NAME

feclos – close a sub-device.

SYNOPSIS

```
#define FE_CLOS          0x4

feclos(dev, req, rpk, cpk)
struct  nd5_dev dev;
struct  nd5_req req;

struct rpk        {
        short    completion;
};
```

GENERIC DEVICES

1	disk
2	tape
3	terminal-in
4	terminal-out
5	invalid(clock)
6	invalid
7	xmsg
8	invalid(si)

SUB-DEVICES

Any subdevice codes corresponding to already open sub-devices on the generic device are allowed. All other sub-devices cause an error.

VALID QUALIFIERS

synchronous and asynchronous.

DESCRIPTION

The *feclose* monitor call closes a currently open sub-device. After this call has been issued no more I/O may be performed until the sub-device has been opened again. The ND-100 should cause all pending I/O to be completed before returning from this call.

COMMAND PACKET FORMAT

None.

RESPONSE PACKET FORMAT

Standard (see § 4.1.3).

NOTES

This call is always used synchronously.

NAME

fedctl – control an open sub-device.

SYNOPSIS

```
#define FE_DCTL         0x9

fedctl(dev, req, rpk, cpk)
struct  nd5_dev dev;
struct  nd5_req req;

struct cpk_intr {
        /* empty */
};

struct cpk_disk {
        short operation;
        short format_no;
};

struct cpk_tape {
        short operation;
        short parameter;
};

struct cpk_term {
        short operation;
        short parameter;
};

struct cpk_xmsg {
        short request;
        short parameter;
};

struct rpk_xxxx {                       /* terminal, si */
        short   completion;
};

struct rpk_disk {
        short   completion;
        short   status;
};

struct rpk_tape {
        short   completion;
        short   status;
        short    ops;
};
```

GENERIC DEVICES

1	disk
2	tape
3	terminal-in
4	invalid(terminal-out)
5	invalid(clock)
6	invalid
7	xmsg
8	software interrupt

SUB-DEVICES

Any open sub-devices on the above generic devices.

VALID QUALIFIERS

Synchronous or asynchronous. Terminal operations are always synchronous. All others are always asynchronous.

DESCRIPTION

The *fedctl* monitor call is used to perform certain, non I/O, functions. The operation is either synchronous or asynchronous. In asynchronous mode the completion is signalled via an interrupt on the appropriate device.

COMMAND PACKET FORMAT

Varies slightly according to the device:

DISK

name	offset/size		meaning
operation	0	2	1 set floppy format (spec by param)
			2 format floppy(80 tracks)
parameter	2	2	format number

These commands will cause an error return if performed on a non-floppy device. The format-floppy (2) command will cause the format to be done according to the currently set format.

TAPE

name	offset/size		meaning
operation	0	2	1 forward space n files (specified by param)
			2 backward space n files
			3 forward space n records
			4 backward space n records
			5 rewind
			6 rewind and unload
			7 get drive h/w status
			8 write EOF (currently using 0)
parameter	2	2	Optional parameter n if required by func.

TERMINAL

name	offset/size		meaning
operation	0	2	1 Flush output (generic device 4)
			2 change baudrate (specified by param).
			3 change flags (param specifies new flags).
			4 invalid
			5 drop dtr
			6 define xon/xoff characteristics (spec by parameter & pchars)
parameter	2	2	Optional parameter as required by func:
			baudrate #define B0 0
			#define B50 0000001
			#define B75 0000002
			#define B110 0000003
			#define B134 0000004
			#define B150 0000005
			#define B200 0000006
			#define B300 0000007
			#define B600 0000010
			#define B1200 0000011
			#define B1800 0000012
			#define B2400 0000013
			#define B4800 0000014
			#define B9600 0000015
			#define EXTA 0000016
			#define EXTB 0000017
			Control Flags #define EVEN 00 (even parity)
			#define ODD 01 (odd parity)
			#define RAW 02 (raw mode)
			Xon/Xoff #define X_RAW 00
			#define X_TANDEM 01
			#define X_DECCTQ 02
pchar1	4	2	Optional parameter - xoff character
pchar2	6	2	Optional parameter - xon character

