

Interrupt List, part 17 of 18

Copyright (c) 1989-1999,2000 Ralf Brown

-----N-6B-----

INT 6B - DECnet DOS - PORT DRIVER

InstallCheck: test for a signature area (see #03729) immediately preceding  
the interrupt handler

SeeAlso: INT 6A"DECnet",INT 6C"DECnet"

Index: installation check;DECnet DOS Port Driver

Format of DECnet DOS signature area:

Offset Size Description (Table 03729)

-5 BYTE major version number

-4 BYTE minor version number

-3 3 BYTES signature (ASCII "PDV")

-----v-6B-----

INT 6B - VIRUS - "Saddam" - ORIGINAL INT 21h VECTOR

SeeAlso: INT 21/AX=FFFFh,INT 61"VIRUS",INT 70"VIRUS"

-----h-6B-----

INT 6B C - HP Vectra AT - IRQ19 - RESERVED HARDWARE INTERRUPT

SeeAlso: INT 0B"IRQ3",INT 6A"HP Vectra",INT 6C"HP Vectra"

-----S-6B0000-----

INT 6B - Novell NASI/NACS, Ungermann-Bass Net One SERIAL I/O - BUFFERED WRITE

AX = 0000h

CX = length

ES:BX -> buffer

Return: CX = number of bytes written

Program: NASI is Novell's NetWare Asynchronous Services Interface (purchased  
from Network Products Corp, who call it NCSI) which runs on  
workstations; NACS is the NetWare Asynchronous Communications  
Services module which runs on servers

InstallCheck: test for the signature string "NCSI" three bytes past the  
interrupt handler; see also AH=02h.

As of version 3.0, Novell's NASI can be distinguished from NPC's  
NCSI by the presence of an 'A' immediately following the signature

Notes: this function is also supported by TelAPI, NPC NCSI, and Connection  
Manager CLIENT.EXE; for TelAPI, nonzero values in AL specify a  
connection ID

Connection Manager returns CF set/AL=FFh if called while an INT 6B  
call is already in progress

SeeAlso: AX=0100h,AH=18h,INT 14/AH=19h,INT 14/AH=E3h

-----S-6B0100-----

INT 6B - Novell NASI/NACS, Ungermann-Bass Net One SERIAL I/O - BUFFERED READ

AX = 0100h  
CX = length of buffer  
ES:BX -> buffer

Return: CX = number of bytes read

Note: also supported by TelAPI and NPC NCSI; for TelAPI, nonzero values in  
AL specify a connection ID

SeeAlso: AX=0000h,AH=19h,INT 14/AH=18h,INT 14/AH=E2h,INT 14/AX=FF02h

-----S-6B02-----

INT 6B - Novell NASI/NACS, Ungermann-Bass Net One SERIAL I/O - INSTALL CHECK

AH = 02h  
AL nonzero

Return: AL = 00h if present and OK

Note: this function is also supported by TelAPI and NPC NCSI

SeeAlso: AX=0700h

-----S-6B0600-----

INT 6B - Novell NASI/NACS, Ungermann-Bass Net One SERIAL I/O - CONTROL

AX = 0600h  
CX = command  
02h send break  
04h disconnect  
06h hold

Return: CF clear if successful

AL = 00h  
CF set on error  
AX < 0

Note: this function is also supported by TelAPI and NPC NCSI

-----S-6B0700-----

INT 6B - Novell NASI/NACS, Ungermann-Bass Net One SERIAL I/O - GET STATUS

AX = 0700h

Return: CH <> 00h if connection active

Notes: this function is also supported by TelAPI and NPC NCSI

Novell TelAPI returns CX=FF01h and CF clear

SeeAlso: AH=02h,AH=10h

-----N-6B08-----

INT 6B - TelAPI - CHECK FOR BREAK SIGNAL

AH = 08h  
AL = circuit number

Return: CF clear if no breaks

AL = 00h  
CX = 0000h

CF set if one or more breaks received

Note: this function also clears ??? flag

SeeAlso: INT 14/AX=FF00h

-----S-6B10-----

INT 6B - NPC NCSI EXTENDED SERIAL I/O - GET STATUS

AH = 10h

AL = connection ID (Novell TELAPI.EXE)

CX = ???

ES:BX -> buffer for status (see #03730)

Return: CF clear if successful

CL = ???

CH = ???

CF set on error

???

Notes: this function is also supported by TelAPI

when shelled out to an external protocol from Novell NASI, the circuit whose status indicates that it is connected is the currently active circuit

SeeAlso: AX=0700h,AH=12h,AH=1Fh

Format of NASI status:

Offset Size Description (Table 03730)

00h BYTE number of allocated virtual circuits

01h BYTE reserved

02h 9 BYTES states of emulated circuits 0-8

00h idle

01h attached Telnet session, in command state

02h attached Telnet session, connected with host

03h-06h ??? (not returned by TelAPI)

Note: this description is derived from the Novell TelAPI emulation of NASI

-----S-6B11--DX0001-----

INT 6B - NPC NCSI EXTENDED SERIAL I/O - ALLOCATE A VIRTUAL CIRCUIT

AH = 11h

DX = 0001h

AL = 00h

ES:BX -> service name string (8 characters, blank-padded)

Return: CF clear if successful

AL = virtual circuit number allocated (01h for Novell TELAPI.EXE)

CL = ??? (01h for Novell TELAPI.EXE)

CH = ??? (01h for Novell TELAPI.EXE)

CF set on error

???

Note: this function is also supported by TelAPI

SeeAlso: AH=12h,AH=15h,AH=16h,AH=17h,AH=18h

-----S-6B12-----

INT 6B - NPC NCSI EXTENDED SERIAL I/O - VIRTUAL CIRCUIT STATUS

AH = 12h

AL = virtual circuit number (0-8)

Return: CF clear if successful

AL = virtual circuit number

CL = virtual circuit's state

00h idle

01h Telnet session, in command state

02h Telnet session, connected to host

03h-06h ??? (not returned by TelAPI)

CF set on error

AL = error code

E2h invalid virtual circuit number

E4h specified virtual circuit not allocated

Note: this function is also supported by TelAPI

SeeAlso: AH=10h,AH=15h,AH=1Ah,AH=1Bh,AH=1Fh

-----S-6B13-----

INT 6B - NPC NCSI EXTENDED SERIAL I/O - SET/RETRIEVE REQUEST/REPLY SERVICE NAME

AH = 13h

AL = virtual circuit number

CL = direction (00h get, nonzero set)

ES:BX -> buffer for/containing service name

Return: ???

Note: this function is also supported by TelAPI

SeeAlso: AH=14h,AH=15h

-----S-6B14-----

INT 6B - NPC NCSI EXTENDED SERIAL I/O - SET/RETRIEVE SERVICE ADDRESS

AH = 14h

AL = virtual circuit number

ES:BX -> buffer for/containing service address

Return: ???

Note: this function is also supported by TelAPI, which only supports  
retrieving the address

SeeAlso: AH=13h,AH=15h,AH=21h

-----S-6B15-----

INT 6B - NPC NCSI EXTENDED SERIAL I/O - SET/RETRIEVE VIRTUAL CIRCUIT CONFIG

AH = 15h

AL = virtual circuit number  
CL = direction (00h get, nonzero set)  
ES:BX -> buffer for/containing virtual circuit config (see #03731)

Return: ES:BX buffer filled

Note: this function is also supported by TelAPI

SeeAlso: AH=13h"NCSI",AH=14h"NCSI"

Format of virtual circuit configuration:

Offset Size Description (Table 03731)

00h	WORD	buffer length
02h	WORD	port ID
04h	WORD	receive rate
06h	WORD	receive word length
08h	WORD	receive stop bits
0Ah	WORD	receive parity
0Ch	WORD	transmit rate
0Eh	WORD	transmit word length
10h	WORD	transmit stop bits
12h	WORD	transmit parity
14h	WORD	DTR
16h	WORD	RTS

-----S-6B16-----

INT 6B - NPC NCSI EXTENDED SERIAL I/O - LOG AND/OR INITIALIZE VIRTUAL CIRCUIT

AH = 16h  
AL = virtual circuit number  
CL = ??? switch (00h, ???)

Return: CF clear if successful

AL = virtual circuit number  
CF set on error  
???

Note: this function is also supported by TelAPI, which always returns CF  
clear and AL=00h

SeeAlso: AH=11h,AH=12h,AH=17h

-----S-6B17-----

INT 6B - NPC NCSI EXTENDED SERIAL I/O - DISCONNECT A VIRTUAL CIRCUIT

AH = 17h  
AL = virtual circuit number

Return: CF clear if successful

CF set on error  
???

Note: this function is also supported by TelAPI, which always returns CF

clear and AL=00h

SeeAlso: AH=11h,AH=16h

-----S-6B18-----

INT 6B - NPC NCSI EXTENDED SERIAL I/O - WRITE DATA ON A VIRTUAL CIRCUIT

AH = 18h

AL = virtual circuit number

CX = number of characters to send

ES:BX -> buffer containing characters to be sent

Return: CF clear if successful

CF set on error

???

