SINTRAN III
REAL TIME LOADER

System Documentation

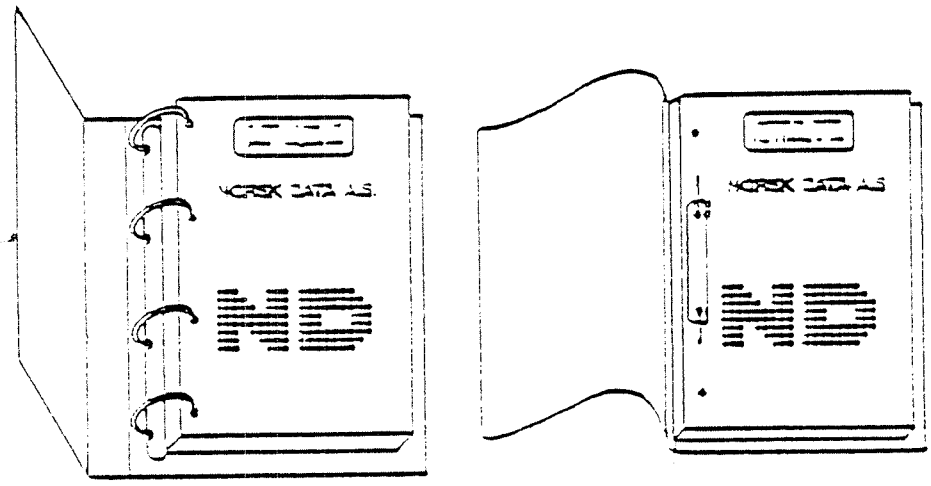
SINTRAN III
REAL TIME LOADER

System Documentation

This manual is in loose leaf form for ease of updating. Old pages may be removed and new pages easily inserted if the manual is revised.

The loose leaf form also allows you to place the manual in a ring binder (A) for greater protection and convenience of use. Ring binders with 4 rings corresponding to the holes in the manual may be ordered in two widths, 30 mm and 40 mm. Use the order form below.

The manual may also be placed in a plastic cover (B). This cover is more suitable for manuals of less than 100 pages than for large manuals. Plastic covers may also be ordered below.



A Ring Binder B Plastic Cover

Please send your order to the local ND office or (in Norway) to:

Documentation Department
Norsk Data A.S
P.O. Box 4, Lindeberg gård
Oslo 10

ORDER FORM

I would like to order

..... Ring Binders, 30 mm, at nkr 20,- per binder

..... Ring Binders, 40 mm, at nkr 25,- per binder

..... Plastic Covers at nkr 10,- per cover

Name .....
Company .....
Address .....
.....
City .....

REVISION RECORD

Revision	Notes
09/76	Original Printing
08/77	Revised Edition, superseding the original printing

SINTRAN III Real Time Loader – System Documentation
Publication No. ND-60.072.02

NORSK DATA A.S.

Lørenveien 57, Postboks 163 Økern, Oslo 5, Norway

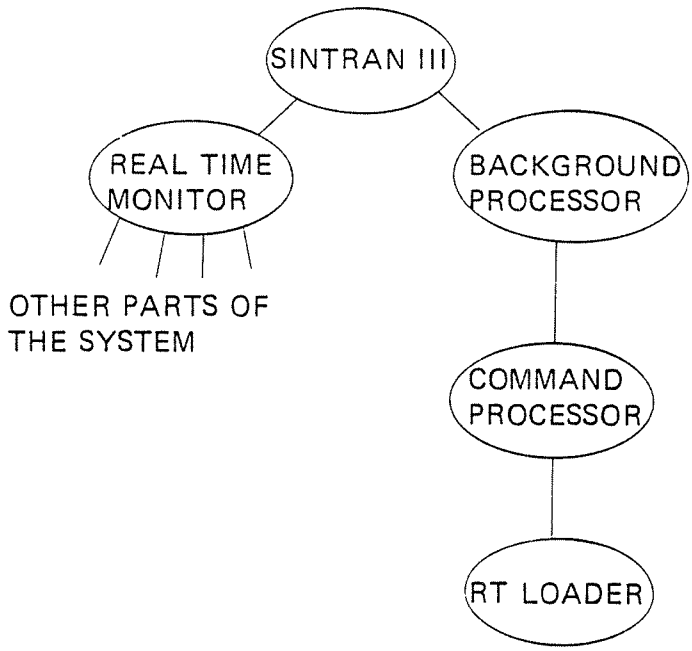
TABLE OF CONTENTS

\+ + +

1 INTRODUCTION

The SINTRAN III Real Time Loader (hereafter called the RT Loader) is a subsystem included in all SINTRAN III mass storage systems. This manual will describe the internal architecture of the RT Loader. For details about the commands available refer to the manuals: SINTRAN III Real Time Loader (ND-60.051) and SINTRAN III Users Guide (ND-60.050).

1.1 *THE RT LOADER'S PLACE IN THE SINTRAN III SYSTEM*



The RT Loader is a subsystem which can be activated by the SINTRAN III command @RT-LOADER.

1.2 *MEMORY LAYOUT*

With respect to the RT Loader's layout in memory, the RT Loader consists of the following three parts:

1. The RT Loader segment.

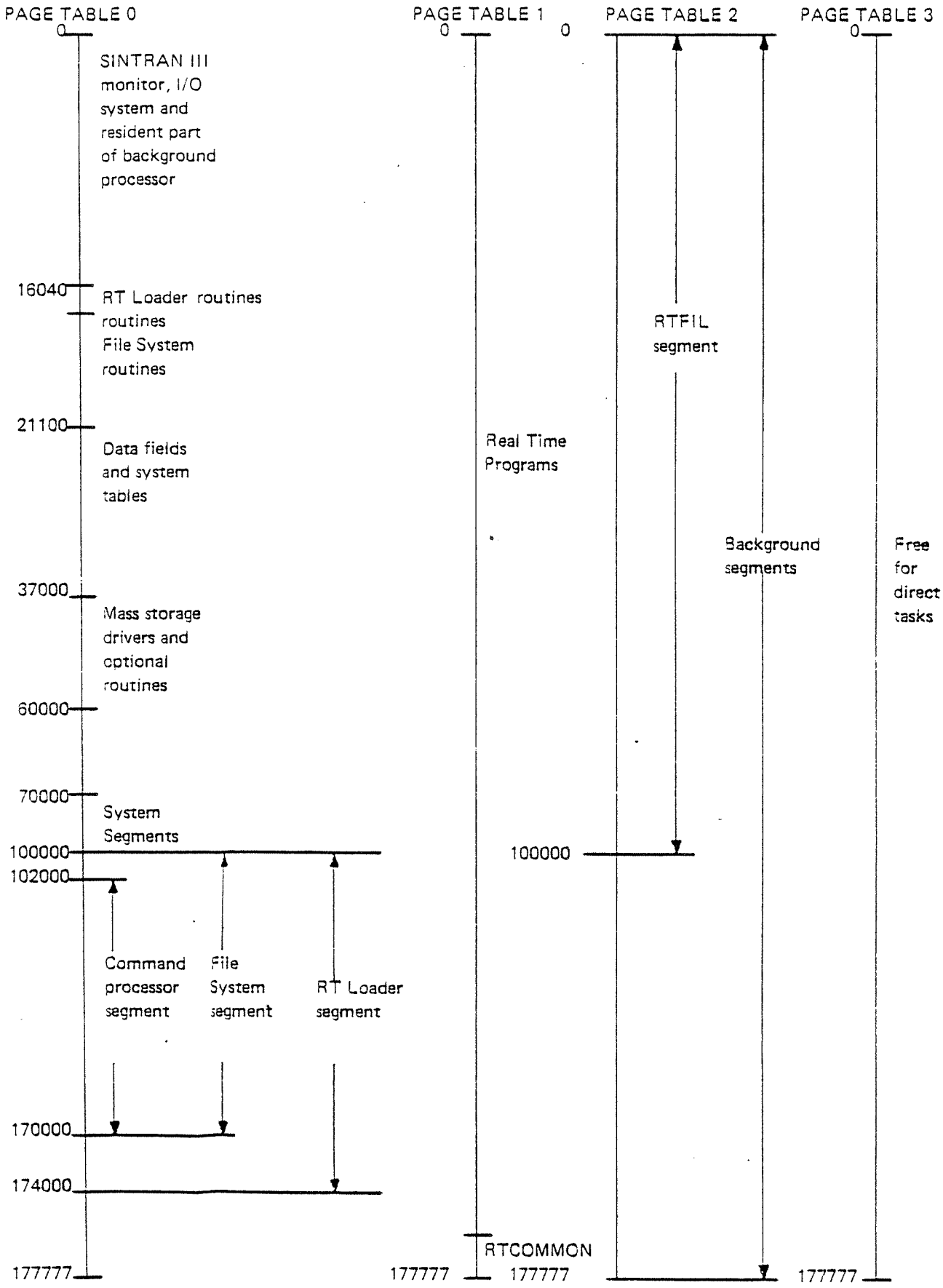
   Segment number 4, protection ring number 2, page table 0, virtual address area: 100000_8 - 173777_8.

2. The RTFIL segment.

   Segment number 8, protection ring 2, page table 2, virtual address area: 0 - 77777_8.

3. The routines and paramter lists for data transfer between the RTFIL segment and the file RTFIL.

   Memory resident, protection ring 2, page table 0, address area: approx. 16040_8 - 16060_8.


PAGE TABLE 0
0
SINTRAN III monitor, I/O system and resident part of background processor
16040
RT Loader routines
routines
File System routines
21100
Data fields and system tables
37000
Mass storage drivers and optional routines
60000
70000
System Segments
100000
102000
Command processor segment
File System segment
RT Loader segment
170000
174000
177777
PAGE TABLE 1
0
Real Time Programs
RTCOMMON
177777
177777
PAGE TABLE 2
0
RTFIL segment
100000
Background segments
177777
PAGE TABLE 3
0
Free for direct tasks
177777

1.3 *GENERATION OF THE RT LOADER*

The source code of the RT Loader is a file which may be compiled/assembled independent of the rest of the SINTRAN III system, and dumped in a binary format onto a file. The "LIST" and "ULIST" symbols are dumped onto another file.

When generating a SINTRAN III system, the binary RT Loader is linked to the SINTRAN III and dumped onto a file together with the rest of the SINTRAN III system.