Terminal fedctl(if) calls will be issued to the input subdevice only, but will affect both input and output devices, and must always be synchronous. The X_TANDEM parameter is always accompanied by 2 optional characters to specify the xoff character and the xon. The X_DECCTQ parameter is accompanied by one xoff character as in this mode any character acts as an xon character. Due to speed considerations, all flow-control handling must be performed in the ND-100.

XMSG

name	offset/size		meaning
request	0	2	1 Kick
			2 Wait for buffer to have n bytes free
parameter	2	2	number of bytes required by request 2 above.

SOFTWARE INTERRUPT

A fedctl call from this device is used to schedule an interrupt at ipl 4 from the ND-100 on the software interrupt generic device. The interrupt is queued to be handled by NDIX when the current ipl drops below 4. (i.e. no action except to generate a response packet).

RESPONSE PACKET FORMAT

Varies slightly according to the device:

DISK, TAPE
Standard (see § 4.1.3).

TERMINALS, SOFTWARE INTERRUPT
Standard (completion code only).

XMSG
None.

NAME

feerrm – send error code to ND-100

SYNOPSIS

```
#define FE_ERRM        0xE

feerrm(dev, req, req, cpk)
struct  nd5_dev dev;
struct  nd5_req req;

struct cpk {
        short error_code;
};
```

GENERIC DEVICES

Valid on all Generic Devices (even non-existent ones). The Generic device/subdevice combination specifies the device causing the error.

VALID QUALIFIERS

synchronous only.

DESCRIPTION

The *feerrm* monitor call is used to send an error code to the ND-100, following the detection of an inconsistency or non-fatal error during ND-100/ND-500 communications. The error is associated with a particular sub-device.

COMMAND PACKET FORMAT

name	offset/size		meaning
error_code	0	2	NDIX Error Code

RESPONSE PACKET FORMAT

Standard (see §4.1.3).

NAME

feexit – shutdown the NDIX system.

SYNOPSIS

```
#define FE_EXIT         0xB

feexit(dev, req, rpk, cpk)
struct  nd5_dev dev;
struct  nd5_req req;

struct cpk {
        short   s3_how;
        long    howto, rootdev;
        short   contigno, pageno, condev;
        long    crshdev, crshst, crshend;
        char    message[80];
        char    bootme[256];
};
```

GENERIC DEVICES

None, ignored.

SUB-DEVICES

None, ignored.

VALID QUALIFIERS

synchronous, it never returns.

DESCRIPTION

The *feexit* monitor call reverses the effect of the feinit(if) call by shutting down the NDIX system running in the ND-500 processor. According to the value of *s3_how* NDIX is either disabled totally or rebooted from the SINTRAN-III file whose pathname is given in bootme.

The number of contiguous pages required by the NDIX kernel may be indicated, if the number previously allocated was not sufficient, and the minimum number of pages that should be allocated for use by NDIX in the initial bitmap. These numbers should be retained, and used, by the interface in between warm starts.

The values of how and rootdev should be also be stored in a static area, and passed back to NDIX at the next feinit(if) call. This monitor call never returns to the caller.