Note: this function is also supported by TelAPI, which always returns CF

clear and AL=30h

SeeAlso: AX=0000h,AH=12h,AH=19h

-----S-6B19-----

INT 6B - NPC NCSI EXTENDED SERIAL I/O - READ DATA ON A VIRTUAL CIRCUIT

AH = 19h

AL = virtual circuit number

CX = number of characters to read

ES:BX -> buffer for received characters

Return: CX = 0000h if failed

CX = nonzero (possibly number of characters received) if successful

Note: this function is also supported by TelAPI

SeeAlso: AX=0100h,AH=12h,AH=18h

-----S-6B1A-----

INT 6B - NPC NCSI EXTENDED SERIAL I/O - RECEIVE STATUS

AH = 1Ah

???

Return: ???

SeeAlso: AH=12h,AH=1Bh

-----S-6B1B-----

INT 6B - NPC NCSI EXTENDED SERIAL I/O - TRANSMIT STATUS

AH = 1Bh

???

Return: ???

SeeAlso: AH=12h,AH=1Ah

-----S-6B1C-----

INT 6B - NPC NCSI EXTENDED SERIAL I/O - CLEAR RECEIVE BUFFER

AH = 1Ch

AL = circuit number

Return: nothing

SeeAlso: AH=1Dh"NCSI",AH=1Eh"NCSI"

-----S-6B1D-----

INT 6B - NPC NCSI EXTENDED SERIAL I/O - TRANSMIT BUFFER CONTROL

AH = 1Dh

???

Return: ???

SeeAlso: AH=1Ch,AH=1Eh

-----S-6B1E-----

INT 6B - NPC NCSI EXTENDED SERIAL I/O - ISSUE CONTROL REQUEST

AH = 1Eh

???

Return: ???

SeeAlso: AH=1Ch,AH=1Dh

-----S-6B1F-----

INT 6B - NPC NCSI EXTENDED SERIAL I/O - EXTERNAL STATUS

AH = 1Fh

???

Return: ???

SeeAlso: AH=10h,AH=12h

-----S-6B20-----

INT 6B - Connection Manager CLIENT.EXE - ???

AH = 20h

???

Return: ???

Program: Connection Manager by Softwarehouse Corp. permits the sharing of serial ports over an IPX or NetBIOS-based network

Note: CLIENT.EXE returns CF set/AL=F9h if AH is not 00h to 21h on entry

-----S-6B21-----

INT 6B - NPC NCSI EXTENDED SERIAL I/O - QUERY NAME SERVICE

AH = 21h

AL = virtual circuit number

CL = ??? (00h or 01h)

ES:BX -> buffer for service name structure (see #03732)

Return: CF clear if successful

ES:BX buffer filled

CF set on error

Notes: a program should call this function after allocating a virtual circuit and check that the general name matches the requested service

prior to version 3.0, Novell's NASI returned the first available port;

v3.0+ returns the first port found--check the returned status to

determine whether the port is available

SeeAlso: AH=14h"NCSI"

Format of NCSI service name structure:

Offset Size Description (Table 03732)

00h WORD buffer length  
02h 8 BYTES service name  
0Ah 8 BYTES general name  
12h 8 BYTES specific name  
1Ah 4 BYTES ???

---Novell NASI v3.0+ ---

1Eh BYTE port number  
1Fh BYTE port status  
    00h idle (available)  
    01h allocated (available)  
    02h connected  
    03h on hold

-----6B6B-----

INT 6B - Tandy SCHOOLMATE PLUS - API

AH = 6Bh  
AL = E0h to FFh

Note: details not yet available

-----6C-----

INT 6C - system resume vector (CONVERTIBLE)

-----6C-----

INT 6C - DOS 3.2 Realtime Clock update

-----N-6C-----

INT 6C - DECnet DOS network scheduler

InstallCheck: test for a signature area immediately preceding the interrupt  
handler (see #03733)

Note: this function is also supported by DEC Pathworks for DOS

SeeAlso: INT 6B"DECnet",INT 6D"DECnet",INT 6E"DECnet"

Index: installation check;DECnet DOS scheduler

Format of DECnet DOS signature area:

Offset Size Description (Table 03733)

-5 BYTE major version number  
-4 BYTE minor version number  
-3 3 BYTES signature (ASCII "SCH")

-----h-6C-----

INT 6C C - HP Vectra AT - IRQ20 - HP-HIL CONTROLLER INTERRUPT

SeeAlso: INT 0C"IRQ4",INT 6B"HP Vectra",INT 6D"HP Vectra"



## -----V-6D-----

INT 6D - VIDEO - many VGA - VIDEO BIOS ENTRY POINT

Desc: points at the original INT 10 entry point set up by the VGA BIOS

Notes: used by IBM, ATI VGA Wonder, Paradise, Video7, and NCR, and many others

the Diamond Stealth64 BIOS uses this vector to store the INT 42 vector

of the inactive video (see INT 10/AH=12h/BL=35h)

SeeAlso: INT 10/AH=00h,INT 10/AH=0Eh,INT 10/AH=12h/BL=10h

## -----V-6D-----

INT 6D - Trident SVGA - VIDEO BIOS HANDLER

Note: the BIOS INT 10 handler on various Trident VGA cards consists merely

of a call to INT 6D followed by an IRET.

## -----N-6D-----

INT 6D - DECnet DOS (before 2.1) - DATA LINK LAYER PROGRAM

AH = function

???

Return: ???

InstallCheck: test for a signature area immediately preceding the interrupt  
handler (see #03734)

SeeAlso: INT 69/AH=0Ah,INT 6C"DECnet",INT 6E"DECnet"

Index: installation check;DECnet DOS Data Link Layer

Format of DECnet DOS signature area:

Offset Size Description (Table 03734)

-5 BYTE major version number

-4 BYTE minor version number

-3 3 BYTES signature (ASCII "DLL")

## -----h-6D-----

INT 6D C - HP Vectra AT - IRQ21 - RESERVED HARDWARE INTERRUPT

SeeAlso: INT 0D"IRQ5",INT 6C"HP Vectra",INT 6E"HP Vectra"

## -----V-6DBD60-----

INT 6D U - Matrox Multiple Monitor Support v1.0 - ???

AX = BD60h

DS:SI -> buffer containing ??? (see #03735)

???

Return: AX = 00BDh if function executed

???:SI -> 22-byte buffer (inside TSR) filled with ??? (see #03735)

AX unchanged if function disabled

Note: this function is a NOP if AX=BD6Fh/BL=01h was called

SeeAlso: AX=BD61h,AX=BD6Fh,AX=BDFFh

Format of Matrox ??? buffer:

```

Offset  Size  Description (Table 03735)
00h    4 BYTES ???
04h   WORD  video mode width in pixels
06h   WORD  video mode height in pixels
08h   BYTE  ???
09h   BYTE  ???
0Ah   BYTE  ???
0Bh   BYTE  flags
        bit 1: ???
        bit 4: ??? (corresponds to bit 14 for AX=BD61h)
        bit 5: ??? (corresponds to bit 15 for AX=BD61h)
0Ch   DWORD -> ??? buffer
10h   WORD  ???
        bit 4: ??? (corresponds to bit 14 for AX=BD61h)
        bit 5: ??? (corresponds to bit 15 for AX=BD61h)
12h   4 BYTES ???

```

SeeAlso: #03736

-----V-6DBD61-----

INT 6D U - Matrox Multiple Monitor Support v1.0 - GET VIDEO MODE INFORMATION

AX = BD61h

BX = width in pixels (bits 13-0)

bit 14: ???

bit 15: ???

CX = height in pixels

Return: AL = BDh if function supported

AH = status

00h successful

01h failed (resolution/flags not supported)

CX:BX -> video mode data structure (see #03736)

(first word is FFFFh if unsupported resolution)

EAX high word cleared

SeeAlso: AX=BD60h,AX=BD6Fh,AX=BDFFh

Format of Matrox video mode information (VCE structure):

```

Offset  Size  Description (Table 03736)

```

00h WORD width in pixels

02h WORD height in pixels

04h WORD ???

06h DWORD dot clock frequency

0Ah 25 BYTES ???

SeeAlso: #03735

-----V-6DBD6F-----

INT 6D U - Matrox Multiple Monitor Support v1.0 - INSTALLATION CHECK

AX = BD6Fh

BL = subfunction

00h installation check

01h disable TSR

02h enable TSR (see AX=BD60h)

Return: AX = 00BDh if installed

SeeAlso: AX=BD60h,AX=BD61h,AX=BDFh

-----V-6DBDFF-----

INT 6D U - Matrox Multiple Monitor Support v1.0 - UNHOOK INTERRUPT (UNINSTALL)

AX = BDFh

Return: AX = 00BDh if supported

Note: this function reverts INT 6D without checking that it points at the

TSR, and does not release the memory occupied by the TSR

SeeAlso: AX=BD60h,AX=BD61h,AX=BD6Fh

-----N-6E-----

INT 6E - DECnet DOS - DECnet NETWORK PROCESS API

InstallCheck: test for the signature/data area (see #03737) immediately prior  
to the interrupt handler

Note: this is the main DECnet DOS access, and is described in Digital manual

AA-EB46B-TV ("DECnet-DOS Programmer's Reference Manual")

Index: installation check;DECnet DOS

Format of DECnet DOS signature area:

Offset Size Description (Table 03737)

-5 BYTE major version number

-4 BYTE minor version number

-3 3 BYTES signature (ASCII "DNP")

-----h-6E-----

INT 6E C - HP Vectra AT - IRQ22 - RESERVED HARDWARE INTERRUPT

SeeAlso: INT 0E"IRQ6",INT 6D"HP Vectra",INT 6F/AH=00h"HP Vectra"