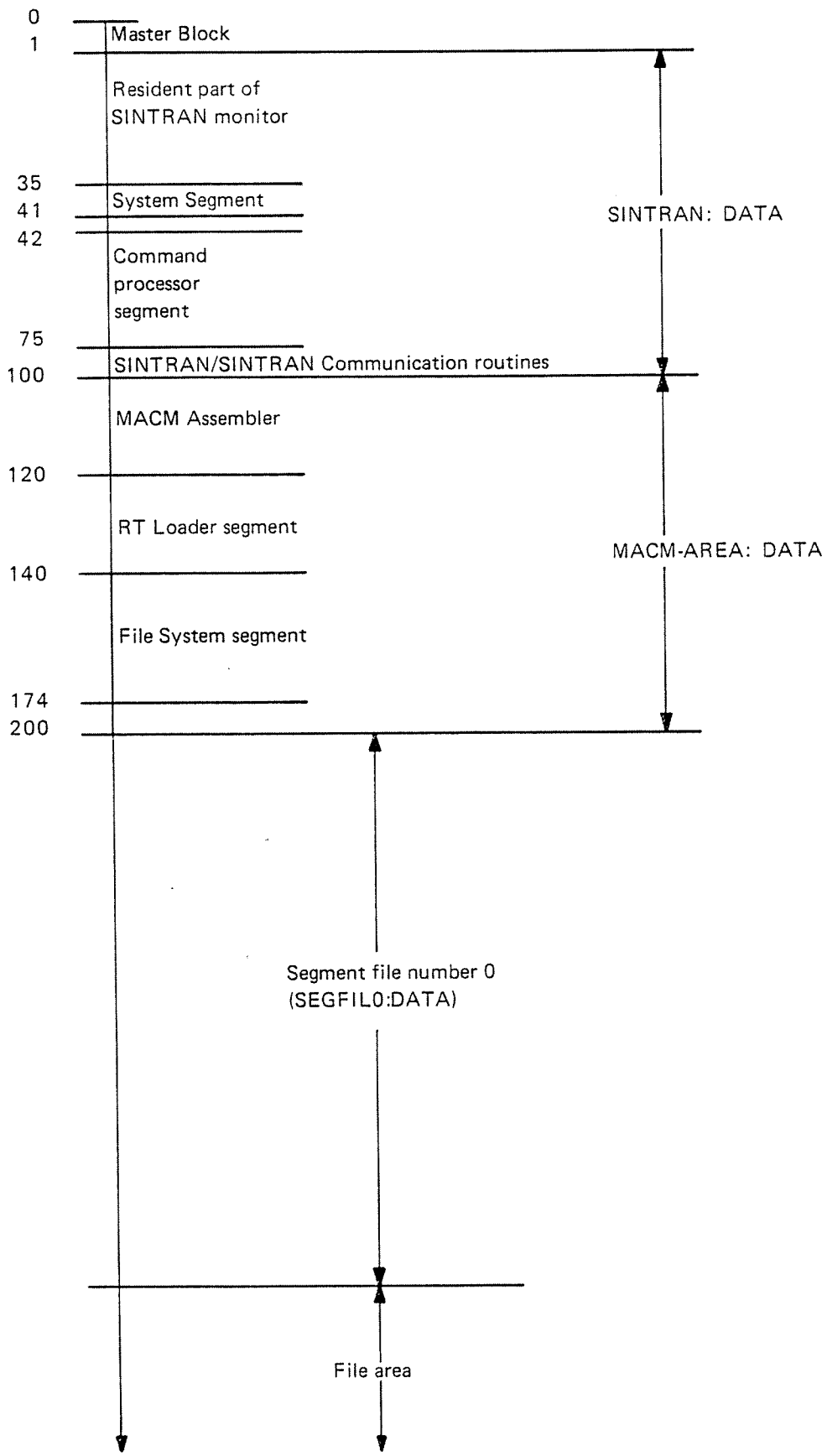
The RT Loader consists of approximately 15K of code from the beginning of the RT Loader segment. The rest of the RT Loader segment and the whole RTFIL segment is working space for the RT Loader.

The memory resident routines of the RT Loader are included in the source file of SINTRAN III.

1.4 *LAYOUT ON MASS STORAGE*

The code of the RT Loader is placed on the "MACM-AREA" file on the mass storage. Every time the SINTRAN III system is started after the MACM command ")HENT" the RT Loader code will be moved by the SINTRAN III start program for the "MACM-AREA" to the segment file number 0.

Mass Storage Layout (address in 1K words):
(on following page)


0
1
Master Block
Resident part of
SINTRAN monitor
35
41
System Segment
42
Command
processor
segment
75
100
SINTRAN/SINTRAN Communication routines
MACM Assembler
120
RT Loader segment
140
File System segment
174
200
SINTRAN: DATA
MACM-AREA: DATA
Segment file number 0
(SEGFIL0:DATA)
File area

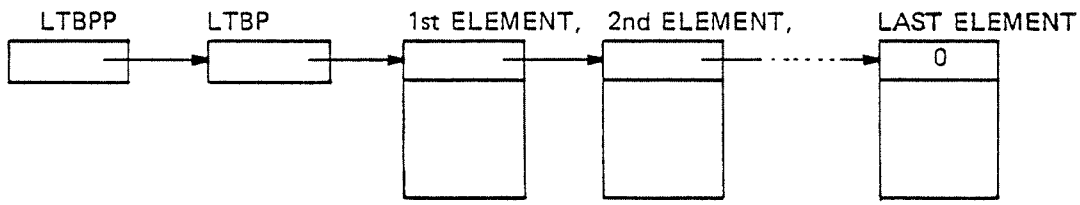
2 THE RT LOADER'S TABLES

2.1 *THE LINKING TABLE*

The linking table is a table where all information about a symbol is kept during a load operation. A symbol may be an RT program name, an entry point, a common label, etc.

Examples of information about a symbol is the value of a symbol, the address where the symbol is referred to, etc.

The linking table is a linked table which always starts in the location LTBPP and ends with a link equals 0.



The symbols in the linking table are:

- defined RT program names
- declared RT program names
- defined RT COMMON labels
- declared RT COMMON labels
- symbols defined on segment number 0
- symbols defined or referred to on the one or eventually two segments currently being built
- symbols defined on an already existing segment currently used as linking segment.
- symbols defined on an already existing segment specified in the FIX-SEGMENT command.

2.1.1 *The Linking Table Element*

The linking table element consists of 8 continuous locations.

Word No.:	Explanation:
0	The element link. This location contains the address of the next element in the linking table. If the element link is zero, then this element is the last in the linking table.
1, 2, 3	The name of the symbol in ASCII. Six bits per character. The symbol is right justified, i.e., the last character in the symbol is in bit 0-5 in word number 3. Maximum of 7 characters in a symbol.
4	Segment number(s) of the symbol. Bit 0-7: first segment number Bit 8-15: eventually second segment number
5	Element descriptor. Bit 0-7: element flag Bit 8-15: eventually priority of an RT program
6	Symbol value or address of a sublist
7	Address of a sublist of linking table elements or program start address if symbol is a defined RT program

The element descriptor values:

Bit 0-2:
0 - defined symbol (entry point)
1 - referenced symbol (undefined symbol)
2 - defined common label
3 - declared common label
4 - not used
5 - not used
6 - declared RT program
7 - defined RT program

Bit 3=1 (BPRINT): This symbol has been printed in the current "list command". This is used in the "list commands" after the commands LIST–IN–ALPHABETICAL–ORDER or LIST–AFTER–INCREASING–ADDRESSES have been given.

Bit 4=1 (INRTF): This symbol is in the RTFIL table.

Bit 5=1 (TEMP): This symbol is not available in the current "load command".

Bit 6=1 (BPREP): This symbol is an RT program name which will replace an already existing RT program with the same name and the same entry in the RT description table.

Bit 7=1 (BDECP): This symbol is a declared RT program which shall be defined now.

Bit 8-15: contains the priority of the RT program if the symbol is an RT program, otherwise, these bits are not used.

Examples of the different linking table element types follows.

2.1.2 *Element Descriptor Bit 0-2 Equals 0, Defined Symbol*

Word No.:	Explanation:
0,1,2,3	Standard
4	Segment number of the symbol
5	Element descriptor, bit 0-2 = 0
6	Value of the symbol
7	not used

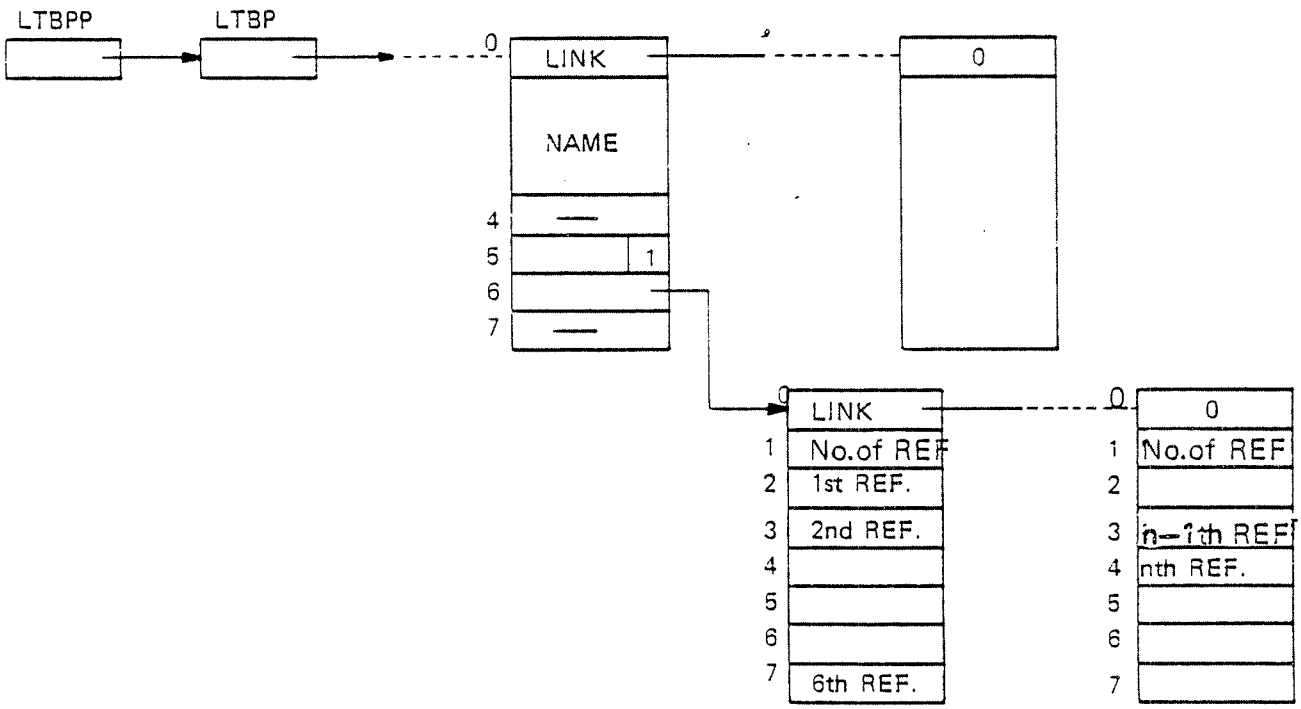
2.1.3 *Element Descriptor bit 0-2 Equals 1, Referenced Symbol*

Word No.:	Explanation:
0,1,2,3	Standard
4	Not used
5	Element descriptor, bit 0-2 = 1
6	address of sublist
7	not used

Sublist element:

Word No.:	Explanation:
0	Element link. Address of next element in sublist, link = 0 means end of sublist
1	number of references in this element, maximum 6
2	address of reference
3	address of next reference
4,5,6,7	address of references

Linking table with a referenced symbol:



2.1.4 *Element Descriptor Bit 0-2 Equals 2, Defined Common Label*

Word No.:	Explanation:
0,1,2,3	Standard
4	Segment number
5	Element descriptor, bit 0-2 = 2
6	Address of common label descriptor
7	eventually address of "data" sublist

Common label descriptor:

Word No.:	Explanation:
0	Eventually address of "9ADS" sublist
1	Address of common block
2	Size of common block
3	Eventually current end of "data" sublist if linking table is expanded to the scratch file
4	Eventually current end of "9ADS" sublist if linking table is expanded to the scratch file
5	Eventually current end of "character" sublist if linking table is expanded to the scratch file
6	Not used
7	Eventually address of "character data" sublist

The address of the common block is not decided before the END-LOAD command is given, so the address of the common block (word number 1 in the common label descriptor) will be relative to the first common address on the actual segment, after the common label is defined until the END-LOAD command is given, then the correct value is inserted.

