.10B      X B A D M    S T A B A    S T O B A
154325
154325   % COMMANDS TO ACTIVATE TADADM
154325   % ENTRY
154325   %       X=1  - INITIALISE AND START (STABA)
154325   %       X=2  - STOP TADADM          (STOBA)
154325   %       X=3  - SUPERVISOR MODE      (XBADM)
154325
154325   % EXIT FROM ABADMI
154325   %       SEGMENT     LOADED: EXIT X=0
154325   %       SEGMENT NOT LOADED: EXIT X><0
154325
154325   SUBR XBADM,STABA,STOBA
154325   INTEGER RESPAR:=("5BADSEM",NULL,"1")
154330   INTEGER NOTLO:='TADADM NOT LOADED'
154341   INTEGER ALRES:='TADADM ALREADY IN USE'
154354   INTEGER XMNST:='NO MORE XT-BLOCKS (TASK BLOCKS) AVAILABLE'
154401   INTEGER NOBAD:='TADADM IS NOT INCLUDED IN THIS SYSTEM'
154424
154424   DISP -200; INTEGER BADFUN; PSID
154424
154424   STABA: A:=1; GO FELLS          % COMMAND: START-TADADM
154426   STOBA: A:=2; GO FELLS          % COMMAND: STOP-TADADM
154430   XBADM: A:=3                    % COMMAND: TADADM
154431   FELLS: L=:D; CALL SMENTER; A=:BADFUN
154434          IF "BADM"=0 GO MSNOB                        % TADADM NOT INCLUDED
154436          T:=0; *MON 2XMSG                            % XT-BLOCKS? (XFDUM)
154440          IF T=0 GO MSXNS
154442          "RESPAR"; *MON 2RESR                        % RESERVE TADADM SEMAPHORE
154444          IF A<0 GO MSALR                             % ALREADY IN USE
154445          X:=BADFUN; CALL ABADMI; IF X=0 GO RETU
154450          "NOTLO"; CALL SMOUTTEXT                     % SEGMENT NOT LOADED
154452   RETU:  "RESPAR"; *MON 2RELE                        % RELEASE TADADM SEMAPHORE
154454          GO SMLEAVE
154455
154455   MSNOB: "NOBAD"; GO ERR
154457   MSXNS: "XMNST"; GO ERR
154461   MSALR: "ALRES"
```

```
154462   ERR:    CALL SMOUTTEXT; GO SMLEAVE
154464   RBUS
154500
154500
154500   %=========================================================================
154500   % 22.10C          S R M M O          S L C M O
154500   %        SUBROUTINES TO SET REMOTE OR LOCAL MODE ON FILE ACCESS OPERATIONS
154500   %
154500   SUBR SRMMO,SLCMO
154500   SRMMO: A:=1; GO FELLS  % SET REMOTE MODE
154502   SLCMO: A:=0            % SET LOCAL MODE
154503   FELLS: L=:D; CALL SMENTER; *MON 2SLRM
154506          GO SMLEAVE
154507   RBUS
154511
154511
154511
154511   %=========================================================================
154511   %              T O O P C S E G
154511   %
154511   % RETURN TO A SPESIFIC ADDR ON OPCOM SEGMENT
154511   %
154511   % ENTRY: X=ADDR OF ROUTINE ON OPCOM SEGMENT
154511   %
154511   SUBR TOOPCSEG
154511   TOOPCSEG: X=:S3RETU; T:="S3RETU"; *MON 2MCAL
154514   RBUS
154515
154515
154515
154515
154515   %=========================================================================
154515   % 28.50       D T S L I C E   -   C M L T S L   -   C P T S L C L A S S
154515   %
154515   % DEFINE-TIME-SLICE
154515   % LIST-TIME-SLICE-CLASS
154515   % LIST-TIME-SLICE-PARAMETERS
154515   %
154515   SUBR DTSLICE,CMLTSL,CPTSLCLASS
154515
154515   DISP -200;
154515           INTEGER LEGAREA
154515           INTEGER CINDX,VALUE,CCLASS,CCADR,CELEM,SVALUE
154515           INTEGER CNEXT=CCLASS,XBT1=CCADR,XBT2=CELEM
154515           INTEGER POINTER C2LRG,ERRUT,C1LRG
154515           INTEGER 5LPR1=CINDX,5LPR2=VALUE
154515           INTEGER POINTER CROUTINE=ERRUT
154515   PSID
154515
154515
154515   INTEGER GLOBAL:='N CHANGE TIMESLICE PARAMETERS (YES/NO) (DEFAULT IS NO): '
154552   INTEGER TSYSP:='PRIORITY FOR OWNER OF SYSTEM RESOURCES WHICH$'
154601   INTEGER TSYS1:='          ARE WAITED FOR BY OTHER PROGRAMS      (1B -  77B) /'
154640   %INTEGER THED1:='
154640   INTEGER TTUNI:='NO. OF BASIC TIME UNITS IN ONE TIMESLICE UNIT  (1B - 400B) /'
154677   INTEGER TLOWL:='LOWEST PRIORITY BEFORE GETTING RAISED ON BREAK (1B -  70B) /'
154736   INTEGER THTIM:='LOWEST TIME COUNT BEFORE GETTING HASHED        (1B - 400B) /'
154775   INTEGER THASH:='BIT MASK USED WHEN HASHING                     (1B - 177B) /'
155034   INTEGER TCELM:='N $CHANGE TIMESLICE ELEMENTS (YES/NO) (DEFAULT IS NO)? '
155070   %INTEGER THED2:='
```

```
155070   INTEGER TTMSL:='TIMESLICE CLASS                (0B -   7B) /'
155116   INTEGER TESEL:='ESCAPE ELEMENT FOR THIS CLASS (0B -  37B) /'
155144   INTEGER TBREL:='BREAK ELEMENT FOR THIS CLASS  (0B -  37B) /'
155172   INTEGER TXELN:='TIMESLICE ELEMENT TO CHANGE   (0B -  37B) /'
155220   INTEGER TEPRI:='PRIORITY FOR THIS ELEMENT     (1B -  77B) /'
155246   INTEGER TETIM:='TIME COUNT FOR THIS ELEMENT   (1B - 400B) /'
155274   INTEGER TNEXT:='POINTER TO NEXT ELEMENT       (0B -  37B) /'
155322   INTEGER TSELN:='TIMESLICE ELEMENT NO.'
155335   INTEGER TMSLI:='N MORE CLASSES (YES/NO) (DEFAULT IS NO): '
155362   INTEGER TCHNE:='N CHANGE NEXT ELEMENT (YES/NO) (DEFAULT IS NO): '
155413   INTEGER TIMAG:='IMAGE', TSAVE:='SAVE'
155421   INTEGER IOINP:='IO/: '
155424   INTEGER TERR1:='    VALUE OUTSIDE RANGE$'
155441
155441   INTEGER CBSKP(0); *BSKP ZRO DT
155442   INTEGER CBSET(0); *BSET ONE 00 DT
155443
155443   % LOCAL SUBROUTINE TO TEST IF A BIT IN A BIT-ARRAY IS SET, IF NOT
155443   % THE BIT WILL BE SET
155443   % ENTRY:      A=BIT NUMBER
155443   % EXIT:       BIT IS ALREADY SET
155443   % EXIT+1:     THE BIT IS SET THIS TIME
155443   XSTBT: X:="XBT1"+B
155445          IF A>17 THEN X+1; A-20 FI
155452          T:=X.S0; A SH 3+CBSKP; *EXR SA; EXIT
155457          A-CBSKP+CBSET; *EXR SA
155462          T=:X.S0; EXITA
155464
155464   % LOCAL SUBROUTINE TO FIND THE HIGHEST "LOW-TIMESLICE-PRIORITY" USED
155464   % BY THE ND-500 TIMESLICER (N5SCHEDULER)
155464   UPLT1: A=:"CROUTINE":=L=:"C1LRG"
155467          0=:XBT1=:XBT2; 377=:5LPR1=:5LPR2
155474          "TSLBRKELEM"+CCLASS
155476          DO
155476             X:=A; CALL CROUTINE; A=:SVALUE; CALL XSTBT; GO UPLUT
155503             SVALUE+"TSLPRITAB"; X:=A; CALL CROUTINE
155507             IF A<5LPR1 THEN T=:5LPR2; A=:5LPR1 FI
155514             SVALUE+"TSLNEXTAB"
155516          OD
155517   UPLUT: IF 5LPR1><377 AND 5LPR2><T THEN
155526             CCLASS+"TSLLPRITAB"=:X; A:=5LPR2
155532             IF T:=LEGAREA BIT BIML THEN CALL W1IMAGE
155536                ELSE IF T BIT BSAVE THEN CALL W1SAVE
155542             FI; FI
155542          FI; GO C1LRG
155543
155543   UPLSTL: IF PN500=0 THEN EXIT FI                    % ND-500 NOT IMPLEMENTED
155546          A:=L=:"C2LRG"
155550          IF LEGAREA BIT BIML THEN "R1IMAGE"; CALL UPLT1; CALL WIMBACK FI
155556          IF LEGAREA BIT BSAVE THEN CALL 6SOPEN; "R1SAVE"; CALL UPLT1; CALL WXSAVE FI
155565          GO C2LRG
155566   +)FILL
155604
155604
155604   INTEGER ARRAY CADR:=(TSLSVPRI,TSLTUNIT,TSLLOWLG,TSLHTIME,TSLHASHM,TSLESCELEM,TSLBRKELEM,TSLPRITAB,TSLTIMTA
155616   *TSYSP
155617   INTEGER ARRAY TADR:=(TSYS1,TTUNI,TLOWL,THTIM,THASH,TESEL,TBREL,TEPRI,TETIM,TNEXT)
155631   INTEGER ARRAY CHADR:=(0,3,6,3,5,1,1,0,4,1)
155643
155643   DTSLICE: L=:D; CALL SMENTER
```

```
155645          6; CALL SELAR; IF A=0 GO SMLEAVE; A=:LEGAR
155652          IF A BIT BSAVE THEN CALL 6SOPEN FI
155655          "GLOBAL"; CALL SMGPAR; CALL SMYESNO
155660          IF A=1 THEN
155663             73; CALL SMSPACE
155665             IF LEGAR BIT BIML THEN "TIMAG"; CALL SMOUTTEXT FI
155672             IF LEGAREA BIT BIML AND A BIT BSAVE THEN 2; CALL SMSPACE FI
155701             IF LEGAR BIT BSAVE THEN "TSAVE"; CALL SMOUTTEXT FI
155706             CALL SMCRLF
155707   CE0:      -1=:CINDX; "CE0"=:"ERRUT"
155713             DO WHILE CINDX><5
155717   CE01:        TADR(CINDX); CALL SMOUTTEXT
155722                IF CINDX>=0 THEN
155724                   X:=CADR(CINDX)
155726                   CALL GETVAL; "IOINP"; CALL SMAGPAR; VALUE
155732                   X:=CINDX; CALL CHKRNG; GO SSMON; X:=CADR(CINDX); CALL SAVE
155740                FI; MIN CINDX; 0/\0; "CE01"=:"ERRUT"
155744             OD
155745          FI; GO BYPAS; *)FILL
155774
155774   BYPAS: CALL SMCRLF; "TCELM"; CALL SMGPAR; CALL SMYESNO
156000          IF A=1 THEN
156003             0=:CCLASS
156004             DO
156004                5=:CINDX; "CE1"=:"ERRUT"
156010   CE1:         "TTMSL"; CALL SMOUTTEXT; CCLASS; CALL SMOCTU
156014                "IOINP"; CALL SMAGPAR; CCLASS; A=:CCLASS
156020                CALL CHCLASS; GO SSMON
156022                53; CALL SMSPACE
156024                IF LEGAR BIT BIML THEN "TIMAG"; CALL SMOUTTEXT FI
156031                IF LEGAREA BIT BIML AND A BIT BSAVE THEN 2; CALL SMSPACE FI
156040                IF LEGAR BIT BSAVE THEN "TSAVE"; CALL SMOUTTEXT FI; CALL SMCRLF
156046                FOR CINDX DO WHILE CINDX<7
156052                   "CE2"=:"ERRUT"
156054   CE2:            TADR(CINDX); CALL SMOUTTEXT; CADR(CINDX)+CCLASS=:X=:CCADR
156064                   CALL GETVAL; "IOINP"; CALL SMAGPAR; VALUE
156070                   IF X:=CINDX=5 THEN A=:CELEM FI
156075                   X:=CINDX; CALL CHKRNG; GO SSMON; X:=CCADR; CALL SAVE
156102                OD; CALL DEFELEM; CALL FAR UPLTSL
156106                CALL SMCRLF; "TMSLI"; CALL SMGPAR; CALL SMYESNO
156112             WHILE A=1
156115                CCLASS+1; IF A>7 THEN A:=0 FI; A=:CCLASS
156124             OD
156125          FI
156125          IF LEGAREA BIT BIML THEN CALL WIMBACK FI
156131          IF LEGAREA BIT BSAVE THEN CALL WXSAVE FI
156135          GO SMLEAVE
156136   *)FILL
156171
156171
156171   % ==============================================
156171   %            D E F E L E M
156171   %   DEFINE ONE WHOLE TIMESLICE ELEMENT
156171
156171   DEFELEM: A:=L=:"C2LRG"
156173          "DEFT"=:"ERRUT"; CALL SMCRLF
156176   DEFT:  "TXELN"; CALL SMOUTTEXT; CELEM; CALL SM3OCTU
156202          "IOINP"; CALL SMAGPAR; CELEM; A=:CELEM
156206          DO
156206             CALL SMCRLF; 7=:CINDX; "LDAGAIN"=:"ERRUT"
```

```
156213              FOR CINDX DO WHILE CINDX<12
156217   LDAGAIN:      TADR(CINDX); CALL SMOUTTEXT; CADR(CINDX)+CELEM=:X=:CCADR
156227                 CALL GETVAL; "IOINP"; CALL SMAGPAR; VALUE
156233                 IF X:=CINDX=11 THEN A=:CELEM FI
156240                 X:=CINDX; CALL CHKRNG; GO LDAGAIN; X:=CCADR; CALL SAVE
156245              OD
156247              "TCHNE"; CALL SMGPAR; CALL SMYESNO
156252            WHILE A=1
156255               "TSELN"; CALL SMOUTTEXT; CELEM; CALL SM3OCTUT
156261            OD; GO C2LRG
156263   *)FILL
156304
156304
156304   % ==============================================
156304   %              G E T V A L
156304   %    GET OLD VALUE FROM IMAGE/SAVE-AREA
156304   %    VALUE IS SAVED IN LOCATION VALUE
156304   %
156304   GETVAL: T:=L=:"C1LRG"
156306           IF T:=LEGAREA BIT BIML THEN CALL R1IMAGE; A=:VALUE; CALL SM3OCTU FI
156314           IF T:=LEGAREA BIT BIML AND T BIT BSAVE THEN 2; CALL SMSPACE FI
156323           IF T:=LEGAREA BIT BSAVE THEN CALL R1SAVE; A=:VALUE; CALL SM3OCTU FI
156331           GO C1LRG
156332
156332   % ==============================================
156332   %              S A V E
156332   %    SAVE VALUE IN  SAVE/IMAGE ACORDING TO LEGAL AREAS
156332
156332   SAVE: T:=L=:"C1LRG"; A=:SVALUE
156335         IF T:=LEGAREA BIT BSAVE THEN CALL W1SAVE FI
156341         IF T:=LEGAR BIT BIML THEN A:=SVALUE; CALL W1IMAG FI
156346         GO C1LRG
156347
156347   % ==============================================
156347   %              C H E C K S I Z E
156347   %    CHECK IF INSIDE LEGAL RANGE
156347
156347   CHKRNG:     X:=CHADR(X)
156350               X GOSW CHPRIOR,CHELEM,CHCLASS,CHTUNIT,CHTIME,CHHASH,CHP70
156360   CHPRIOR:    IF A>>77 OR A<1 GO ERR; EXITA
156367   CHELEM:     IF A>>37 GO ERR; EXITA
156373   CHCLASS:    IF A>>7 GO ERR; EXITA
156377   CHTUNIT:    IF A>>400 OR A<1 GO ERR; EXITA
156406   CHTIME:     IF A>>400 OR A<1 GO ERR; EXITA
156415   CHHASH:     IF A<1 OR A>177 GO ERR; EXITA
156424   CHP70:      IF A<1 OR A>>70 GO ERR; EXITA
156433   ERR:        "TERR1"; CALL SMOUTTEXT; GO ERRUT
156436   *)FILL
156451
156451   INTEGER LTSYP:='$PRIORITY FOR OWNER OF SYSTEM RESOURCES'
156475   INTEGER LTSY1:='$          WHICH ARE WAITED FOR BY OTHER PROGRAMS:  '
156527   INTEGER LNBTU:='$NO. OF BASIC TIME UNITS IN ONE TIMESLICE UNIT:    '
156561   INTEGER LLTUB:='$LOWEST PRIORITY BEFORE GETTING RAISED ON BREAK:   '
156613   INTEGER LLHAS:='$LOWEST TIME COUNT BEFORE GETTING HASHED:          '
156645   INTEGER LTBIT:='$BIT MASK USED WHEN HASHING:                       '
156677   %INTEGER LTHEA:='$
156677   INTEGER ARRAY TLTSX:=(LTSYP,LTSY1,LNBTU,LLTUB,LLHAS,LTBIT)
156705   INTEGER ARRAY LCADR:=(0,TSLSYPRI,TSLTUNIT,TSLLOWLG,TSLHTIME,TSLHASHM)
156713
156713   CMLTSL: L=:D; CALL SMENTR
```

```
156715          6; CALL SELAREA; A=:LEGAREA
156720          IF A><0 THEN
156721             IF A BIT BSAVE THEN CALL 6SOPEN FI
156724             61; CALL SMSPACE
156726             IF LEGAREA BIT BIML THEN "TIMAG"; CALL SMOUTTEXT FI
156733             IF LEGAREA BIT BIML AND A BIT BSAVE THEN 2; CALL SMSPACE FI
156742             IF LEGAREA BIT BSAVE THEN "TSAVE"; CALL SMOUTTEXT FI
156747             0=:CINDX
156750             DO WHILE X:=CINDX><6
156754                TLTSX(X); CALL SMOUTTEXT
156756                IF X><0 THEN X:=LCADR(X); CALL FAR GETVAL FI
156761                MIN CINDX
156762             OD
156763          FI; GO SMLEAVE
156764   *)FILL
156777
156777   INTEGER CTX8:='$$TIMESLICE   PRIORITY OF   NO. OF TIMESLICE'
157026   INTEGER CTX9:='$ELEMENT NO.   ELEMENT     UNITS ON ELEMENT$'
157055   INTEGER PTSLCLASS:='IDTIMESLICE CLASS: '
157067   INTEGER ARRAY TLGAREA:=(XTIMI,2,0,0, XTSAV,4,0,0, -1)
157100   INTEGER XTIMI:='IMAGE',XTSAV:='SAVE-AREA'
157110   INTEGER TSELAREA:='N IMAGE OR SAVE-AREA: '
157124   INTEGER TXESC:='  (ESCAPE ELEMENT)'
157136   INTEGER TXBRK:='  (BREAK ELEMENT)$'
157150
157150   CPTSLCLASS: L=:D; CALL SMENTR
157152          "PTSLCLASS"; CALL SMGPAR; A=:CCLASS
157155          "TSELAREA"; CALL SMSGPAR; GO SSMON
157160          X:="TLGAREA"; CALL SMABLOOK
157162          IF A><0 GO SMLEAVE; T.S1=:LEGAREA
157167          IF A BIT BSAVE THEN CALL 6SOPEN FI
157172          0=:XBT1=:XBT2
157174          "CTX8"; CALL SMOUTTEXT; "CTX9"; CALL SMOUTTEXT
157200          3; CALL SMSPACE; "TSLESCELEM"+CCLASS=:X; CALL FAR GETVAL
157206          VALUE=:SVALUE
157210          11; CALL SMSPACE; "TSLPRITAB"+SVALUE=:X; CALL FAR GETVAL
157216          14; CALL SMSPACE; "TSLTIMTAB"+SVALUE=:X; CALL FAR GETVAL
157224          "TXESC"; CALL SMOUTTEXT; CALL SMCRLF
157227          3; CALL SMSPACE; "TSLBRKELEM"+CCLASS=:X; CALL FAR GETVAL
157235          VALUE=:SVALUE; CALL FAR XSTBT; 0/\0
157241          11; CALL SMSPACE; "TSLPRITAB"+SVALUE=:X; CALL FAR GETVAL
157247          14; CALL SMSPACE; "TSLTIMTAB"+SVALUE=:X; CALL FAR GETVAL
157255          "TXBRK"; CALL SMOUTTEXT
157257          0=:CINDX
157260          FOR CINDX DO WHILE CINDX<<40
157264             3; CALL SMSPACE
157266             "TSLNEXTAB"+SVALUE=:X; CALL FAR GETVAL
157272             VALUE=:SVALUE; CALL FAR XSTBT; GO SMLEAVE
157276             11; CALL SMSPACE; "TSLPRITAB"+SVALUE=:X; CALL FAR GETVAL
157304             14; CALL SMSPACE; "TSLTIMTAB"+SVALUE=:X; CALL FAR GETVAL
157312             CALL SMCRLF
157313          OD; GO SMLEAVE
157316
157316   RBUS
157346
157346
157346   %=============================================================================
157346   %      L I S T  -  T I M E  -  S L I C E D  -  P R O G R A M S
157346   %
157346   %
```