COMMAND PACKET FORMAT

```
name          offset/size      meaning

s3_how           0     2        Bit coded field indicating the reboot
action
                               to be taken by the ND-100.
                               Bit 0:  0  shutdown completely
                                       1  reboot
                                   1:  reset stored values
                                   2:  reboot with specified contiguous
area
                                       (not implemented)
                                   3:  reboot with specified no of xtra
pages
```

```
                                             (not implemented)
                                         4:  take crash dump before reboot
                                             (not      implemented)      howto
2    4       how to reboot NDIX (single/multiuser) rootdev               6
4       NDIX root filesystem device contigno       10    2             no
of  contig  pages  required  by  NDIX  (unused)  pageno        12     2
min  no  of  pages  in  first  bitmap  (unused)  condev        14     2
NDIX    console    minor    device    no    crshdev         16        4
device/subdevice  for  crash  dump(not  implemented)  crshst          20
4       block  offset  for  start  of  crash  dump  crshend     24     4
crash  dump  must  not  go  beyond  this  block  message       28    80
console message.  bootme         108  256       system file pathname.
```

In all cases the null terminated string 'message' should be printed once on the NDIX system console and once on the SINTRAN-III system console. If bit 1 is set in *s3_how* then NDIX should be rebooted from the image contained in the SINTRAN-III data file whose name is given in the null terminated string 'bootme'. For example 'bootme' might contain the string:

(SYSTEM)VMUNIX

In this case NDIX would be booted from the files (SYSTEM)VMUNIX:PSEG and (SYSTEM)VMUNIX:DSEG, which must be present in the main directory on the SINTRAN-III filesystem. If bit 1 is set then the values of *howto and rootdev* should be saved in a manner that associates them permanently with the file named in *bootme* and passed back in the feinit call when the file is booted.

RESPONSE PACKET FORMAT

None.

NAME

feidev – initialise a generic device.

SYNOPSIS

```
#define FE_IDEV          0x2

feidev(dev, req, rpk, cpk)
struct  nd5_dev dev;
struct  nd5_req req;

struct cpk_xxxx {
        short ipl;
};

struct cpk_term {
        short ipl;
        long   pparam;
};

struct cpk_xmsg {
        short ipl;
        long   paralist;
};

struct rpk_xxx   {
        short    completion;
        short    subdevc;
};

struct rpk_terminal {
        short    completion;
        long     locdevm;
        short    remdevc;
};

struct rpk_clock        {
        short    completion;
        struct s3_date now;       /* the date in SIII format */
};
```

GENERIC DEVICES

number	meaning
1	disk
2	tape
3	terminal input
4	terminal output
5	clock
6	invalid
7	xmsg
8	software interrupt

SUB-DEVICES

None, ignored.

VALID QUALIFIERS

synchronous

DESCRIPTION

The *feidev* monitor call is used to initialise a generic device, prior to I/O and before any *feopen(if)* calls are issued on its subdevices. The monitor call may only be used synchronously.

COMMAND PACKET FORMAT

Varies according to the dievice:

name	offset/size		meaning
DISK, TAPE, SI			
ipl	0	2	Interrupt priority level for this generic device
TERMINAL			
ipl	0	2	Interrupt priority level for this generic device
pparam	2	4	ND100 address of shared character counters
XMSG			
ipl	0	2	Interrupt priority level for this generic device
paralist	2	4	ND100 address of xmsg parameter list buffer

RESPONSE PACKET FORMAT

Varies according to the device:

name	offset/size		meaning
CLOCK			
completion	0	2	0 command completed ok n interface error code
now	2	12	SINTRAN-III format date (must be GMT)
TERMINAL-INPUT			
completion	0	2	0 command completed ok n interface error code
locdevm	2	4	32 bit mask of local devices present. Each bit set on indicates 4 terminal lines physically present.
remdev	6	2	count of remote devices
DISK TAPE XMSG			
completion	0	2	0 command completed ok n interface error code
subdevc	2	2	count of valid logical subdevices available for use

If the generic device being initialised is the clock then it will begin interrupting immediately, at 40ms intervals. This differs from the other devices which must be opened using the FE_OPEN call before any I/O can be performed.