The "9ADS" sublist (references to the common label):

Word No.:	Explanation:
0	Element link, address of next element in the sublist, link = 0 means end of sublist.
1	Number of references in this element, maximum 6
2	Address of reference
3	Address of next reference
4,5,6,7	Address of references

The "data" sublist (DATA statements):

Word No.:	Explanation:
0	Element link. Address of next element in the sublist, link = 0 means end of sublist.
1	Number of data elements in this element, maximum 3
2	Displacement (address relative to the start of the common block)
3	Data value
4	Displacement of next data element
5	Data value
6	Displacement of next data element
7	Data value

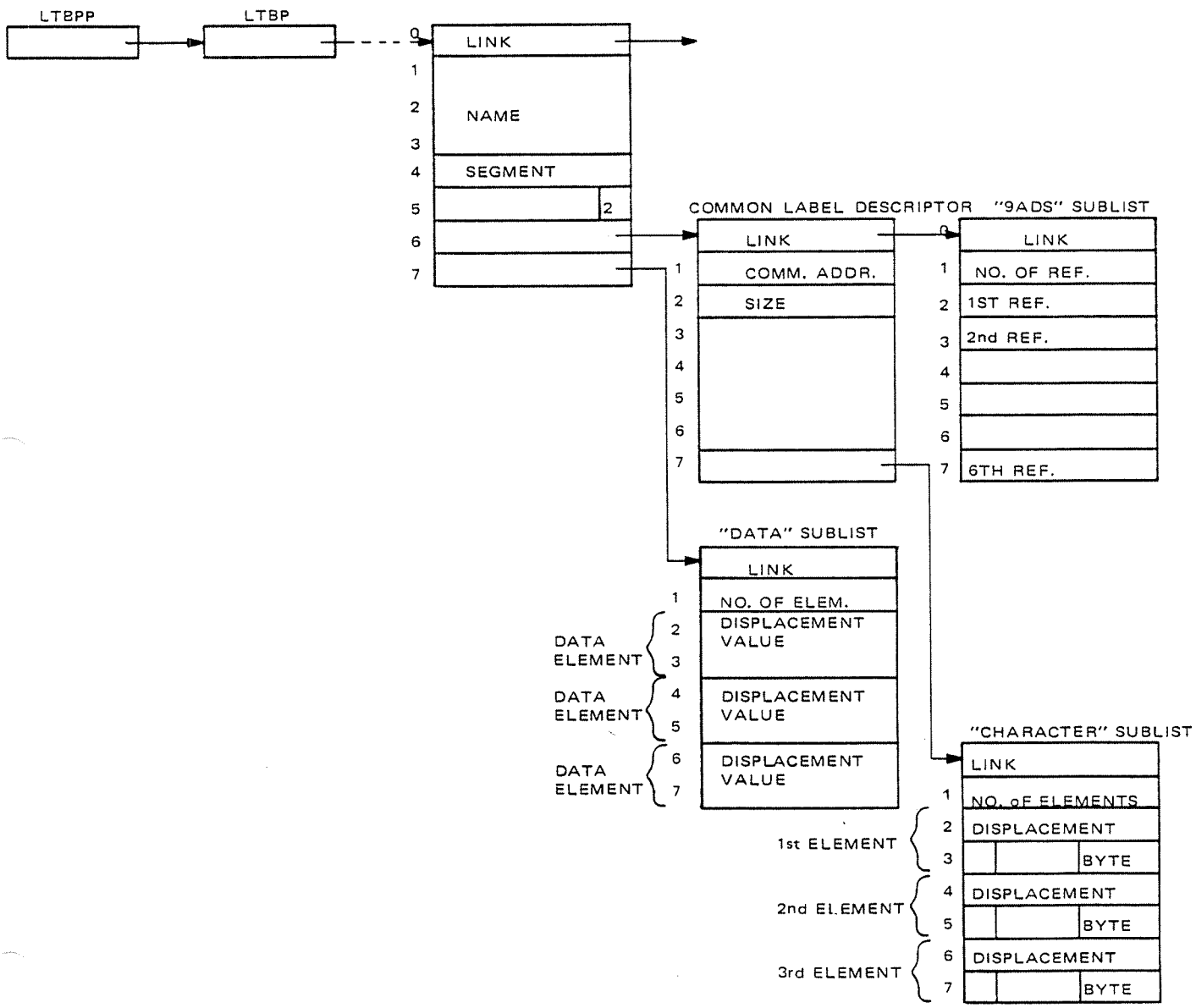
The "CHARACTER" sublist (CHARACTER statements):

Word No.:	Explanation:
0	Element link. Address of next element in the sublist, link = 0 means end of sublist.
1	Number of "character elements" in this sublist element
2	Displacement (address relative to the start of the common block)

The "CHARACTER" sublist continued:

Word No.:	Explanation:
3	Bit 0 – 7: character byte Bit 15 = 0: left byte Bit 15 = 1: right byte
4	Displacement of next "character element"
5	Next character byte
6	Displacement of next "character element"
7	Next character byte

Linking table with a defined common label:



2.1.5 *Element Descriptor Bit 0-2 Equals 3, Declared Common Label*

Word No.:	Explanation:
0,1,2,3	Standard
4	Segment Number. Segment number equals –1 means segment common, segment number equals 0 means RTCOMMON.
5	Element descriptor, bit 0-2 = 3
6	Eventually address of common label descriptor
7	Not used

2.1.6 *Element Descriptor Bit 0-2 Equals 6, Declared RT Program*

Word No.:	Explanation:
0,1,2,3	Standard
4	Not used
5	Element descriptor, bit 0-2 = 6
6	Address of RT description
7	Not used

2.1.7 *Element Descriptor Bit 0-2 Equals 7, Defined RT Program*

Word No.:	Explanation:
0,1,2,3	Standard
4	Segment Number. Bit 0-7, segment number 1; bit 8-15, segment number 2.
5	Element descriptor. Bit 0-2 = 7; bit 8-15, program priority.
6	Address of RT description
7	Start address of RT program

2.1.8 *Insert an Element in the Linking Table*

A new element will always be inserted as the first element in the linking table by setting in the content of the linking table head location LTBP in the LINK of the new element, and set the address of the new element in the table head location LTBP.

2.1.9 *Remove an Element from the Linking Table*

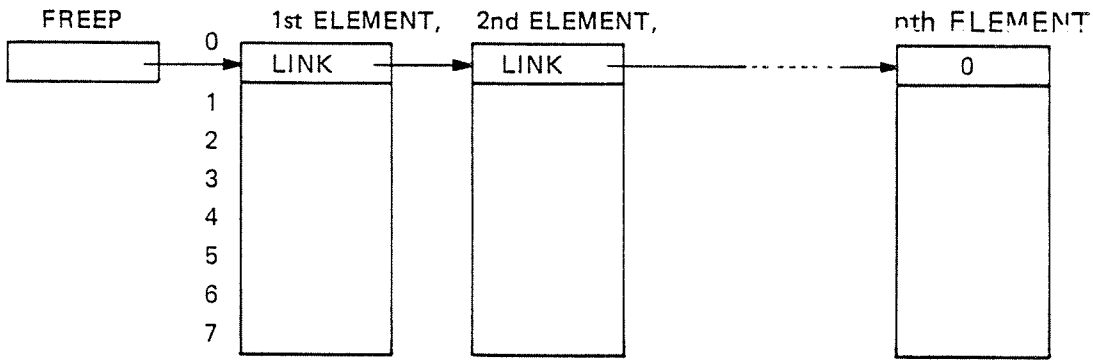
An element in the linking table is removed by taking the content of the LINK location of the element, and storing it in the LINK location of the table element in front of the element to be removed.

2.2 *THE FREELIST*

All the available elements for the linking table are initially set up in a freelist. The freelist is a linked table as in the linking table. When an element is removed from the linking table, it is inserted in the freelist again.

The freelist starts in a location named FREEP.

The freelist:



2.2.1 *Remove an Element from the Freelist*

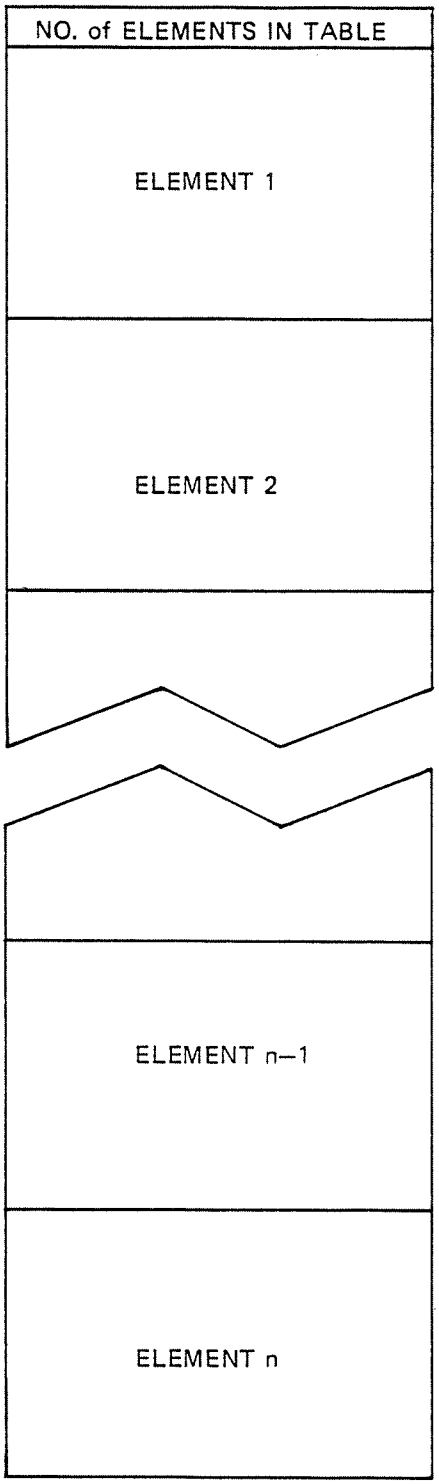
An element is removed from the freelist by taking the LINK of the first element and inserting it in the freelist header FREEP. The first element in the freelist is always the element to remove from the freelist.

2.2.2 *Insert an Element in the Freelist*

A new element in the freelist will always be inserted before the current first element by taking the content of the freelist header FREEP and inserting it as the LINK in the new element, and inserting the address of the new element as the content of the freelist header FREEP.

2.3 *THE RTFIL TABLE*

The RTFIL table is a continuous sequential table containing information about symbols of all existing segments which have been built by the RT loader. Each table element in the RTFIL table consist of 7 locations. The RTFIL table has a header location containing the number of elements in the table.



2.3.1 *The RTFIL Table Element*

Word No.:	Explanation:
0,1,2	The name of the symbol in ASCII. Same format as in the linking table (see Section 2.1.1)
3	segment number(s)
4	Element descriptor. Bit 4 (INRTF) is always set
5	symbol value
6	other information

2.3.2 *Element Descriptor Bits 0-4 Equals 20, Defined Symbol*

Word No.:	Explanation:
0,1,2	Standard
3	Segment number
4	Element descriptor. Bit 0-4 = 20
5	Symbol value
6	not used

2.3.3 *Element Descriptor Bits 0-4 Equals 22, Defined Common Label*

Word No.:	Explanation:
0,1,2	Standard
3	Segment number. Segment number equals 0 means RTCOMMON.
4	Element descriptor. Bit 0-4 = 22.
5	Common label address (value).
6	Size of the common area.

2.3.4 *Element Descriptor Bits 0-4 Equals 26, Declared RT Program*

Word No.:	Explanation:
0,1,2	Standard
3	Not used
4	Element descriptor. Bit 0-4 = 26
5	Address of RT description
6	Not used

2.3.5 *Element Descriptor Bits 0-4 Equals 27, Defined RT Programs*

Word No.:	Explanation:
0,1,2	Standard
3	Segment number(s) Bit 0-7, segment number 1; Bit 8-15, segment number 2.
4	Element Descriptor. Bit 0-4 = 27; Bit 8-15, program priority.
5	Address of RT description
6	Start address of RT program

2.3.6 *Insert an Element in the RTFIL Table*

A new element will always be inserted at the top of the RTFIL table, i.e., after the current last element in the table. The RTFIL header and the element counter will be incremented.

2.3.7 *Remove an Element from the RTFIL Table*

An element will be removed from the RTFIL table by moving all the elements following the element to remove, 7 locations down in the RTFIL table. The RTFIL header and the element counter will be decremented by one.