```
157346   SUBR LTSPR
157346
157346
157346   DISP -200
157346   INTEGER ARRAY POINTER PCADDR
157346   INTEGER CINDX,DYNAL,CDFELT,CDBPROG,CTSLCLASS,HEADCOUNT
157346   PSID
157346
157346   INTEGER THED:='$LOG. DEV.  BACKG. PROGR.  TIMESLICE CLASS$'
157374   INTEGER TDYNAL:='NO FIXED PROGR.'
157404   LTSPR: L=:D; CALL SMENTER
157406          0=: HEADCOUNT
157407          "BACKT"=:"PCADDR"; 0=:CINDX
157412          DO WHILE X:=PCADDR(CINDX)><-1
157417             X=:CDFELT; 0=:DYNAL
157421             IF HEADCOUNT=0 THEN
157423               CALL SMCRLF;"THED"; CALL SMOUTTEXT;24=:HEADCOUNT
157430             FI; HEADCOUNT -1=: HEADCOUNT
157433
157433             CALL SMDYNAL; GO NDYN
157435             CALL GBTINDX; GO NDYN
157437             A*5PRVTSIZE+APRVTTABLE=:X; T:=MBPRVTTABLE; *LDATX TXSVT
157444             A=:CTSLCLASS
157445             *LDATX TXSVB
157446             IF A BIT BPCFIXED THEN
157450                X:=ASBPRTAB
157451                DO WHILE X<<AEBPRTAB
157454                   T:=MBSPRTAB; *LDDTX TXBBP            % A=PROG;D=DATAF.
157456                   IF D=CDFELT THEN A=:CDBPROG; GO L1 FI
157463                   X+BPRTSIZE
157464                OD
157465             FI; 1=:DYNAL; GO L1
157470
157470   NDYN:     X:=CDFELT; T:="DBPROG"; CALL XGTDFADDR
157473             IF A=0 GO NXT; A=:CDBPROG
157475             X:=A; CALL GTSLPINDEX; GO NXT; TSLSTATUS(X)=:CTSLCLASS
157502   L1:       IF CTSLCLASS BIT 5NOSLICE GO NXT; A SHZ -7CUTY/\7=:CTSLCLASS
157510             CDFELT; CALL SMFFLOGDV; GO NXT; A=:CDFELT
157514             CALL SMCRLF; CDFELT; CALL SMDECUT; 4; CALL SMSPACE
157521             IF DYNAL=0 THEN
157523                "CDBPROG"+B; CALL OPSYS(INAMS); GO NXT; X:=D; CALL OPSYS(PRIRT)
157533                7;CALL SMSPACE
157535             ELSE
157536                " TDYNAL"; CALL SMOUTTEXT
157540             FI;  CTSLCLASS; CALL SMDECUT
157542   NXT:      MIN CINDX
157543          OD; GO SMLEAVE
157545   RBUS
157576
157576
157576
157576
157576   %=================================================================
157576   % 28.56       I T S L I S   X R T S L I S
157576   %
157576   % INSERT-IN-TIME-SLICE
157576   % REMOVE-FROM-TIME-SLICE
157576   %
157576
157576   SUBR ITSLIS,XRTSLIS
```

```
157576
157576   DISP -200
157576          INTEGER ROUFLG,CLOGU,CDFLT,LEGAREA,CPROG,CFPTSL,TSLCLASS,DYNAL
157576          INTEGER POINTER LREG,CRAREA,CWAREA
157576          INTEGER CSG=CPROG,CPRVADDR=CFPTSL
157576   PSID
157576
157576   INTEGER PWCSGB:=BFIELD+CSG
157577   INTEGER TMEMO:='MEMORY',TIMAG:='  IMAGE', TSAVE:='   SAVE'
157613   INTEGER TGRST:='TIMESLICE CLASS/'
157624   %INTEGER TGSPA:='
157624   INTEGER TILLOG:='$ILLEGAL LOGICAL UNIT NUMBER$'
157643   INTEGER TILPROG:='$NOT TIMESLICED PROGRAM$'
157660   INTEGER TILTCLASS:='$ILLEGAL TIMESLICE CLASS (LEGAL: 0-7)$'
157704   INTEGER IOINP:='IO/: '
157707   INTEGER ETX4:='$ERROR IN ACESSING PIT3 SEGMENT$'
157730   SYMBOL 0ANDMASK=377,1ANDMASK=176777
157730
157730
157730   UPDAREA: A=:"CRAREA"; T=:"CWAREA":=L=:"LREG"
157734          IF DYNAL><0 THEN
157736             IF "CRAREA"="W1IMAGE" THEN
157742                T:=5PT3S; X:=CPRVADDR
157744             ELSE
157745                CPRVADDR+"XSVTSLSTATUS-SVTSLSTATUS"=:X
157750                T:=5IPT3S
157751             FI; T=:CSG; CALL GET1L; GO FAR ERR4; GO UPDA1
157755          FI
157755   @LIB CXCPU
157755          "TYPRING"; X:=CDFLT+A; CALL CRAREA
157761          IF A BIT 5TERM THEN T:="ZDBPROG" ELSE T:="DBPROG" FI
157766   @ELIB
157766   @LIB CXCPU-,
157766          X:=CDFLT+T; CALL CRAREA; A=:CPROG
157772          X:="FPTSLICE"; CALL CRAREA; A=:CFPTSL; X+1; CALL CRAREA
157777          A=:D:=CFPTSL; X:=CPROG; CALL CGTTSLPROC; GO FAR ERR2
160004          "TSLSTATUS"; X+A; CALL CRAREA
160007   UPDA1: IF T:=ROUFLG=0 THEN
160012             A/\0ANDMASK; T:=TSLCLASS SH 7CUTY; A\/T; T SH 3; A\/T BZERO 5NOSLICE
160021          ELSE
160022             A BONE 5NOSLICE
160023          FI
160023          IF T:=DYNAL=0 THEN CALL CWAREA ELSE T:=CSG; CALL PUT1L; GO FAR ERR4 FI
160033          GO LREG
160034
160034   GTTSLCLASS: A=:"CRAREA":=L=:"LREG"
160037          IF DYNAL><0 THEN
160041             IF "CRAREA"="R1IMAGE" THEN
160045                X:=CPRVADDR; T:=5PT3S
160047             ELSE
160050                CPRVADDR+"XSVTSLSTATUS+SVTSLSTATUS"=:X; T:=5IPT3S
160054             FI; T=:CSG; CALL GET1L; GO FAR ERR4; GO GTTS1
160060          FI
160060   @LIB CXCPU
160060          "TYPRING"; X:=CDFLT+A; CALL CRAREA
160064          IF A BIT 5TERM THEN T:="ZDBPROG" ELSE T:="DBPROG" FI
160071   @ELIB
160071   @LIB CXCPU-,
160071          X:=CDFLT+T; CALL CRAREA; A=:CPROG
160075          X:="FPTSLICE"; CALL CRAREA; A=:CFPTSL; X+1; CALL CRAREA
```

```
160102          A=:D:=CFPTSL; X:=CPROG; CALL CGTTSLPROC; GO FAR ERR2
160107          "TSLSTATUS"; X+A; CALL CRAREA
160112   GTTS1: A SHZ -7CUTY/\7=:TSLCLASS
160115          CALL SMOCTU
160116          GO LREG
160117
160117   CGTTSLPROC: IF X>>=A AND X<<D THEN
160123                  X-A=:D; A:=0; T:=5RTSIZE; *RDIV ST
160130                  A=:X; EXITA
160132               FI; EXIT
160133   *)FILL
160155   ITSLIS: A:=0; GO L1
160157   XRTSLIS: A:=1
160160   L1:     L=:D; CALL SMENTER; A=:ROUFLG; 0=:DYNAL
160164          "6PLOGU"; CALL SMSGPAR; GO SSMON; A=:CLOGU
160170          CALL LOGPH; IF A=0 GO FAR ERR1; A=:CDFLT
160174          X:=A; CALL SMDYNALLOC; GO NDYN; A+1=:DYNAL
160201          A-1*5PRVT+"PRVTTABLE"+"SVTSLSTATUS"=:CPRVADDR
160206   NDYN:  7; CALL SELAREA; IF A=0 GO SMLEAVE; A=:LEGAREA
160213          IF A BIT BSAVE THEN CALL 6SOPEN FI
160216          IF ROUFLG=0 THEN
160220             21; CALL SMSPACE
160222             IF LEGAREA BIT BMEMO THEN "TMEMO"; CALL SMOUTTEXT FI
160227             IF LEGAREA BIT BIML THEN "TIMAG"; CALL SMOUTTEXT FI
160234             IF LEGAREA BIT BSAVE THEN "TSAVE"; CALL SMOUTTEXT FI
160241             CALL SMCRLF; "TGRST"; CALL SMOUTTEXT
160244             IF LEGAREA BIT BMEMO THEN
160247                IF DYNAL=0 THEN
160251                   X:=CDFLT; T:="DBPROG"; CALL XGTDFADDR
160254                   X:=A; AD:=DFPTSLICE; CALL CGTTSLPROC; GO FAR ERR2
160260                   A:=TSLSTATUS(X)
160261                ELSE
160262                   T:=5PT3S; X:=CPRVADDR; CALL GET1L; GO FAR ERR4
160266                FI; A SHZ -7CUTY/\7=:TSLCLASS; CALL SMOCTU
160272             FI
160272             IF LEGAREA BIT BIML THEN "R1IMAGE"; CALL FAR GTTSLCLASS FI
160277             IF LEGAREA BIT BSAVE THEN "R1SAVE"; CALL FAR GTTSLCLASS FI
160304             "IOINP"; CALL SMAGPAR; TSLCLASS; A=:TSLCLASS
160310             IF A>>7 GO FAR ERR3; GO LABL1; *)FILL
160360          FI
160360   LABL1: IF LEGAREA BIT BMEMO THEN
160363             IF DYNAL=0 THEN
160365                X:=CDFLT; T:="DBPROG"; CALL XGTDFADDR
160370                X:=A; AD:=DFPTSLICE; CALL CGTTSLPROC; GO ERR2
160374                IF ROUFLG=0 THEN
160376                   T:=TSLCLASS SH 7CUTY=:D SH 3 \/ D BONE 5ESCF BZERO 5NOSLICE
160405                   A:=0ANDMASK
160406                ELSE
160407                   T:=1000; A:=1ANDMASK
160411                FI; CALL TSLANDOR
160412             ELSE
160413                IF ROUFLG=0 THEN
160415                   A:=TSLCLASS SH 7CUTY=:D SH 3 \/D BONE 5ESCF BZERO 5NOSLICE
160424                ELSE
160425                   A:=TSLCLASS/\1ANDMASK BONE 5NOSLICE
160430                FI; X:=CPRVADDR; T:=5PT3S; CALL PUT1L; GO ERR4
160434             FI
160434          FI; GO L2; *)FILL
160446   L2:    IF LEGAREA BIT BIML THEN
160451             A:="R1IMAGE"; T:="W1IMAGE"; CALL FAR UPDAREA
```

```
160454          FI
160454          IF LEGAREA BIT BSAVE THEN
160457             A:="R1SAVE"; T:="W1SAVE"; CALL FAR UPDAREA
160462          FI
160462          IF DYNAL=0 THEN
160464             IF LEGAREA BIT BIML THEN CALL WIMBACK FI
160470             IF LEGAREA BIT BSAVE THEN CALL WXSAVE FI
160474          ELSE
160475             5PT3S=:CSG; "PWCSGB"; *MON 2WSEG
160501             5IPT3S=:CSG; "PWCSGB"; *MON 2WSEG
160505          FI; GO SMLEAVE
160506   ERR1:  "TILLOGU"; CALL SMOUTTEXT; GO SMLEAVE
160511   ERR2:  "TILPROG"; CALL SMOUTTEXT; GO SMLEAVE
160514   ERR3:  "TILTCLASS"; CALL SMOUTTEXT; GO SMLEAVE
160517   ERR4:  "ETX4"; CALL SMOUTTEXT; GO SMLEAVE
160522   RBUS
160540
160540   %=========================================================================
160540   %              L O G D I S C
160540   %
160540   SUBR LOGDISC
160540
160540   INTEGER TFU3:='DEFINE-DISC-ACCESS-LOG'
160554   INTEGER TFU4:='START-DISC-ACCESS-LOG'
160567   INTEGER TFU5:='STOP-DISC-ACCESS-LOG'
160602   INTEGER TFU6:='START-DISC-ACCESS-COUNTER'
160617   INTEGER TFU7:='STOP-DISC-ACCESS-COUNTER'
160634   INTEGER TFU10:='CLEAR-DISC-ACCESS-COUNTER'
160651   INTEGER TFU11:='DISC-ACCESS-COUNTER'
160663   INTEGER TFU12:='DISC-DRIVER-ERROR-INFORMATION'
160702   INTEGER TFU13:='DISC-ERROR-STATUS'
160713   INTEGER TFCDL:='LOG-DISC-ACCESS-COUNTER'
160727   INTEGER TFEXI:='EXIT'
160732   INTEGER TFHLP:='HELP'
160735
160735   @ICR;
160735   INTEGER ARRAY FUNTAB:=(
160735           TFU3,3,FU3,0,          TFU4,4,FU4,0,
160745           TFU5,5,FU5,0,          TFU6,6,FU6,0,
160755           TFU7,7,FU7,0,          TFU10,10,FU10,0,
160765           TFU11,11,FU11,0,       TFU12,12,FU12,0,
160775           TFU13,13,FU13,0,       TFEXI,0,FUEXI,0,
161005           TFHLP,0,FUHLP,0,       TFCDL,11,FUCDL,0,
161015           -1);
161016   @CR;
161016   INTEGER TNIMPL:='$DISC ACCESS LOG NOT IMPLEMENTED'
161037   INTEGER TXELOG:='$ILLEGAL LOGICAL DEVICE NUMBER'
161057   INTEGER TXEUNIT:='$ILLEGAL DRIVE NUMBER'
161072   INTEGER TXEDADDR:='$ILLEGAL DISC ADDRESS'
161105   INTEGER TXEYN:='$ONLY YES OR NO IS LEGAL ANSWER'
161125   INTEGER TXEABS:='$ERROR FROM MON ABSTR ON DISC ACCESS LOG'
161152   INTEGER PDILFILE:='S DISC ACCESS LOG FILE: '
161167   INTEGER TFNCONT:='$FILE IS NOT CONTIGUOUS'
161203   INTEGER TEWRDISC:='$DISC ACCESS LOG FILE NOT LEGAL ON THIS DISC TYPE'
161234   INTEGER TQX:='N SMALL OR BIG RECORD SIZE ON DISC LOG FILE (DEFAULT IS BIG)? '
161274   INTEGER TQ1:='N LOG ALL DISC ACCESSES (DEFAULT IS YES)? '
161322   INTEGER TQ2:='N LOG DISC ACCESSES TO ONE CONTROLLER ONLY (DEFAULT IS NO)? '
161361   INTEGER TQ3:='N LOG DISC ACCESSES TO ONE DRIVE ONLY (DEFAULT IS NO)? '
161415   INTEGER TQ4:='N LOG ONLY WRITE ACCESSES (DEFAULT IS NO)? '
161443   INTEGER TQ5:='N LOG ONLY READ ACCESSES (DEFAULT IS NO)? '
```

```
161471   INTEGER TQ6:='N LOG ONLY ACCESSES TO A LIMITED PART OF THE DISC (DEFAULT IS NO)? '
161533   INTEGER TXPLOG:='IOLOGICAL DEVICE NUMBER OF DISC TO LOG (OCTAL): '
161564   INTEGER TXPUN:='IODRIVE NUMBER OF DRIVE TO LOG: '
161605   INTEGER TXLG1ADD:='DOFIRST DISC ADDRESS IN THE DISC PART TO LOG (OCTAL ONLY): '
161643   INTEGER TXLG2ADD:='DOLAST DISC ADDRESS IN THE DISC PART TO LOG (OCTAL ONLY): '
161701   INTEGER TXNDACC:='$TOTAL NUMBER OF DISC ACCESSES: '
161722   INTEGER TXNWDACC:='$NUMBER OF DISC WRITE ACCESSES: '
161743   INTEGER TXNRDACC:='$NUMBER OF DISC READ ACCESSES.: '
161764   INTEGER ARRAY SMBIG:=(TXBIG,0,0,0, TXSMA,1,0,0, -1)
161775   INTEGER TXBIG:='BIG',TXSMALL:='SMALL'
162002
162002   DISP -200
162002           INTEGER CFUNC,CFILNO,CLOGU,CUNIT,NEGFLG,CLUNIT
162002           DOUBLE CFSIZE,CFISTART
162002           DOUBLE DFLGADDR=CFSIZE,DLLGADDR=CFISTART
162002           DOUBLE CNDACCESS=CFSIZE,CNWDACC=CFISTART
162002           INTEGER CDILGFLAG=NEGFLG
162002   PSID
162002   DISP 0
162002           DOUBLE ZFSTART
162002           INTEGER ZFNBLCK
162002           DOUBLE ZFEND
162002           INTEGER ZFLOGU,ZFUNIT,ZDILGFLAG,ZLLOGU,ZLUNIT
162002           DOUBLE ZFLGADDR,ZLLGADDR
162002   PSID
162002   DISP 0; INTEGER TFFTX,TFFVAL,TFFADDR; PSID
162002
162002   INTEGER ARRAY CDILBU(20)
162022   INTEGER ARRAY PPRSV:=("2201","0","0")
162025
162025   CTRYAGAIN:CALL SMCCLEAR;GO LOOP
162027   LOGDISC: L=:D; CALL SMENTER
162031           IF "DFDIL"=0 THEN "TNIMPL"; CALL SMOUTTEXT; GO SMLEAVE FI
162036   LOOP:   "6CFUPAR"; CALL SMSGPAR;GO CTRYAGAIN; X:="FUNTAB"; CALL SMABLOOK
162043           IF A><0 THEN
162044              IF A=-1 THEN "SMTXFILL" ELSE "SMFAMBIG" FI
162052              CALL SMOUTTEXT; GO CTRYAGAIN
162054           FI; T.TFFVAL=:CFUNC; X.TFFADDR=:P          % GO TO FUNCTION  ROUTINE
162061
162061   FU4:
162061   FU5:
162061   FU7:
162061   FU10:   "PPRSV"; *MON 2RESR
162063   FUABS:  CFUNC=:XDILF; T:=2200; A:="XP131"; *MON 2ABST
162070   ERABS:  IF A<0 THEN "TXEABS"; CALL SMOUTTEXT; FI
162073           "PPRSV"; *MON 2RELE
162075           GO CTRYAGAIN
162076
162076   FU11:   "PPRSV"; *MON 2RESR
162100           CFUNC=:XDILF; T:=2200; A:="XP131"; *MON 2ABST
162105           IF A<0 GO ERABS
162106           "DILBU".DS0=:CNDACC; X.DS1=:CNWDACC
162113           "PPRSV"; *MON 2RELE
162115           "TXNDACC"; CALL SMOUTTEXT; CNDACC; CALL SMDDECUT
162121           "TXNWDACC"; CALL SMOUTTEXT; CNWDACC; CALL SMDDECUT
162125           "TXNRDACC"; CALL SMOUTTEXT
162127           CNWDACC; A=:T; D=:L; CNDACC; D-L; A:=A+C-1-T; CALL SMDDECUT
162140           CALL SMCRLF
162141           GO CTRYAGAIN
162142
```

```
162142   FUHLP: X:="FUNTAB"
162143          DO
162143             CALL SMCRLF
162144          WHILE X.TFFTX><-1; CALL SMOUTTEXT; X+4
162152          OD; CALL SMCRLF; GO CTRYAGAIN
162155
162155   FUEXI: GO SMLEAVE
162156   *)FILL
162207
162207   @LIB HKD
162207   FU3:   0=:CFILNO
162210          "PDILFILE"; CALL SMSGPAR; GO CTRYAGAIN
162213          X:=A; CALL SMGFINFO; GO FAR ERET; 9CFNO=:CFILNO
162220   %%     AD:=CFISTART SH 1; IF T:=NEGFLG><0 THEN D BONE 17 FI
162220          IF 9CABLPAGE><2 THEN "TEWRDISK"; CALL SMOUTTEXT; GO FAR ERET FI
162227          AD:=9CFSTART=:"CDILBU".ZFSTART; A=:T; D=:L
162234          AD:=9CFSIZE; AD SHZ 1; D-1; A:=A+C-1; D+L; A:=A+C+T; AD=:X.ZFEND
162245          2=:X.ZFNBLCK; 9CLOGU=:X.ZFLOGU; 9CUNIT=:X.ZFUNIT
162253          GO L1; *)FILL
162271   L1:    0=:CDILGFLAG
162272          "TQX"; CALL SAGPAR; GO FAR CTRYAGAIN; GO LABL1; X:="SMBIG"; CALL SMABLOOK
162300          IF A><0 THEN
162301             IF A=-1 THEN "SMCILLP" ELSE "SMCAMBIGP" FI; CALL SMOUTTEXT; GO FAR CTRYAGAIN
162311          FI; IF T.S1><0 THEN CDILGFLAG BONE DILSMALL=:CDILGFLAG FI
162317   LABL1: "TQ1"; CALL SAGPAR; GO FAR CTRYAGAIN; GO FAR DODEF; CALL SMYESNO; IF A=1 GO FAR DODEF
162327          IF A<0 GO FAR ERET1; "TQ2"; CALL SAGPAR; GO FAR CTRYAGAIN; GO L2; CALL SMYESNO
162336          IF A<0 GO FAR ERET1
162340          IF A=1 THEN
162343             "TXPLOG"; CALL SMSGPAR; GO FAR CTRYAGAIN; A=:CLOGU; CALL LOGPH
162350             IF A=0 THEN "TXELOG"; CALL SMOUTTEXT; GO FAR ERET FI
162354             CDILGFLAG BONE DIL1CONTROLLER=:CDILGFLAG
162357   L2:    FI; "TQ3"; CALL SAGPAR; GO FAR CTRYAGAIN; GO L3; CALL SMYESNO; IF A<0 GO FAR ERET1
162366          IF A=1 THEN
162371             "TXPUN"; CALL SMSGPAR; GO FAR CTRYAGAIN; A=:CUNIT
162375             IF A>>7 THEN "TXEUNIT"; CALL SMOUTTEXT; GO FAR ERET FI
162403             CDILGFLAG BONE DIL1UNIT=:CDILGFLAG
162406   L3:    FI; "TQ4"; CALL SAGPAR; GO FAR CTRYAGAIN; GO L4; CALL SMYESNO; IF A<0 GO FAR ERET1
162415          IF A=1 THEN CDILGFLAG BONE DILWACCESSES=:CDILGFLAG FI
162423   L4:    "TQ5"; CALL SAGPAR; GO FAR CTRYAGAIN; GO L5; CALL SMYESNO; IF A<0 GO ERET1
162431          IF A=1 THEN CDILGFLAG BONE DILRACCESSES=:CDILGFLAG FI
162437          GO L5;*)FILL
162467   L5:    "TQ6"; CALL SAGPAR; GO FAR CTRYAGAIN; GO DODEF; CALL SMYESNO; IF A<0 GO ERET1
162475          IF A=1 THEN
162500             "TXLG1ADDR"; CALL SMSGPAR; GO FAR CTRYAGAIN; AD=:DFLGADDR
162504             "TXLG2ADDR"; CALL SMSGPAR; GO FAR CTRYAGAIN; AD=:DLLGADDR
162510             A=:L; D=:T
162512             AD:=DFLGADDR; IF L<<A GO ILDADDR
162515             IF L=A AND T<<=D THEN
162521   ILDADDR:     "TXEDADDR"; CALL SMOUTTEXT; GO ERET; *)FILL
162535             FI; CDILGFLAG BONE DILLIMIT=:CDILGFLAG
162540          FI
162540   DODEF: CDILGFLAG=:"CDILBU".ZDILGFLAG
162543          CLOGU=:X.ZLLOGU; CUNIT=:X.ZLUNIT
162547          DFLGADDR=:X.ZFLGADDR; DLLGADDR=:X.ZLLGADDR
162553          "PPRSV"; *MON 2RESR
162555          CFUNC=:XDILF; X:="DILBU"; A:="CDILBU"; T:=34=:D; *MOVB; JMP *
162565          T:=2200; A:="XP131"; *MON 2ABST
162570          IF A<0 THEN "TXEABST"; CALL SMOUTTEXT FI
162573   ERET:  "PPRSV"; *MON 2RELE
162575          T:=CFILNO; *MON 2CLOS; JMP *+1
```