-----N-6F-----

INT 6F - Novell NetWare - PCOX API (3270 PC terminal interface)

AX = function

0000h enter terminal mode

Return: AX = status

0000h no action requested

0001h screen save

0001h reset interface and set configuration parameters

DX = bitfields

bits 2-0: model number  
bits 4-3: I/O address  
bits 6-5: DMA channel  
Return: nothing  
0002h set display parameters  
DX = bitfields  
bits 1-0: OIA mode  
bits 4-2: monitor support  
Return: nothing  
0003h read status  
Return: AX = status word (see #03738)  
0004h read cursor position  
Return: AX = cursor position  
0005h get character from device buffer  
DX = cursor position  
Return: AH = type (00h data, 01h attribute)  
AL = data or attribute character  
0006h send character  
DH = type (00h ASCII, 01h extended code)  
DL = ASCII character or extended code  
Return: nothing  
0007h set timeout  
DX = timeout in seconds  
Return: nothing  
0008h wait for location to be modified  
DX = cursor position  
Return: AX = status (0000h modified, nonzero timeout)  
0009h NOP  
000Ah restore display  
Return: nothing  
000Bh update device buffer  
Return: AX = cursor position  
000Ch write string to add information area  
DS:DX -> string  
Return: nothing  
000Dh maintenance operations  
DX = maintenance operation code  
Return: AX = operation status  
000Eh get control program version  
Return: AH = release number (major version)  
AL = level number (minor version)

000Fh get microcode version  
Return: AH = release number (major version)  
AL = level number (minor version)  
0010h save or display graphics  
BX = length of data buffer  
CX = subfunction request code  
DS:DX -> data buffer  
Return: AX = return code  
CX = length of PIF data  
0011h perform structured field operation  
CX = request number  
DS:DX -> parameter list  
Return: AX = status word (see #03738)  
CX = error number  
0012h set cursor position for direct write buffer  
DX = new cursor position  
Return: AX = status word (see #03738)  
0013h write direct to buffer  
DL = character to be written  
DH = translation option  
Return: AX = status word (see #03738)  
0014h write direct to buffer without echo  
DL = character to be written  
DH = translation option  
Return: AX = status word (see #03738)  
0015h set direct write string length  
DX = string value  
Return: nothing  
0016h write string direct to buffer  
DS:DX -> string  
Return: AX = status word (see #03738)  
0017h write string direct to buffer, untranslated  
DS:DX -> string  
Return: AX = status word (see #03738)  
0018h get direct-write cursor position  
Return: AX = cursor position  
0019h convert row/column to cursor position  
DH = display row (1-43)  
DL = display column (1-132)  
Return: AX = cursor position  
001Ah convert cursor position to row/column

DX = cursor position  
Return: AH = display row  
AL = display column  
001Bh find next field  
DX = initial cursor position  
Return: AX = field cursor position  
001Ch find previous field  
DX = initial cursor position  
Return: AX = field cursor position  
001Dh find next unprotected field  
DX = initial cursor position  
Return: AX = field cursor position  
001Eh find previous unprotected field  
DX = initial cursor position  
Return: AX = field cursor position  
001Fh find next protected field  
DX = initial cursor position  
Return: AX = field cursor position  
0020h find previous protected field  
DX = initial cursor position  
Return: AX = field cursor position  
0021h masked search forward  
DH = mask  
DL = search pattern  
Return: AX = cursor position or 0000h  
0022h masked search backward  
DH = mask  
DL = search pattern  
Return: AX = cursor position or 0FFFh  
0023h find field length  
DX = cursor position  
Return: AX = field length  
0024h read field  
DS:DX -> buffer for field contents  
Return: AX = status word (see #03738)  
0025h read screen  
DS:DX -> buffer for screen contents  
Return: AX = status word (see #03738)  
0026h read buffer untranslated  
DX = cursor position  
Return: AX = buffer code

CX:BX -> 3278/79 device buffer image  
 0027h enable/disable keyboard  
 DL = new state of keyboard breaks (00h enabled, 01h disabled)  
 Return: nothing  
 0028h select host session  
 DL = session short name  
 Return: AX = session information  
 0029h retrieve host session name  
 AX = short name (DFT) or 0000h (not available, CUT mode)  
 002Ah get current device buffer size  
 Return: AX = device buffer size  
 CX = segment of EAB  
 002Bh arm modified location trigger  
 DX = cursor position  
 Return: AX = status (0000h not available, 0001h successful)

Bitfields for PCOX status word:

Bit(s) Description (Table 03738)

0,1 cursor type  
 2 cursor inhibited  
 3 display inhibited  
 4 feature step inhibited  
 5 480-character format code  
 6,7 unused  
 8-10 model number (2-5)  
 11 unit has been reset by controller (cleared after status returned)  
 12 buffer has been written into (cleared after status returned)  
 13 alarm has been sounded (cleared after status returned)  
 14-15 monitor type (01 mono, 10 color, 11 hybrid)

-----N-6F00-----

INT 6F - 10NET - LOGIN

AH = 00h

DS:DX -> login record (see #03739)

Return: CL = security level

AX = status (see #03740)

SeeAlso: AH=01h,AH=80h,INT 21/AX=4402h"10MEMMGR"

Format of 10NET login record:

Offset Size Description (Table 03739)

00h 8 BYTES user name

08h 8 BYTES password

10h 12 BYTEs name of SuperStation

(Table 03740)

Values for 10NET status:

0000h successful  
01FFh "RTO\_NERR" transmit interrupt lost (time out on response)  
02FFh "NET\_NERR" network (hardware) error  
03FFh "PAS\_NERR" invalid password  
04FFh "LRN\_NERR" local resource not available  
05FFh "SRN\_NERR" server resource not available  
06FFh "LNM\_NERR" already logged in under different name  
07FFh "LSF\_NERR" login security failure (node)  
08FFh "NLI\_NERR" not logged in  
09FFh "DIVZ\_NERR" position calc error  
0AFFh "NT1\_NERR" receive subfunction not = send subfunction (i.e. read,write)  
0BFFh "RFNC\_NERR" request function not in range  
0CFFh "NSFH\_NERR" no more server file handle entries left  
0DFFh "NFTAB\_NERR" no more shared file table entries left  
0EFFh "NUFH\_NERR" no more user file handle entries left  
0FFFh "CHAT\_NERR" chat permit not on  
10FFh "NSRV\_NERR" not a server on request  
11FFh "NOBD\_NERR" no transporter board error  
12FFh "STO\_NERR" time out on send  
13FFh "INF\_NERR" item not found (spool item not on queue)  
14FFh "DACS\_NERR" DOS access incompatible  
15FFh "RLOCK\_NERR" record already locked  
16FFh "IVP\_NERR" invalid parameter  
17FFh "RLTO\_NERR" record lock time out error  
18FFh "CSPL\_NERR" currently spooling to named device  
19FFh "DRP\_NERR" dropped receive message (throttle)  
1AFFh "SOPV\_NERR" open sharing violation  
1BFFh "NTUF\_NERR" no more tuf entries left  
1CFFh "NOWN\_NERR" not file owner on open  
1DFFh "RSEC\_NERR" read security not passed  
1EFFh "WSEC\_NERR" write security not passed  
1FFFh "GSEC\_NERR" group security not passed  
20FFh "SEC1\_NERR" security file failure  
21FFh "ACT1\_NERR" activity file failure  
22FFh "SPL1\_NERR" spool control file failure  
23FFh "NMT\_NERR" device not mounted (spooling)  
24FFh "RSPL\_NERR" spool file has not been terminated



```

25FFh "DNSH_NERR" device not mounted or is not being shared
26FFh "DUP_NERR" duplicate node ID
27FFh "FNF_NERR" file not found error
28FFh "NMF_NERR" no more files
29FFh "UN_NERR" unknown internal system error
2AFFh "QCP_NERR" print queue is full or corrupted
2BFFh "IFNC_NERR" invalid function
2CFFh "IVH_NERR" invalid handle
2DFFh "TOF_NERR" too many files opened
2EFFh "PNF_NERR" path not found
2FFFh "SACT_NERR" named file is active
---10NET v5.0+ ---
30FFh "NAK_NERR" received NAK on send (destination out of buffers)
31FFh "RENT_NERR" reentrancy in driver F_SEND
32FFh "RECV_NERR" driver could not be put in receive mode
33FFh "NRLT_NERR" no more RLTAB entries left
34FFh "DIAL_NERR" function requires an unsupported dialect
35FFh "IVD_NERR" invalid device
36FFh "NALV_NERR" netname access level violated
37FFh "NPIDNF_NERR" network path not found
38FFh "SP_NERR" server is paused
39FFh "TMNM_NERR" too many remote user names
3AFFh "DUPD_NERR" duplicate network device
3BFFh "DIU_NERR" shared device in use, can't delete
3CFFh "NNWD_NERR" network name was deleted
3DFFh "NPFS_NERR" not enough space for print file
3EFFh "NNNF_NERR" network name not found (can't find the call name)
3FFFh "NB_NERR" network busy
40FFh "NDNLE_NERR" network device no longer exists
41FFh "NBCLE_NERR" NetBIOS command limit exceeded
42FFh "FINT24_NERR" Fail on INT 24h
43FFh "PEXP_NERR" password expired
44FFh "NPUP_NERR" new password error
45FFh "MAXS_NERR" maximum allowed disk space exceeded
46FFh "TDOW_NERR" time-of-day/day-of-week error