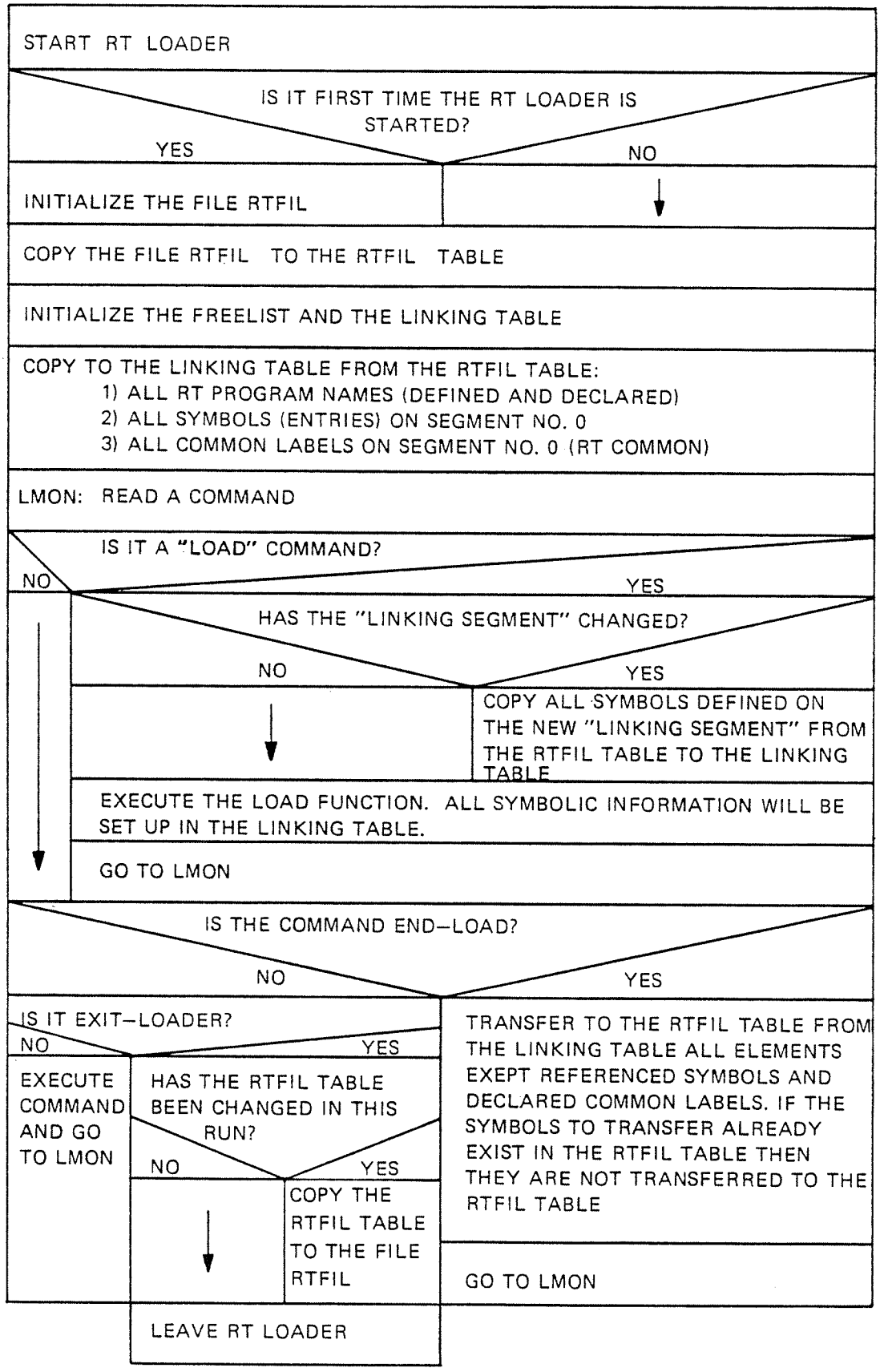
2.4 *THE FILE RTFIL*

The RT loader uses a file named RTFIL to keep a copy of the RTFIL table. The content of the file RTFIL is exactly the same as the content of the RTFIL table.

The content of the RTFIL table is written onto the file RTFIL on the EXIT-LOADER command if changes in the RTFIL table have been made in the current run of the RT loader.

The file RTFIL is initialized by setting the element counter the first location in the file, to zero, the first time the RT loader is started.

2.5 *THE FLOW OF SYMBOLIC INFORMATION IN THE RT LOADER*



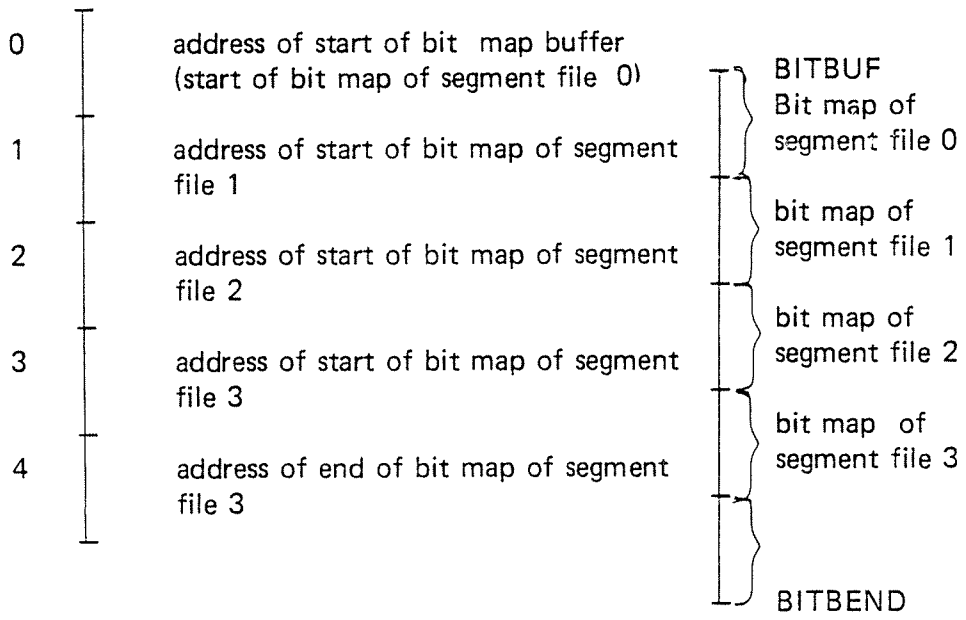
2.6 *THE SEGMENT FILES BIT MAP*

The RT loader keeps a bit map of the segment files, to see which pages are free and which pages are already in use by segments. The bit map of the segment files are set up in the array BITBUF, and the addresses to the bit map of the respective segment files are set up in the array SEGTBIT.

Bit set to one means page is occupied, bit set to zero means page is free.

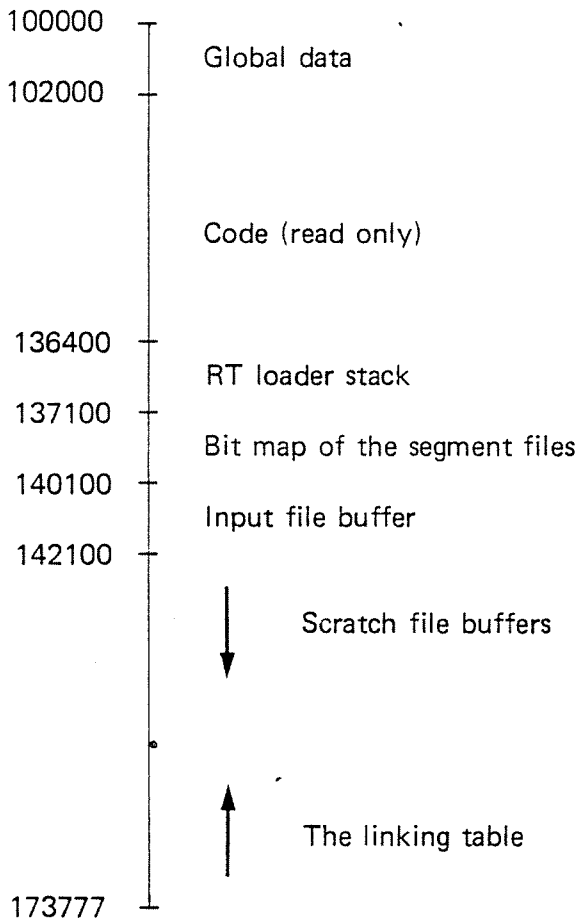
Bit number 15 in the first word in a segment file bit map corresponds to page number 0, bit 14 corresponds to page number 1, etc. (Bit 15 in the second word corresponds to page number 16.)

The size of the different bit maps depends upon the sizes of the corresponding segment files.



3 BUILDING SEGMENTS

3.1 *MEMORY LAYOUT OF THE RT LOADER SEGMENT*



3.2 *HOW TO BUILD A SEGMENT*

While loading a segment, the code is stored on the terminal's scratch file. The address area used in the scratch file is the same as the actual segment's virtual address area.

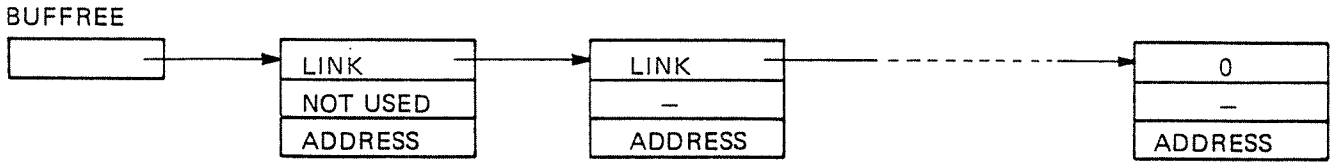
There are buffers for the scratch file on the RT loader segment. The number of buffers available depends upon the size of the linking table, as the linking table and the scratch file buffers share a common area.

In the END-LOAD command, the scratch file buffers are written onto the scratch file and the segment area on the scratch file is transferred to the actual segment file.

3.3 *THE SCRATCH FILE BUFFERS*

The available scratch file buffers are set up in a list of used and unused buffers. Each scratch file buffer consists of 1k (2000_8) words.

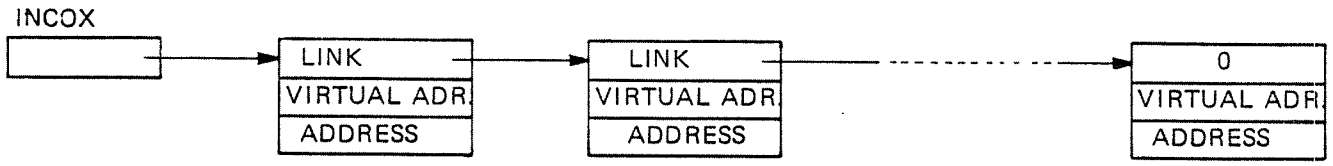
The list of unused buffers:



The list of unused scratch file buffers starts in the location BUFFREE. Each element in the list consists of 3 locations.