```
162600          GO FAR CTRYAGAIN
162601   CFERR: *MON 64
162602          GO ERET
162603   ERET:: "TXEYN"; CALL SMOUTTEXT; GO ERET
162606   *)FILL
162620
162620   INTEGER TQC1:='N COUNT ALL DISC ACCESSES (DEFAULT IS YES)? '
162647   INTEGER TQC2:='N COUNT DISC ACCESSES TO ONE CONTROLLER ONLY (DEFAULT IS NO)? '
162707   INTEGER TQC3:='N COUNT DISC ACCESSES TO ONE DISC UNIT NUMBER ONLY (DEFAULT IS NO)? '
162752
162752   FU6:    0=:CDILGFLAG
162753           "TQC1"; CALL SAGPAR; GO FAR CTRYAGAIN; GO FU6F; CALL SMYESNO
162760           IF A<0 GO FAR ERET
162762           IF A=1 GO FU6F
162765           "TQC2"; CALL SAGPAR; GO FAR CTRYAGAIN; GO FU6A; CALL SMYESNO
162772           IF A<0 GO FAR ERET1
162774           IF A=1 THEN
162777              "TXPLOG"; CALL SMSGPAR;GO FAR CTRYAGAIN; A=:CLOGU; CALL LOGPH
163004              IF A=0 THEN "TXELOG"; CALL SMOUTTEXT; GO FAR CTRYAGAIN FI
163010              CDILGFLAG BONE DAC1CONTROLLER=:CDILGFLAG
163013   FU6A:   FI; "TQC3"; CALL SAGPAR; GO FAR CTRYAGAIN; GO FU6F; CALL SMYESNO
163020           IF A<0 GO FAR ERET1
163022           IF A=1 THEN
163025              "TXPUN"; CALL SMSGPAR; GO FAR CTRYAGAIN; A=:CUNIT
163031              IF A>>3 THEN "TXEUNIT"; CALL SMOUTTEXT; GO FAR CTRYAGAIN FI
163037              CDILGFLAG BONE DAC1UNIT=:CDILGFLAG
163042           FI
163042   FU6F:   "PPRSV"; *MON 2RESR
163044           CDILGFLAG=:"DILBU".S0; CLOGU=:X.S1; CUNIT=:X.S2
163053           GO FAR FUABS
163054   *)FILL
163076
163076   INTEGER TXF12:='$FOR EXPLANATION SEE THE LISTING OF THE DISC DRIVER (BDISK).$'
163135   INTEGER ARRAY F12TX:=(F12TA,F12TB,F12TC,F12TD,F12TE,F12TF,F12TG,F12TH,F12TI,F12TJ,F12TK)
163150   INTEGER F12TA:='DRIAR: ',F12TB:='QQQST: ',F12TC:='RSCON: ',F12TD:='TREGI: '
163170   INTEGER F12TE:='CONTR: ',F12TF:='DEVNO: ',F12TG:='BANKN: ',F12TH:='ADRES: '
163210   INTEGER F12TI:='QQQCY: ',F12TJ:='SRSFC: ',F12TK:='WORDC: '
163224
163224   FU12:   "TPPLOG"; CALL SMSGPAR; GO FAR CTRYAGAIN; A=:CLOGU
163230           CALL LOGPH; IF A=0 THEN "TXELOG"; CALL SMOUTTEXT; GO FAR CTRYAGAIN FI
163235           "PPRSV"; *MON 2RESR
163237           CLOGU=:"DILBU".S0
163242           CFUNC=:XDILF; T:=2200; A:="XP131"; *MON 2ABST
163247           IF A<0 GO FAR ERABS
163251           X:="CDILBU"; A:="DILBU"; T:=26=:D; *MOVB; JMP *
163257           "PPRSV"; *MON 2RELE
163261           "TXF12";CALL SMOUTTEXT
163263           X:=0
163264           DO WHILE X<13
163267              CALL SMCRLF; F12TX(X); CALL SMOUTTEXT; CDILBU(X); CALL SMOCTU
163274              X+1
163275           OD; GO FAR CTRYAGAIN
163277   *)FILL
163320   INTEGER THED1:='$   TOTAL DISC ACCESSES        WRITE ACCESSES            READ ACCESSES'
163363   INTEGER THED2:='$  IN INTV. /ACCUMULATED    IN INTV. /ACCUMULATED     IN INTV. /ACCUMULATED'
163431   INTEGER PTINTV:='IDINTERVAL IN SECONDS (DEFAULT IS 60 SECS): '
163460   INTEGER TSLSH:=' / '
163462
163462   DISP -177
163462           INTEGER CINTV,ADA1,ADA2,AWA1,AWA2,ARA1,ARA2,HEADCOUNT
```

```
163462          INTEGER CACC1,CACC2
163462          DOUBLE DADA=ADA1,DAWA=AWA1,DARA=ARA1,DCACC=CACC1
163462   PSID
163462
163462   INTEGER ARRAY PARHOLD:=(CINTV+BFIELD,"2")
163464
163464   CSUBR: D-T; A:=A+C-1-X; AD=:DCACC; EXIT
163472
163472   FUCDL: "PTINTV"; CALL SAGPAR; GO FAR CTRYAGAIN; A:=74; A=:CINTV
163477          0=:HEADCOUNT
163500          A:=0=:D; AD=:DADA=:DAWA=:DARA
163505          CALL ESCON
163506          DO
163506             "PARHOLD"; *MON 2HOLD
163510             "PPRSV"; *MON 2RESR
163512             CFUNC=:XDILF; T:=2200; A:="XP131"; *MON 2ABST
163517             IF A<0 THEN
163520                "PPRSV"; *MON 2RELE
163522                "TXEABS"; CALL SMOUTTEXT; GO SMLEAVE
163525             FI; X:="CDILBU"; A:="DILBU"; T:=14=:D; *MOVB; JMP *
163533             "PPRSV"; *MON 2RELE
163535             IF HEADCOUNT=0 THEN
163537                "THED1"; CALL SMOUTTEXT; "THED2"; CALL SMOUTTEXT; 17=:HEADCOUNT
163545             FI; HEADCOUNT-1=:HEADCOUNT
163550             CALL SMCRLF; "CDILBU".DS0; T:=ADA2; X:=ADA1; CALL CSUBR
163556             CALL SMDDECUT; "TSLSH"; CALL SMOUTTEXT
163561             DCACC; A:=:D+ADA2; D:=D+C; A:=:D+ADA1; AD=:DADA; CALL SMDDECUT
163571             20040; CALL SM2TCO; "CDILBU".DS1; T:=AWA2; X:=AWA1; CALL CSUBR
163600             CALL SMDDECUT; "TSLSH"; CALL SMOUTTEXT
163603             DCACC; A:=:D+AWA2; D:=D+C; A:=:D+AWA1; AD=:DAWA; CALL SMDDECUT
163613             20040; CALL SM2TCO; AD:="CDILBU".DS0; T:=X.S3; X:=X.S2; CALL CSUBR
163622             AD:=DCACC; T:=ARA2; X:=ARA1; CALL CSUBR; CALL SMDDECUT
163627             "TSLSH"; CALL SMOUTTEXT
163631             DCACC; A:=:D+ARA2; D:=D+C; A:=:D+ARA1; AD=:DARA; CALL SMDDECUT
163641          OD; *JMP *; )FILL
163671
163671
163671   @ICR;
163671   INTEGER ARRAY F13TX:=(F13TA,F13TB,F13TC,F13TD,F13TE,F13TF,F13TG,F13TH,
163701                         F13TI,F13TJ,F13TK,F13TL);
163705   INTEGER ARRAY F13TY:=(CTXRE,CTXWR,CTXRE,CTXWR,CTXRE,CTXWR,CTXRE,CTXWR,
163715                         CTXNE,CTXNR,CTXXR,CTXTR);
163721   INTEGER ARRAY F13PT:=(-14,-13,-3,-2,-12,-11,-4,-3,-7,0,-13,-13);
163735   @CR;
163735   INTEGER ARRAY OCTDEC:=(0,0,1,1,0,0,0,0,0,0,1,1)
163751   %INTEGER F13TA:='NUMBER OF ERRORS IN READ OPERATIONS............:  '
163751   %INTEGER F13TB:='NUMBER OF ERRORS IN WRITE OPERATIONS............:  '
163751   %INTEGER F13TC:='INCLUSIVE OR OF ERRORBITS IN READ OPERATIONS...:  '
163751   %INTEGER F13TD:='INCLUSIVE OR OF ERRORBITS IN WRITE OPERATIONS..:  '
163751   %INTEGER F13TE:='NUMBER OF TIMEOUTS IN READ OPERATIONS...........:  '
163751   %INTEGER F13TF:='NUMBER OF TIMEOUTS IN WRITE OPERATIONS..........:  '
163751   %INTEGER F13TG:='NUMBER OF DRIVER RETRIES IN READ OPERATIONS....:  '
163751   %INTEGER F13TH:='NUMBER OF DRIVER RETRIES IN WRITE OPERATIONS...:  '
163751   %INTEGER F13TI:='NUMBER OF ERRORCORRECTIONS IN THE DRIVER.......:  '
163751   %INTEGER F13TJ:='NUMBER OF TIMES REALLOCATED TRACKS ARE ACCESSED:  '
163751   %INTEGER F13TK:='INCLUSIVE OR OF X-REGISTERS ON ERRORS...........:  '
163751   %INTEGER F13TL:='INCLUSIVE OR OF T-REGISTERS ON ERRORS...........:  '
163751   INTEGER ARRAY F13TA(0)
163751   INTEGER ARRAY F13TB:='NUMBER OF ERRORS'
163762   INTEGER ARRAY F13TC(0)
```

```
163762   INTEGER ARRAY F13TD:='INCLUSIVE OR OF ERRORBITS'
163777   INTEGER ARRAY F13TE(0)
163777   INTEGER ARRAY F13TF:='NUMBER OF TIMEOUTS'
164011   INTEGER ARRAY F13TG(0)
164011   INTEGER ARRAY F13TH:='NUMBER OF DRIVER RETRIES'
164026   INTEGER ARRAY F13TI(0)
164026   INTEGER ARRAY F13TJ:='NUMBER OF '
164034   INTEGER ARRAY F13TK(0)
164034   INTEGER ARRAY F13TL:='INCLUSIVE OR OF '
164045   INTEGER ARRAY CTXRE:=' IN READ OPERATIONS'
164057   INTEGER ARRAY CTXWR:=' IN WRITE OPERATIONS'
164072   INTEGER ARRAY CTXNE:='ERRORCORRECTIONS IN THE DRIVER'
164112   INTEGER ARRAY CTXNR:='TIMES REALLOCATED TRACKS ARE ACCESSED'
164135   INTEGER ARRAY CTXXR:='X-REGISTER ON ERRORS'
164150   INTEGER ARRAY CTXTR:='T-REGISTER ON ERRORS'
164163   INTEGER TPPLOG:='IOLOGICAL DEVICE NUMBER: '
164200   INTEGER TPPUN:='IOUNIT NUMBER: '
164210
164210   DISP -177; INTEGER CCOUNT; PSID
164210
164210   FU13:  "TPPLOG"; CALL SMSGPAR; GO FAR CTRYAGAIN; A=:CLOGU
164214          CALL LOGPH; IF A=0 THEN "TXELOG"; CALL SMOUTTEXT; GO FAR CTRYAGAIN FI
164221          "TPPUN"; CALL SMSGPAR; GO FAR CTRYAGAIN; A=:CUNIT
164225          IF A>>3 THEN "TXEUNIT"; CALL SMOUTTEXT; GO FAR CTRYAGAIN FI
164233          "PPRSV"; *MON 2RESR
164235          CLOGU=:"DILBU".S0; CUNIT=:X.S1
164242          CFUNC=:XDILF; T:=2200; A:="XP131"; *MON 2ABST
164247          IF A<0 GO FAR ERABS
164251          X:="CDILBU"; A:="DILBU"; T:=30=:D; *MOVB; JMP *
164257          "PPRSV"; *MON 2RELE
164261          X:=0
164262          DO WHILE X<14
164265             CALL SMCRLF; F13TX(X); CALL SMOUTTEXT; F13TY(X); CALL SMOUTTEXT
164272             IF F13PT(X)><0 THEN
164274                A=:CCOUNT; FOR CCOUNT DO; ##.; CALL SMTCO; OD
164301             FI
164301             IF OCTDEC(X)=0 THEN CDILBU(X); CALL SMDECUT ELSE CDILBU(X); CALL SMOCTUT FI
164310             X+1
164311          OD; GO FAR CTRYAGAIN
164313
164313   RBUS
164342
164342
164342   %=========================================================================
164342   %              M O N C A L L   L O G    ( C M M C L G )
164342   %
164342   SUBR CMMCLG
164342
164342   @ICR
164342   INTEGER ARRAY FUNTAB:=(
164342          TFU0,0,FU0,0,          TFU1,0,FU1,0,
164352          TFU2,0,FU2,0,          TFU3,0,FU3,0,
164362          TFU4,0,FU4,0,          TFU5,0,FU5,0,
164372          -1);
164373   @CR;
164373   INTEGER TFU0:='START-MONCALL-LOG'
164404   INTEGER TFU1:='STOP-MONCALL-LOG'
164415   INTEGER TFU2:='RESTART-MONCALL-LOG'
164427   INTEGER TFU3:='PRINT-MONCALL-LOG'
164440   INTEGER TFU4:='EXIT'
```

```
164443   INTEGER TFU5:='HELP'
164446   INTEGER TNIMPL:='NOT IMPLEMENTED$'
164457   INTEGER FUOPAR:='N LOG MONCALLS FOR ONLY ONE PROGRAM (DEFAULT IS YES)? '
164513   INTEGER TXYN:='$ONLY YES OR NO IS LEGAL ANSWER'
164533   INTEGER TNSTART:='NOT STARTED'
164541   INTEGER TNSTOPED:='NOT STOPPED'
164547   INTEGER TNNSTART:='NEVER STARTED'
164556   INTEGER TXILOGU:='$ILLEGAL LOGICAL UNIT OR NO BACKGROUND PROCESS CONNECTED'
164613   INTEGER THEAD:='$ MONCALL NUMBER  NUMBER OF TIMES USED$'
164637   INTEGER XXTXT:=';      '
164642   INTEGER TMCRUN:='USED BY ANOTHER TERMINAL.'
164657   INTEGER TMCCONT:='N $DO YOU WANT TO CONTINUE (DEFAULT IS NO)? '
164706   INTEGER TXMFELL:='$MONCALL LOG '
164715
164715   DISP -200
164715          INTEGER CLOGU,CBPROG,CMNO
164715          INTEGER POINTER LREG
164715   PSID
164715
164715   SMCCONT: A:=L=:"LREG"
164717          "TXMFELL"; CALL SMOUTTEXT
164721          "TMCRUN"; CALL SMOUTTEXT; "TMCCONT"; CALL SAGPAR; GO LREG; GO LREG
164727          CALL SMYESNO; IF A<0 GO ERET1
164731          IF A><1 GO LREG
164734          MIN "LREG"; GO LREG
164736
164736   CTRYAGAIN: CALL SMCCLEAR;GO LOOP
164740   ERET1: "TXYN"; CALL SMOUTTEXT; GO CTRYAGAIN
164743   EILOGU: "TXILOGU"; CALL SMOUTTEXT; GO FAR CTRYAGAIN
164746
164746   CMMCLG: L=:D; CALL SMENTER
164750          IF "MCLGFLG"=0 THEN
164752             "TXMFELL"; CALL SMOUTTEXT; "TNIMPL"; CALL SMOUTTEXT; GO SMLEAVE
164757          FI; CALL ESCON
164760
164760   LOOP:  "6CFUPAR"; CALL SMSGPAR;GO CTRYAGAIN; X:="FUNTAB"; CALL SMABLOOK
164765          IF A><0 THEN
164766             IF A=-1 THEN "SMTXFILL" ELSE "SMFAMBIG" FI
164774             CALL SMOUTTEXT; GO CTRYAGAIN
164776          FI; T.S2=:P
165001
165001   FUO:   "FUOPAR"; CALL SAGPAR; GO CTRYAGAIN; GO L1; CALL SMYESNO % START MONCALL LOG
165006          IF A<0 GO ERET1
165007          IF A=1 THEN                                          % LOG FOR ONLY ONE PROCESS
165012   L1:       "6PRTNAME" CALL SAGPAR; GO CTRYAGAIN; A:=RTREF; A=:CBPROG
165017             A-RTSTART=:D:=0; T:=5RTSIZE; *RDIV ST
165024             IF D><0 THEN "SMCILLP"; CALL SMOUTTEXT; GO CTRYAGAIN FI
165031          ELSE                                                 % LOG ALL PROGS
165032             -1=:CBPROG
165034          FI; T:=0; X:="MCLGFLG"; *LDATX
165037          IF A><0 THEN
165040             X:="MCLGOWNER"; *LDATX
165042             IF A><TTNO THEN CALL SMCCONT; GO CTRYAGAIN,FI
165047          FI; T:=0; X:="MCLGFLG"; *STZTX                       % STOP MONCALL LOG
165052          X:="MCLGBANK"; *LDATX
165054          A=:D; X:="TNMCALL"; *LDXTX
165057          T:=D; A:=X+1000                                      % CLEAR MONCALL LOG BUFFER
165062          DO
165062             *STZTX; AAX 1
165064          WHILE X><A
```

```
165066          OD; CBPROG; T:=0; X:="CMCLG"; *STATX                    % SET LOG CONDITION
165073          A:=TTNO; X:="MCLGFLG"; *STATX                           % START LOG
165076          X:="MCLGOWNER"; *STATX
165100          GO CTRYAGAIN
165101   *)FILL
165140
165140   FU1:   X:="MCLGFLG"; T:=0; *LDATX                              % STOP-MONCALL LOG
165143          IF A=0 THEN
165144             "TXMFELL"; CALL SMOUTTEXT; "TNSTART"; CALL SMOUTTEXT; GO FAR CTRYAGAIN
165151          FI; X:="MCLGOWNER"; T:=0; *LDATX
165154          IF A><TTNO THEN CALL FAR SMCCONT; GO FAR CTRYAGAIN FI
165161          T:=0; X:="MCLGFLG"; *STZTX
165164          GO FAR CTRYAGAIN
165165   *)FILL
165174
165174   FU2:   X:="MCLGFLG"; T:=0; *LDATX                              % RESTART-MONCALL LOG
165177          IF A><0 THEN
165200             "TXMFELL"; CALL SMOUTTEXT; "TNSTOPED"; CALL SMOUTTEXT; GO FAR CTRYAGAIN
165205          FI; X:="MCLGOWNER"; *LDATX
165207          IF A><TTNO THEN CALL FAR SMCCONT; GO FAR CTRYAGAIN FI
165214          T:=0; X:="CMCLG"; *LDATX
165217          IF A=0 THEN
165220             "TXMFELL"; CALL SMOUTTEXT; "TNNSTART"; CALL SMOUTTEXT; GO FAR CTRYAGAIN
165225          FI; TTNO; X:="MCLGOWNER"; *STATX
165230          A:=1; X:="MCLGFLG"; *STATX
165233          GO FAR CTRYAGAIN
165234
165234   FU3:   CALL SMOOPEN; GO FAR CTRYAGAIN; A=:TDVN                  % PRINT-MONCALL LOG
165237          "THEAD"; CALL SMOUTTEXT;
165241          0=:CMNO
165242          DO WHILE CMNO><400
165246             CALL SMCRLF; CMNO; CALL SMOCTUT; "XXTXT"; CALL SMOUTTEXT
165253             X:="MCLGBANK"; T:=0; *LDATX
165256             A=:D; X:="TNMCALL"; *LDXTX
165261             T:=D; CMNO SH 1; X+A; *LDDTX
165266             CALL SMDDECUT; MIN CMNO
165270          OD; CALL SMCRLF; T:=TDVN; *MON 2CLOS; JMP *+1
165275          1=:TDVN                                          %RESET LIST DEVICE NUMBER
165277          GO FAR CTRYAGAIN
165300
165300   FU4:   GO SMLEAVE                                  % EXIT
165301
165301   FU5:   X:="FUNTAB"                                 % HELP
165302          DO
165302             CALL SMCRLF
165303          WHILE X.S0><-1; CALL SMOUTTEXT; X+4
165311          OD; GO FAR CTRYAGAIN
165313
165313
165313   RBUS
165340
165340
165340   %=================================================================
165340   %              S W A P P I N G   -   L O G
165340   %
165340   SUBR SWPLO
165340
165340   INTEGER TFU0:='START-SWAPPING-LOG'
165352   INTEGER TFU1:='STOP-SWAPPING-LOG'
165363   INTEGER TFU2:='RESTART-SWAPPING-LOG'
```