```

SeeAlso: #03767

Index: error codes;10Net|10Net;error codes

-----b-6F00-----

INT 6F - HP Vectra EX-BIOS - "F\_ISR" (internal) - RESPOND TO LOGICAL ISR

AH = 00h

BP = ID for driver (see #03741)

DH = data type (see #03742)  
DL = physical device driver's index (driver ID / 6)  
BX,CX = data  
ES:0000h -> physical device's Describe Record (see #03749)  
---keypress event data---  
    BH = keyboard state (only if bit 5 of DH set) (see #03743)  
    BL = scancode (if bit 4 of DH clear)  
    CX = number of scancodes in list (if bit 4 of DH set)  
    ES:SI -> scancode list (if bit 4 of DH set)  
---motion event data---  
    BX = new X position (abs device) or X increment (relative device)  
    CX = new Y position (abs device) or Y increment (relative device)  
---button event data  
    BL = button information  
    bits 15-8 reserved  
    bit 7: button state (0 = down, 1 = up)  
    bits 6-0: button number (0-7)  
    BH = reserved

Return: AH = status (see #03744)

BP,DS destroyed

---if device is keyboard translator---

    BL = translated scancode  
    BH = new keyboard state (if DH bit 5 set) (see #03743)  
    DH = new scancode type (see #03742)

Notes: INT 6F corresponds to IRQ23 on the original HP Vectra AT, which is unavailable because of its use as a BIOS extension software interrupt the INT 6F handler consists of an instruction to load DS with the driver's data segment followed by an indexed far jump using BP to select the destination vector; since the interrupt handler is located immediately following the dispatch table, the HP\_VECTOR\_TABLE may be found by looking at offset 0000h in the INT 6F segment, and its size is equal to the offset of the interrupt handler each entry in the HP\_VECTOR table consists of a DWORD for the driver's entry point address and a WORD for the driver's data segment this function is not user-callable, as it is a response to a physical event, and assumes that the caller has already handled the physical interrupt and updated the Describe Record (see #03749) to reflect the event

SeeAlso: INT 6C"HP Vectra",INT 6F/AX=0200h"HP Vectra"

(Table 03741)

Values for HP Vectra EX-BIOS driver ID:

0000h V\_SCOPY (null driver, but DS value points at system copyright string)  
0006h V\_DOLITTLE (null driver)  
000Ch V\_PNULL (null driver)  
0012h V\_SYSTEM  
0018h reserved  
001Eh V\_S8259  
0024h reserved  
002Ah V\_SINPUT  
0030h reserved  
0036h V\_QWERTY (keyboard translator)  
003Ch V\_SOFTKEY (keyboard translator)  
0042h V\_FUNCTION (keyboard translator)  
0048h V\_NUMPAD (keyboard translator)  
004Eh V\_CCP (keyboard translator)  
0054h V\_SVIDEO  
005Ah V\_STRACK  
0060h V\_EVENT\_TOUCH  
0066h V\_EVENT\_TABLET  
006Ch V\_EVENT\_POINTER  
0072h reserved  
0078h reserved  
007Eh reserved  
0084h reserved  
008Ah V\_CCPCUR (keyboard translator)  
0090h V\_RAW (keyboard translator)  
0096h V\_CCPNUM (keyboard translator)  
009Ch V\_OFF (keyboard translator)  
00A2h V\_CCPGID (translator: cursor control pad keys to GID data)  
00A8h V\_SKEY2FKEY (keyboard translator)  
00AEh V\_8041  
00B4h V\_PGID\_CCP  
00BAh C\_LTABLET  
00C0h V\_LPOINTER (pointing device: mouse, etc.)  
00C6h V\_LTOUCH (touch screen)  
00CCh V\_LHPMOUSE  
00D2h ???  
...  
0102h ???  
0108h V\_LNULL  
010Eh reserved

0114h V\_HPHIL  
011Ah-01C2h reserved  
016Eh V\_SCANDOOR (scancode management chip driver) (ES/QS/RS only)  
01C8h-0228h available

(Table 03742)

Values for HP Vectra EX-BIOS ISR data type:

00h reserved "T\_KC\_R0"  
01h reserved "T\_KC\_R1"  
02h ASCII data  
03h reserved "T\_KC\_R3"  
04h HP150 keyboard (ITF) scancode  
05h reserved "T\_KC\_R5"  
06h device-definable type  
07h HP Vectra keyboard set  
08h IBM AT scancode set  
09h button data  
0Ah IBM PC scancode set  
0Bh Softkey keypad (F1-F8)  
0Ch function key keypad (F1-F10)  
0Dh HP Cursor Control Pad keypad  
0Eh Qwerty keypad  
0Fh Numeric keypad  
1xh bit 4 set: string of CX scancode of type 0xh at ES:SI  
2xh bit 5 set: BH contains current keyboard state  
40h signed 8-bit relative data  
41h signed 16-bit relative data  
42h unsigned 8-bit absolute data  
43h unsigned 16-bit absolute data  
45h specially-formed data (80x25) generated by V\_LTOUCH  
46h specially-formed data (640x200) generated by V\_LTABLET  
47h specially-formed data (640x200) generated by V\_LPOINTER

Bitfields for HP Vectra EX-BIOS keyboard state:

Bit(s) Description (Table 03743)

0 Alt pressed  
1 Left Shift pressed  
2 Right Shift pressed  
3 Ctrl pressed  
4 CapsLock active  
5 NumLock active

6 Right unlabeled key pressed (some international keyboards)  
7 Left unlabeled key pressed (some international keyboards)

SeeAlso: #00582

(Table 03744)

Values for HP Vectra EX-BIOS status:

00h successful  
02h unsupported function  
04h not serviced  
06h done (no further processing should be performed on the ISR event)  
F2h device is out of paper  
F4h device is offline  
F6h no more space for more drivers  
F8h driver is busy  
FAh bad parameter  
FEh operation failed

Note: status codes are always even; negative values ( $\geq 80h$ ) indicate errors  
while positive values indicate exceptional conditions

Format of HP EX-BIOS driver header data:

Offset Size Description (Table 03745)

00h WORD driver attributes (see #03746)  
02h WORD string index of driver's name  
04h WORD driver's default logical device vector (see #03741)  
06h WORD driver's parent class (bitset) (see #03747)  
08h WORD driver's child class (bitset)  
0Ah WORD driver's parent vector  
0Ch WORD driver's child vector  
0Eh BYTE major subaddress  
0Fh BYTE minor subaddress

Notes: this structure is located at offset 0 in the driver's data segment,  
which in turn may be read from the HP\_VECTOR\_TABLE (refer to note in  
main entry)

only the first WORD is required, and everything from offset 6 onward  
is only required if the device wishes to perform device mapping

SeeAlso: #03748, #03749

Bitfields for HP EX-BIOS driver header attributes:

Bit(s) Description (Table 03746)

15 this is a complete driver header  
14 "ATR\_DEVCFG" reserved

13 driver can be mapped with the parent vector at offset 0Ah  
12 driver can be mapped with the child vector at offset 0Ch  
11-9 driver type  
000 reserved vector  
001 free vector  
010 EX-BIOS service  
011 logical driver (mapped from parent to child)  
100 mappable driver (cannot be last in driver chain)  
101 mappable driver that is last in driver chain  
110 input driver (mappable)  
111 reserved  
8 "ATR\_STRING" reserved  
7 call SF\_START whenever driver is remapped  
6-5 addressing requirements  
00 no subaddresses required  
01 requires major address be stored at offset 0Eh  
10 requires minor address be stored at offset 0Fh  
11 required major, minor, and mid addresses (minor in low nybble of  
offset 0Fh, mid address in high nybble of 0Fh)  
4 driver can be shared between several parent drivers  
3 driver can be shared between several child drivers  
2 this driver header is in ROM  
1 "ATR\_YIELD" reserved  
0 reserved

SeeAlso: #03745, #03747

Bitfields for HP EX-BIOS driver class:

Bit(s) Description (Table 03747)

15 maps F1 to F8 softkeys  
14 keyboard  
13 cursor pad  
12 console device  
11 serial output device (may be capable of limited input)  
10 "CL\_COMM" reserved  
9 interfaces multiple resources transparent to operating system  
8 serial output device filter (can be mapped between logical and physical  
driver to perform translations)  
7 addressed block device  
6 priority boot device  
5 logical graphics input device  
4 physical graphics input device (can map to child of another driver)

3 "CL\_GID" can map to an event  
 2 physical touch device  
 1 reserved  
 0 class extension bit

Note: special values: FFFFh maps to all other devices (V\_PNULL), and 0000h maps to no other driver

SeeAlso: #03745

Format of HP EX-BIOS global data area:

Offset Size Description (Table 03748)

00h 20 BYTES reserved  
 14h BYTE sound driver status  
 15h BYTE number of pending key clicks (max 4) (see AH=34h/BP=0012h)  
 16h BYTE current tick duration scaling factor  
 17h BYTE current key click volume  
 18h WORD current beep period (10 us increments) (see AH=3Ch/BP=0012h)  
 1Ah WORD current beep duration (10 us increments)  
 1Ch BYTE number of pending beep functions (max 4) (see AH=3Ah/BP=0012h)  
 1Dh BYTE reserved  
 1Eh WORD next unused string index number  
 20h ... reserved