Word No.:	Explanation:
0	LINK. Address of next element in the list, LINK = 0 means end of list.
1	Not used
2	Address of the buffer

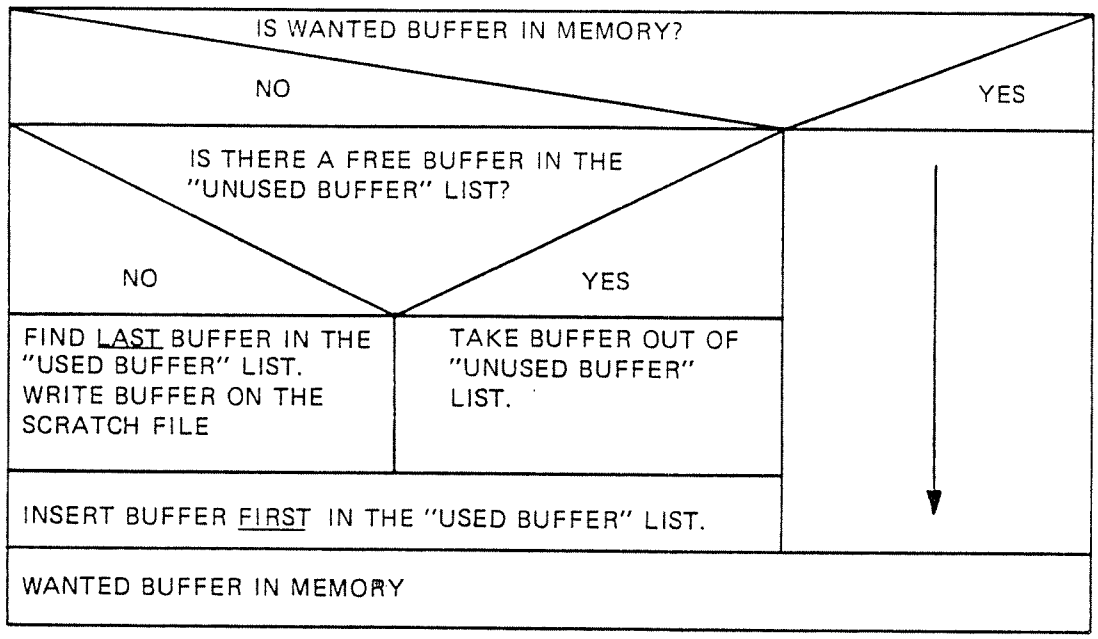
The list of used buffers:



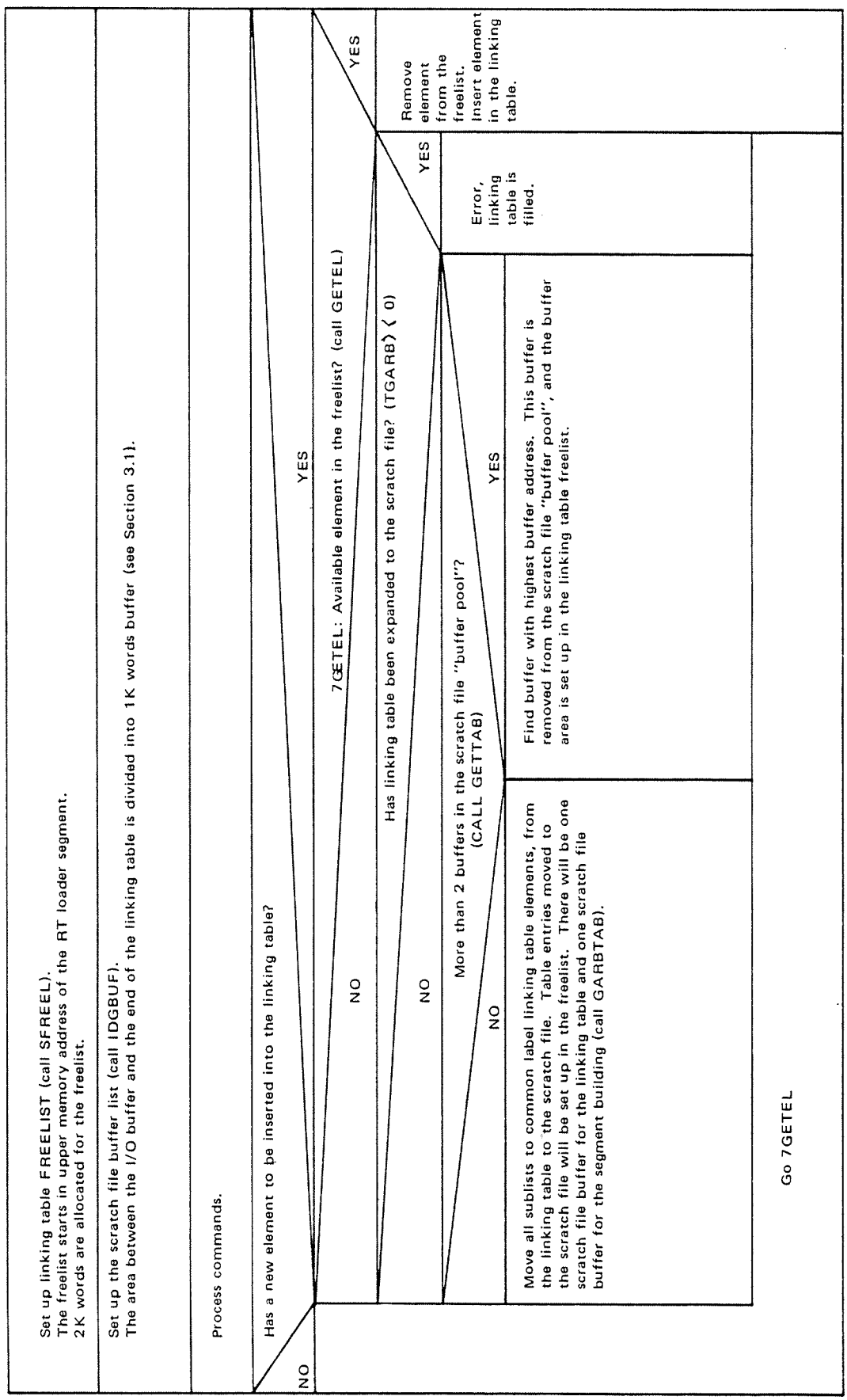
Word No.:	Explanation:
0	LINK. Address of next element in the list. LINK = 0 means end of list.
1	Virt. Addr. Address area of the scratch file for this buffer.
2	Address of the buffer

The list of unused scratch file buffers are set up each time the RT loader is reset.