```
165376   INTEGER TFU3:='READ-SWAPPING-LOG'
165407   INTEGER TFU4:='SWAPPING-LOG'
165416   INTEGER TFU5:='EXIT'
165421   INTEGER TFU6:='HELP'
165424   @ICR
165424   INTEGER ARRAY FUNTAB:=(
165424          TFU0,0,FU0,0,          TFU1,0,FU1,0,
165434          TFU2,0,FU2,0,          TFU3,0,FU3,0,
165444          TFU4,0,FU4,0,          TFU5,0,FU5,0,
165454          TFU6,0,FU6,0,
165460          -1);
165461   @CR;
165461
165461   INTEGER TNIMPL:='NOT IMPLEMENTED$'
165472   INTEGER PTINTV:='IDINTERVAL IN SECONDS (DEFAULT IS 60 SECS): '
165521   INTEGER SWLGRUN:='IS ACTIVE'
165526   INTEGER TPOP1:='N LOG SWAPPING FOR A SPECIFIC PROGRAM (DEFAULT IS YES)? '
165563   INTEGER TXYN:='$ONLY YES OR NO IS LEGAL ANSWER'
165603   INTEGER TACCU:='$ $TOT.ACCUM:          '
165615   INTEGER TINTV:='$IN INTV.:          '
165626   INTEGER PTACC:='$TOT.PROG.ACCUM: '
165637   INTEGER TPTIN:='$PROG.IN.INTV:          '
165650   INTEGER THED1:='$                    PF WITHOUT   PF IN          PF ON          PF ON       PAGES'
165716   INTEGER THED2:='$                    DISC ACCESS  RT-COMMON      LEVEL 4        LEVEL 1     SWAPPED OUT$'
165767   INTEGER TSWLUSED:='ACTIVATED BY ANOTHER TERMINAL'
166006   INTEGER TSWCONT:='N $DO YOU WANT TO CONTINUE (DEFAULT IS NO)? '
166035   INTEGER TNSTA:='NOT STARTED'
166043   INTEGER TXSWLFELLS:='$SWAPPING LOG '
166053
166053   DISP -200
166053          INTEGER CPROG,CINTV,HEADCOUNT,CFUNC
166053          INTEGER POINTER LREG
166053          REAL CRTNA
166053   PSID
166053
166053   INTEGER PARHOLD:=(CINTV+BFIELD,"2")
166055
166055   CTRYAGAIN: CALL SMCCLEAR;GO LOOP
166057
166057   ERET1: "TXYN"; CALL SMOUTTEXT; GO CTRYAGAIN
166062
166062   SWPLO: L=:D; CALL SMENTER
166064          IF "CSWLG"=0 THEN
166066             "TXSWLFELLS"; CALL SMOUTTEXT; "TNIMPL"; CALL SMOUTTEXT; GO SMLEAVE
166073          FI
166073   LOOP:  "6CFUPAR"; CALL SMSGPAR;GO CTRYAGAIN; X:="FUNTAB"; CALL SMABLOOK
166100          IF A><0 THEN
166101             IF A=-1 THEN "SMTXFILL" ELSE "SMFAMBIG" FI
166107             CALL SMOUTTEXT; GO CTRYAGAIN
166111          FI; T.S2=:P
166114
166114   FU0:   0=:CFUNC; GO FUFEL                          % START SWAPPING LOG
166116   FU4:   4=:CFUNC                                    % SWAPPING LOG
166120   FUFEL: "TPOP1"; CALL SAGPAR; GO CTRYAGAIN; GO L0; CALL SMYESNO
166125          IF A<0 GO ERET1
166126          IF A=1 THEN
166131   L0:       "6PRTNAME"; CALL SAGPAR; GO CTRYAGAIN; RTREF; A=:CPROG
166136             A-RTSTART=:D:=0; T:=5RTSIZE; *RDIV ST
166143             IF D><0 THEN "SMCILLP"; CALL SMOUTTEXT; GO CTRYAGAIN FI
166150          ELSE
```

```
166151              -1=:CPROG
166153           FI
166153           IF CFUNC=4 THEN                                % SWAPPING LOG
166157              "PTINTV"; CALL SAGPAR; GO CTRYAGAIN; A:=74; A=:CINTV
166164           FI; T:=0; X:="CSWLG"; *LDATX
166167           IF A><0 THEN
166170              "TXSWLFELLS"; CALL SMOUTTEXT; "SWLGRUN"; CALL SMOUTTEXT; GO CTRYAGAIN
166175           FI; A:=0; D:=0; T:=0; X:="TFPFS"; *STDTX       % CLEAR SWAPPING LOG
166202           X:="CFPFS"; *STDTX
166204           X:="TPFRS"; *STDTX
166206           X:="CPFRS"; *STDTX
166210           X:="TPFL4"; *STDTX
166212           X:="CPFL4"; *STDTX
166214           X:="TPFL1"; *STDTX
166216           X:="CPFL1"; *STDTX
166220           X:="SNWPS"; *STDTX
166222           X:="SWLGOWNER"; TTNO; *STATX
166225           X:="CSWLG"; CPROG; *STATX
166230           X:="CCSWLG"; *STATX
166232           IF CFUNC><4 GO CTRYAGAIN; GO L1; *)FILL
166302
166302   CSUBR: D-T; A:=A+C-1-X
166306   PSU1:  T:=L=:"LREG"; CALL SMDDECUT; 20040; CALL SM2TCO; GO LREG
166314
166314   INTEGER 1FPFS=?,2FPFS=?,1PFRS=?,2PFRS=?,1PFL4=?,2PFL4=?,1PFL1=?,2PFL1=?,1NWPS=?,2NWPS=?
166314   INTEGER 3FPFS=?,4FPFS=?,3PFRS=?,4PFRS=?,3PFL4=?,4PFL4=?,3PFL1=?,4PFL1=?
166314   DOUBLE 9FPFS=?,9PFRS=?,9PFL4=?,9PFL1=?,9NWPS=?
166314   DOUBLE 7FPFS=?,7PFRS=?,7PFL4=?,7PFL1=?
166314   DOUBLE XFPFS=?,XPFRS=?,XPFL4=?,XPFL1=?,XNWPS=?
166314   DOUBLE YFPFS=?,YPFRS=?,YPFL4=?,YPFL1=?
166314
166314   L1:    A:=0; D:=0; 0=:HEADCOUNT
166317          AD=:7FPFS=:7PFRS=:7PFL4=:7PFL1
166323          AD=:9FPFS=:9PFRS=:9PFL4=:9PFL1=:9NWPS
166330          CALL ESCON
166331          %DO
166331   L2:       "PARHOLD"; *MON 2HOLD
166333             IF HEADCOUNT=0 THEN
166335                "THED1"; CALL SMOUTTEXT; "THED2"; CALL SMOUTTEXT; 10=:HEADCOUNT
166343             FI; HEADCOUNT-1=:HEADCOUNT
166346             "TACCUM"; CALL SMOUTTEXT
166350             X:="TFPFS"; T:=0; *LDDTX
166353             AD=:XFPFS; CALL PSU1
166355             X:="TPFRS"; T:=0; *LDDTX
166360             AD=:XPFRS; CALL PSU1
166362             X:="TPFL4"; T:=0; *LDDTX
166365             AD=:XPFL4; CALL PSU1
166367             X:="TPFL1"; T:=0; *LDDTX
166372             AD=:XPFL1; CALL PSU1
166374             X:="SNWPS"; T:=0; *LDDTX
166377             AD=:XNWPS; CALL SMDDECUT
166401             "TINTV"; CALL SMOUTTEXT
166403             X:=1FPFS; T:=2FPFS; AD:=XFPFS; CALL CSUBR
166407             X:=1PFRS; T:=2PFRS; AD:=XPFRS; CALL CSUBR
166413             X:=1PFL4; T:=2PFL4; AD:=XPFL4; CALL CSUBR
166417             X:=1PFL1; T:=2PFL1; AD:=XPFL1; CALL CSUBR
166423             X:=1NWPS; T:=2NWPS; AD:=XNWPS; CALL CSUBR
166427             AD:=XFPFS=:9FPFS:=XPFRS=:9PFRS:=XPFL4=:9PFL4:=XPFL1=:9PFL1:=XNWPS=:9NWPS
166441             IF CPROG=-1 GO L2; GO L3; *)FILL
166465
```

```
166465   INTEGER 1FPFS,2FPFS,1PFRS,2PFRS,1PFL4,2PFL4,1PFL1,2PFL1,1NWPS,2NWPS
166477   INTEGER 3FPFS,4FPFS,3PFRS,4PFRS,3PFL4,4PFL4,3PFL1,4PFL1
166507   DOUBLE 9FPFS=1FPFS,9PFRS=1PFRS,9PFL4=1PFL4,9PFL1=1PFL1,9NWPS=1NWPS
166507   DOUBLE 7FPFS=3FPFS,7PFRS=3PFRS,7PFL4=3PFL4,7PFL1=3PFL1
166507   DOUBLE XFPFS,XPFRS,XPFL4,XPFL1,XNWPS
166521   DOUBLE YFPFS,YPFRS,YPFL4,YPFL1
166531
166531   L3:         HEADCOUNT -1 =: HEADCOUNT
166534               "PTACC"; CALL SMOUTTEXT
166536               X:="CFPFS"; T:=0; *LDDTX
166541               AD=:YFPFS; CALL FAR PSU1
166543               X:="CPFRS"; T:=0; *LDDTX
166546               AD=:YPFRS; CALL FAR PSU1
166550               X:="CPFL4"; T:=0; *LDDTX
166553               AD=:YPFL4; CALL FAR PSU1
166555               X:="CPFL1"; T:=0; *LDDTX
166560               AD=:YPFL1; CALL SMDDECUT
166562               "TPTINTV"; CALL SMOUTTEXT
166564               X:=3FPFS; T:=4FPFS; AD:=YFPFS; CALL FAR CSUBR
166570               X:=3PFRS; T:=4PFRS; AD:=YPFRS; CALL FAR CSUBR
166574               X:=3PFL4; T:=4PFL4; AD:=YPFL4; CALL FAR CSUBR
166600               X:=3PFL1; T:=4PFL1; AD:=YPFL1; CALL FAR CSUBR
166604               AD:=YFPFS=:7FPFS:=YPFRS=:7PFRS:=YPFL4=:7PFL4:=YPFL1=:7PFL1
166614               GO FAR L2; *)FILL
166630           %OD
166630
166630   SWLCONT: A:=L=:"LREG"
166632           "TXSWLFELLS"; CALL SMOUTTEXT
166634           "TSWLUSED"; CALL SMOUTTEXT; "TSWCONT"; CALL SAGPAR; GO LREG; GO LREG
166642           CALL SMYESNO; IF A<0 GO FAR ERET1
166645           IF A><1 GO LREG
166650           MIN "LREG"; GO LREG
166652                                                  % STOP SWAPPING LOG
166652   FU1:    T:=0; X:="CSWLG"; *LDATX
166655           IF A=0 THEN
166656              "TXSWLFELLS"; CALL SMOUTTEXT; "TNSTART"; CALL SMOUTTEXT; GO FAR CTRYAGAIN
166663           FI; X:="SWLGOWNER"; *LDATX
166665           IF A><TTNO THEN CALL SWLCONT; GO FAR CTRYAGAIN FI
166672           T:=0; X:="CSWLG"; *STZTX
166675           GO FAR CTRYAGAIN
166676                                                  % RESTART SWAPPING LOG
166676   FU2:    T:=0; X:="CSWLG"; *LDATX
166701           IF A><0 THEN
166702              "TXSWLFELLS"; CALL SMOUTTEXT; "SWLGRUN"; CALL SMOUTTEXT; GO FAR CTRYAGAIN
166707           FI; X:="SWLGOWNER"; *LDATX
166711           IF A><TTNO THEN CALL SWLCONT; GO FAR CTRYAGAIN FI
166716           T:=0; X:="CCSWLG"; *LDATX
166721           X:="CSWLG"; *STATX
166723           A:=TTNO; X:="SWLGOWNER"; *STATX
166726           GO FAR CTRYAGAIN
166727   *)FILL
166744
166744   INTEGER FU3T1:='$TOTAL NUMBER OF PAGEFAULTS WITHOUT DISC ACCESS
166777   INTEGER FU3T2:='$NUMBER OF PAGEFAULTS WITHOUT DISC ACCESS FOR'
167026   %INTEGER FU3T3:='$TOTAL NUMBER OF PAGEFAULTS IN RT-COMMON
167026   %INTEGER FU3T4:='$NUMBER OF PAGEFAULTS IN RT-COMMON FOR '
167026   %INTEGER FU3T5:='$TOTAL NUMBER OF PAGEFAULTS ON LEVEL 4
167026   %INTEGER FU3T6:='$NUMBER OF PAGEFAULTS ON LEVEL 4 FOR '
167026   %INTEGER FU3T7:='$TOTAL NUMBER OF PAGEFAULTS ON LEVEL 1
167026   %INTEGER FU3T8:='$NUMBER OF PAGEFAULTS ON LEVEL 1 FOR '
```

```
167026   %INTEGER FU3T9:='$TOTAL NUMBER OF PAGES SWAPPED OUT (WRITTEN TO DISC) '
167026   INTEGER ARRAY FU3T3(0)
167026   INTEGER ARRAY FU3T5(0)
167026   INTEGER ARRAY FU3T7(0)
167026   INTEGER ARRAY FU3T9:='$TOTAL NUMBER OF PAGE'
167041   INTEGER ARRAY FU3X3:='FAULTS IN RT-COMMON'
167053   INTEGER ARRAY FU3X5:='FAULTS ON LEVEL 4'
167064   INTEGER ARRAY FU3X7:='FAULTS ON LEVEL 1'
167075   INTEGER ARRAY FU3X9:='S SWAPPED OUT (WRITTEN TO DISC) '
167116   INTEGER ARRAY FU3T4(0)
167116   INTEGER ARRAY FU3T6(0)
167116   INTEGER ARRAY FU3T8:='$NUMBER  OF  PAGEFAULTS '
167133   INTEGER ARRAY FU3X4:=' IN  RT-COMMON  FOR  '
167146   INTEGER ARRAY FU3X6:='  ON  LEVEL  4  FOR  '
167161   INTEGER ARRAY FU3X8:='  ON  LEVEL  1  FOR  '
167174
167174   FU3:    T:=0; X:="CCSWLG"; *LDATX
167177           A=:CPROG
167200           "FU3T1"; CALL SMOUTTEXT
167202           X:="TFPFS"; T:=0; *LDDTX
167205           CALL SMDDECUT
167206           IF CPROG><-1 THEN
167212              "FU3T2"; CALL SMOUTTEXT; "CPROG"+B; CALL OPSYS(INAMS);*JMP * 1
167221              TAD =: CRTNA;D=: X;CALL OPSYS (PRIRT)
167225              X:="CFPFS"; T:=0; *LDDTX
167230              CALL SMDDECUT
167231           FI; "FU3T3"; CALL SMOUTTEXT; "FU3X3"; CALL SMOUTTEXT; 15; CALL SMSPAC
167237           X:="TPFRS"; T:=0; *LDDTX
167242           CALL SMDDECUT
167243           IF CPROG><-1 THEN
167247              "FU3T4"; CALL SMOUTTEXT; "FU3X4"; CALL SMOUTTEXT
167253              TAD:=CRTNA; D=:X; CALL OPSYS(PRIRT)
167257              X:="CPFRS"; T:=0; *LDDTX
167262              CALL SMDDECUT
167263           FI; "FU3T5"; CALL SMOUTTEXT; "FU3X5"; CALL SMOUTTEXT; 17; CALL SMSPACE
167271           X:="TPFL4"; T:=0; *LDDTX
167274           CALL SMDDECUT
167275           IF CPROG><-1 THEN
167301              "FU3T6"; CALL SMOUTTEXT; "FU3X6"; CALL SMOUTTEXT
167305              TAD:=CRTNA; D=:X; CALL OPSYS(PRIRT)
167311              X:="CPFL4"; T:=0; *LDDTX
167314              CALL SMDDECUT
167315           FI; "FU3T7"; CALL SMOUTTEXT; "FU3X7"; CALL SMOUTTEXT; 17; CALL SMSPACE
167323           X:="TPFL1"; T:=0; *LDDTX
167326           CALL SMDDECUT
167327           IF CPROG><-1 THEN
167333              "FU3T8"; CALL SMOUTTEXT; "FU3X8"; CALL SMOUTTEXT
167337              TAD:=CRTNA; D=:X; CALL OPSYS(PRIRT)
167343              X:="CPFL1"; T:=0; *LDDTX
167346              CALL SMDDECUT
167347           FI; "FU3T9"; CALL SMOUTTEXT; "FU3X9"; CALL SMOUTTEXT
167353           X:="SNWPS"; T:=0; *LDDTX
167356           CALL SMDDECUT; CALL SMCRLF
167360           GO FAR CTRYAGAIN
167361   *)FILL
167415
167415   FU5:    GO SMLEAVE
167416
167416   FU6:    X:="FUNTAB"
167417           DO
```

```
167417              CALL SMCRLF
167420           WHILE X.SO><-1; CALL $MOUTTEXT; X+4
167426           OD; CALL SMCRLF
167430           GO FAR CTRYAGAIN
167431
167431   RBUS
167436
167436
167436   %=========================================================================
167436   %              C P U - L O G
167436   %
167436   SUBR CMCPULG,FCPULP
167436
167436   DISP -200
167436           INTEGER CINTV,NLPSECS
167436           DOUBLE CLPTOTAL,CNLOOPS
167436   PSID
167436
167436   INTEGER PXHOLD:=("3","2"),PHOLD:=(CINTV+BFIELD,"2")
167442   INTEGER PTINT:='IDINTERVAL IN SECONDS (DEFAULT IS 30 SECS): '
167471   INTEGER CTX1:='$THIS COMMAND TAKES 30 SECONDS.$'
167512   INTEGER CTX2:='$NUMBER OF "CPULOOPS" PER SECOND:'
167533   *)FILL
167535   CSUBR: A:=0; *IRW DA; IRW DD
167540           A:=175732; *IRW DT                          % LOOPCOUNTER FOR IDLE LOOP
167542           A:="ENTO"; *IRW DP
167544           EXIT
167545
167545   FCPULP: L=:D; CALL SMENTER
167547           "CTX1"; CALL SMOUTTEXT; CALL ESCON
167552           36=:CINTV; "PXHOLD"; *MON 2HOLD
167556           CALL CSUBR; "PHOLD"; *MON 2HOLD
167561           *IRR DD; COPY SA DD; IRR DA
167564           T:=36; *RDIV ST
167566           IF D BIT 17 THEN A+1 FI
167571           A=:NLPSECS; "CTX2"; CALL SMOUTTEXT; NLPSECS; CALL SMDECUT
167576           GO SMLEAVE
167577
167577   CMCPULG: L=:D; CALL SMENTER
167601           "PTINTV"; CALL SMAGPAR;  A:=36; A=:CINTV
167605           IF A=0 THEN "SMCILLP"; CALL SMOUTTEXT; GO SMLEAVE FI
167611           CALL 6OOPEN; GO SMLEAVE; A=:TDVN
167614           A:=CPULOOPTIME=:X; A:=0 ; D:=0; T:=CINTV
167621
167621           DO WHILE T><0; D+X; A:=A+C; T-1; OD
167627           AD=:CLPTOTAL                                % TOTAL NO. OF LOOPS IN INTERVAL
167630           "PXHOLD"; *MON 2HOLD
167632           CALL ESCON; CALL CSUBR
167634           DO                                          % UNTIL ESCAPE
167634              "PHOLD"; *MON 2HOLD
167636              *IRR DD; COPY SA DD; IRR DA
167641              T:=D; X:=A; AD:=CLPTOTAL                 % NUMBER OF LOOPS IN THIS INTERVAL
167644              D-T; A:=A+C-1-X; AD=:CNLOOPS
167651              CALL CSUBR; CALL SMCRLF
167653              "CLPTOTAL"+"BFIELD"=:X; AD:=CNLOOPS; CALL SMPERCENT
167660           OD
167661   RBUS
167706
167706   %=========================================================================
167706   %              C H A N G E - T A B L E
```