SeeAlso: #03745

Format of HP EX-BIOS Driver Describe Record:

Offset Size Description (Table 03749)

00h 16 BYTES EX-BIOS driver header data (see #03745)  
 10h BYTE device GID type  
     bits 7-4: device type  
     bits 3-0: physical device link address  
 11h BYTE physical device ID (see #03751)  
 12h WORD logical device status bits (see #03750)  
 14h BYTE physical device vector number (driver ID / 6)  
 15h BYTE maximum number of axes reported (0-2)  
 16h BYTE device class  
     bits 7-4: current class  
     bits 3-0: default class  
 17h BYTE number of buttons/prompts  
     bits 7-4: number of prompts  
     bits 3-0: number of buttons  
 18h BYTE reserved  
 19h BYTE (physical devices only) maximum output burst length

1Ah BYTE (physical devices only) number of write registers  
 1Bh BYTE (physical devices only) number of read registers  
 1Ch BYTE button transition flags (bit 0 = button0, etc.)  
 1Dh BYTE current button states (bit 0 = button0, etc.)  
 1Eh WORD device resolution  
 20h WORD maximum x-axis count  
 22h WORD maximum y-axis count  
 24h WORD X position data for absolute devices  
 26h WORD Y position data for absolute devices  
 28h WORD X delta for relative devices  
 2Ah WORD Y delta for relative devices  
 2Ch WORD (logical devices only) X-axis scaling accumulator  
 (fraction of one logical unit)  
 2Eh WORD (logical devices only) Y-axis scaling accumulator  
 (fraction of one logical unit)

Bitfields for HP logical device status flags:

Bit(s) Description (Table 03750)

15-5 reserved  
 4 event enabled  
 3 tracking enabled  
 2 clipping enabled  
 1 button error occurred  
 0 interrupt in progress

SeeAlso: #03749

(Table 03751)

Values for HP-HIL device ID:

00h-02h reserved  
 03h Swiss-French keyboard  
 04h-06h reserved  
 07h Canadian-English keyboard  
 08h-0Ah reserved  
 0Bh Italian keyboard  
 0Ch reserved  
 0Dh Dutch keyboard  
 0Eh Swedish keyboard  
 0Fh German keyboard  
 10h-12h reserved  
 13h Spanish keyboard  
 14h reserved



15h Belgian (Flemish) keyboard  
16h Finnish keyboard  
17h UK keyboard  
18h French-Canadian keyboard  
19h Swiss-German keyboard  
1Ah Nerwegian keyboard  
1Bh Frensh keyboard  
1Ch Danish keyboard  
1Dh Katakana keyboard  
1Eh Latin American-Spanish keyboard  
1Fh US-American keyboard  
20h-2Bh reserved  
2Ch-2Fh tone generator  
30h-3Fh reserved  
40h-5Bh reserved (character entry)  
5Ch-5Fh barcode reader  
60h-67h reserved (relative positions)  
68h-6Bh mouse  
6Ch-6Fh trackball  
70h-7Fh reserved (relative positions)  
80h-87h reserved (absolute positions)  
88h-8Bh touchpad  
8Ch-8Fh touch screen  
90h-97h graphics tablet  
98h-9Fh reserved (absolute positions)  
A0h-BFh compressed keyboard (91-93 keys)  
C0h-DFh extended keyboard (107-109 keys)  
E0h-FFh standard keyboard (85-87 keys)

-----N-6F01-----  
INT 6F - 10NET - LOGOFF  
  AH = 01h  
  DS:DX -> superstation ID or nulls (12 bytes)  
Return: CX = number of files closed  
  AX = status (see also #03740)  
      08FFh superstation ID not already logged in  
SeeAlso: AH=00h"10NET",AH=81h

-----N-6F02-----  
INT 6F - 10NET - STATUS OF NODE  
  AH = 02h  
  DS:DX -> 512-byte status record (see #03753)  
Return: CF clear if successful

CF set on error

AX = error code (see #03740)

SeeAlso: INT 21/AX=5E01h"10NET"

(Table 03752)

Values for 10NET station type:

00h workstation

01h superstation

02h gateway station

03h gateway active

04h logged into multiple superstations

05h reserved

Format of 10NET node status record:

Offset Size Description (Table 03753)

00h 8 BYTES user name (0 if none)

08h BYTE station type (see #03752)

09h 24 BYTES list of superstations logged into more than one superstation

21h 12 BYTES node ID

2Dh WORD message count for this station (send for user node, receive for superstations)

---for superstations only---

2Fh WORD drives allocated (bit 0=A:, bit 1=B:,...)

31h BYTE user service flags (see #03754)

32h BYTE printers allocated (bit 0=LPT1,...)

33h BYTE number of unprinted spool files

34h BYTE number of opened files

35h BYTE number of logged on nodes

36h BYTE primary drive (1=A:)

37h BYTE reserved

38h N BYTES list of logged on node IDs (each 12 bytes, max 37 IDs)

1F4h 3 BYTES time: sec/min/hrs

1F7h 3 BYTES date: day/mon/year-1980

Bitfields for 10NET user service flags:

Bit(s) Description (Table 03754)

7 gate

6 print permit on

4 SUBMIT is on

3 mail waiting for node

2 calendar waiting for you

1 news waiting for you

0 mail waiting for you

-----b-6F0200-----

INT 6F - HP Vectra EX-BIOS - "F\_SYSTEM SF\_INIT" - START INITIALIZING DRIVER

AX = 0200h

BP = driver ID (see #03741)

BX = "last-used DS" value from HP global data area (see #03748)

Return: AH = status (see #03744)

BX = new "last-used DS" value

BP,DS destroyed

SeeAlso: AH=00h"HP Vectra",AX=0202h,AH=0Ch"F\_INS\_FIXOWNDS"

-----b-6F0202-----

INT 6F - HP Vectra EX-BIOS - "F\_SYSTEM SF\_START" - COMPLETE DRIVER INITIALIZ.

AX = 0202h

BP = driver ID (see #03741)

Return: AH = status (see #03744)

BP,DS destroyed

SeeAlso: AH=00h"HP Vectra",AX=0200h,AX=0204h"HP Vectra"

-----W-6F0204-----

INT 6F C - MS Windows 3.0 - DOS APPLICATION SWITCH HOOK

AX = 0204h

Return: AX = status

0000h switch is allowed

other switch not allowed

Note: intercepting this call will allow a DOS application to ensure that

Windows will not switch away from it.

-----b-6F0204-----

INT 6F - HP Vectra EX-BIOS - "F\_SYSTEM SF\_REPORT\_STATE" - GET DRIVER STATE

AX = 0204h

BP = driver ID (see #03741)

Return: AH = status (see #03744)

DX = device state from Logical Describe Record (see #03749,#03755)

BP,DS destroyed

SeeAlso: AH=00h"HP Vectra",AX=0200h,AX=0206h"HP Vectra"

Bitfields for HP Vectra V\_HPHIL driver state:

Bit(s) Description (Table 03755)

0 link configuration in progress

1 reserved

2 no devices attached

3 general failure

4,5 reserved  
6 HP-HIL driver open  
7 reserved  
8 link has been reconfigured  
9 error during output request  
10 reserved  
11 output request complete  
12 timeout  
13-15 reserved

-----b-6F0206-----

INT 6F - HP Vectra EX-BIOS - "F\_SYSTEM SF\_VERSION\_DESC" - GET DRIVER VERSION

AX = 0206h

BP = driver ID (see #03741)

Return: AH = status (see #03744)

BX = driver release date code

(BL = BCD years since 1960, BH = BCD week number within year)

CX = length of version string

ES:DI -> version string

BP,DS destroyed

SeeAlso: AH=00h"HP Vectra",AX=0200h,AX=0204h"HP Vectra",AX=0208h

-----b-6F0208-----

INT 6F - HP Vectra EX-BIOS - "F\_SYSTEM SF\_DEF\_ATTR" - RESET TO DEFAULT ATTR

AX = 0208h

BP = driver ID (see #03741)

Return: AH = status (see #03744)

BP,DS destroyed

SeeAlso: AH=00h"HP Vectra",AX=0200h,AX=020Ah,AX=020Ch

-----b-6F020A-----

INT 6F - HP Vectra EX-BIOS - "F\_SYSTEM SF\_GET\_ATTR" - GET CURRENT ATTRIBUTES

AX = 020Ah

BP = driver ID (see #03741)

Return: AH = status (see #03744)