NAME

feinit – system initialisation

SYNOPSIS

```
#define FE_INIT        0x1

feinit(dev, req, rpk, cpk)
struct   nd5_dev dev;
struct   nd5_req req;

struct cpk {
         short   ux_vers;
         long    cxbadd;
         long    errcxb,errdata;
         long    intvec;
         long    errvec;
};

struct rpk {
         short   completion;
         long    howto, rootdev;
         short   condev;
         long    spst, scont, sndix;
         long    stext, sdata, sstack, sfree, sbuffer, sphys;
         long    private;
         short   cputype, s3_vers;
         long    sharedseg;
         short   contigno, pageno;
         char    booted[256];
};
```

GENERIC DEVICES

Always zero. Ignored.

SUB-DEVICES

None, ignored.

VALID QUALIFIERS

synchronous

DESCRIPTION

Feinit is a monitor call made once at system startup to tell the ND-100 that the NDIX program has been correctly loaded and to pass basic system parameters between NDIX and the SINTRAN-III ND-500 driver. Unlike all the other monitor calls supported by the ND100/NDIX interface the addresses of the response and command packets are **ND-500 data segment logical addresses (KVA)** and not **ND-100 physical addresses.** This is necessary because the NDIX kernel does not yet know the extent of the ND-100 **private** memory and therefore cannot perform the relocation itself. All other monitor calls use ND-100 physical addresses, i.e. they relocate all the ND-500 physical addresses by the value **private.** The ND-100 should regard the failure of the NDIX program to make an feinit call within a reasonable time (say 10 seconds) as a fatal error. The operation of this call is always synchronous and the *dev* and *req.qualifier* call parameters should be ignored by the ND-100.

COMMAND PACKET FORMAT

name	offset/size		meaning
ux_vers	0	2	A version number for the NDIX system.
cxbadd	2	4	ND-500 logical addr interrupt cxb array
errcxb	6	4	ND-500 logical addr context block for traps
errdata	10	4	ND-500 logical addr trap data area
intvec	14	4	ND-500 logical addr of interrupt vector
errvec	18	4	ND-500 logical addr of fatal trap vector

RESPONSE PACKET FORMAT

name	offset/size		meaning
completion	0	2	0 command completed successfully
			n interface error code
howto	2	4	how to reboot NDIX (passed from previous exit)
rootdev	6	4	NDIX root filesystem device (passed from exit)
condev	10	2	NDIX console minor-device no (from exit or LOAD)
spst	12	4	ND100 phys addr of PST
scont	16	4	ND100 phys addr of NDIX contig area (ignored)
sndix	20	4	ND100 phys addr of NDIX process segment (ignored)
stext	24	4	ND100 phys addr of NDIX program segment
sdata	28	4	ND100 phys addr of NDIX data segment
sstack	32	4	ND100 phys addr of NDIX stack segment
sfree	36	4	ND100 phys addr of start of free memory
sbuffer	40	4	ND100 phys addr of end of contig area (ignored)
sphys	44	4	ND100 phys addr of start of non-existent memory
private	48	4	size of ND-100 private memory in bytes
cputype	52	2	The ND500 cputype on which system is running
s3_vers	54	2	SIII version in use
sharedseg	56	4	ND-100 phys addr of shared segment index page
contigno	60	2	Length of contiguous area in pages (ignored)
pageno	62	2	Number of pages in primary bitmap (ignored)
booted	64	256	system file pathname.

Booted This is a null-terminated string which specifies the name of the SINTRAN-III data file from which the NDIX system is being booted. For example 'bootme' might contain the string:

(SYSTEM)VMUNIX

In this case NDIX would have been booted from the files (SYSTEM)VMUNIX:PSEG and (SYSTEM)VMUNIX:DSEG.

NAME

feopen – open a subdevice for I/O.