3.4 *FLOW CHART OF THE SCRATCH FILE BUFFER SYSTEM*



3.5 *FLOW CHART OF HOW THE LINKING TABLE EXPANDS AND HOW THE SCRATCH FILE BUFFER LISTS ARE REDUCED*



4 GLOBAL VARIABLES

```
%=========================================================================
% -1,0-      E X P L A N A T I O N  O F  T H E  G L O B A L  V A R I A B L E S
%                          ( A L P H A B E T I C A L )


% ACTPRI      - LOCATION IN RT DESCRIPTION ENTRY
%             - BIT 0=1: PROTECT RING
%             - BIT 14: THE RTOFF BIT (PROGRAM CANNOT BE STARTED)
% ACTSEG      - LOCATION IN THE RT DESCRIPTION ENTRY CONTAINING THE
%             - RT PROGRAMS CURRENT SEGMENT NUMBERS
% ALOGNO      - LOCATION IN THE CORE MAP ENTRY, LOGICAL PAGE NUMBER
% AREG        - A REGISTER FOR THE SUBROUTINES, SAVED IN STACK BY ENTER
% BALPH       - BIT 7 IN COMFLAG (SEE COMFLAG)
% BBACK       - BIT 1 IN COMFLAG (SEE COMFLAG)
% BBOOT       - BIT 2 IN COMFLAG (SEE COMFLAG)
% BDECP       - BIT 7 IN LINKING TABLE AND RTFIL TABLE ENTRY FLAG
%             - BDECP=1 MEANS DECLARED RT PROGRAM NAME
% BDEMAND     - BIT 1 IN SEGMENT TABLE ENTRY FLAG WORD
%             - IF BDEMAND=1 THEN DEMAND SEGMENT, ELSE NON-DEMAND SEGMENT
% BDREL       - BIT 9 IN COMFLAG (SEE COMFLAG)
% BECBT       - BRF CONTOL BYTE FOR START OF PROGRAM UNIT (17 OCT)
% BECFL       - FLAG FOR MARKING THAT LOADING OF A BRF UNIT IS STARTED
%             - BECFL=1 WHEN THE LOADING IS STARTED (BEG-BYTE FOUND, END-BYTE
%             - NOT YET FOUND)
% BFTNL       - BIT 6 IN COMFLAG (SEE COMFLAG)
% BINL        - BIT 0 IN COMFLAG (SEE COMFLAG)
% BITBEND     - ADDRESS OF THE END OF THE SEGMENT FILES BIT MAPS
% BITBUF      - THE SEGMENT FILE BIT MAP BUFFER
% BLCKI       - BLOCK NUMBER TO ACCESS ON THE SCRATCH FILE
% BLOCM       - BIT 13 IN COMFLAG (SEE COMFLAG)
% BLST        - BUFFER IN SINTRAN - CONTAINING THE MASS STORAGE START
%             - ADDRESSES OF THE DEFINED SEGMENT FILES
% BNREL       - BIT 4 IN COMFLAG (SEE COMFLAG)
% BOOTS       - START ADDRESS OF THE BOOTSTRAP TO OUTPUT IN THE END-LOAD
%             - COMMAND IN THE "IMAGE-LOAD" MODE
% BPADR       - BIT 8 IN COMFLAG (SEE COMFLAG)
% BPAGLINK    - LOCATION IN THE SEGMENT TABLE ENTRY, CONTAINING THE ADDRESS
%             - OF THIS SEGMENT'S FIRST PAGE IN CORE-MAP
% BPREP       - BIT 6 IN LINKING TABLE ELEMENT FLAG
%             - BPREP=1 MEANS RT PROGRAM SHALL BE REPLACED
% BPRINT      - BIT 3 IN LINKING TABLE AND RTFIL TABLE ELEMENT FLAG
%             - BPRINT=1 MEANS THAT ELEMENT HAS BEEN PRINTED
%             - IN THE CURRENT "LIST-COMMAND"
% BPR1CO      - BIT 10 IN COMFLAG (SEE COMFLAG)
% BPR2CO      - BIT 11 IN COMFLAG (SEE COMFLAG)
% BREL        - BIT 3 IN COMFLAG (SEE COMFLAG)
% BRESLINK    - LOCATION IN THE RT DESCRIPTION ENTRY, START OF THE
%             - RT PROGRAMS RESERVATION LINK (QUEU)
% BRFDEV      - LOGICAL DEVICE NUMBER FOR THE INPUT STREAM
% BRFTP       - POINTER TO THE PARAMETER "BRF INPUT FILE: "
```

```
% BRTDSTART  - START OF THE RT-DESCRIPTION TABLE IN THE "BACKUP-SYSTEM",
%           - USED IN THE "BACKUP-LOAD" MODE ONLY.
% BSEGLINK   - IN SINTRAN - START OF "SEGMENT-IN-MEMORY" QUEU
% BTABCONST  - ADDRESS MODIFIER FOR THE SEGMENT TABLE AND THE RT-DESCRIPTION
%           - TABLE IN THE "BACKUP-LOAD MODE"
% BTLIST     - BIT 12 IN COMFLAG (SEE COMFLAG)
% BUFADR     - START OF "SCRATCH FILE BUFFER AREA"
% BUFFREE    - START OF FREELISTE FOR THE "SCRATCH FILE BUFFERS"
% BUFLINK    - BUFFER CONTAING THE "USED-SCRATCH-FILE-BUFFER" LISTE AND THE
%           - "UNUSED-SCRATCH-FILE-BUFFER" LISTE
% BUFS       - ADDRESS OF THE "SCRATCH FILE BUFFER" HEADERS (BUFLINK)
% BWIP       - BIT 12 IN PCR REGISTER (WRITTEN IN PAGE)
% BXLO       - BIT 5 IN COMFLAG (SEE COMFLAG)
% CCFPAGE    - IN SINTRAN - FIRST PAGE NO. OF RTCOMMON
% CCLPAGE    - IN SINTRAN - LAST PAGE NO. OF RTCOMMON
% CCSGFNO    - SAVED CURRENT SEGMENT FILE NUMBER IN USE
% CHCI       - BYTE POINTER IN THE INPUT BUFFER IOBUF
% CHSUM      - LOCATION FOR GENERATING CHECKSUM WHILE LOADING
% CIMSEG     - SEGMENT NUMBER USED IN THE
%           - "WRITE-SEGMENT-BACK" MONITOR CALL
% CLC        - CURRENT LOAD ADDRESS FOR THE CURRENT LOAD SEGMENT
% CLC0       - CURRENT LOAD ADDRESS FOR RTCOMMON
% CLC1       - CURRENT LOAD ADDRESS FOR THE FIRST NEW-SEGMENT (SEGN1)
% CLC2       - CURRENT LOAD ADDRESS FOR THE SECOND NEW-SEGMENT (SEGN2)
% CLITEND    - CURRENT END OF THE LINKING TABLE (LOWEST ADDRESS IN LISTE)
% CLITSTART  - CURRENT START OF THE  LINKING TABLE (HIGHEST ADDR. IN LISTE)
% CNZERO     - FLAG WORD MARKING THAT AN ADDRESS AREA HAS REACHED
%           - THE LIMIT OF 64K
%           -    BIT 0 (SXS0)=1: RTCOMMON AREA
%           -    BIT 1 (SXS1)=1: COMMON DEFINED TO THE END OF SEGN1
%           -    BIT 2 (SXS2)=1: COMMON DEFINED TO THE END OF SEGN2
%           -    BIT 3 (SXCLC)=1: ADDRESS AREA OF  THE CURRENT
%           -         LOAD SEGMENT HAS REACHED 64K
%                BIT 4 (SXCL1)=1: ADDRESS AREA OF SEGN1 HAS REACHED 64K
%                BIT 5 (SXCL2)=1: ADDRESS AREA OF SEGN2 HAS REACHED 64K
% COMFLAG    - FLAG WORD USED BY THE COMMANDS
%           - BIT 0 (BIML)=1: "IMAGE-LOAD MODE"
%           - BIT 1 (BBACK)=1: "BACKUP-LOAD MODE"
%           - BIT 2 (BBOOT)=1: BOOTSTRAP START ADDRESS IS NOT
%                              GIVEN IN THE IMAGE-LOAD COMMAND
%           - BIT 3 (BREL)=1: REENTRANT-LOAD COMMAND IS THE LAST
%                             GIVEN LOAD-COMMAND
%           - BIT 4 (BNREL)=1: NREENTRANT-LOAD COMMAND IS THE LAST
%                              GIVEN LOAD-COMMAND
%           - BIT 5 (BXLO)=1: X-LOAD COMMAND IS THE LAST GIVEN LOAD-COMMAND
%           - BIT 6 (BFTNL)=1: THE FILE NAMED "(SYSTEM)FTNLIBR" WILL BE
%                              SCANNED WHEN THE END-LOAD COMMAND IS GIVEN
%                              AND NREENTRANT-LOAD IS THE LAST GIVEN LOAD-
%                              COMMAND
%                         OR
%                              THE FILE NAMED "(SYSTEM)FTNRTLIBR" WILL BE
%                              SCANNED WHEN THE BRF CONTROL BYTE EOF
%                              (23 OCT) IS FOUND IN THE INPUT STREAM, AND
%                              REENTRANT-LOAD IS THE LAST GIVEN LOAD-COMMAND
```

```
%              - BIT 7 (BALPH)=1: THE SYMBOL OUTPUT IN THE "LIST-COMMANDS"
%                                 WILL BE IN ALPHABETICAL ORDER
%              - BIT 8 (BPADR)=1: THE SYMBOL OUTPUT IN THE "LIST-COMMANDS"
%                                 WILL BE IN THE ORDER OF INCREASING ADDRESSES
%              - BIT 9 (BDREL)=1: "DUMP-REENTRANT" COMMAND
%              - BIT 10 (BPR1CO)=1: "PRESET-COMMON" COMMAND IS GIVEN FOR SEGN1
%              - BIT 11 (BPR2CO)=1: "PRESET-COMMON" COMMAND IS GIVEN FOR SEGN2
%              - BIT 12 (BTLIST)=1: TABLE CONTENT SHOULD BEE PRINTED IN THE
%                                   "TERMINATE-LOAD-AND-PRINT" COMMAND
%              - BIT 13 (BLOCM)=1: MARKING THAT THE CURRENT INPUT-STREAM
%                                  IS IN BRF FORMAT AND NOT IN BINARY FORMAT
% COMINDEV     - LOGICAL DEVICE NUMBER FOR THE COMMAND INPUT DEVICE
% COMOUTDEV    - LOGICAL DEVICE NUMBER FOR THE COMMAND OUTPUT DEVICE
% COMSTRING    - COMMAND INPUT LINE BUFFER
% CORMSTART    - IN SINTRAN - ADDRESS OF START OF CORE MAP TABLE (END OF
%              - SEGMENT TABLE)
% CPNT         - BYTE POINTER IN BUFFER CSTRING
% CRLFC        - CONSTANT=6412, ASCII CODE FOR CARRIAGE RETURN AND LINE FEED
% CSCFNO       - CURRENT SEGMENT FILE NUMBER (0 - 3)
% CSTRING      - POINTER TO ARRAY COMSTRING CONTAINING COMMAND LINE
% CURPROG      - IN SINTRAN - RT DESCRIPTION ADDRESS OF THE CURRENT RUNNING
%              - RT PROGRAM
% CURSLLIM     - LOWEST ADDRESS USED ON THE CURRENT LOAD-SEGMENT (CUR1S)
% CURSULIM     - UPPER ADDRESS USED ON THE CURRENT LOAD-SEGMENT (CUR2S)
% CURTP        - POINTER TO FIRST FREE ELEMENT IN RTFIL-BUFFER
% CUR1S        - CURRENT LOAD-SEGMENT
% CUR2S        - CURRENT LINK-SEGMENT
% C3COM        - ADDRESS OF COMMON LABEL IN RESIDENT CORE RELATIVE TO
%              - UPPER RTCOMMON ADDRESS WHEN THE CURRENT LOAD
%              - OPERATION STARTED
% C1COM        - ADDRESS OF COMMON LABEL IN SEGN1 RELATIVE TO THE
%              - START OF COMMON AREA ON SEGN1
% C2COM        - ADDRESS OF COMMON LABEL IN SEGN2 RELATIVE TO THE
%              - START OF COMON AREA ON SEGN2
% DGPARL       - POINTS TO PARAMETER LISTE FOR ACCESSING THE SCRATCH FILE
% DREG         - RETURN ADDRESS TO THE SUBROUTINES, SAVED IN STACK BY ENTER
% DSEGTYP      - DEFAULT SEGMENT TYPE. (RFW & ND)
% DSTACKL      - CONSTANT=2000, DEFAULT STACK LENGTH USED IN THE
%              - REENTRANT-LOAD COMMAND
% ELRTFIL      - POINTER TO THE BUFFER XELRTFIL, USED FOR MOVING TABLE ELEMENTS
%              - TO/FROM RTIL FROM/TO THE LINKING TABLE
% ENDBT        - BRF CONTOL BYTE FOR END OF PROGRAM UNIT (21 OCT)
% ENTLREG      - L REGISTER WHEN ENTERING THE RT LOADER
% ENTTREG      - T REGISTER WHEN ENTERING THE RT LOADER
% EOFBT        - BRF CONTOL BYTE FOR END OF FILE (23 OCT)
% FACCODE      - ACCESS CODE TO USE WHEN OPENING FILES
% FALLO        - BIT 5 IN OPEN FILE TABLE OFTYP
%              - FALLO=1 MEANS ALLOCATED FILE
% FCONT        - BIT 4 IN OPEN FILE TABLE OFTYP
%              - FCONT=1 MEANS CONTINUOUS FILE
```

```
% FCURTP      - CONSTANT=1, FIRST ADDRESS IN THE RTFIL-BUFFER
% FFTNFIL     - BUFFER CONTAINING THE FILE NAME '(SYSTEM)FTNLIBR:BRF'
% FIX         - BIT 2 IN FLAG IN SEGMENT TABLE ENTRY
%             - FIX=1 MEANS SEGMENT IS FIXED IN MEMORY (FIX OR FIXC)
% FIXCLC0     - IN SINTRAN - CURRENT TOP OF USED RTCOMMON AREA
% FLAG        - LOCATION IN THE SEGMENT TABLE ENTRY
%             - BIT 1=1: DEMAND SEGMENT, BIT 1=0: NON DEMAND SEGMENT
%             - BIT 2=1: SEGMENT IS FIXED IN MEMORY
%             - BIT 3 (INHBT)=1: SEGMENT CANNOT BE SWAPPED IN OR OUT
%             - BIT 9-15: SAME AS IN THE PAGE INDEX TABLE ENTRY
% FP          - BIT 13 IN PCR REGISTER (FETCH PERMITTED)
% FREEP       - POINTER TO FIRST ELEMENT IN FREELISTE OF THE LINKING TABLE
% FRTFTN      - BUFFER CONTAINING THE FILE NAME '(SYSTEM)FTNRTLIBR:BRF'
% FTMP0       - (REAL) WORKING LOCATIONS FOR THE SUBROUTINES,
%             - SAVED IN STACK BY ENTER
% FTMP1       - (REAL) WORKING LOCATIONS FOR THE SUBROUTINES,
%             - SAVED IN STACK BY ENTER
% FTMP2       - (REAL) WORKING LOCATIONS FOR THE SUBROUTINES,
%             - SAVED IN STACK BY ENTER
% IBLCK       - BLOCK NUMBER TO USE WHEN ACCESSING THE INPUT FILE
% INCBT       - BRF CONTROL BYTE FOR INTEGER COMMON DATA CONSTANT (40 OCT)
% INCOX       - START OF LISTE OF USED BUFFERS IN THE "SCRATCH FILE BUFFER POOL"
% INHBT       - BIT 3 IN FLAG IN SEGMENT TABLE ENTRY
%             - INHBT=1 MEANS SEGMENT FREE OR TEMPORARLY UNAVAILABLE
% INITZFLAG   - FLAG SET TO ONE WHEN AN ERROR IS DETECTED IN THE
%             - INITIALIZING OF THE RT-LOADER, RETURNS TO THE
%             - SINTRAN III OPERATOR COMMUNICATION
% INLBT       - BRF CONTROL BYTE FOR INTEGER LOCAL DATA CONSTANT (34 OCT)
% INPFP       - POINTER TO THE PARAMETER "INPUT FILE: "
% INRTF       - BIT 4 IN ELEMENT DESCRIPTOR (FLAG)
%             - IN LINKING TABLE ELEMENT AND IN RTFIL ELEMENT
%             - INRTF=1 MEANS THAT SYMBOL EXIST IN RTFIL TABLE
% IOBUF       - BUFFER OF THE BRF/BINARY INPUT FILE (1K)
% IRTDSIZE    - RT DESCRIPTION SIZE IN SINTRAN III C
% IPTAB       - PAGE INDEX TABLE NUMBER FOR THE RT-PROGRAMS AND THE SEGMENTS
%             - CURRENTLY BEEING BUILDT, BIT 11-12: PAGE INDEX TABLE NUMBER
%             - PAGE INDEX TABLE NO, 1 IS DEFAULT
% KEYDEV      - CONSTANT=1, LOGICAL DEVICE NUMBER OF THE TERMINALS
% KGPFL       - FLAG FOR GPAR CALLED DIRECTLY OR BY KGPAR
%             - KGPFL=1 WHEN GPAR CALLED BY KGPAR
% LAREG       - A-REGISTER FOR THE SUBROUTINES CALLING
%             - LENTER & LLEAVE
% LCORMSTART  - ADDRESS OF END OF SEGMENT TABLE
% LDREG       - RETURN ADDRESS FOR THE SUBROUTINES CALLING
%             - LENTER & LLEAVE
% LFILTABL    - NEXT FREE LINKING TABLE ELEMENT ON THE SCRATCH FILE
% LFIXCLC0    - UPPER ADDRESS USED IN RTCOMMON
% LIBNO       - FLAG TO USE WHEN CHECK FOR SKIP/NOT SKIP AFTER
%             - A STRING OF SYMBOLS AFTER THE BRF CONTROL
%             - BYTE LIBR
% LINK1       - LOCAL SAVING OF RETURN ADDRESS FOR SUBROUTINES NOT CALLING
%             - ENTER & LEAVE
```

```
% LINK2       - LOCAL SAVING OF RETURN ADDRESS FOR SUBROUTINES NOT CALLING
%             - ENTER & LEAVE
% LITESIZE    - LINKING TABLE ENTRY SIZE (8 LOC)
% LOADI       - IN SINTRAN - FLAG SET TO ONE THE FIRST TIME THE RT LOADER
%             - IS STARTED
% LOGADR      - LOCATION IN THE SEGMENT TABLE ENTRY
%             - BIT 0-7: FIRST LOGICAL PAGE IN SEGMENT
%             - BIT 8-15: NUMBER OF PAGES IN SEGMENT
% LONGFL      - FLAG FOR 4 OR 6 BYTES BRF SYMBOLS, LONGFL=1 THEN 6 BYTES
% LOWPRI      - CONSTANT=1, PRIORITY GIVEN THE RT PROGRAMS
%             - WHO HAS NO PRIORITY SPECIFIED
% LRTDSTART   - ADDRESS OF START OF RT-DESCRIPTION TABLE
% LRTFPAGE    - FIRST LEGAL PAGE TO LOAD INTO ON THE CURRENT
%             - PAGE INDEX TABLE
% LRTLPAGE    - LAST LEGAL PAGE TO LOAD INTO ON THE CURRENT
%             - PAGE INDEX TABLE