BX,CX = current attributes

BP,DS destroyed

---for V\_LTOUCH, V\_LPOINTER---

BX = logical width

CX = logical height

SeeAlso: AH=00h"HP Vectra",AX=0200h,AX=0208h,AX=020Ch

-----b-6F020C-----

INT 6F - HP Vectra EX-BIOS - "F\_SYSTEM SF\_SET\_ATTR" - SET DEVICE ATTRIBUTES

AX = 020Ch

BP = driver ID (see #03741)  
BX,CX = new attributes  
---for V\_LTOUCH, V\_LPOINTER---  
BX = new logical width  
CX = new logical height  
Return: AH = status (see #03744)  
BP,DS destroyed  
SeeAlso: AH=00h"HP Vectra",AX=0200h,AX=0208h,AX=020Ah  
-----b-6F020E-----  
INT 6F - HP Vectra EX-BIOS - "F\_SYSTEM SF\_OPEN" - RESERVE DRIVER FOR ACCESS  
AX = 020Eh  
BP = driver ID (see #03741) for V\_HPHIL, others  
Return: AH = status (see #03744)  
BP,DS destroyed  
Desc: place device in open state, which allows output to the device  
SeeAlso: AH=00h"HP Vectra",AX=0200h,AX=0208h,AX=0210h  
-----b-6F0210-----  
INT 6F - HP Vectra EX-BIOS - "F\_SYSTEM SF\_CLOSE" - RELEASE DRIVER  
AX = 0210h  
BP = driver ID (see #03741) for V\_HPHIL, others  
Return: AH = status (see #03744)  
BP,DS destroyed  
Desc: place device in closed state, which disallows output to the device  
SeeAlso: AH=00h"HP Vectra",AX=0200h,AX=0208h,AX=020Eh  
-----b-6F0212-----  
INT 6F - HP Vectra EX-BIOS - "F\_SYSTEM SF\_TIMEOUT" - REPORT REQUESTED TIMEOUT  
AX = 0212h  
BP = driver ID (see #03741)  
Return: AH = status (see #03744)  
BP,DS destroyed  
Desc: report to driver that a previously-requested timeout interval has  
expired  
SeeAlso: AH=00h"HP Vectra",AX=0200h,AX=0208h,AX=020Eh,AX=0214h,AX=0216h  
-----b-6F0214-----  
INT 6F - HP Vectra EX-BIOS - "F\_SYSTEM SF\_INTERVAL" - 60 HZ INTERVAL EXPIRED  
AX = 0214h  
BP = driver ID (see #03741)  
Return: AH = status (see #03744)  
BP,DS destroyed  
Desc: report to driver that a previously-requested 60 Hz interval has  
expired

SeeAlso: AH=00h"HP Vectra",AX=0200h,AX=0208h,AX=020Eh,AX=0212h,AX=0216h

-----b-6F0216-----

INT 6F - HP Vectra EX-BIOS - "F\_SYSTEM SF\_TEST" - PERFORM HARDWARE TEST

AX = 0216h

BP = driver ID (see #03741)

Return: AH = status (see #03744)

BP,DS destroyed

---on failure---

CX = length of reason string

ES:DI -> string describing error

SeeAlso: AH=00h"HP Vectra",AX=0200h,AX=0208h,AX=020Eh,AX=0212h,AX=0214h

-----N-6F03-----

INT 6F - 10NET - GET ADDRESS OF CONFIGURATION TABLE

AH = 03h

DS:DI -> node ID (optional)

Return: ES:BX -> configuration table (see #03756)

SeeAlso: AH=13h,INT 21/AX=5E01h"10NET"

Format of 10NET configuration table:

Offset Size Description (Table 03756)

-41 WORD local device table address

-39 WORD extended network error mapping table address

-37 WORD shared device table address

-35 WORD mounted device table address

-33 BYTE receive buffer counter

-32 BYTE collect buffer counter

-31 WORD TUF address

-29 BYTE enable flag

-28 BYTE FCB keep flag

-27 WORD reserved

---up to here, 10NET v3.3---

-25 WORD count of dropped Send6F

-23 WORD buffer start address

-21 WORD comm driver base address

-19 WORD send/receive retry count

-17 BYTE number of 550ms loops before timeout

-16 WORD UFH address

-14 WORD CDIR address

-12 WORD LTAB address

-10 WORD SFH address

-8 WORD FTAB address

```

-6 WORD  RLTAB address
-4 WORD  SMI address
-2 WORD  NTAB address
00h WORD  address of first CT_DRV
02h BYTE  number of DRV entries
03h 8 BYTES login name
0Bh 12 BYTES node ID (blank-padded)
17h 6 BYTES node address
1Dh BYTE  flag
1Eh BYTE  CT_CFLG (chat permit)
    bit 1: sound bell
    bit 0: CHAT permit
1Fh BYTE  CT_PSFLG (see #03757)
20h BYTE  in 10Net flag
21h WORD  receive message count
23h WORD  send message count
25h WORD  retry count
27h WORD  failed count
29h WORD  driver errors
2Bh WORD  dropped responses/CHATs
2Dh 9 BYTES LIST ID/NTAB address (3 entries--LPT1-3)
36h 6 BYTES AUX ID/NTAB address (2 entries--COM1-2)
3Ch BYTE  active CB channel
3Dh BYTE  received 6F messages on queue
3Eh 9 BYTES activity counters for channels 1-9
---beyond here, 10NET v3.3---
47h BYTE  bit 0: RS232 gate
    bit 1: Send6F gate (user set)
48h DWORD pointer into gate (user set)
4Ch DWORD pointer into 10Net send
50h N WORDs addresses of timer blocks

```

## Bitfields for CT\_PSFLG:

Bit(s) Description (Table 03757)

```

5 PRINT permit
4 KB initiated
3 CHAT called FOXPTRM
2 SUBMIT active
1 SUBMIT received
0 SUBMIT permit

```

-----N-6F04-----

INT 6F - 10NET - SEND

AH = 04h

DS:BX -> send record (see #03758)

DS:DX -> data (max 1024 bytes)

Return: CF clear if successful

CF set on error

AX = error code (see #03740)

SeeAlso: AH=05h"10NET",AH=09h"10NET",AH=0Ah"10NET"

Format of 10NET send record:

Offset Size Description (Table 03758)

00h 12 BYTES receiving node's ID

if first byte has high-order bit set, message is directed to  
the CT\_RGATE vector at the receiver

if second byte is 00h, first byte is taken as a CB

channel number and delivered to all nodes on same channel

0Ch WORD length of data at DX

-----b-6F04--BP0012-----

INT 6F - HP Vectra EX-BIOS - "F\_INS\_BASEHPVT" - GET HP\_VECTOR\_TABLE ADDRESS

AH = 04h

BP = 0012h (driver ID for V\_SYSTEM)

Return: AH = status (see #03744)

ES = segment of HP\_VECTOR\_TABLE (see also AH=00h"HP Vectra")

BP,DS destroyed

SeeAlso: AH=00h"HP Vectra",AH=06h"F\_INS\_XCHGFIX",AH=08h"F\_INS\_XCHGRSVD"

-----b-6F04--BP005A-----

INT 6F - HP Vectra EX-BIOS - "F\_TRACK\_INIT" - RESET TRACKING DRIVER TO DEFAULTS

AH = 04h

BP = 005Ah (driver ID for V\_STRACK)

Return: AH = status (see #03744)

BP,DS destroyed

SeeAlso: AH=00h"HP Vectra",AH=06h"F\_TRACK\_ON",AH=08h"F\_TRACK\_OFF"

SeeAlso: AH=0Ah"F\_DEF\_MASKS",AH=0Ch"F\_SET\_LIMITS\_X"

-----b-6F0400-----

INT 6F - HP Vectra EX-BIOS - "F\_IO\_CTRL\_SF\_LOCK" - RESERVE SUB-ADDRESSES

AX = 0400h

BP = driver ID (see #03741)

DH,DL = major,minor address (optional)

Return: AH = status (see #03744)

BP,DS destroyed

Desc: reserve specified sub-addresses on an already-allocated driver for



exclusive access

Note: this function is documented as present but unsupported in the drivers

V\_LTABLET, V\_LPOINTER, and V\_LTOUCH

SeeAlso: AX=0200h"HP Vectra",AX=0402h"SF\_UNLOCK"

-----b-6F0400BP001E-----

INT 6F - HP Vectra EX-BIOS - "F\_IO\_CTRL SF\_ENABLE\_SVC" - UNMASK 8041 SVC INT

AX = 0400h

BP = 001Eh (driver ID for V\_S8259)

Return: AH = status (see #03744)

BP,DS destroyed

SeeAlso: AH=00h"HP Vectra",AX=0402h"SF\_DISABLE\_SVC",AX=0404h"SF\_ENABLE\_KBD"

SeeAlso: AX=0408h"SF\_ENABLE\_HPHIL"

-----b-6F0400BP002A-----

INT 6F - HP Vectra EX-BIOS - "F\_IO\_CTRL SF\_DEF\_LINKS" - SET DEFAULT PARENTS

AX = 0400h

BP = 002Ah (driver ID for V\_SINPUT)

Return: AH = status (see #03744)

BP,DS destroyed

Desc: set the parent vectors in the physical device driver headers to their  
system defaults

SeeAlso: AH=00h"HP Vectra",AX=0402h/BP=002Ah,AX=0404h/BP=002Ah

SeeAlso: AH=06h"F\_INQUIRE"

-----b-6F0400BP0054-----

INT 6F - HP Vectra EX-BIOS - "F\_IO\_CTRL SF\_VID\_ID\_HP" - INSTALLATION CHECK

AX = 0400h

BP = 0054h (driver ID for V\_SVIDEO)

???

Return: AH = status (see #03744)

BX = 4850h ('BP')

BP,DS destroyed

???