SYNOPSIS

```
#define FE_OPEN          0x3

feopen(dev, req, rpk, cpk)
struct  nd5_dev dev;
struct  nd5_req req;

struct cpk_disk {
        short    format;
};

struct cpk_xmsg {
        long     datbuf;
};

struct rpk_disk {
        short   completion;
        long     devtype;
        long     frmsiz;
        short    secsiz;
};

struct rpk_tape {
        short   completion;
        short   status;
};

struct rpk_xxxx     {
        short   completion;
};
```

GENERIC DEVICES

number	meaning
1	disk
2	tape
3	terminal-in
4	terminal-out
5	invalid(clock)
6	invalid
7	xmsg
8	invalid(si)

SUB-DEVICES

Any sub-devices passed back when the generic device was feidev'd are allowed. All other sub-devices cause an error.

VALID QUALIFIERS

Synchronous and asynchronous.

DESCRIPTION

The *feopen* monitor call prepares a sub-device on a previously feidev'd generic device for I/O. The sub-device must be one of those passed back to this feidev call. Until an feopen call has been performed on a sub-device all I/O should be disallowed, following an feopen all I/O requests should be enabled.

For terminal devices configured for hard carrier an asynchronous *feopen* call is used. The open only succeeds when the carrier is present. Shared terminal devices must also use the aynchronous call, the open succeeds only when the user requests access to NDIX. For other (soft carrier) devices the synchronous call is used on open.

COMMAND PACKET FORMAT

DISK

name	offset/size		meaning
format	0	4	Format type for floppy disk.

XMSG

name	offset/size		meaning
datbuf	0	4	Physical ND-100 address of data buffer

RESPONSE PACKET FORMAT

Standard apart from disk.

DISK

name	offset/size		meaning
completion	0	2	0 command completed ok n interface error code
devtype	2	4	Code indicating type of disk
frmsiz	6	4	Size in sectors if code indicated SCSI disk or format number if code indicated floppy disk
secsiz	10	2	unused

The ***devsiz*** parameter allows NDIX to determine the maximum capacity of the disk drive in sectsiz byte blocks, which represent the smallest read or write request size performed by the NDIX disk drivers. The NDIX system assumes that these blocks can be addressed linearly in the range 0 -> devsiz-1.

NAME

fercon – synchronous read from system console

SYNOPSIS

```
#define FE_RCON          0x6

fercon(dev, req, rpk, cpk)
struct  nd5_dev dev;
struct  nd5_req req;

struct cpk {
        long     physaddr;
};

struct rpk       {
        short    completion;
};
```

GENERIC DEVICES

Only valid on generic device number 3. All other generic devices cause an error.

SUB-DEVICES

Unused.

VALID QUALIFIERS

synchronous only.

DESCRIPTION

The *fercon* monitor call is used to read single bytes from the terminal which has been designated the NDIX system console by SINTRAN-III. The operation of this command is always synchronous.

COMMAND PACKET FORMAT

name	offset/size		meaning
physaddr	0	4	ND-100 physical address input buffer.

The input byte should be placed in the least significant eight bits of the input buffer longword. This is identical to the functioning of the INBT Sintran III monitor call.

RESPONSE PACKET FORMAT

Standard (completion code only).

NAME

feread – read data from a sub-device.

SYNOPSIS

```
#define FE_READ          0x5

feread(dev, req, rpk, cpk)
struct  nd5_dev dev;
struct  nd5_req req;

struct cpk_disk {
        long     nbytes;
        long     physaddr;
        long     devaddr;
};

struct cpk_tape {
        long     maxbytes;
        long     physaddr;
};

struct cpk_terminal {
        long     dummy;
        long     physaddr;
};

struct rpk_xxxx {
        short    completion;
        short    status;
        long     nbytes;
};

struct rpk_term {
        short    completion;
        long     nbytes;
};
```

GENERIC DEVICES

type	meaning
1	disk
2	tape
3	terminal-in
4	invalid(terminal-out)
5	invalid(clock)
6	invalid
7	invalid(xmsg)
8	invalid(si)

SUB-DEVICES

Any open sub-devices are valid, otherwise an error return is generated.

VALID QUALIFIERS

Asynchronous only.