```
167706   %
167706   SUBR CMCHTABLE
167706
167706   DISP -200
167706           INTEGER CARRAY,CMXINDX,LEGAREA,CFUNADDR,CFUTYPE,CINDX
167706           INTEGER CFP,CLP; DOUBLE CDVAL=CFP
167706           INTEGER POINTER LREG
167706   PSID
167706
167706   INTEGER PARTABLE:='N $USER-RESERVED-DEVICE-NUMBERS,   '
167730   * *-1/'$USER-RESERVED-MEMORY-AREA,   '
167746   * *-1/'$MEMORY-AREA-UNAVAILABLE-FOR-SWAPPING,   '
167772   * *-1/'$MEMORY-AREA-INVISIBLE-FOR-THIS-SYSTEM   ';
170016   * *-1/'$TABLE: '
170022   @ICR
170022   INTEGER ARRAY TABARR:=(
170022           TPA1,USIOXTAB,USDVSIZE,1,        TPA2,NSWPAGE,NNSWSZ,0,
170032           TPA3,NINITPAGE,NINSZ,0,          TPA4,CUMTABLE,CUMSIZE,0,
170042           -1);
170043   INTEGER ARRAY FUNTAB:=(
170043           TFU1,FULIST,1,0,       TFU2,FUCHTB,0,0,
170053           TFU3,FUCHEL,1,0,       TFU4,FUDEL,1,0,
170063           TFU5,FUINS,1,0,        TFU6,FUCLTB,1,0,
170073           TFU7,FUEXI,0,0,        TFU8,FUHLP,0,0,
170103           -1);
170104   INTEGER ARRAY TSELAREA:=(
170104           TXIMA,2,0,0,
170110           TXSAV,4,0,0,           -1);
170115   INTEGER ARRAY LIHEAD:=(LOHEAD,L1HEAD);
170117   @CR;
170117   INTEGER TXIMA:='IMAGE'
170122   INTEGER TXSAV:='SAVE-AREA'
170127   INTEGER PSELAREA:='N IMAGE OR SAVE-AREA (DEFAULT IS IMAGE): '
170154   INTEGER TFU1:='LIST-TABLE'
170162   INTEGER TFU2:='CHANGE-TABLE'
170171   INTEGER TFU3:='CHANGE-ELEMENT'
170201   INTEGER TFU4:='DELETE-ELEMENT'
170211   INTEGER TFU5:='INSERT-ELEMENT'
170221   INTEGER TFU6:='CLEAR-TABLE'
170227   INTEGER TFU7:='EXIT'
170232   INTEGER TFU8:='HELP'
170235   INTEGER TPA1:='USER-RESERVED-DEVICE-NUMBERS'
170254   INTEGER TPA2:='MEMORY-AREA-UNAVAILABLE-FOR-SWAPPING'
170277   INTEGER TPA3:='MEMORY-AREA-INVISIBLE-FOR-THIS-SYSTEM'
170322   INTEGER TPA4:='USER-RESERVED-MEMORY-AREA'
170337   INTEGER LOHEAD:='$ELEMENT NO.     FIRST PAGE      LAST PAGE'
170365   INTEGER L1HEAD:='$ELEMENT NO.     FIRST DEVNO     LAST DEVNO'
170413   INTEGER PELENO:='IDELEMENT NUMBER: '
170425   INTEGER ARRAY 1TXPAR:=(1TXT0,1TXT1),2TXPAR:=(2TXT0,2TXT1)
170431   INTEGER 1TXT0:='IOFIRST PAGE (OCT): ',1TXT1:='IOFIRST DEVNO. (OCT): '
170460   INTEGER 2TXT0:='IOLAST PAGE (OCT): ',2TXT1:='IOLAST DEVNO. (OCT): '
170505   INTEGER TTFLL :='TABLE IS FULL'
170514
170514   CTRYAGAIN:  CALL SMCCLEAR;GO LOOP        %  CLEAR COMMAND BUFFER
170516   C1TRYAGAIN: CALL SMCCLEAR;GO LOOP1       %  BEFORE READING NEW SUB-COMMAND
170520
170520   CMCHTABLE: L=:D; CALL SMENTER
170522   FUCHTB:
170522   LOOP:  "PARTABLE"; CALL SMSGPAR;GO CTRYAGAIN; X:="TABARR"; CALL SMABLOOK
170527           IF A><0 THEN
```

```
170530              IF A=-1 THEN "SMTXFILL" ELSE "SMFAMBIG" FI
170536              CALL SMOUTTEXT; GO CTRYAGAIN
170540           FI; T.S1=:CARRAY; X.S2=:CMXINDX; X.S3=:CFUTYPE
170547
170547   LOOP1:  "6CFUPAR"; CALL SMSGPAR; GO C1TRYAGAIN; X:="FUNTAB"; CALL SMABLOOK
170554           IF A><0 THEN
170555   L1:        IF A=-1 THEN "SMCILLP" ELSE "SMCAMBIGP" FI
170563   L11:       CALL SMOUTTEXT; GO C1TRYAGAIN
170565           FI; T.S1=;CFUNADDR
170570           IF X.S2><0 THEN
170572              2=:LEGAREA
170574              "PSELAREA"; CALL SAGPAR; GO C1TRYAGAIN; GO L2
170600              X:="TSELAREA"; CALL SMABLOOK
170602              IF A><0 GO L1; T.S1=:LEGAREA
170606           FI
170606   L2:     CFUNADDR=:P
170610   USMEM:  1=:LEGAREA; GO L2
170613   *)FILL
170633
170633   FULIST: CALL 6OOPEN; GO SMLEAVE; A=:TDVN
170636           IF LEGAREA BIT BSAVE THEN CALL 6SOPEN FI
170642           LIHEAD(CFUTYP); CALL SMOUTTEXT
170645           0=:CINDX
170646           DO WHILE CINDX<<CMXINDX
170652              CALL SMCRLF; 3; CALL SMSPACE; CINDX ; CALL SMDECUT
170657              CINDX SH 1+CARRAY=:X
170663              IF LEGAREA BIT BIML THEN
170666                 CALL R1IMAGE; A=:CFP; X+1; CALL R1IMAGE; A=:CLP
170673              ELSE
170674                 CALL R1SAVE; A=:CFP; X+1; CALL R1SAVE; A=:CLP
170701              FI; 10; CALL SMSPACE; CFP; CALL SMOCTUT
170705              10; CALL SMSPACE; CLP; CALL SMOCTUT
170711              CINDX+1=:CINDX
170714           OD; T:=TDVN; *MON 2CLOS; JMP *+1
170720           IF LEGAREA BIT BSAVE THEN CALL 6CLOSE FI
170724           1=:TDVN
170726           GO FAR C1TRYAGAIN; *)FILL
170745
170745   CGELNO: A:=L=:"LREG"
170747           "PELENO"; CALL SMSGPAR; GO FAR C1TRYAGAIN; A=:CINDX
170753           IF A>>=CMXINDX THEN "SMCILLP"; CALL SMOUTTEXT; GO FAR C1TRYAGAIN FI
170761           GO LREG
170762
170762   FUDEL:  CALL CGELNO
170763           CINDX SH 1+CARRAY=:X
170767           IF LEGAREA BIT BSAVE THEN
170772              CALL 6SOPEN; A:=0; CALL W1SAVE; A:=0; X+1; CALL W1SAVE
171000              CALL WXSAVE; CALL 6CLOSE
171002           ELSE
171003              A:=0; CALL W1IMAGE; X+1; A:=0; CALL W1IMAGE; CALL WIMBACK
171011           FI; GO FAR C1TRYAGAIN; *)FILL
171025
171025   FUCHEL: CALL CGELNO
171026   FUFEL:  1TXPAR(CFUTYPE); CALL SMSGPAR; GO FAR C1TRYAGAIN; A=:CFP
171033           2TXPAR(CFUTYPE); CALL SMSGPAR; GO FAR C1TRYAGAIN; A=:CLP
171040           IF A<<CFP THEN
171043              "SMCILLP";GO FAR L11
171045           FI
171045           CINDX SH 1+CARRAY=:X
171051           IF LEGAREA BIT BSAVE THEN
```

```
171054              CALL 6SOPEN; CFP; CALL W1SAVE; X+1; CLP; CALL W1SAVE
171062              CALL WXSAVE; CALL 6CLOSE
171064           ELSE
171065              CFP; CALL W1IMAGE; X+1; CLP; CALL W1IMAGE; CALL WIMBACK
171073           FI; GO FAR C1TRYAGAIN
171074
171074   FUINS: 0=:CINDX
171075           IF LEGAREA BIT BSAVE THEN CALL 6SOPEN FI
171101           DO WHILE CINDX<<CMXINDX
171105              CINDX SH 1+CARRAY=:X
171111              IF LEGAREA BIT BSAVE THEN
171114                 CALL R1SAVE; IF A><0 GO EDO; X+1; CALL R1SAVE; IF A=0 GO ELFOUND
171121              ELSE
171122                 CALL R1IMAGE; IF A><0 GO EDO; X+1; CALL R1IMAGE; IF A=0 GO ELFOUND
171127              FI
171127   EDO:       CINDX+1=:CINDX
171132           OD; "TTFLL"; GO FAR L11                    %TABLE IS FULL
171135   ELFOUND: IF LEGAREA BIT BSAVE THEN CALL 6CLOSE FI
171141           GO FUFEL
171142   *)FILL
171161
171161   FUCLTB: 0=:CINDX
171162           IF LEGAREA BIT BSAVE THEN CALL 6SOPEN FI
171166           DO WHILE CINDX<<CMXINDX
171172              CINDX SH 1+CARRAY=:X
171176              IF LEGAREA BIT BSAVE THEN
171201                 A:=0; CALL W1SAVE; X+1; A:=0; CALL W1SAVE
171206              ELSE
171207                 A:=0; CALL W1IMAGE; X+1; A:=0; CALL W1IMAGE
171214              FI; CINDX+1=:CINDX
171217           OD
171220           IF LEGAREA BIT BSAVE THEN CALL WXSAVE; CALL 6CLOSE FI
171225           IF LEGAREA BIT BIML THEN CALL WIMBACK FI
171231           GO FAR C1TRYAGAIN
171232
171232   FUEXI: GO SMLEAVE
171233
171233   FUHLP: X:="FUNTAB"
171234           DO
171234              CALL SMCRLF
171235           WHILE X.S0><-1; CALL SMOUTTEXT; X+4
171243           OD; GO FAR C1TRYAGAIN
171245
171245   RBUS
171260
171260
171260
171260
171260   %=================================================================
171260   %      B A C K G R O U N D   A L L O C A T I O N   U T I L I T I E S
171260   %
171260   SUBR BAUTIL
171260
171260   DISP -200
171260           INTEGER LEGAREA,ROUTFLAG,CDFELT,CTIME,DYNAL,CSEG,CT1,CT2
171260           INTEGER POINTER LREG,CROUTINE,C2ROUTINE
171260           INTEGER CINDX=CDFELT,CCOUNTER=CTIME,CPROG=DYNAL
171260           INTEGER CADDR=CSEG,CBPRFLG=CT1,CPT3OFFS=CT2
171260           INTEGER CCLOG=CTIME
171260           INTEGER HEADCOUNT=ROUTFLAG
```

```
171260   PSID
171260
171260   @ICR
171260   INTEGER ARRAY FUNTAB:=(
171260          TF1, 1,FU1,0,          TF2, 2,FU2,0,
171270          TF3, 3,FU3,0,          TF4, 4,FU4,0,
171300          TF5, 5,FU5,0,          TF6, 6,FU6,0,
171310          TF7, 7,FU7,0,          TF8,10,FU8,0,
171320          TF9,11,FU9,0,          TF10,12,FU10,0,
171330          TF11,13,FU11,0,        TFH,0,FUH,0,
171340          TFE, 0,FUE,0,
171344          -1);
171345   @CR;
171345   INTEGER TF1:='SET-PERMANENT-CONNECTION'
171362   INTEGER TF2:='RESET-PERMANENT-CONNECTION'
171400   INTEGER TF3:='ENABLE-TIMEOUT'
171410   INTEGER TF4:='DISABLE-TIMEOUT'
171420   INTEGER TF5:='CHANGE-WARNING-TIME'
171432   INTEGER TF6:='CHANGE-LOGOUT-TIME'
171444   INTEGER TF7:='DISPLAY'
171450   INTEGER TF8:='TIMEOUT-OFF'
171456   INTEGER TF9:='TIMEOUT-ON'
171464   INTEGER TF10:='LIST-PARAMETERS'
171474   INTEGER TF11:='FREE-BACKGROUND-PROGRAMS'
171511   INTEGER TFH:='HELP'
171514   INTEGER TFE:='EXIT'
171517   INTEGER TNIMPL:='$BACKGROUND PROGRAM ALLOCATION SYSTEM NOT IMPLEMENTED$'
171553   INTEGER TXEROP:='$ERROR IN ACCESSING IMAGE OR SAVE'
171574   INTEGER CETX1:='$ILLEGAL LOGICAL UNIT'
171607   INTEGER CETX2:='$NO FREE BACKGROUND PROGRAM FOUND'
171630   INTEGER CETX3:='$ILLEGAL ON TAD'
171640   INTEGER CET56:='$ERROR, LOGOUT TIME MUST BE LONGER THAN WARNING TIME'
171673   INTEGER 6PTIM:='IDNUMBER OF MINUTES (DEC.): '
171712   INTEGER LTXF:='$NUMBER OF MINUTES INACTIVE BEFORE '
171734   INTEGER LTX0:='$TIMEOUT:'
171741   INTEGER LTX1:='LOGGED OUT:'
171747   INTEGER LTX2:='WARNING   :'
171755   INTEGER TMEM:=' MEMORY'
171761   INTEGER TIMA:='  IMAGE'
171765   INTEGER TSAV:='   SAVE'
171771   INTEGER TXOFF:='    OFF'
171775   INTEGER TXON:= '    ON '
172001   INTEGER PWSEG:=BFIELD+CSEG
172002   INTEGER PCABRT:=("BPTMP","17","2")
172005   INTEGER THED1:='$LOG.DEV.  TYPE      STATUS      BACKG.  DYN.   TIME-   PERM.'
172044   INTEGER THED2:='$ (DEC)                          PROGR.  ALLOC.  OUT    CONNECTION'
172105   INTEGER T5TERM:='   TERMINAL'
172113   INTEGER T5BAD:= '   TAD     '
172121   INTEGER TBATCH:='   BATCH   '
172127   INTEGER TNDNET:='   ND-NET  '
172135   INTEGER TRESER:='  RESERVED'
172143   INTEGER TFREE:= '  FREE    '
172151   INTEGER TYES:=  '    YES'
172155   INTEGER TNO:=   '    NO '
172161   *)FILL
172164
172164   CSUB3:    T:=ROUTFLG GOSW LAB0,LAB1,LAB2
172171   LAB3:     A BZERO BPRTMOUT; EXIT
172173   LAB0:     A BONE  BPCFIXED; EXIT
172175   LAB1:     A BZERO BPCFIXED; EXIT
```

```
172177   LAB2:     A BONE  BPRTMOUT; EXIT
172201
172201   CSUBR: A=:CSEG:=L=:"LREG"
172204          DYNAL*5PRVTSIZE+"YSVBPRFL"+"PRVTT"-CPT3OFFS=:X
172212          T:=CSEG;CALL GET1L;GO ERROP
172215          CALL CSUB3
172216          T:=CSEG;CALL PUT1L;GO ERROP
172221
172221          "SBPRTAB"+"IMCBPTERM"=:X
172224          DO WHILE X<<"EBPRT"
172227             A:=CPT3OFFS ;X-A
172231             T:=CSEG; CALL GET1L; GO ERROP
172234             IF A=CDFELT THEN
172237                "IMBPRFLG-IMCBPTERM"; X+A; T:=CSEG; CALL GET1L; GO ERROP
172244                CALL CSUB3
172245   LABF:        T:=CSEG; CALL PUT1L; GO ERROP
172250   OKRET:       MIN "LREG"; GO LREG
172252             FI; X+BPRTSIZE;A:=CPT3OFFS;X+A
172255          OD
172256          IF ROUTFLAG><0 GO OKRET
172260          "SBPRTAB"+"IMCBPTERM"=:X
172263          DO WHILE X<<"EBPRT"
172266             A:=CPT3OFFS;X-A
172270             T:=CSEG; CALL GET1L; GO ERROP
172273             IF A=0 THEN
172274                A:=CDFELT; CALL PUT1L; GO ERROP
172277                "IMBPRFLG-IMCBPTERM"; X+A; CALL GET1L; GO ERROP
172303                A BONE BPCFIXED; CALL PUT1L; GO ERROP
172306                GO OKRET
172307             FI; X+BPRTSIZE;A:=CPT3OFFS;X+A
172312          OD; GO LREG
172314   ERROP: "TXEROP"; CALL SMOUTTEXT; GO CTRYAGAIN
172317
172317   CWSEG: A=:CSEG; "PWSEG"; *MON 2WSEG
172322          EXIT
172323
172323   +)FILL
172337
172337   BAUTIL: L=:D; CALL SMENTER
172341          IF "BPTMP"=0 THEN "TNIMPL"; CALL SMOUTTEXT; GO SMLEAVE FI
172346          CALL ESCON
172347   LOOP:  -1=:6BLCKI;T:=-1; *MON 2CLOS; JMP * +1
172354          "6CFUPAR"; CALL SMSGPAR; GO CTRYAGAIN; X:="FUNTAB"; CALL SMABLOOK
172361          IF A><0 THEN
172362             IF A=-1 THEN "SMTXFILL" ELSE "SMFAMBIG" FI
172370             CALL SMOUTTEXT; GO CTRYAGAIN
172372          FI; T.S2=:P
172375
172375   CTRYAGAIN: CALL SMCCLEAR; GO LOOP
172377
172377   FU1:                                          % SET-PERMANENT-CONNECTION
172377   FU2:                                          % RESET-PERMANENT-CONNECTION
172377   FU3:                                          % ENABLE-TIMEOUT
172377   FU4:                                          % DISABLE-TIMEOUT
172377          X.S1-1=:ROUTFLAG
172402          0=:DYNAL; "6PLOGU"; CALL SMSGPAR; GO CTRYAGAIN
172406          CALL LOGPH; IF A=0 GO ERR1; A=:CDFELT
172411          IF A.TYPRING BIT 5BAD THEN
172415             IF ROUTFLAG=2  GO ERR3
172421          FI
```

```
172421          X:=CDFELT; CALL SMDYNALLOC; GO ERR1; A=:DYNAL
172425          7; CALL SELAREA; IF A=0 GO CTRYAGAIN; A=:LEGAREA
172431          IF A BIT BMEMO THEN
172433             DYNAL*5PRVTSI+APRVTTABLE=:X; T:=MBPRVTTABLE; *LDATX TXSVB
172441             CALL FAR CSUB3
172442   FELLS:    T:=MBPRVTTABLE; *STATX TXSVB
172444             A:=ROUTFLAG; T:=CDFELT; CALL XSBPRTAB; GO ERR2
172450          FI
172450          IF LEGAREA BIT BIML THEN
172453              0=:CPT3OFFS
172454              5PT3S;CALL FAR CSUBR;GO ERR2
172457          FI
172457          IF LEGAREA BIT BSAVE THEN
172462              "SG41".LOGADR /\ 77 SH 12 =: CPT3OFFS
172467              5IPT3S; CALL FAR CSUBR; GO ERR2
172472          FI
172472          5PT3S; CALL CWSEG; 5IPT3S; CALL CWSEG
172476          GO CTRYAGAIN
172477   *)FILL
172531   ERR1:  "CETX1"; CALL SMOUTTEXT; GO FAR CTRYAGAIN
172534   ERR2:  "CETX2"; CALL SMOUTTEXT; GO FAR CTRYAGAIN
172537   ERR3:  "CETX3"; CALL SMOUTTEXT; GO FAR CTRYAGAIN
172542   CSUBTST: IF ROUTFLAG=0 THEN
172544                IF CTIME>>CT1 GO ERR56
172550            ELSE
172551                IF CTIME<<CT2 GO ERR56
172555            FI; EXIT
172556   ERR56: "CET56"; CALL SMOUTTEXT; GO FAR CTRYAGAIN
172561
172561   FU5:                                          % CHANGE-WARNING-TIME
172561   FU6:                                          % CHANGE-LOGOUT-TIME
172561          "6PTIME"; CALL SMSGPAR; GO FAR CTRYAGAIN; A SH 2=:CTIME
172566          7; CALL SELAREA; IF A=0 GO FAR CTRYAGAIN; A=:LEGAREA
172573          X.S1-5=:ROUTFLAG
172576          IF LEGAREA BIT BMEMO THEN
172601             ONTTMCOUNT=:CT1; TTMWARNING=:CT2; CALL CSUBTST
172606          FI
172606          IF LEGAREA BIT BIML THEN
172611             X:="ONTTMCOUNT"; CALL R1IMAGE; A=:CT1; X:="TTMWARNING"; CALL R1IMAGE; A=:CT2
172617             CALL CSUBTST
172620          FI
172620          IF LEGAREA BIT BSAVE THEN
172623             CALL 6SOPEN
172624             X:="ONTTMCOUNT"; CALL R1SAVE; A=:CT1; X:="TTMWARNING"; CALL R1SAVE; A=:CT2
172632             CALL CSUBTST
172633          FI; IF ROUTFLAG=0 THEN X:="TTMWARNING" ELSE X:="ONTTMCOUNT" FI
172640          IF LEGAREA BIT BMEMO THEN CTIME=:X.SO FI
172645          IF LEGAREA BIT BIML THEN CTIME; CALL W1IMAGE; CALL WIMBACK FI
172653          IF LEGAREA BIT BSAVE THEN CTIME; CALL W1SAVE; CALL WXSAVE FI
172661          GO FAR CTRYAGAIN
172662   *)FILL
172704
172704   CCROUT: A=:"CROUTINE":=L=:"LREG"; T=:"C2ROUTINE"
172710           "ACTPRI"+"BPTMP"=:X; CALL CROUTINE
172714           IF T:=ROUTFLAG=0 THEN A BONE 5RTOFF ELSE A BZERO 5RTOFF FI
172722           CALL C2ROUTINE; GO LREG
172724
172724   FU8:                                          % TIMEOUT-OFF
172724   FU9:                                          % TIMEOUT-ON
172724          X.S1-10=:ROUTFLAG
```