Note: only partially documented

SeeAlso: AX=0402h/BP=0054h,AX=0404h/BP=0054h,AX=0406h/BP=0054h

SeeAlso: AX=0408h/BP=0054h,AX=040Ah/BP=0054h

-----b-6F0400BP00CC-----

INT 6F - HP Vectra EX-BIOS - "F\_IO\_CTRL SF\_MOUSE\_COM" - INIT INT 33 HANDLER

AX = 0400h

BP = 00CCh (driver ID for V\_LHPMOUSE)

Return: AH = status (see #03744)

BP,DS destroyed

SeeAlso: AH=00h"HP Vectra",AX=0402h/BP=00CCh

-----b-6F0402-----

INT 6F - HP Vectra EX-BIOS - "F\_IO\_CTRL\_SF\_UNLOCK" - RELEASE SUB-ADDRESSES

AX = 0402h

BP = driver ID (see #03741)

DH,DL = major,minor address (optional)

Return: AH = status (see #03744)

BP,DS destroyed

Desc: release specified sub-addresses from exclusive access

Note: this function is documented as present but unsupported in the drivers

V\_LTABLET, V\_LPOINTER, and VLTOUCH

SeeAlso: AX=0200h"HP Vectra",AX=0400h"SF\_LOCK"

-----b-6F0402BP001E-----

INT 6F - HP Vectra EX-BIOS - "F\_IO\_CTRL\_SF\_DISABLE\_SVC" - MASK 8041 SVC INT

AX = 0402h

BP = 001Eh (driver ID for V\_S8259)

Return: AH = status (see #03744)

BP,DS destroyed

SeeAlso: AH=00h"HP Vectra",AX=0400h/BP=001Eh,AX=0406h/BP=001Eh

SeeAlso: AX=040Ah/BP=001Eh

-----b-6F0402BP002A-----

INT 6F - HP Vectra EX-BIOS - "F\_IO\_CTRL\_SF\_GET\_LINKS" - GET DRIVER LINK FIELDS

AX = 0402h

BP = 002Ah (driver ID for V\_SINPUT)

ES:SI -&gt; buffer for link table (see #03759)

Return: AH = status (see #03744)

ES:SI buffer filled

BP,DS destroyed

Desc: get the parent and child vectors for the seven supported physical  
device drivers

SeeAlso: AH=00h"HP Vectra",AX=0400h"SF\_DEF\_LINKS",AX=0404h"SF\_SET\_LINKS"

SeeAlso: AH=06h"F\_INQUIRE"

Format of HP Vectra EX-BIOS driver link table:

Offset Size Description (Table 03759)

00h BYTE vector number of HP-HIL device #1's child

01h BYTE vector number of HP-HIL device #1's parent

...

0Ch BYTE vector number of HP-HIL device #7's child

0Dh BYTE vector number of HP-HIL device #7's parent

-----b-6F0402BP0054-----

INT 6F - HP Vectra EX-BIOS - "F\_IO\_CTRL\_SF\_VID\_GET\_INFO" - GET DISPLAY INFO

```
AX = 0402h
BP = 0054h (driver ID for V_SVIDEO)
???
```

Return: AH = status (see #03744)  
BP,DS destroyed  
???

Note: only partially documented  
SeeAlso: AX=0400h/BP=0054h,AX=0404h/BP=0054h,AX=0406h/BP=0054h  
SeeAlso: AX=0408h/BP=0054h,AX=040Ah/BP=0054h

-----b-6F0402BP00CC-----

INT 6F - HP Vectra EX-BIOS - "F\_IO\_CTRL SF\_MOUSE\_COM" - FORCE INT 33 INIT  
AX = 0402h  
BP = 00CCh (driver ID for V\_LHPMOUSE)  
Return: AH = status (see #03744)  
BP,DS destroyed  
Desc: set up INT 33 even if no mouse is installed  
SeeAlso: AH=00h"HP Vectra",AX=0400h/BP=00CCh

-----b-6F0404-----

INT 6F - HP Vectra EX-BIOS - "F\_IO\_CTRL SF\_TRACK\_ON" - GRAPHICS CURSOR ON  
AX = 0404h  
BP = driver ID (see #03741) for V\_LPOINTER, V\_LTOUCH, V\_LTABLET  
Return: AH = status (see #03744)  
BP,DS destroyed  
Desc: enables graphics cursor tracking of the logical device's movement  
SeeAlso: AH=00h"HP Vectra",AX=0400h"SF\_LOCK",AX=0406h"SF\_TRACK\_OFF"

-----b-6F0404BP001E-----

INT 6F - HP Vectra EX-BIOS - "F\_IO\_CTRL SF\_ENABLE\_KBD" - UNMASK 8041 OBF INT  
AX = 0404h  
BP = 001Eh (driver ID for V\_S8259)  
Return: AH = status (see #03744)  
BP,DS destroyed  
SeeAlso: AH=00h"HP Vectra",AX=0400h"SF\_ENABLE\_SVC",AX=0406h"SF\_DISABLE\_KBD"  
SeeAlso: AX=0408h"SF\_ENABLE\_HPHIL"

-----b-6F0404BP002A-----

INT 6F - HP Vectra EX-BIOS - "F\_IO\_CTRL SF\_SET\_LINKS" - SET DRIVER LINK FIELDS  
AX = 0404h  
BP = 002Ah (driver ID for V\_SINPUT)  
ES:SI -> buffer containing link table (see #03759)  
Return: AH = status (see #03744)  
BP,DS destroyed  
Desc: specify the parent and child vectors for the seven supported physical

device drivers

SeeAlso: AH=00h"HP Vectra",AX=0400h"SF\_DEF\_LINKS",AX=0402h"SF\_GET\_LINKS"

SeeAlso: AH=06h"F\_INQUIRE"

-----b-6F0404BP0054-----

INT 6F - HP Vectra EX-BIOS - "F\_IO\_CTRL\_SF\_VID\_SET\_INFO" - SET EXT-CTRL INFO

AX = 0404h

BP = 0054h (driver ID for V\_SVIDEO)

???

Return: AH = status (see #03744)

BP,DS destroyed

???

Note: only partially documented

SeeAlso: AX=0400h/BP=0054h,AX=0402h/BP=0054h,AX=0406h/BP=0054h

SeeAlso: AX=0408h/BP=0054h,AX=040Ah/BP=0054h

-----b-6F0406-----

INT 6F - HP Vectra EX-BIOS - "F\_IO\_CTRL\_SF\_TRACK\_OFF" - GRAPHICS CURSOR OFF

AX = 0406h

BP = driver ID (see #03741) for V\_LPOINTER, V\_LTOUCH, V\_LTABLET

Return: AH = status (see #03744)

BP,DS destroyed

Desc: disables graphics cursor tracking of the logical device's movement

SeeAlso: AH=00h"HP Vectra",AX=0400h"SF\_LOCK",AX=0404h"SF\_TRACK\_ON"

SeeAlso: AX=0408h"SF\_CREATE\_EVENT"

-----b-6F0406BP001E-----

INT 6F - HP Vectra EX-BIOS - "F\_IO\_CTRL\_SF\_DISABLE\_KBD" - MASK 8041 OBF INT

AX = 0406h

BP = 001Eh (driver ID for V\_S8259)

Return: AH = status (see #03744)

BP,DS destroyed

SeeAlso: AH=00h"HP Vectra",AX=0402h"SF\_DISABLE\_SVC",AX=0404h"SF\_ENABLE\_KBD"

SeeAlso: AX=040Ah"SF\_DISABLE\_HPHIL"

-----b-6F0406BP0054-----

INT 6F - HP Vectra EX-BIOS - "F\_IO\_CTRL\_SF\_VID\_MOD\_INFO" - SET EXT-CTRL REG

AX = 0406h

BP = 0054h (driver ID for V\_SVIDEO)

???

Return: AH = status (see #03744)

BP,DS destroyed

???

Note: only partially documented

SeeAlso: AX=0400h/BP=0054h,AX=0402h/BP=0054h,AX=0404h/BP=0054h

SeeAlso: AX=0408h/BP=0054h,AX=040Ah/BP=0054h

-----b-6F0406BP0114-----

INT 6F - HP Vectra EX-BIOS - "F\_IO\_CTRL\_SF\_CRV\_RECONFIGURE" - RECONFIG LINK

AX = 0406h

BP = 0114h (driver ID for V\_HPHIL)

Return: AH = status (see #03744)

BP,DS destroyed

SeeAlso: AH=00h"HP Vectra",AX=020Eh,AX=0408h"SF\_CRV\_WR\_PROMPTS"

SeeAlso: AX=0410h"SF\_CRV\_SELF\_TEST"

-----b-6F0408-----

INT 6F - HP Vectra EX-BIOS - "F\_IO\_CTRL\_SF\_CREATE\_EVENT" - SET EVENT HANDLER

AX = 0408h

BP = driver ID (see #03741) for V\_LPOINTER, V\_LTOUCH, V\_LTABLET

ES:SI -> new event handler (see AH=00h"HP Vectra")

DX = new handler's DS

Return: AH = status (see #03744)

ES:SI -> old event handler

DX = old handler's DS

BP,DS destroyed

Desc: specify routine to be called on logical device events

Note: when events occur, the event handler is called with the same registers

as the "F\_ISR" function on AH=00h"HP Vectra"

SeeAlso: AH=00h"HP Vectra",AX=0400h"SF\_LOCK",AX=0406h"SF\_TRACK\_OFF"

SeeAlso: AX=040Ah"SF\_EVENT\_ON",AX=040Ch"SF\_EVENT\_OFF",AH=06h"F\_SAMPLE"

-----b-6F0408BP001E-----

INT 6F - HP Vectra EX-BIOS - "F\_IO\_CTRL\_SF\_ENABLE\_HPHIL" - UNMASK HP-HIL INT

AX = 0408h

BP = 001Eh (driver ID for V\_S8259)

Return: AH = status (see #03744)

BP,DS destroyed

SeeAlso: AH=00h"HP Vectra",AX=0400h/BP=001Eh,AX=0404h/BP=001Eh

SeeAlso: AX=040Ah/BP=001Eh

-----b-6F0408BP0054-----

INT 6F - HP Vectra EX-BIOS - "F\_IO\_CTRL\_SF\_VID\_GET\_RES" - GET VIDMODE RESOLUTN

AX = 0408h

BP = 0054h (driver ID for V\_SVIDEO)

???