DESCRIPTION

The *feread* monitor call is used to transfer data between peripherals, attached to the ND-100, and the ND-500 physical memory. A completion interrupt should be generated.

COMMAND PACKET FORMAT

`name offset/size meaning`

DISK

nbytes	0	4	Number of bytes to read.
physaddr	4	4	ND-100 phys address of buffer to be read into.
devaddr	8	4	Logical 1024 byte block on the disk where the data is to be read from.

TAPE

maxbytes	0	4	Number of bytes to be read from the tape (the expected record size).
physaddr	4	4	ND-100 phys address of buffer to be read into.

TERMINAL

dummy	0	4	unused
physaddr	4	4	ND-100 phys address of buffer to be read into

RESPONSE PACKET FORMAT

Varies according to the device:

name offset/size meaning

DISK, TAPE

completion	0	2	0 command completed ok n interface error code
status	2	4	h/w status
nbytes	6	4	The number of bytes read

TERMINAL

completion	0	2	0 command completed ok n interface error code
nbytes	2	4	The number of bytes read

NAME

fewcon – synchronous write to system console

SYNOPSIS

```
#define FE_WCON          0x8

fewcon(dev, req, rpk, cpk)
struct  nd5_dev dev;
struct  nd5_req req;

struct cpk {
        long      physaddr;
};

struct rpk       {
        short    completion;
};
```

GENERIC DEVICES

Only valid on generic device number 4. All other generic devices cause an error.

SUB-DEVICES

Unused.

VALID QUALIFIERS

synchronous only.

DESCRIPTION

The *fewcon* monitor call is used to write single bytes to the terminal which has been designated the NDIX system console by SINTRAN-III. The operation of this command is always synchronous.

COMMAND PACKET FORMAT

name	offset/size		meaning
physaddr	0	4	ND-100 physical address output buffer.

The ND-100 should write the character, specified in the least significant eight bits of the 32 bit longword specified by *physaddr* to the system console. This mirrors the functionality of the OUTBT Sintran III monitor call.

RESPONSE PACKET FORMAT

Standard (completion code only).

NAME

fewrit – write data to a sub-device.

SYNOPSIS

```
#define FE_WRIT          0x7

fewrit(dev, req, rpk, cpk)
struct  nd5_dev dev;
struct  nd5_req req;

struct cpk_disk {
        long    nbytes;
        long    physaddr;
        long    devaddr;
};

struct cpk_tape {
        long    nbytes;
        long    physaddr;
};

struct cpk_terminal {
        long    nbytes;
        long    physaddr;
};

struct rpk_xxxx {
        short   completion;
        short   status;
};

struct rpk_term {
        short   completion;
};
```

GENERIC DEVICES

type	meaning
1	disk
2	tape
3	invalid
4	terminal-out
5	invalid(clock)
6	invalid
7	invalid(xmsg)
8	invalid(si)

SUB-DEVICES

Any open sub-devices are valid, otherwise an error return in generated.

VALID QUALIFIERS

Asynchronous only.

DESCRIPTION

The *fewrit* monitor call is used to transfer data between peripherals, attached to the ND-100, and the ND-500 physical memory. A completion interrupt should be generated on the appropriate subdevice.

COMMAND PACKET FORMAT

	name	offset/size		meaning
DISK				
	nbytes	0	4	Number of bytes to write.
	physaddr	4	4	ND-100 physical address of buffer containing data to write.
	devaddr	8	4	Logical 1024 byte block on the disk where the data is to be written to.
TAPE				
	nbytes	0	4	The number of bytes to be written to the tape (the record size).
	physaddr	4	4	ND-100 physical address of buffer containing data to write.
TERMINAL				
	nbytes	0	4	Number of bytes to be written to the terminal.
	physaddr	4	4	ND-100 physical address of buffer containing data to write.

RESPONSE PACKET FORMAT

Standard (see §4.1.3)