```
172727          7; CALL SELAREA; IF A=0 GO FAR CTRYAGAIN; A=:LEGAREA
172734          IF A BIT BMEMO THEN
172736             IF ROUTFLAG=0 THEN
172740                "PCABRT"; *MON 2ABOR
172742                "PCABRT"; *MON 2RTOF
172744             ELSE
172745                X:="BPTMP"; CALL SRTON
172747                "PCABRT"; *MON 2INTV
172751                "PCABRT"; *MON 2RT
172753             FI
172753          FI
172753          IF LEGAREA BIT BIML THEN "R1IMAGE"; T:="W1IMAGE"; CALL CCROUT; CALL WIMBACK FI
172762          IF LEGAREA BIT BSAVE THEN CALL 6SOPEN; "R1SAVE"; T:="W1SAVE"; CALL CCROUT; CALL WXSAVE FI
172772          GO FAR CTRYAGAIN
172773   *)FILL
173010
173010   CSUB1: IF A BIT 5RTOFF THEN "TXOFF" ELSE "TXON" FI
173015          GO SMOUTTEXT
173016
173016   CSUB2: A:=L=:"LREG"
173020          IF LEGAREA BIT BMEMO THEN X.S0; A SHZ -2;CALL SMDECUT;FI
173026          IF LEGAREA BIT BIML THEN CALL R1IMAGE; A SHZ -2; CALL SMDECUT;FI
173034          IF LEGAREA BIT BSAVE THEN CALL R1SAVE; A SHZ -2; CALL SMDECUT;FI
173042          GO LREG
173043
173043
173043   FU10:                                          % LIST-PARAMETERS
173043          7; CALL SELAREA; IF A=0 GO FAR CTRYAGAIN; A=:LEGAREA
173050          55; CALL SMSPACE;
173052          IF LEGAREA BIT BMEMO THEN "TMEM"; CALL SMOUTTEXT FI
173057          IF LEGAREA BIT BIML THEN "TIMA"; CALL SMOUTTEXT FI
173064          IF LEGAREA BIT BSAVE THEN CALL 6SOPEN;"TSAV"; CALL SMOUTTEXT FI
173072          "LTX0"; CALL SMOUTTEXT; 46; CALL SMSPACE
173076          "BPTMP"+"ACTPRI"=:X
173101          IF LEGAREA BIT BMEMO THEN X.S0; CALL CSUB1 FI
173106          IF LEGAREA BIT BIML THEN CALL R1IMAGE; CALL CSUB1 FI
173113          IF LEGAREA BIT BSAVE THEN CALL R1SAVE; CALL CSUB1 FI
173120          "LTXF"; CALL SMOUTTEXT; "LTX1"; CALL SMOUTTEXT
173124          X:="ONTTMCOUNT"; CALL CSUB2
173126          "LTXF"; CALL SMOUTTEXT; "LTX2"; CALL SMOUTTEXT
173132          X:="TTMWARNING"; CALL CSUB2
173134          GO FAR CTRYAGAIN
173135   *)FILL
173162
173162   FU11:                                          % FREE-BACKGROUND-PROGRAMS
173162          0=:CINDX; -1=:CCOUNTER
173165          DO
173165             CINDX*BPRTSIZE+ASBPRTAB
173170             IF A>>=AEBPRTAB THEN CALL SMCRLF; GO FAR CTRYAGAIN FI
173175             T:=MBSPRTAB; A=: X; *LDATX TXBPR
173200             IF A NBIT BPCFIXED THEN              %  NOT PERMANENT-CONNECTED
173202                *LDATX TXBBP
173203                A=:CPROG
173204                IF A.TLINK=0 AND X.WLINK=0 THEN   %  PROGRAM IS FREE
173211                   MIN CCOUNTER; GO NXT; CALL SMCRLF; -5=:CCOUNTER
173216   NXT:          A:="CPROG"+B
173220                   CALL OPSYS(INAMS); GO FAR CTRYAGAIN; X:=D; CALL OPSYS(PRIRT)
173226                   4; CALL SMSPACE
173230                FI
173230             FI; MIN CINDX
```

```
173231          OD
173232    *)FILL
173243
173243    FU7:                                                  % DISPLAY
173243          "BACKT"=:CADDR ;0=:HEADCOUNT
173246          DO WHILE CADDR.S0><-1
173253             A=:CDFELT; CALL SMFFLOGDV; GO FU7B;A=:CCLOG
173257             IF HEADCOUNT =0 THEN
173261                CALL SMCRLF
173262                "THED1"; CALL SMOUTTEXT; "THED2"; CALL SMOUTTEXT;24=: HEADCOUNT
173270             FI;HEADCOUNT -1 =: HEADCOUNT
173273             CALL SMCRLF;CCLOG; CALL SMDECUT
173276             IF CDFELT.TYPRING BIT 5TERM THEN "T5TERM"
173303             ELSE IF A BIT 5BAD THEN "T5BAD"
173307             ELSE IF CADDR>>="BBCHT" AND A<<"EBCHT" THEN "TBATCH"
173320             ELSE "TNDNET"
173322             FI; FI; FI; CALL SMOUTTEXT
173323             T:="DBPROG"; CALL XGTDFADDR
173325             IF A=X.RTRES AND A><0 THEN
173331                A=:CPROG; "TRESER"; CALL SMOUTTEXT
173334                "CPROG"+B; CALL OPSYS(INAMS); GO FU7B; X:=D; CALL OPSYS(PRIRT)
173344             ELSE
173345                "TFREE"; CALL SMOUTTEXT;10;CALL SMSPACE
173351             FI; X:=CDFELT; CALL SMDYNAL; GO FU7B
173354             "TYES"; CALL SMOUTTEXT; CALL GBTINDX; GO FU7B
173360             A*5PRVTSIZE+APRVTTABLE=:X; T:=MBPRVTTABLE; *LDATX TXSVB
173365             A=:CBPRFLG
173366             IF A BIT BPRTMOUT THEN "TXON" ELSE "TXOFF" FI; CALL SMOUTTEXT
173374             IF CBPRFLG BIT BPCFIXED THEN "TYES" ELSE "TNO" FI; CALL SMOUTTEXT
173403    FU7B:    MIN CADDR
173404          OD; GO FAR CTRYAGAIN
173406          *)FILL
173444
173444    FUH:    X:="FUNTAB"                                   % HELP
173445            DO
173445               CALL SMCRLF
173446            WHILE X.S0><-1; CALL SMOUTTEXT; X+4
173454            OD; GO FAR CTRYAGAIN
173456
173456    FUE:    GO SMLEAVE                                    % EXIT
173457    RBUS
173464
173464    %=====================================================================
173464    %
173464    % SET-COLDSTART-MODE-FILE
173464    % RESET-COLDSTART-MODE-FILE
173464    %
173464    SUBR SSCLDMODE,RSCLDMODE
173464
173464    DISP -200
173464    INTEGER CBYTP,CHAR,CADDR,CCBYTP,CCADDR,CMXCHAR
173464    INTEGER POINTER LREG,LREG2
173464    PSID
173464
173464    INTEGER TXED1:='PARAMETERS TO THE ENTER-DIRECTORY COMMAND'
173511    INTEGER TXED2:='$WHEN ENTERING MAIN DIRECTORY: '
173531    INTEGER HNTIFILE:='S COLDSTART INPUT FILE: '
173546    INTEGER HNTOFILE:='S COLDSTART OUTPUT FILE: '
173563    INTEGER TOOLSTR:='$TOO LONG PARAMETER STRING'
173601
```

```
173601   STCHAR: X:=CADDR; A=:CHAR:=L=:"LREG":=CBYTP SHZ -1; X+A
173610           IF CBYTP>>=CMXCHAR THEN "TOOLSTR"; CALL SMOUTTEXT; GO SMLEAVE FI
173617           IF T:=CBYTP BIT "0" THEN
173622              CALL R1SAVE; A/\177400\/CHAR
173625           ELSE
173626              A:=CHAR SHZ 10
173630           FI; CALL W1SAVE; MIN CBYTP; GO LREG
173633   STFNAM: 0=:CCBYTP; A=:CCADDR:=L=:"LREG2"
173637           DO
173637              T:=CCADDR; X:=CCBYTP; *LBYT
173642              CALL STCHAR
173643           WHILE CHAR><##'
173647              MIN CCBYTP
173650           OD; GO LREG2
173652
173652   SSCLDMODE: L=:D; CALL SMENTER
173654           CALL 6MOPEN
173655           0=:CBYTP; 43=:CMXCHAR; "X9HNTCOMM"-BCSTA=:CADDR
173663           CALL SMCREAD
173664           IF A/\177=15 THEN
173670              "TXED1"; CALL SMOUTTEXT; "TXED2"; CALL SMOUTTEXT
173674              CALL SMGCOM
173675           FI; DO CALL SMCREAD WHILE A><15; CALL STCHAR OD
173703           ##'; CALL STCHAR; CALL SMBACKSP
173706           74=:CMXCHAR; 0=:CBYTP; "9IHENTMODE"-BCSTA=:CADDR
173714           "HNTIFILE"; CALL SMGPAR; CALL STFNAM
173717           0=:CBYTP; "9OHENTMODE"-BCSTA=:CADDR
173723           "HNTOFILE"; CALL SMGPAR; CALL STFNAM
173726           T:=1; GO FELLS
173730   RSCLDMODE: L=:D; CALL SMENTER
173732           CALL 6MOPEN; T:=0
173734   FELLS: "9MOCOLDSTART"-BCSTA=:X
173737           A:=T; CALL W1SAVE; CALL WXSAVE; CALL 6CLOSE
173743           GO SMLEAVE
173744   RBUS
173774
173774
173774   *OP2EN=*
173774   +OP2BG/OP2BG;173777;OP2EN     % LOWER LIMIT, UPPER LIMIT, FIRST FREE LOCATION
110003
110003   @EOF
110003
```

```
174001
174001   %==========================================================================
174001   %       S I N F
174001   %==========================================================================
174001
174001
174001   *ERRFL/;*<*+3;)ZERO;*+3/
110003   %==========================================================================
110003   %
110003   %  30.0        E R R O R   H A N D L I N G   ( E R R O R   P R O G R A M )
110003   %
110003   %==========================================================================
110003   SUBR ELEAV
110003   RBUS
110003
110003   %==========================================================================
110003   % 17.9        S I M I N B T
110003   %
110003   % SUBROUTINE TO SIMULATE INBT, CALLED FROM SRTERR
110003   %
110003   %%%%%% M U S T   N O T   C R O S S   A   P A G E   L I M I T %%%%%%%%%%%%%%%
110003   %
110003   % A=LOG.NO
110003   % RETURN: A=VALUE
110003
110003
110003   SUBR SIMINBT
110003
110003   INTEGER POINTER LREG; INTEGER XREG,BREG
110006
110006   SIMINBT: X=:XREG:=L=:"LREG"
110011           CALL LOGPH; IF =0 THEN CALL ERRFATAL FI; A:=:B=:BREG; *IOF
110017           DO                        % WHILE WAITING FOR ERRORS
110017                 CALL ERIOTRANS; GO IWAIT; *ION
110022                 X:=BREG=:B;=XREG; GO LREG
110026   IWAIT:        X:=RTREF; CALL WDATA; "STUPR"; *IRW MLEVB DP
110032                 MLEV; *MST PID; ION; IOF
110036           OD
110037   RBUS
110045   %==========================================================================
110045   % 30.1        S R T E R R
110045
110045   % ERROR RT-PROGRAM FOR REAL-TIME ERRORS
110045
110045   SUBR SRTERR
110045
110045   INTEGER RISIPAR:=(ERRILOG,NULL,"1"),ERRILOG:=500
110051   INTEGER ARRAY POINTER ERRARR:=RERNUM
110052
110052   SRTERR: "BFIELD"=:B; "STBEG"=:STPNT; "RISIPAR"; *MON 2RESR
110060           IF <0 THEN CALL ERRFATAL FI
110062           DO
110062                 FOR X:=0 TO 4 DO ERRILOG; CALL SIMINBT;.A=:ERRARR(X) OD
110073                 CALL RERR
110074           OD
110075   RBUS
110104
110104   %==========================================================================
110104   % 30.2        B E R R   R E R R   E R I N I T   E R P R I N T
110104   %
```

```
110104   %
110104   %       SUBROUTINE TO WRITE ERROR MESSAGES
110104
110104   SUBR BERR,RERR,XERPRINT
110104
110104   INTEGER ARRAY ERRDESC=?,ERRTEXT=?,ERATEXT=?,ERBTEXT=?
110104   INTEGER RUNERR:=' ERROR ',IN:=' IN ',AT:=' AT ',BPST:='BP.$'
110121   INTEGER APPR:='APPROACHING END OF ERROR SEGMENT'
110142   INTEGER FILLED:='ERROR LOG SEGMENT FILLED'
110157
110157   DISP -200; INTEGER BFLAG,ERTYPE,XRNUM,SAVSTATE; DOUBLE ADREG; PSID
110157
110157   BERR:  1; GO BRERR
110161   RERR:  "0"
110162   BRERR: L=:D; CALL EENTR; A=:BFLAG;
110165          IF A=1 THEN
110170             IF RERNUMB=#98 THEN T:=RN1 ELSE T:=1 FI
110177             A:=15; *MON 2SYCN; JMP * 1          % (BAD)
110202          FI
110202          IF RERNUM=#98 GO FAR TMALERR              % NO BACKGROUND PROCESS AVAILABLE
110206          CALL ECRLF; 40; CALL ETCO2
110211          IF RERNUM=-1 THEN "BPST"; CALL EOUTTEXT; GO ELEAVE FI
110220          IF BFLAG=0 THEN CALL EOPSYS(CLFORM) FI
110224          "RUNERR"; CALL EOUTTEXT; RERNUM; CALL ETCO2
110230          RERNUM; AD SH -10; D SH -10; A-60*12+D-60
110237          IF <0 OR >144 THEN 144 FI                 % FROM FTN LIBRARY
110244          IF =30 THEN                               % 24, INTERNAL INTERRUPT
110247                36+RN1; IF A=43 THEN A-5 FI
110255          FI; A=:XRNUM; ERRDESC(A)=:ERTYPE
110261          IF >=0 AND BFLAG=0 THEN
110264             "IN"; CALL EOUTTEXT; RRTPROG
110267             CALL EOPSYS(RTOUT)
110271          FI
110271          IF ERTYPE NBIT 6 THEN "AT"; CALL EOUTTEXT; RERPREG; CALL EOCTU FI
110300          IF ERRTEXT(XRNUM)><0 THEN
110303                X:=#; :=:A; CALL ETCO2; X=:A; CALL EOUTTEXT
110310          FI; CALL ECRLF
110311          IF ERATEXT(XRNUM)><0 THEN CALL EOUTTEXT; #: ; CALL ETCO2 FI
110317          IF ERTYPE SHZ -3/\3=1 THEN RN1; CALL EOCTU
110327          ELSE IF =2 THEN RN1; CALL EDECU FI
110335          FI; GO BYPAS; *)FILL
110363
110363   DISP -200; INTEGER SVTDVN,SVTYPRING; PSID
110363
110363   TMALERR: RN1; CALL LOGPH
110365          IF D=0 GO ELEAV
110367          X:=D; T:="TYPRING"; CALL XGTDFADDR; A=:SVTYPRING
110373          A BONE 5NORESERVE=:X.TYPRING; T:="TYPRING"; CALL XSTDFADDR
110377          TDVN=:SVTDVN; RN1=:TDVN
110403          "ERT98"; CALL EOUTTEXT; T:=RN1; A:=142; *MON 2SYCN; JMP * 1
110411          SVTDVN=:TDVN; SVTYPRING=:X.TYPRING; T:="TYPRING"; CALL XSTDFADDR
110417          GO ELEAV
110420   *)FILL
110430
110430   INTEGER POINTER OFLD:=OFLCK    % IN FILE SYSTEM
110431   INTEGER WSPAR:=LOGRSEG,LOGRSEG:=5LOGRSEG
110433   @DEC; INTEGER WARN:=3000,ERMAX:=4080; @OCT    % 4K SEGMENT
110435
110435   BYPAS: IF ERBTEXT(XRNUM)><0 THEN
110440                X:=#; :=:A; CALL ETCO2; X=:A
```

162113 | 17 | 17 |

30464 "14"
132112 ← P
74
74
46145 COSPO

Outside segment bounds

```
110444                  CALL EOUTTEXT; #: ; CALL ETCO2
110447            FI
110447            IF ERTYPE/\3=1 THEN RN2; CALL EOCTU
110456            ELSE IF =2 THEN RN2; CALL EDECU FI
110464            FI
110464            IF ERTYPE BIT 7 THEN
110467                  TTNO; CALL LOGPH
110471                  X:=A; T:="BSTATE"; CALL XGTDFADDR; A=:SAVSTATE
110475                  T:="BSTATE"; 5BCOMM; CALL XSTDFADDR
110500                  OFLD; CALL XLOCK; RN1; *MON 2ERMS
110504                  OFLD; CALL XUNLOCK; SAVSTATE; T:="BSTATE"; CALL XSTDFADDR
110511            FI; CALL ECRLF
110512            IF ERTYPE BIT 10 THEN       % SAVE ERROR INF.
110515                  T:=LOGRSEG; X:=0; CALL GET1L; CALL ERRFATAL; A=:X
110522                  IF =WARN THEN "APPR"; CALL EOUTTEXT
110527                  ELSE IF =ERMAX THEN "FILLED"; CALL EOUTTEXT
110535                       ELSE IF >ERMAX OR <5 GO OUT; A=:D:=0; T:=5; *RDIV ST
110550                         IF D><0 GO OUT
110552                  FI   FI
110552                  T:=LOGRSEG; XRNUM; CALL PUT1L; CALL ERRFATAL % ERROR NUMBER
110556                  X+1; RN1; CALL PUT1L; CALL ERRFATAL          % PARAM. 1
110562                  X+1; RN2; CALL PUT1L; CALL ERRFATAL          % PARAM. 2
110566                  CALL DATGT; AD=:ADREG; X+1; CALL PUT1L; CALL ERRFATAL
110573                  X+1; ADREG; D=:A; CALL PUT1L; CALL ERRFATAL  % DATE&CLOCK
110600                  X+1=:A:=0; CALL PUT1L; CALL ERRFATAL         % FIRST FREE LOC.
110605                  "WSPAR"; *MON 2WSEG
110607   OUT:     FI; GO ELEAVE
110610   *)FILL
110637
110637   %=========================================================================
110637   % PRINT-ERROR-LOG
110637   %
110637   INTEGER NOTI:='NOT INITIALIZED'
110647   INTEGER POINTER XLOGR:=LOGRSEGM,XERMAX:=ERMAX
110651
110651   DISP -200
110651            INTEGER INDXX,XNUM,ERPNT,AREG,ERTP
110651   PSID
110651
110651   XERPRINT: L=:D; CALL EENTR
110653            T:=XLOGR; X:=0; CALL GET1L; CALL ERRFATAL; A=:ERPNT
110660            IF <5 OR >XERMAX+5 THEN "NOTI"; CALL EOUTTEXT; CALL EOPSYS(OPCOR) FI
110673            5=:INDXX
110675            FOR INDXX STEP 5 TO ERPNT-5 DO
110702                  X:=INDXX+3; T:=XLOGR; CALL GET1L; CALL ERRFATAL; A=:AREG
110710                  X+1; T:=XLOGR; CALL GET1L; CALL ERRFATAL; A=:D:=AREG
110716                  CALL LDATE; 40; CALL ETCO
110721                  X:=INDXX; T:=XLOGR; CALL GET1L; CALL ERRFATAL; A=:X=:XNUM
110727                  IF >0 AND <114 THEN IF ERRTEXT(X)><0 THEN CALL EOUTTEXT FI;CALL ECRLF
110740                    ERRDESC(X)=:ERTP
110742                    IF ERATEXT(X)><0 THEN
110744                      CALL EOUTTEXT; #: ; CALL ETCO2
110747                      T:=XLOGR; X:=INDXX+1; CALL GET1L; CALL ERRFATAL; A=:T
110755                      IF ERTP SHZ -3/\3-1=0 THEN A:=T; CALL EOCTU
110764                      ELSE IF ERTP SHZ -3/\3-2=0 THEN A:=T; CALL EDECU; FI
110774                      FI
110774                    FI
110774                    IF ERBTEXT(XNUM)><0 THEN
110777                      X:=#; :=:A; CALL ETCO2; X=:A; CALL EOUTTEXT; #: ; CALL ETCO2
111006                      T:=XLOGR; X:=INDXX+2; CALL GET1L; CALL ERRFATAL; A=:T
```

```
111014                    IF ERTP/\3-1=0 THEN A:=T; CALL EOCTU
111022                    ELSE IF ERTP/\3-2=0 THEN A:=T; CALL EDECU; FI
111031                    FI
111031                  FI; CALL ECRLF; CALL ESCON; CALL ESCOF
111034                FI
111034          OD
111040          IF T:=TDVN><A:=1 THEN; *MON 2CLOS; 0
111046          FI; GO ELEAV
111047    *)FILL
111076
111076    @ICR
111076    % BIT 15: NO RT-PROG
111076    % BIT 8: SAVE INF. ON SEGMENT, BIT 7: FILE ERROR, BIT 6: NO ADDRESS
111076    % BIT 3-5: PARAM.1(IN A), BIT 0-2: PARAM.2(IN T): 1=OCTAL, 2=DECIMAL
111076    INTEGER ARRAY ERRDESC:=(
111076          10,0,0,100510,100011,0,310,310,311,0,
111110          0,0,0,10,10,10,10,10,10,0,
111122          100411,100511,100421,100411,0,10,100511,0,10,210,
111134          0,0,0,0,0,0,0,100512,511,511,
111146          400,0,10,100110,511,0,11,11,510,510,
111160          20,20,20,20,20,20,20,20,20,20,
111172          20,20,20,20,20,20,20,20,20,20,
111204          510,100510,100510,100511,100511,100511,100511,0,0,0,
111216          0,0,0,0,0,0,0,0,0,0,
111230          10,210,100111,100111,100,100,100,100,100,100520,
111242          100),
111243    ERRTEXT:=(
111243          ERT0,ERT1,ERT2,ERT3,ERT4,0,ERT6,ERT7,ERT8,ERT9,
111255          ERT10,ERT11,ERT12,ERT13,ERT14,ERT15,ERT16,ERT17,ERT18,ERT19,
111267          ERT20,ERT20,ERT22,ERT23,0,ERT25,ERT26,ERT27,ERT28,ERT29,
111301          ERT30,ERT31,ERT32,ERT33,ERT34,ERT35,ERT36,ERT37,ERT38,ERT39,
111313          ERT40,ERT41,ERT42,ERT37,ERT44,ERT45,ERT46,ERT47,ERT48,ERT49,
111325          ERT50,ERT50,ERT50,ERT50,ERT50,ERT50,ERT50,ERT50,ERT50,ERT50,
111337          ERT50,ERT50,ERT50,ERT50,ERT50,ERT50,ERT50,ERT50,ERT50,ERT50,
111351          ERT70,ERT71,ERT72,ERT73,ERT74,ERT75,ERT76,0,0,0,
111363          0,0,0,0,0,0,0,0,0,0,
111375          ERT90,ERT91,ERT92,ERT93,0,0,0,ERT97,ERT98,ERT99,
111407          ET100),
111410    ERATEXT:=(
111410          0,0,0,ERA3,ERA4,0,ERA6,ERA6,ERA6,0,
111422          0,0,0,ERA3,ERA3,ERA15,ERA15,ERA15,ERA15,0,
111434          ERA20,ERA21,ERA22,ERA20,0,ERA15,ERA20,0,ERA15,ERA6,
111446          0,0,0,0,0,0,0,ERA37,ERA38,ERA38,
111460          0,0,ERA42,ERA20,ERA38,0,ERA46,ERA46,ERA48,ERA48,
111472          ERA50,ERA50,ERA50,ERA50,ERA50,ERA50,ERA50,ERA50,ERA50,ERA50,
111504          ERA50,ERA50,ERA50,ERA50,ERA50,ERA50,ERA50,ERA50,ERA50,ERA50,
111516          ERA50,ERA71,ERA71,ERA71,ERA74,ERA71,ERA71,0,0,0,
111530          0,0,0,0,0,0,0,0,0,0,
111542          ERA90,ERA6,ERA92,ERA93,0,0,0,0,0,ERA99,
111554          0),
111555    ERBTEXT:=(                    % PARAMETER 2, IN T
111555          0,0,0,0,ERB4,0,0,0,ERA37,0,
111567          0,0,0,0,0,0,0,0,0,0,
111601          ERB20,ERB21,ERB22,ERB21,0,0,ERB20,0,0,0,
111613          0,0,0,0,0,0,0,ERA22,ERB38,ERB38,
111625          0,0,0,0,ERB38,0,ERB46,ERB47,0,0,
111637          0,0,0,0,0,0,0,0,0,0,
111651          0,0,0,0,0,0,0,0,0,0,
111663          0,0,0,ERB73,ERB74,ERB73,ERB76,0,0,0,
111675          0,0,0,0,0,0,0,0,0,0,
```