% LSEGTSTART  - ADDRESS OF START OF SEGMENT TABLE
% LSTACK      - THE LOADER STACK ARRAY
% LSTASIZE    - SIZE OF RT-LOADER STACK
% LTBP        - POINTER TO FIRST ELEMENT IN LINKING TABLE
% LTBPP       - CONSTANT, ALLWAYS POINTS TO LTBP
%             - (LINKING-TABLE-HEADER)
% LTREG       - T-REGISTER FOR THE SUBROUTINES CALLING
%             - LENTER & LLEAVE
% LUSEGM      - FIRST SEGMENT AVAILABLE FOR RT-LOADER
% LUSFADR     - FIRST FREE PAGE ON SEGFIL0 IN BACKUP-LOAD MODE
% LXREG       - X-REGISTER FOR THE SUBROUTINES CALLING
%             - LENTER & LLEAVE
% MASSADR     - LOCATION IN THE SEGMENT TABLE ENTRY
%             - BIT 0-13: ADDRESS OF THE SEGMENT ON THE SEGMENT FILE
%             - IN PAGES RELATIVE TO THE START OF THE SEGMENT FILE
%             - BIT 14-15: SEGMENT FILE NUMBER
% MASSNO      - BUFFER CONTAING THE "OPEN-FILE-NUMBER" OF THE SPECIFIED
%             - SEGMENT FILES
% MAXCBT      - HIGHEST BRF CONTROL NUMBER (45 OCT)
% MAXLTABLE   - UPPER MEMORY ADDRESS OF LINKING TABLE (END OF RT LOADER SEGMENT)
% MAXPAGE     - ACTUAL MAX. NUMBERS OF BUFFERS IN THE "SCRATCH FILE BUFFER POOL"
% MCURTP      - UPPER ADDRESS ON RTFIL SEGMENT
% NBYT1K      - CONSTANT=4000, NUMBER OF BYTES IN 1K WORDS
% NCOSTR      - NUMBER OF CHARACTERS IN THE OSTRING-BUFFER
% NELRTFIL    - ACTUAL NUMBER OF ELEMENT IN THE RTFIL-BUFFER
% NEWPAGE     - FLAG FOR STARTING/NOT STARTING ON A NEW PAGE FOR EACH
%             - NEW BRF UNIT, NEWPAG=1 WHEN STARTING ON A NEW PAGE
% NOBS        - CONSTANT=2000, SCRATCH FILE BUFFER SIZE
% NOBSW       - NUMBER OF WORDS IN TRANSFER TO/FROM SEGMENT FILE AND SCRATCH FILE
% NULL        - CONSTANT=0
% NWRTF       - IN SINTRAN - NO. OF WORDS TO TRANSFER TO/FROM THE RTFIL FILE
%             - FROM/TO THE RTFIL SEGMENT
% OFPAG       - LOCATION IN THE OPEN FILE TABLE ENTRY - NUMBER OF PAGES
%             - IN A CONTINUOUS FILE
% OFTYP       - LOCATION IN THE OPEN FILE TABLE ENTRY - FILE TYPE
% OLDCHAR     - USED FOR SAVING ONE ASCII CHAR,
% OPNT        - BYTE POINTER FOR OSTRING
```

```
% OSTRING     - ADDRESS OF OLD STRING, CONTAINING TEXT TO OUTPUT
% OUTFP       - POINTER TO THE PARAMETER "OUTPUT FILE: "
% OUTDEV      - LOGICAL DEVICE NUMBER FOR GENERAL OUTPUT FROM THE RT-LOADER
% OUTFIL      - LOGICAL DEVICE NUMBER OF THE OUTPUT FILE,
%             - USED BY THE "LIST-COMMANDS"
% PACCERR     - ADDRESS OF THE TEXT STRING "ERROR IN ACCESSING "
% PAGLINK     - LOCATION IN THE CORE MAP ENTRY - POINTER TO THE NEXT ENTRY
%             - IN THE CORE MAP FOR THIS SEGMENT
% PAGPHYS     - LOCATION IN THE CORE MAP ENTRY CONTAINING INFORMATION
%             - TO BE PUT INTO THE PAGE INDEX TABLE
% PARDEF      - FLAG TELLING WHICH PARAMETERS HAVE BEEN SKIPPED BY THE
%             - USER. THE RT-LOADER WILL SET IN DEFAULT PARAMETER VALUES
%             - FOR THE SKIPPED PARAMETERS IF DEFAULT VALUES ARE AVAILABLE
%             - BIT 0 (P0D)=1 :PARAMETER NO. 1 IS SKIPPED
%             - BIT 1 (P1D)=1 :PARAMETER NO. 2 IS SKIPPED
%             - BIT 2 (P2D)=1 :PARAMETER NO. 3 IS SKIPPED
%             - BIT 3 (P3D)=1 :PARAMETER NO. 4 IS SKIPPED
%             - BIT 4 (P4D)=1 :PARAMETER NO. 5 IS SKIPPED
%             - BIT 5 (P5D)=1 :PARAMETER NO 6 IS SKIPPED
%             - PARDEF IS RESET FOR EACH CALL TO GCOM, AND IT IS UPDATED
%             - FOR EACH CALL TO GPAR/KGPAR
% PARDGET     - PARAMETER LISTE USED WHEN ACCESSING THE SCRATCH FILE
% PARNO       - CONTAINS THE PARAMETER NUMBER IN A COMMAND, PARNO IS UPDATED
%             - FOR EACH CALL TO GPAR/KGPAR, AND PARNO IS RESET FOR EACH CALL
%             - TO GCOM
% PARRTF      - IN SINTRAN - PARAMETER LISTE FOR ACCESSING THE RTFIL FILE
% PB          - PROGRAM BASE (FIRST ADDRESS-1 OF A BRF UNIT)
% PCIMSEG     - ADDRESS OF CIMSEG, PARAMETER LISTE FOR THE
%             - "WRITE-SEGMENT-BACK" MONITOR CALL
% PRIADR      - ADDRESS OF SYMBOL TO PRINT NOW IN THE "LIST COMMANDS"
%             - AFTER THE LIST-IN-ALPHABETICAL-ORDER COMMAND OR
%             - THE LIST-AFTER-INCREASING-ADDRESSES COMMAND HAS BEEN USED
% PRIBT       - BRF CONTROL BYTE FOR PROGRAM PRIORITY (25 OCT)
% PRMCALL     - ADDRESS OF THE ENTRY IN RESIDENT CORE TO ENTER WHEN
%             - READING FROM THE RTFIL FILE. (PARAMETER LISTE FOR MCALL)
% PR1COMADR   - ADDRESS OF THE START OF THE COMMON AREA ON SEGN1
% PR2COMADR   - ADDRESS OF THE START OF THE COMMON AREA ON SEGN2
% PSY1,PSY2,PSY3 - LOCATIONS USED FOR COMPARING A SYMBOL IN SY1-SY3
%             - AGAINST A SYMBOL IN PSY1-PSY3 TO SEE IF THIS SYMBOL (IN SY1-SY3)
%             - SHALL BE PRINTED NOW
% PWMCALL     - ADDRESS OF THE ENTRY IN RESIDENT CORE TO ENTER WHEN
%             - WRITING ON THE RTFIL FILE (PARAMETER LISTE FOR MCALL)
% PWORKA      - POINTER TO THE ARRAY WORKA, USED BY THE COMMAND DECODING ROUTINES
% P0D         - BIT 0 IN PARDEF (SEE PARDEF)
% P3LLIM      - LOWEST LEGAL ADDRESS ON PAGE INDEX TABLE 0
% P0ULIM      - UPPER LEGAL ADDRESS ON PAGE INDEX TABLE 0
% P1D         - BIT 1 IN PARDEF (SEE PARDEF)
% P2D         - BIT 2 IN PARDEF (SEE PARDEF)
% P3D         - BIT 3 IN PARDEF (SEE PARDEF)
% P4D         - BIT 4 IN PARDEF (SEE PARDEF)
% P5D         - BIT 5 IN PARDEF (SEE PARDEF)
```

```
% RENTSEGM   - LOCATION IN RT DESCRIPTION TABLE ENTRY, CONTAINING
%            - THE SEGMENT NUMBER OF THE RT-PROGRAMS'S CURRENT
%            - REENTRANT SEGMENT
% RTFILSIZE  - SIZE OF ONE RTFIL-TABLE ELEMENT (7 LOCATIONS)
% RINOPAR    - PARAMETER LISTE USED WHEN ACCESSING THE INPUT FILES
% RMCSEG     - SEGMENT NUMBERS USED IN THE MCALL MONITOR CALL WHEN
%            - ENTERING THE SUBROUTINE IN RESIDENT CORE TO READ FROM
%            - THE RTFIL FILE TO THE RTFIL BUFFER
% RP         - BIT 14 IN THE PCR REGISTER (READ PERMITTED)
% RTFBL      - IN SINTRAN - BLOCK NUMBER TO ACCESS ON THE RTFIL FILE
% RTFIL      - IN SINTRAN - "OPEN FILE NUMBER" OF THE RTFIL FILE
% RTDSIZE    - SIZE OF ONE RT-DESCRIPTION TABLE ELEMENT
% RTFBU      - ADDRESS TO START OF RTFIL-BUFFER ON THE RTFIL SEGMENT
% RTFPAGE    - IN SINTRAN - FIRST PAGE IN SEGMENT ADDRESS AREA
%            - ON PAGE INDEX TABLE 1
% RTFSEG     - SEGMENT NUMBER OF RTFIL SEGMENT
% RTFRESPAR  - PARAMETER LISTE USED FOR RESERVE/RELEASE OF THE RTFIL FILE
% RTLPAGE    - IN SINTRAN - LAST PAGE IN SEGMENT ADDRESS AREA
%            - ON PAGE INDEX TABLE 1
% RTLSEG     - SEGMENT NUMBER OF RT LOADER
% RTOFF      - BIT 14 IN ACTPRI IN RT DESCRIPTION TABLE ELEMENT
% RTSTART    - IN SINTRAN - ADDRESS OF START OF RT DESCRIPTION TABLE
% SAVTAD     - (REAL) WORKING LOCATIONS FOR THE ENTER AND LEAVE SUBROUTINES
% SAVX       - WORKING LOCATION FOR THE SUBROUTINES ENTER & LEAVE
% SCRFIL     - CONSTANTS=100, FILE NUMBER OF THE SCRATCH FILE
% SECBL      - BLOCK NUMBER USED WHEN ACCESSING THE CURRENT SEGMENT FILE
% SECFPAR    - PARAMETER LISTE USED WHEN ACCESSING THE SEGMENT FILES
% SECLINK    - LOCATION IN THE SEGMENT TABLE ENTRY
%            - LINK FOR THE "SEGMENTS-IN-MEMORY" QUEU, AND
%            - WHICH MAY BE SWAPPED OUT (NOT FIXED)
% SECLLIM    - LOWEST LOGICAL ADDRESS LEGAL TO LOAD INTO
% SECM       - LOCATION IN THE RT DESCRIPTION ENTRY
%            - BIT 0-7: 1ST SEGMENT OF RT PROGRAM
%            - BIT 8-15: SECOND SEGMENT OF RT PROGRAM
% SECMLIM    - CONSTANT=100000, DEFAULT LOWEST LOGICAL ADDRESS OF THE
%            - SECOND NEW-SEGMENT (SEGN2)
% SECNP      - POINTS TO PARAMETER STRING "SEGMENT NO,:"
% SECN1      - FIPST NEW-SEGMENT SPECIFIED IN A LOAD OPERATION
% SECN2      - SECOND NEW-SEGMENT SPECIFIED IN A LOAD OPERATION
% SECSIZE    - SIZE OF A SEGMENT-TABLE ENTRY (5 LOCATIONS)
% SECSTART   - IN SINTRAN - ADDRESS OF START OF THE SEEGMENT TABLE
%            - (END OF RT DESCRIPTION TABLE)
% SECTBIT    - BUFFER CONTAINIG THE ADDRESSES OF THE SEGMENT FILES
%            - BIT MAPS
% SECULIM    - UPPER LOGICAL ADDRESS LEGAL TO LOAD INTO
% SEC0LLIM   - LOWER ADDRESS OF RTCOMMON AREA
% SEC0ULIM   - UPPER ADDRESS OF RTCOMMON AREA
% SEC1LLIM   - LOWEST ADDRESS USED ON SEGN1
% SEC1TYP    - SEGMENT TYPE OF THE FIRST NEW-SEGMENT (ND/DM & RING)
% SEC1ULIM   - UPPER ADDRESS USED ON SEGN1
% SEC2LLIM   - LOWEST ADDRESS USED ON SEGN2
% SEC2TYP    - SEGMENT TYPE OF THE SECOND NEW-SEGMENT (ND/DM & RING)
% SEC2ULIM   - UPPER ADDRESS USED ON SEGN2
```

```
% SERRFL      - FLAG SET EQAL ONE WHEN AN SERIOUS ERROR IS DETECTED
%             - NO MORE LOADING IS LEGAL WHEN THIS FLAG IS SET
%             - SERRFL IS RESET BY THE RESET-LOADER COMMAND
% SGFILNO     - FILE NUMBER OF THE CURRENTLY USED SEGMENT FILE
% SG1ADR      - POINTER TO ADDRESS OF ENTRY NO, 1 IN THE SEGMENT TABLE
% SGFTABLE    - 4 FIRST LOCATIONS MARK IF THE RESPECTIVE SEGMENT FILE
%             - IS USED OR NOT, CONTENT EQUAL 0 MEANS SEGMENT FILE NOT DEFINED
%             - LOCATION EQUAL 1 MEANS SEGMENT FILE IS DEFINED
%             - THE 4 NEXT LOCATIONS CONTAINS THE ADDRESSES OF THE RESPECTIVE
%             - SEGMENT FILE NAMES
% SIZF        - BUFFER CONTAINING THE SIZE IN PAGES OF THE DEFINED SEGMENT
%             - FILES
% SKIPFL      - FLAG FOR SKIP/NOT SKIP BRF LIBRARY UNITS, SKIPFL=1 IF SKIP
% SPACE       - CONSTANT=40, ASCII CODE FOR SPACE
% SRTFIL      - BUFFER CONTAINING THE FILE NAME '(SYSTEM)RTFIL:DATA'
% SSTEND      - BUFFER CONTAING THE SYMBOL "STEND" IN "BRF FORMAT"
% STACKL      - CURRENT STACK LENGTH, USED IN THE REENTRANT-LOAD COMMAND
% STADR       - LOCATION IN THE RT DESCRIPTION ENTRY
%             - START ADDRESS OF RT PROGRAM
% STATUS      - LOCATION IN THE RT DESCRIPTION ENTRY
%             - BIT 0-7: RT PROGRAM PRIORITY
%             - BIT 8-9: PROTECTION RING
% STBEG       - START OF THE LOADER STACK
% STDELTA     - SIZE OF ONE ELEMENT IN THE LOADER STACK (15 OCT)
% STEND       - END OF THE LOADER STACK
% STPNT       - POINTER TO CURRENT STACK TOP
% SXCLC       - BIT 3 IN CNZERO (SEE CNZERO)
% SXCL1       - BIT 4 IN CNZERO (SEE CNZERO)
% SXCL2       - BIT 5 IN CNZERO (SEE CNZERO)
% SXS0        - BIT 0 IN CNZERO (SEE CNZERO)
% SXS1        - BIT 1 IN CNZERO (SEE CNZERO)
% SXS2        - BIT 2 IN CNZERO (SEE CNZERO)
% SYMBP       - POINTER TO THE PARAMETER FILE TYPE :SYMB
% SY1,SY2,SY3- LOCATIONS FOR UNPACKING PACKED SYMBOLS BEFORE PRINTING THEM
% S1LOGADR    - FIRST LOGICAL PAGE AND NUMBER OF PAGES IN THE
%             - FIRST NEW-SEGMENT (SEGN1), BIT 0-7: FIRST LOGICAL PAGE IN SEGN1
%             - BIT 8-15: NUMBER OF PAGES IN SEGN1
% S1LONG      - LENGTH OF SEGN1 IN PAGES
% S1MADR      - PAGE ADDRESS OF THE FIRST NEW-SEGMENT (SEGN1) ON
%             - THE SEGMENT FILE, RELATIVE TO THE START OF THE SEGMENT FILE
% S2LOGADR    - FIRST LOGICAL PAGE AND NUMBER OF PAGES IN THE
%             - SECOND NEW-SEGMENT (SEGN2), BIT 0-7: FIRST LOGICAL PAGE IN SEGN2
%             - BIT 8-15: NUMBER OF PAGES IN SEGN2
% S2LONG      - LENGTH OF SEGN2 IN PAGES
% S2MADR      - PAGE ADDRESS OF THE SECOND NEW-SEGMENT (SEGN2) ON
%             - THE SEGMENT FILE, RELATIVE TO THE START OF THE SEGMENT FILE
% TEMP        - BIT 5 IN FLAG IN LINKING TABLE ELEMENT
%             - TEMP=1 MEANS SYMBOL TEMPORARLY NOT AVAILABLE FOR LOADING
% TGARB       - FLAG SET TO ONE WHEN THE LINKING TABLE IS EXPANDED TO
%             - USE THE SCRATCH FILE FOR REFERENCES TO COMMON LABELS
```

```
%  TREG          - T REGISTER FOR THE SUBROUTINES, SAVED IN STACK BY ENTER
%  USEGM         - IN SINTRAN - FIRST SEGMENT THE RT LOADER MAY USE
%  WAREG         - WORKING LOCATION FOR THE SUBROUTINES NOT USING ENTER AND LEAVE
%  WBRTFLAG      - FLAG SET TO ONE WHEN THE RTFIL-BUFFER IS CHANGED,
%                - THEN THE RTFIL-BUFFER IS WRITTEN TO THE FILE RTFIL
%                - IN THE EXIT-LOADER COMMAND
%  WLINK         - LOCATION IN THE RT DESCRIPTION ENTRY
%                - USED AS QUEU LINK IF RT PROGRAM IS IN THE EXECUTION QUEU
%                - OR IN A WAITING QUEU
%  WMCSEG        - SEGMENT NUMBERS USED IN THE MONITOR CALL MCALL WHEN
%                - ENTERING THE SUBROUTINE IN RESIDENT CORE TO WRITE THE RTFIL
%                - BUFFER TO THE RTFIL FILE
%  WORKA         - CHARACTER STRING BUFFER
%  WP            - BIT 15 IN THE PCR REGISTER (WRITE PERMITTED)
%  WXREG         - WORKING LOCATION USED BY THE SUBROUTINES NOT CALLING ENTER/LEAVE
   W1,W2,W3      - SAVES SYMBOLS WHILE READING BRF INFORMATION
                 - AND CONTAINS THE NAME OF THE SYMBOLS WHILE
%                - SEARCHING FOR SYMBOLS WHITH A SPECIFIED NAME IN THE
%                - LINKING TABLE OR IN THE RTFIL TABLE
%  XREG          - X REGISTER FOR THE SUBROUTINES, SAVED IN STACK BY ENTER
%  XSGF0         - BUFFER CONTAING THE NAME OF SEGMENT FILE NO. 0
%  XSGF1         - BUFFER CONTAING THE NAME OF SEGMENT FILE NO. 1
%  XSGF2         - BUFFER CONTAING THE NAME OF SEGMENT FILE NO. 2
%  XSGF3         - BUFFER CONTAING THE NAME OF SEGMENT FILE NO. 3
%  5CBUFSIZE     - SIZE OF COMMAND-LINE BUFFER
%  5CHCOM        - FLAG FOR MATCHING/NOT MATCHING 5 CHARACTERS MAC-ADS-SYMBOL
%                - WITH 6/7 CHARACTERS FORTRAN COMMON LABEL
%  5MXCHAR       - MAX CHARS ON ONE LINE
%  5NASEM        - LOG. DEV. NO. OF SEMAPHORE TO RESERVE WHEN ACCESSING RTFIL FILE
%  9ERW1         - FIRST OCTAL NUMBER TO USE IN A SERIOUS ERROR MESSAGE
%  9ERW2         - SECOND OCTAL NUMBER TO USE IN A SERIOUS ERROR MESSAGE
%
```