```
111707          0,0,ERB92,ERB92,0,0,0,0,0,0,
111721          0),
111722    ERT0:='ILL.MONITOR CALL',
111733    ERT1:='ILL.RT-PROG',
111741    ERT2:='ILL.PRIORITY',
111750    ERT3:='BAD MEMORY PAGE', ERA3:='PAGE NO.',
111765    ERT4:='INTERNAL INTERRUPT ON DIRECT TASK LEVEL',
112011          ERA4:='LEVEL (OCT.): ',ERB4:='IBITNO (OCT.): ',
112031    ERT6:='BATCH INPUT ERROR', ERA6:='FILE ERROR NO.',
112052    ERT7:='BATCH OUTPUT ERROR',
112064    ERT8:='BATCH SYSTEM ERROR',
112076    ERT9:='ILL. PARAM. IN CLOCK',
112111    ERT10:='ILL. PAR IN ABSET',
112122    ERT11:='ILL. PAR IN UPDAT',
112133    ERT12:='ILL. TIME PAR.',
112143    ERT13:='PAGE FAULT FOR NON-DEMAND',
112160    ERT14:='OUTSIDE SEGMENT BOUNDS',
112174    ERT15:='ILL. SEGMENT NO.',ERA15:='SEGMENT NO.',
112213    ERT16:='SEGMENT NOT LOADED',
112225    ERT17:='FIXING DEMAND',
112234    ERT18:='TOO MANY FIXED PAGES',
112247    ERT19:='TOO BIG SEGMENT',
112257    ERT20:='DISC TRANSFER ERROR',ERA20:='HARDW.DEV.NO.',ERB20:='UNIT',
112303                ERA21:='DISC ADDRESS',  ERB21:='STATUS',
112316    ERT22:='FALSE INTERRUPT', ERA22:='LEVEL(DEC.)',ERB22:='IDENT CODE(OCT.)',
112345    ERT23:='DEVICE ERROR',
112354    ERT25:='ALREADY FIXED',
112363    ERT26:='DEVICE TIMEOUT',
112373    ERT27:='ILL. PAR. IN CONCT',
112405    ERT28:='SPACE NOT AVAILABLE',
112417    ERT29:='FILE SYSTEM ERROR',
112430    ERT30:='DIVIDE BY ZERO',
112440    ERT31:='PERMIT VIOLATION',
112451    ERT32:='RING VIOLATION',
112461    ERT33:='HDLC DRIVER FATAL ERROR',
112475    ERT34:='ILL. INSTR',
112503    ERT35:='REENTRANT-FTN STACK ERROR',
112520    ERT36:='PRIV. INSTR',
112526    ERT37:='IOX ERROR', ERA37:='ADDRESS',
112537    ERT38:='MEMORY PARITY ERROR', ERA38:='PEA', ERB38:='PES',
112555    ERT39:='MEMORY OUT OF RANGE',
112567    ERT40:='POWER FAIL',
112575    ERT41:='ILLEGAL ERRORCODE IN ERMON',
112613    ERT42:='OVERLAPPING SEGMENTS',ERA42:='SEGMENTS',
112633    ERT44:='CORRECTED MEMORY ERROR',
112647    ERT45:='NOT DEMAND SEGMENTS',
112661    ERT46:='XMSG FATAL ERROR - INTERNAL ERROR OR INCONSISTENCY',
112713            ERA46:='XMSG ERROR CODE',ERB46:='PHYSICAL ADDRESS',
112734    ERT47:='XMSG USER ERROR',ERB47:='CALLING LEVEL',
112753    ERT48:='FALSE BEX INTERRUPT',ERA48:='BEX NO: ',
112772    ERT49:='REMOTE POWER FAIL INTERRUPT',
113010    ERT50:='USER ERROR', ERA50:='SUBERROR',
113023    ERT70:='BEX PARITY ERROR ',
113034    ERT71:='FALSE MPM4 INTERRUPT', ERA71:='BUSC NO',
113053    ERT72:='MPM4 POWER FAIL INTERRUPT',
113070    ERT73:='MPM4 MEMORY OUT OF RANGE', ERB73:='LOWER LIMIT',
113113    ERT74:='MPM4 MEMORY ERROR', ERA74:='LOCAL PES', ERB74:='LOCAL PEA',
113136    ERT75:='MPM4 PARITY ERROR',
113147    ERT76:='MPM4 WRITE PARITY ERROR', ERB76:='PORT CODE',
113170    ERT90:='FORTRAN RUNTIME ERROR',ERA90:='FORTRAN ERROR NUMBER',
```

```
113216   ERT91:='FORTRAN I/O ERROR',
113227   ERT92:='FATAL ERROR IN GPIB DRIVER. CONTROLLER STOPPED',
113257            ERA92:='CONTROLLER NUMBER', ERB92:='ERROR CODE',
113276   ERT93:='GPIB ERROR',
113304            ERA93:='FUNCTION',
113311   ERT94:='                                                  ',
113336   ERT95:='                                                  ',
113363   ERT96:='                                                  ',
113410   ERT97:='$TRYING TO START UN-INITIALIZED BACKGROUND PROGRAM',
113442   ERT98:='$$NO BACKGROUND PROCESS AVAILABLE$$',
113464   ERT99:='OCTOBUS ERROR',ERA99:='OCTOBUS ERROR CODE:',
113505   ET100:='FTN LIBRARY ERROR';
113516   @CR;
113516   RBUS
113516
113516   %=================================================================
113516   %       L D A T E
113516   %
113516   % OUTPUT DATE ON <TDVN> FROM A PACKED DATE/TIME DOUBLE WORD
113516   %
113516   % ENTRY:      AD=PACKED DATE/TIME
113516   %
113516   SUBR LDATE
113516
113516   DISP -200
113516           INTEGER POINTER LLRG
113516           INTEGER YARG,XARG,XDRG; DOUBLE DXADRG=XARG
113516           INTEGER AARG,DDRG; DOUBLE DADRG=AARG
113516           INTEGER REST
113516   PSID
113516
113516   INTEGER ARRAY MTAB:=(M1,M2,M3,M4,M5,M6,M7,M8,M9,M10,M11,M12,M12,M12)
113534   INTEGER ARRAY M1:=('JANUARY ')
113541   INTEGER ARRAY M2:=('FEBRUARY ')
113546   INTEGER ARRAY M3:=('MARCH ')
113552   INTEGER ARRAY M4:=('APRIL ')
113556   INTEGER ARRAY M5:=('MAY ')
113561   INTEGER ARRAY M6:=('JUNE ')
113564   INTEGER ARRAY M7:=('JULY ')
113567   INTEGER ARRAY M8:=('AUGUST ')
113573   INTEGER ARRAY M9:=('SEPTEMBER ')
113601   INTEGER ARRAY M10:=('OCTOBER ')
113606   INTEGER ARRAY M11:=('NOVEMBER ')
113613   INTEGER ARRAY M12:=('DECEMBER ')
113620
113620
113620   MTWOD:  T:=L=:"LLRG":=D=:XDRG; A=:YARG
113625           D:=0
113626           DO A-12; WHILE A>=0; D+1; OD
113632           A:=D=:REST+##0; CALL ETCO
113636           A:=REST*12=:D:=YARG-D+##0; CALL ETCO
113645           GO LLRG
113646
113646   LDATE:  T:=D; L=:D; CALL EENTR; T=:D; AD=:DADRG; 0=:XARG
113654           AD SHZ 4; A/\37; CALL MTWOD                % HOUR
113657           A:=##.; CALL ETCO
113661           AD:=DXADRG SHZ 6; CALL MTWOD               % MINUTE
113664           A:=##.; CALL ETCO
113666           AD:=DXADRG SHZ 6; CALL MTWOD               % SECOND
113671           A:=#   ; CALL ETCO2
```

```
113673          AARG SHZ -6/\17-1                          % MONTH
113677          IF A<0 OR A>14 THEN ##?; CALL ETCO ELSE MTAB(A); CALL EOUTTEXT FI
113711          AARG SHZ -1/\37; CALL MDECI                % DAY
113715          A:=#,. ; CALL ETCO2
113717          AARG SHZ -12+3636; CALL MDECI              % YEAR
113723          GO ELEAV
113724   RBUS
113741
113741   %=========================================================================
113741   %             M D E C I
113741   %
113741   % SUBROUTINE TO OUTPUT A DECIMAL NUMBER
113741   % LEADING ZEROES WILL BE SKIPPED
113741   %
113741   % ENTRY:      A=NUMBER
113741   %
113741   SUBR MDECI
113741
113741   DISP -200
113741   INTEGER NUMBR
113741   INTEGER FLAGR
113741   INTEGER CNT
113741   PSID
113741
113741   INTEGER ARRAY XPOT10:=(23420,1750,144,12,1)
113746
113746   MDECI: L=:D; CALL EENTR
113750          A=:NUMBR; 0=:FLAGR
113752          FOR X:=0 TO 4 DO
113756             0=:CNT
113757   LOOP:     IF NUMBR>=XPOT10(X) THEN
113763                NUMBR-XPOT10(X)=:NUMBR; MIN CNT; GO LOOP
113770             ELSE
113771                IF CNT><0 OR FLAGR><0 THEN
113775                   CNT+##0; CALL ETCO
114000                   1=:FLAGR
114002                FI
114002             FI
114002          OD; IF FLAGR=0 THEN ##0; CALL ETCO FI
114010          GO ELEAVE
114011   RBUS
114015
114015
114015   %=========================================================================
114015   % 30.3        E R L I S T
114015   %
114015   %  E R R O R   M E S S A G E   R O U T I N E
114015
114015
114015   % WRITE ERROR MESSAGE.
114015
114015   % A-REG: ERROR CODE
114015
114015   % RETURN
114015
114015
114015   @ICR
114015   INTEGER ARRAY ERSTB:=(ERR0,ERR0,ERS2,ERS3,ERS4,ERS5,ERR0,ERS7,ERR0,
114026                 ERR0,ERS12,ERR0,ERR0,ERR0,ERR0,ERR0,ERR0);
114036   INTEGER ARRAY ERTAB:=(ERR0, ERR1, ERR20,ERR2, ERR3, ERR4, ERR5, ERR6, ERR7, ERR8, ERR9,
```

```
114051                          ERR10,ERR11,ERR12,ERR13,FRR14,ERR15,ERR16,ERR17,ERR18,
114062                          ERR19,ERR21,FRR22,ERR23,ERR24,ERR25,ERR26,ERR27,
114072                          ERR28,ERR29,ERR30,ERR31,ERR32,ERR33,ERR34,ERR35,ERR36,
114103                          ERR37,ERR38,ERR39,ERR40,ERR41,ERR42,ERR43,ERR44,ERR45,
114114                          ERR46,ERR47,ERR48,ERR49,ERR50,ERR51,ERR52,ERR53,ERR54,
114125                          ERR55,ERR56,ERR57,ERR58,ERR59,ERR60,ERR61,ERR62,ERR63,
114136                          ERR64,ERR65,ERR66,ERR67,ERR68,ERR69,ERR70,ERR72,
114146                          ERR73,ERR74,ERR75,ERR76,ERR77,ERR78,ERR79,ERR80,ERR81,
114157                          ERR82,ERR83,ERR84,ERR85,ERR86,ERR87,ERR88,ERR89,ERR90,
114170                          ERR91,ERR92,ERR93,ERR94,ERR95,ERR96,ERR97,ERR98,ERR99,
114201                          RR100,RR101,RR102,RR103,RR104,RR105,RR106,RR107,RR108,
114212                          RR109,RR110,RR111,RR112,RR113,RR114,RR115,RR116,RR117,
114223                          RR118,RR119,RR120,RR121,RR122,RR123,RR124,RR125,RR126,
114234                          RR127,RR128,RR129,RR130,RR131,RR132,RR133,RR134,
114244                          RR135,RR136,RR137,RR138,RR139,RR140,RR141,RR142,
114254                          RR143,RR144,RR145,RR146,RR147,RR148,RR149,RR150,RR151,
114265                          RR152,RR153,RR154,RR155,RR156,RR157,RR158,RR159,RR160,
114276                          RR161,RR162,RR163,RR164,RR165,RR166,RR167,RR168,RR169,
114307                          RR170,RR171,RR172,RR173,RR174,RR175,RR176,RR177,RR178,
114320                          RR179,RR180,RR181,RR182,RR183,RR184,RR185,RR186,RR187,
114331                          RR188,RR189,RR190,RR191,RR192,RR193,RR194,RR195,RR196,
114342                          RR197,RR198,RR199,RR200,RR201,RR202,RR203,RR204,RR205,
114353                          RR206,RR207,RR208,RR209,RR210,RR211,RR212);
114362   @CR;
114362
114362   SUBR ERLIST
114362
114362   INTEGER ARRAY ERS2:=('BAD FILE NUMBER')
114372   INTEGER ARRAY ERS3:=('END OF FILE')
114400   INTEGER ARRAY ERS4:=('CARD READER ERROR (CARD READ)')
114417   INTEGER ARRAY ERS5:=('DEVICE NOT RESERVED')
114431   INTEGER ARRAY ERS7:=('CARD READER ERROR (CARD NOT READ)')
114452   INTEGER ARRAY ERS12:=('END OF DEVICE (TIMEOUT)')
114466
114466
114466   INTEGER ARRAY ERR0:=('ERROR CODE OUTSIDE RANGE')
114503   INTEGER ARRAY ERR1:=('ILLEGAL CHARACTER IN PARAMETER')
114523   INTEGER ARRAY ERR2:=('NOT DECIMAL NUMBER')
114535   INTEGER ARRAY ERR3:=('NOT OCTAL NUMBER')
114546   INTEGER ARRAY ERR4:=('YOU ARE NOT AUTHORIZED TO DO THIS')
114567   INTEGER ARRAY ERR5:=('DIRECTORY NOT ENTERED')
114602   INTEGER ARRAY ERR6:=('AMBIGUOUS DIRECTORY NAME')
114617   INTEGER ARRAY ERR7:=('NO SUCH DEVICE NAME')
114631   INTEGER ARRAY ERR8:=('AMBIGUOUS DEVICE NAME')
114644   INTEGER ARRAY ERR9:=('DIRECTORY ENTERED')
114655   INTEGER ARRAY ERR10:=('NO SUCH LOGICAL UNIT')
114670   INTEGER ARRAY ERR11:=('UNIT OCCUPIED')
114677   INTEGER ARRAY ERR12:=('MASTER BLOCK TRANSFER ERROR')
114715   INTEGER ARRAY ERR13:=('BIT FILE TRANSFER ERROR')
114731   INTEGER ARRAY FRR14:=('NO MORE TRACKS AVAILABLE')
114746   INTEGER ARRAY ERR15:=('DIRECTORY NOT ON SPECIFIED UNIT')
114766   INTEGER ARRAY ERR16:=('FILES OPEN ON THIS DIRECTORY')
115005   INTEGER ARRAY ERR17:=('MAIN DIRECTORY NOT LAST ONE RELEASED')
115030   INTEGER ARRAY ERR18:=('NO MAIN DIRECTORY')
115041   INTEGER ARRAY ERR19:=('TOO LONG PARAMETER')
115053   INTEGER ARRAY ERR20:=('NO SUCH PAGE')
115062   INTEGER ARRAY ERR21:=('AMBIGUOUS USER NAME')
115074   INTEGER ARRAY FRR22:=('NO SUCH USER NAME')
115105   INTEGER ARRAY ERR23:=('NO SUCH USER NAME IN MAIN DIRECTORY')
115127   INTEGER ARRAY ERR24:=('ATTEMPT TO CREATE TOO MANY USERS')
```

```
115150   INTEGER ARRAY ERR25:=('USER ALREADY EXISTS')
115162   INTEGER ARRAY ERR26:=('USER HAS FILES')
115172   INTEGER ARRAY ERR27:=('USER IS ENTERED')
115202   INTEGER ARRAY ERR28:=('NOT SO MUCH SPACE UNRESERVED IN DIRECTORY')
115227   INTEGER ARRAY ERR29:=('RESERVED SPACE ALREADY USED')
115245   INTEGER ARRAY ERR30:=('NO SUCH FILE NAME')
115256   INTEGER ARRAY ERR31:=('AMBIGUOUS FILE NAME')
115270   INTEGER ARRAY ERR32:=('WRONG PASSWORD')
115300   INTEGER ARRAY ERR33:=('USER ALREADY ENTERED')
115313   INTEGER ARRAY ERR34:=('NO USER ENTERED')
115323   INTEGER ARRAY ERR35:=('FRIEND ALREADY EXISTS')
115336   INTEGER ARRAY ERR36:=('NO SUCH FRIEND')
115346   INTEGER ARRAY ERR37:=('ATTEMPT TO CREATE TOO MANY FRIENDS')
115370   INTEGER ARRAY ERR38:=('ATTEMPT TO CREATE YOURSELF AS FRIEND')
115413   INTEGER ARRAY ERR39:=('CONTINUOUS SPACE NOT AVAILABLE')
115433   INTEGER ARRAY ERR40:=('NOT DIRECTORY ACCESS')
115446   INTEGER ARRAY ERR41:=('SPACE NOT AVAILABLE TO EXPAND FILE')
115470   INTEGER ARRAY ERR42:=('SPACE ALREADY ALLOCATED')
115504   INTEGER ARRAY ERR43:=('NO SPACE IN DEFAULT DIRECTORIES')
115524   INTEGER ARRAY ERR44:=('NO SUCH FILE VERSION')
115537   INTEGER ARRAY ERR45:=('NO MORE PAGES AVAILABLE FOR THIS USER')
115562   INTEGER ARRAY ERR46:=('FILE ALREADY EXISTS')
115574   INTEGER ARRAY ERR47:=('ATTEMPT TO CREATE TOO MANY FILES')
115615   INTEGER ARRAY ERR48:=('OUTSIDE DEVICE LIMITS')
115630   INTEGER ARRAY ERR49:=('NO PREVIOUS VERSION')
115642   INTEGER ARRAY ERR50:=('FILE NOT CONTINUOUS')
115654   INTEGER ARRAY ERR51:=('FILE TYPE ALREADY DEFINED')
115671   INTEGER ARRAY ERR52:=('NO SUCH ACCESS CODE')
115703   INTEGER ARRAY ERR53:=('FILE ALREADY OPEN')
115714   INTEGER ARRAY ERR54:=('NOT WRITE ACCESS')
115725   INTEGER ARRAY ERR55:=('ATTEMPT TO OPEN TOO MANY FILES')
115745   INTEGER ARRAY ERR56:=('NOT WRITE AND APPEND ACCESS')
115763   INTEGER ARRAY ERR57:=('NOT READ ACCESS')
115773   INTEGER ARRAY ERR58:=('NOT READ, WRITE AND COMMON ACCESS')
116014   INTEGER ARRAY ERR59:=('NOT READ AND WRITE ACCESS')
116031   INTEGER ARRAY ERR60:=('NOT READ AND COMMON ACCESS')
116047   INTEGER ARRAY ERR61:=('FILE RESERVED BY ANOTHER USER')
116066   INTEGER ARRAY ERR62:=('FILE ALREADY OPEN FOR WRITE BY YOU')
116110   INTEGER ARRAY ERR63:=('NO SUCH USER INDEX')
116122   INTEGER ARRAY ERR64:=('NOT APPEND ACCESS')
116133   INTEGER ARRAY ERR65:=('ATTEMPT TO OPEN TOO MANY MASS STORAGE FILES')
116161   INTEGER ARRAY ERR66:=('ATTEMPT TO OPEN TOO MANY FILES')
116201   INTEGER ARRAY ERR67:=('NOT OPEN FOR SEQUENTIAL WRITE')
116220   INTEGER ARRAY ERR68:=('NOT OPEN FOR SEQUENTIAL READ')
116237   INTEGER ARRAY ERR69:=('NOT OPEN FOR RANDOM WRITE')
116254   INTEGER ARRAY ERR70:=('NOT OPEN FOR RANDOM READ')
116271   INTEGER ARRAY ERR72:=('FILE NUMBER OUT OF RANGE')
116306   INTEGER ARRAY ERR73:=('FILE NUMBER ALREADY USED')
116323   INTEGER ARRAY ERR74:=('NO MORE BUFFER SPACE')
116336   INTEGER ARRAY ERR75:=('NO FILE OPEN WITH THIS NUMBER')
116355   INTEGER ARRAY ERR76:=('NOT MASS STORAGE FILE')
116370   INTEGER ARRAY ERR77:=('FILE USED FOR WRITE')
116402   INTEGER ARRAY ERR78:=('FILE USED FOR READ')
116414   INTEGER ARRAY ERR79:=('FILE ONLY OPEN FOR SEQUENTIAL READ OR WRITE')
116442   INTEGER ARRAY ERR80:=('NO SCRATCH FILE OPEN')
116455   INTEGER ARRAY ERR81:=('FILE NOT RESERVED BY YOU')
116472   INTEGER ARRAY ERR82:=('TRANSFER ERROR')
116502   INTEGER ARRAY ERR83:=('FILE ALREADY RESERVED')
116515   INTEGER ARRAY ERR84:=('NO SUCH BLOCK')
116524   INTEGER ARRAY ERR85:=('SOURCE AND DESTINATION EQUAL')
```

```
116543   INTEGER ARRAY ERR86:=('ILLEGAL ON TAPE DEVICE')
116557   INTEGER ARRAY ERR87:=('END OF TAPE')
116565   INTEGER ARRAY ERR88:=('DEVICE UNIT RESERVED FOR SPECIAL USE')
116610   INTEGER ARRAY ERR89:=('MAIN DIRECTORY MUST BE DEFAULT')
116630   INTEGER ARRAY ERR90:=('NOT LAST FILE ON TAPE')
116643   INTEGER ARRAY ERR91:=('NOT TAPE DEVICE')
116653   INTEGER ARRAY ERR92:=('ILLEGAL ADDRESS REFERENCE IN MONITOR CALL')
116700   INTEGER ARRAY ERR93:=('SOURCE EMPTY')
116707   INTEGER ARRAY ERR94:=('FILE ALREADY OPEN BY ANOTHER USER')
116730   INTEGER ARRAY ERR95:=('FILE ALREADY OPEN FOR WRITE BY ANOTHER USER')
116756   INTEGER ARRAY ERR96:=('MISSING PARAMETER')
116767   INTEGER ARRAY ERR97:=('TWO PAGES MUST BE LEFT UNRESERVED')
117010   INTEGER ARRAY ERR98:=('NO ANSWER FROM REMOTE COMPUTER')
117030   INTEGER ARRAY ERR99:=('DEVICE CANNOT BE RESERVED')
117045   INTEGER ARRAY RR100:=('OVERFLOW IN READ')
117056   INTEGER ARRAY RR101:=('DMA ERROR')
117063   INTEGER ARRAY RR102:=('BAD DATABLOCK')
117072   INTEGER ARRAY RR103:=('CONTROL/MODUS WORD ERROR')
117107   INTEGER ARRAY RR104:=('PARITY ERROR')
117116   INTEGER ARRAY RR105:=('LRC ERROR')
117123   INTEGER ARRAY RR106:=('DEVICE ERROR (READ-LAST-STATUS TO GET STATUS)')
117152   INTEGER ARRAY RR107:=('DEVICE BUFFER OF REQUESTED SIZE NOT AVAILABLE')
117201   INTEGER ARRAY RR108:=('ILLEGAL MASS STORAGE UNIT NUMBER')
117222   INTEGER ARRAY RR109:=('ILLEGAL PARAMETER')
117233   INTEGER ARRAY RR110:=('WRITE-PROTECT VIOLATION')
117247   INTEGER ARRAY RR111:=('ERROR DETECTED BY READ AFTER WRITE')
117271   INTEGER ARRAY RR112:=('NO EOF MARK FOUND')
117302   INTEGER ARRAY RR113:=('CASSETTE NOT IN POSITION')
117317   INTEGER ARRAY RR114:=('ILLEGAL FUNCTION CODE')
117332   INTEGER ARRAY RR115:=('TIME OUT (NO DATABLOCK FOUND)')
117351   INTEGER ARRAY RR116:=('PAPER FAULT')
117357   INTEGER ARRAY RR117:=('DEVICE NOT READY')
117370   INTEGER ARRAY RR118:=('DEVICE ALREADY RESERVED')
117404   INTEGER ARRAY RR119:=('NOT PERIPHERAL FILE')
117416   INTEGER ARRAY RR120:=('NO SUCH QUEUE ENTRY')
117430   INTEGER ARRAY RR121:=('NOT SO MUCH SPACE LEFT')
117444   INTEGER ARRAY RR122:=('NO SPOOLING FOR THIS DEVICE')
117462   INTEGER ARRAY RR123:=('NO SUCH QUEUE')
117471   INTEGER ARRAY RR124:=('QUEUE EMPTY')
117477   INTEGER ARRAY RR125:=('QUEUE FULL')
117505   INTEGER ARRAY RR126:=('NOT LAST USED BY YOU')
117520   INTEGER ARRAY RR127:=('NO SUCH CHANNEL NAME')
117533   INTEGER ARRAY RR128:=('NO REMOTE CONNECTION')
117546   INTEGER ARRAY RR129:=('ILLEGAL CHANNEL')
117556   INTEGER ARRAY RR130:=('CHANNEL ALREADY RESERVED ON REMOTE COMPUTER')
117604   INTEGER ARRAY RR131:=('NO REMOTE FILE PROCESSOR')
117621   INTEGER ARRAY RR132:=('FORMATTING ERROR')
117632   INTEGER ARRAY RR133:=('INCOMPATIBLE DEVICE SIZES')
117647   INTEGER ARRAY RR134:=('REMOTE PROCESSOR NOT AVAILABLE')
117667   INTEGER ARRAY RR135:=('TAPE FORMAT ERROR')
117700   INTEGER ARRAY RR136:=('BLOCK COUNT ERROR')
117711   INTEGER ARRAY RR137:=('VOLUME NOT ON SPECIFIED UNIT')
117730   INTEGER ARRAY RR138:=('NOT DELETED RECORD')
117742   INTEGER ARRAY RR139:=('DEVICE ERROR')
117751   INTEGER ARRAY RR140:=('ERROR IN OBJECT ENTRY')
117764   INTEGER ARRAY RR141:=('ODD NUMBER OF BYTES (RIGHT BYTE IN LAST WORD INSIGNIFICANT)')
120022   INTEGER ARRAY RR142:=('ERROR IN BACKSPACE/FORWARD SPACE PRINT')
120046   INTEGER ARRAY RR143:=('BLOCK FORMAT ERROR')
120060   INTEGER ARRAY RR144:=('OVERFLOW IN WRITE')
120071   INTEGER ARRAY RR145:=('ILLEGAL DEVICE TYPE')
```

```
120103   INTEGER ARRAY RR146:=('SEGMENT NOT CONTIGUOUSLY FIXED')
120123   INTEGER ARRAY RR147:=('SEGMENT NOT FIXED')
120134   INTEGER ARRAY RR148:=('APPROACHING END OF ACCOUNTING FILE')
120156   INTEGER ARRAY RR149:=('ACCOUNTING FILE FULL')
120171   INTEGER ARRAY RR150:=('NO MORE UNUSED SPOOLING FILES AVAILABLE')
120215   INTEGER ARRAY RR151:=('INCONSISTENT DIRECTORY')
120231   INTEGER ARRAY RR152:=('OBJECT ENTRY NOT USED')
120244   INTEGER ARRAY RR153:=('USER DOES NOT EXIST')
120256   INTEGER ARRAY RR154:=('DIRECTORY NOT RESERVED')
120272   INTEGER ARRAY RR155:=('NOT A MULTIPLE OF HARDWARE BLOCK SIZE')
120315   INTEGER ARRAY RR156:=('NOT INDEXED FILE')
120326   INTEGER ARRAY RR157:=('ILLEGAL FLOPPY FORMAT')
120341   INTEGER ARRAY RR158:=('FILE NOT OPEN')
120350   INTEGER ARRAY RR159:=('FILE ALREADY OPEN FOR READ OR WRITE BY YOU')
120376   INTEGER ARRAY RR160:=('USER DOES NOT EXIST IN THE SAME MAIN DIRECTORY AS YOU')
120431   INTEGER ARRAY RR161:=('FILE ACCESS REENTRANT SEGMENTS NOT LOADED')
120456   INTEGER ARRAY RR162:=('ILLEGAL ACCESS CODE FOR REMOTE FILE')
120500   INTEGER ARRAY RR163:=('FILE ACCESS CONNECTION ABORTED BY FILE SERVER')
120527   INTEGER ARRAY RR164:=('FILE ACCESS CONNECTION ABORTED BY FILE SERVER ADMINISTRATOR')
120565   INTEGER ARRAY RR165:=('NO ANSWER FROM REMOTE SYSTEM, FILE ACCESS CONNECTION ABORTED')
120624   INTEGER ARRAY RR166:=('FILE ACCESS INITIALIZE FAILED')
120643   INTEGER ARRAY RR167:=('UNKNOWN REMOTE SYSTEM NAME')
120661   INTEGER ARRAY RR168:=('FILE ACCESS PROTOCOL ERROR, CONNECTION ABORTED')
120711   INTEGER ARRAY RR169:=('FILE ACCESS INTERNAL ERROR, CALL NOT VALID IN CURRENT STATE')
120747   INTEGER ARRAY RR170:=('ILLEGAL RANGE OF LAMU ID')
120764   INTEGER ARRAY RR171:=('LAMU IN USE')
120772   INTEGER ARRAY RR172:=('LAMU TABLE FULL')
121002   INTEGER ARRAY RR173:=('ILLEGAL RT PROGRAM')
121014   INTEGER ARRAY RR174:=('MAXIMUM NUMBER OF LAMUS PR. RT PROGRAM REACHED')
121044   INTEGER ARRAY RR175:=('LAMU NOT CONNECTED')
121056   INTEGER ARRAY RR176:=('NO LAMU AREA BIG ENOUGH')
121072   INTEGER ARRAY RR177:=('LAMU NOT DEFINED')
121103   INTEGER ARRAY RR178:=('ILLEGAL LOGICAL PAGE NUMBER')
121121   INTEGER ARRAY RR179:=('LOGICAL LAMU OVERLAP')
121134   INTEGER ARRAY RR180:=('CAN ONLY LOG IN ON A MAIN DIRECTORY')
121156   INTEGER ARRAY RR181:=('DIRECTORY INDEX TOO LARGE')
121173   INTEGER ARRAY RR182:=('OBJECT INDEX TOO LARGE')
121207   INTEGER ARRAY RR183:=('RECOVER/DUMP OF 2-BANK PROGRAMS NOT ALLOWED IN NORD-NET')
121243   INTEGER ARRAY RR184:=('WARNING; 2-BANK PROG. FILE, BUT SEGMENT IS ONLY 1-BANK')
121277   INTEGER ARRAY RR185:=('WARNING; NO SUCH PAGE IN DATA BANK, PROGRAM STARTS AS 1-BANK')
121336   INTEGER ARRAY RR186:=('THE SPECIFIED DEVICE IS NOT A TERMINAL')
121362   INTEGER ARRAY RR187:=('NO TERMINATION HANDLING DEFINED')
121402   INTEGER ARRAY RR188:=('NO MORE REMOTE FILE ACCESS DATA SEGMENTS AVAILABLE')
121434   INTEGER ARRAY RR189:=('INPUT WHILE ESCAPE/LOCAL OFF IS ILLEGAL')
121460   INTEGER ARRAY RR190:=('T.A.D PROTOCOLL ERROR, ILLEGAL OR INCONSISTENT MESSAGE')
121514   INTEGER ARRAY RR191:=('TERMINAL LINE IS NOT CONNECTED')
121534   INTEGER ARRAY RR192:=('ILLEGAL COMBINATION OF DENTE AND DTUSE BITS')
121562   INTEGER ARRAY RR193:=('DIRECTORY NOT RESERVED BY YOU')
121601   INTEGER ARRAY RR194:=('WRONG PROJECT PASSWORD')
121615   INTEGER ARRAY RR195:=('FILE ACCESS TRANSPORT LAYER ERROR, ALL CONNECTIONS ABORTED')
121653   INTEGER ARRAY RR196:=('FILE ACCESS INTERNAL ERROR, INVALID PARAMETER VALUE')
121705   INTEGER ARRAY RR197:=('FILE ACCESS NOT RUNNING OR CRASHED, ALL CONNECTIONS ABORTED')
121743   INTEGER ARRAY RR198:=('WRONG FORMAT IN FILE')
121756   INTEGER ARRAY RR199:=('DIRECTORY ALREADY RESERVED FOR SPECIAL USE')
122004   INTEGER ARRAY RR200:=('UNKNOWN USER-CONTROL CODE')
122021   INTEGER ARRAY RR201:=('NO SERVICE IS AVAILABLE FOR THIS CODE')
122044   INTEGER ARRAY RR202:=('PLEASE TERMINATE CURRENT SERVICE BEFORE REQUESTING NEW SERVICE')
122104   INTEGER ARRAY RR203:=('THIS CODE IS ONLY LEGAL WITHIN A SERVICE')
122131   INTEGER ARRAY RR204:=('REMOTE FILE SERVER IS NOT AVAILABLE')
122153   INTEGER ARRAY RR205:=('ND-100 PANEL CLOCK INCORRECT')
```

```
122172   INTEGER ARRAY RR206:=('BLOCK SIZE TOO BIG FOR BUFFER PREVIOUSLY OBTAINED')
122223   INTEGER ARRAY RR207:=('ILLEGAL SEGMENT NAME')
122236   INTEGER ARRAY RR208:=('NOT OCTAL NUMBER')
122247   INTEGER ARRAY RR209:=('NOT CONTIGUOUS FILE')
122261   INTEGER ARRAY RR210:=('AMBIGUOUS COMMAND')
122272   INTEGER ARRAY RR211:=('PROTECTED COMMAND')
122303   INTEGER ARRAY RR212:=('AMBIGUOUS SUBSYSTEM')
122315
122315   DISP -200; INTEGER AREG; PSID
122315   DISP 30; INTEGER IDERO=DERO; PSID
122315
122315   ERLIST: L=:D
122316           IF T:=CRTREF.ACTPRI BIT 5BACKGR THEN
122322              CALL EENTR; A=:TTIFIELD.IDERO=:RERNUM
122326           ELSE
122327              X:=X.RTDLGADDR; T:=0; A=:D; *LDATX
122333              T:="DF2".SSREF; X:=D; L=:D; CALL 9ERR(#29); D=:P
122342           FI
122342           A=:AREG; T:=1; A:=14; *MON 2SYCN; JMP * 1            % (BAD)
122347           IF UEFLG NBIT 5UEIE THEN                    % DISABLED ERROR MESSAGE
122352              CALL ECRLF
122353              IF AREG<ERO  AND A>0 THEN
122361                 A=:X; X:=ERSTB(X)
122363              ELSE
122364                 IF A<ERM AND A>0 THEN A-ERO=:X ELSE X:=0 FI
122375                 X:=ERTAB(X)
122376              FI
122376              IF "ERRO"=X THEN
122401                 A:=##"; CALL ETCO; AREG; CALL EOCTU; A:=##"; CALL ETCO
122407                 A:=## ; CALL ETCO
122411              FI; A:=X; CALL EOUTTEXT; CALL ECRLF
122414           FI; GO ELEAVE
122415
122415   RBUS
122433
122433
122433   %=========================================================================
122433   % 19.8        3 O U T T E X T
122433
122433   %       SUBROUTINE TO OUTPUT TEXT ON TELETYPE
122433   %       A-REG. POINTS TO STRING TERMINATED WITH'
122433   %       THE VARIABLE "TEXTADR" IS MOVED TO THE LOCATION AFTER THE STRING
122433
122433   SUBR EOUTTEXT
122433
122433   DISP -200; INTEGER PNT,XREG; PSID
122433
122433   EOUTTEXT: L=:D; CALL EENTR; A=:PNT; X=:XREG
122437           FOR X:=0 TO 1000 DO
122443                 T:=PNT; *LBYT
122445                 WHILE ><##' AND A NBIT 7
122452                 IF A=##$ THEN CALL ECRLF ELSE CALL ETCO FI
122460           OD; A:=X SHZ -1+1+PNT=:TEXTADR
122467           X:=XREG; GO ELEAVE
122471   RBUS
122476
122476   %=========================================================================
122476   % 19.10       3 O C T U   3 D E C U   D T D E C
122476
122476   %       SUBROUTINES FOR INTEGER OUTPUT; VALUE IN A-REG.
```

```
122476
122476   SUBR EOCTU,EDECU,EDTDEC
122476
122476   @DEC
122476   DATA(10000,1000,100,10,1); INTEGER ARRAY CONST(0)
122503   @OCT
122503
122503   DISP -200; DOUBLE DVAL; INTEGER VALUE,NULLFLAG; PSID
122503
122503   CHBLANK: IF =0 AND X><-1 AND T:=NULLFLAG=0 THEN
122512                40
122513            ELSE
122514                A+60; MIN NULLFLAG
122516            FI
122516            EXIT
122517
122517   EOCTU: L=:D; CALL EENTR; AD SHZ -17=:DVAL; 40; CALL ETCO; 0=:NULLFLAG
122526          FOR X:=-6 DO DVAL; CALL CHBLANK; CALL ETCO
122532                DVAL SH 20 SHZ -15=:DVAL
122536          OD; GO ELEAVE
122540
122540   EDTDEC: L=:D; CALL EENTR; A=:VALUE; GO DOUT
122544   EDECU: L=:D; CALL EENTR; A=:X; 40;CALL ETCO
122551          IF X<0 THEN X-; ##- ELSE 40 FI; CALL ETCO; X=:VALUE
122561   DOUT:  0=:NULLFLAG
122562          FOR X:=-5 DO VALUE=:D; A:=0; T:=CONST(X); *RDIV ST
122570                T:=D=:VALUE; CALL CHBLANK; CALL ETCO
122574          OD; GO ELEAVE
122576   RBUS
122602
122602   %==========================================================================
122602   % 19.11       3 C R L F   3 T C O   3 T C I   3 T C O 2   W I N B
122602   %
122602   %       T E R M I N A L   I / O
122602
122602   %TT I/O SUBROUTINES
122602
122602   SUBR ECRLF,ETCO,ETCO2
122602
122602   ECRLF: L=:D; CALL EENTR; 15; CALL ETCO; 12; CALL ETCO; GO ELEAVE
122611   ETCO2: L=:D; CALL EENTR; A=:X SHZ -10; CALL ETCO
122616          377/\X; CALL ETCO; GO ELEAVE
122622   ETCO:  L=:D; T:=TDVN; CALL SETPARITY; *MON 2OUTB; 0; COPY SD DP
122630
122630   RBUS
122641
122641   %==========================================================================
122641   % 19.13       3 E N T E R   3 L E A V E   3 L E V 2   L E A V X   L E A V 3
122641
122641   %SUBROUTINES FOR SUBROUTINE EENTR AND ELEAVE
122641   SUBR EENTR,ELEAVE,ELEV2,ELEAX,ELEA3
122641   INTEGER STKERR:='STACK ERROR'
122647   DISP 0; DOUBLE ADSV; REAL V0SV,V1SV,V2SV; PSID
122647
122647   EENTR: TAD=:SAVTAD; X=:INDEX
122651          IF STPNT-"STEND">=0 GO ERR
122654          X=:A; AD=:STPNT.ADSV
122657          FV0=:X.V0SV; FV3=:X.V1SV; FV6=:X.V2SV
122665          X+STDELTA=:STPNT; SAVTAD; X:=INDEX; EXIT
122672
```

```
122672   ELEA3: TAD=:SAVTAD; X=:INDEX; STPNT-STDELTA=:STPNT=:X
122700          IF A-"STBEG"<0 GO ERR
122702          MIN X.S1; MIN X.S1; GO LEV
122705   ELEAX: TAD=:SAVTAD; T:=X; STPNT-STDELTA=:STPNT=:X
122713          IF A-"STBEG"<0 GO ERR
122715          MIN X.S1; T=:X.S0; GO LEV
122720   ELEA2: TAD=:SAVTAD; STPNT-STDELTA=:STPNT=:X
122725          IF A-"STBEG"<0 GO ERR; MIN X.S1; GO LEV
122731   ELEAVE: TAD=:SAVTAD; STPNT-STDELTA=:STPNT=:X
122736          IF A-"STBEG"<0 GO ERR
122740   LEV:   X.V0SV=:FV0; X.V1SV=:FV3; X.V2SV=:FV6
122746          X.ADSV; A=:X; D=:L; SAVTAD; EXIT
122753   ERR:   "STBEG"=:STPNT; "STKERR"; CALL EOUTTEXT; CALL EOPSYS(OPCOR)
122761   RBUS
122766
122766
122766   %==========================================================================
122766   % 19.14        DATGT
122766   %
122766   %             GET DATE
122766   %             FIND CURRENT TIME AND DATE AND PACK
122766   %             RETURN - A&D-REG: PACKED TIME AND DATE
122766
122766   SUBR DATGT
122766
122766   INTEGER BUNIT,SECON,MINUT,HOUR,DAY,MONT,YEAR
122775   INTEGER PARAM
122776   INTEGER REGT
122777
122777   DATGT: T=:REGT; A:="BUNIT"=:PARAM:="PARAM"; *MON 2CLOC
123004          A:=YEAR-3636=:D; AD SHZ 4
123010          T:=MONT; D+T; AD SHZ 5
123013          T:=DAY; D+T; AD SHZ 5
123016          T:=HOUR; D+T; AD SHZ 6
123021          T:=MINUT; D+T; AD SHZ 6
123024          T:=SECON; D+T; T:=REGT
123027          EXIT
123030   RBUS
123033
123033
123033
123033   *EESYS=*
123033   *ERRFL/ERRFL;123777;EESYS    % LOWER LIMIT, UPPER LIMIT, FIRST FREE LOCATION
110003   *EESYS/
123033
123033   @EOF
123033
```