* * * * * * * * * * SEND US YOUR COMMENTS!!! * * * * * * * * * *

Are you frustrated because of unclear information in this manual? Do you have trouble finding things? Why don't you join the Reader's Club and send us a note? You will receive a membership card - and an answer to your comments.

Please let us know if you

* find errors
* cannot understand information
* cannot find information
* find needless information

Do you think we could improve the manual by rearranging the contents? You could also tell us if you like the manual!!

* * * * * * * * * HELP YOURSELF BY HELPING US!! * * * * * * * * *

Manual name: Sintran III REAL TIME LOADER-SYSTEM DOCUMENTATION

Manual number: ND-60.072.02

What problems do you have? (use extra pages if needed) _____

Do you have suggestions for improving this manual? _____

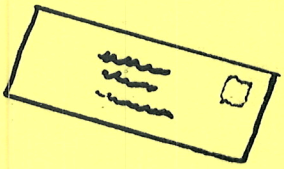
Your name: _____ Date: _____

Company: _____ Position: _____

Address: _____

What are you using this manual for? _____

Send to: Norsk Data A.S.
Documentation Department
P.O. Box 4, Lindeberg Gård
Oslo 10, Norway



Norsk Data's answer will be found on reverse side

Answer from Norsk Data: _____

Answered by: _____ Date: _____

Norsk Data A.S
Documentation Department
P.O. Box 25 BOGERUD
N - 0621 OSLO 6 - Norway

Systems that put people first