Return: AH = status (see #03744)

BP,DS destroyed

???

Note: only partially documented

SeeAlso: AX=0400h/BP=0054h,AX=0402h/BP=0054h,AX=0404h/BP=0054h

SeeAlso: AX=0406h/BP=0054h,AX=040Ah/BP=0054h

-----b-6F0408BP0114-----

INT 6F - HP Vectra EX-BIOS - "F\_IO\_CTRL\_SF\_CRV\_WR\_PROMPTS" - WRITE PROMPT

AX = 0408h

BP = 0114h (driver ID for V\_HPHIL)

BX = device address indicator (see #03760)

DH = HP HIL device major address

DL = prompt number

Return: AH = status (see #03744)

BP,DS destroyed

SeeAlso: AH=00h"HP Vectra",AX=020Eh,AX=0406h/BP=0114h,AX=040Ah/BP=0114h

Bitfields for HP Vectra V\_HPHIL device address indicator:

Bit(s) Description (Table 03760)

15-14 reserved

13 address in DH is valid (must be set, clear is reserved)

12 valid value in DL

11-0 reserved

-----b-6F040A-----

INT 6F - HP Vectra EX-BIOS - "F\_IO\_CTRL\_SF\_EVENT\_ON" - ENABLE EVENT HANDLER

AX = 040Ah

BP = driver ID (see #03741) for V\_LPOINTER, V\_LTOUCH, V\_LTABLET

Return: AH = status (see #03744)

BP,DS destroyed

SeeAlso: AH=00h"HP Vectra",AX=0408h"SF\_CREATE\_EVENT",AX=040Ch"SF\_EVENT\_OFF"

SeeAlso: AX=040Eh"SF\_CLIPPING\_ON"

-----b-6F040ABP001E-----

INT 6F - HP Vectra EX-BIOS - "F\_IO\_CTRL\_SF\_DISABLE\_HPHIL" - MASK HP-HIL INT

AX = 040Ah

BP = 001Eh (driver ID for V\_S8259)

Return: AH = status (see #03744)

BP,DS destroyed

SeeAlso: AH=00h"HP Vectra",AX=0402h/BP=001Eh,AX=0406h/BP=001Eh

SeeAlso: AX=0408h/BP=001Eh

-----b-6F040ABP0054-----

INT 6F - HP Vectra EX-BIOS - "F\_IO\_CTRL\_SF\_VID\_SET\_MODE" - SET VIDEO MODE

AX = 040Ah

BP = 0054h (driver ID for V\_SVIDEO)

???

Return: AH = status (see #03744)

BP,DS destroyed

???

Note: only partially documented

SeeAlso: AX=0400h/BP=0054h,AX=0402h/BP=0054h,AX=0404h/BP=0054h

SeeAlso: AX=0406h/BP=0054h,AX=0408h/BP=0054h

-----b-6F040ABP00AE-----

INT 6F - HP Vectra EX-BIOS - "F\_IO\_CTRL\_SF\_CREATE\_INTR" - CREATE INTERVAL ENTRY

AX = 040Ah

BP = 00AEh (driver ID for V\_8041)

BH = vector number of driver to be called regularly (vector addr / 6)

Return: AH = status (see #03744)

BP,DS destroyed

Desc: add a driver to the list (max eight) of drivers which will be called

by the HP's 8041 at 60 Hz

Note: the newly added entry will not be called until enabled with

AX=040Eh/BP=00AEh

SeeAlso: AH=00h"HP Vectra",AX=040Ch/BP=00AEh,AX=040Eh/BP=00AEh

SeeAlso: AX=0412h/BP=00AEh,AX=0416h/BP=00AEh,AX=041Ah/BP=00AEh

-----b-6F040ABP0114-----

INT 6F - HP Vectra EX-BIOS - "F\_IO\_CTRL\_SF\_CRV\_WR\_ACK" - ACKNOWLEDGE DEVICE

AX = 040Ah

BP = 0114h (driver ID for V\_HPHIL)

BX = device address indicator (see #03760)

DH = HP HIL device major address

DL = acknowledge number (specific if 01h-07h, generic otherwise)

Return: AH = status (see #03744)

BP,DS destroyed

SeeAlso: AH=00h"HP Vectra",AX=0406h/BP=0114h,AX=040Ch/BP=0114h

-----b-6F040C-----

INT 6F - HP Vectra EX-BIOS - "F\_IO\_CTRL\_SF\_EVENT\_OFF" - DISABLE EVENT HANDLER

AX = 040Ch

BP = driver ID (see #03741) for V\_LPOINTER, V\_LTOUCH, V\_LTABLET

Return: AH = status (see #03744)

BP,DS destroyed

SeeAlso: AH=00h"HP Vectra",AX=0408h"SF\_CREATE\_EVENT",AX=040Ah"SF\_EVENT\_ON"

-----b-6F040CBP00AE-----

INT 6F - HP Vectra EX-BIOS - "F\_IO\_CTRL\_SF\_DELET\_INTR" - DELETE INTERVAL ENTRY

AX = 040Ch

BP = 00AEh (driver ID for V\_8041)

BH = vector number of driver to be removed (vector addr / 6)

Return: AH = status (see #03744)

BP,DS destroyed

Desc: remove a driver to the list (max eight) of drivers which will be called  
by the HP's 8041 at 60 Hz

SeeAlso: AH=00h"HP Vectra",AX=040Ah/BP=00AEh,AX=0410h/BP=00AEh  
SeeAlso: AX=0412h/BP=00AEh,AX=0416h/BP=00AEh,AX=041Ah/BP=00AEh

-----b-6F040CBP0114-----

INT 6F - HP Vectra EX-BIOS - "F\_IO\_CTRL\_SF\_CRV\_REPEAT" - SET KEY REPEAT RATE

AX = 040Ch

BP = 0114h (driver ID for V\_HPHIL)

BX = device address indicator (see #03760)

DH = HP HIL device major address

DL = repeat rate (00h = 30 Hz, 01h = 60 Hz)

Return: AH = status (see #03744)

BP,DS destroyed

Note: this function is only available if the driver is in the open state  
(see AX=020Eh"SF\_OPEN")

SeeAlso: AH=00h"HP Vectra",AX=040Ah/BP=0114h,AX=040Eh/BP=0114h

-----b-6F040E-----

INT 6F - HP Vectra EX-BIOS - "F\_IO\_CTRL\_SF\_CLIPPING\_ON" - ENABLE CLIPPING

AX = 040Eh

BP = driver ID (see #03741) for V\_LPOINTER, V\_LTOUCH, V\_LTABLET

Return: AH = status (see #03744)

BP,DS destroyed

Desc: causes the physical device's motion to be clipped to prevent overflow  
or underflow of the logical position

SeeAlso: AH=00h"HP Vectra",AX=040Ah"SF\_EVENT\_ON",AX=0410h"SF\_CLIPPING\_OFF"

-----b-6F040EBP00AE-----

INT 6F - HP Vectra EX-BIOS - "F\_IO\_CTRL\_SF\_ENABL\_INTR" - START INTERRUPT SRVICE

AX = 040Eh

BP = 00AEh (driver ID for V\_8041)

BH = vector number of driver to be called regularly (vector addr / 6)

Return: AH = status (see #03744)

BP,DS destroyed

Desc: enable 60 Hz calling of a driver previously added to the list of  
drivers to be called by the HP's 8041

SeeAlso: AH=00h"HP Vectra",AX=040Ah/BP=00AEh,AX=0410h/BP=00AEh  
SeeAlso: AX=0412h/BP=00AEh,AX=0416h/BP=00AEh,AX=041Ah/BP=00AEh

-----b-6F040EBP0114-----

INT 6F - HP Vectra EX-BIOS - "F\_IO\_CTRL\_SF\_CRV\_DISABLE\_REPEAT" - KEY REPT OFF

AX = 040Eh

BP = 0114h (driver ID for V\_HPHIL)



BX = device address indicator (see #03760)

DH = HP HIL device major address

Return: AH = status (see #03744)

BP,DS destroyed

Note: this function is only available if the driver is in the open state

(see AX=020Eh"SF\_OPEN")

SeeAlso: AH=00h"HP Vectra",AX=040Ch/BP=0114h,AX=0410h/BP=0114h

SeeAlso: AX=0416h/BP=0114h

-----b-6F0410-----

INT 6F - HP Vectra EX-BIOS - "F\_IO\_CTRL SF\_CLIPPING\_OFF" - DISABLE CLIPPING

AX = 0410h

BP = driver ID (see #03741) for V\_LPOINTER, V\_LTOUCH, V\_LTABLET

Return: AH = status (see #03744)

BP,DS destroyed

Desc: disables clipping of the physical device's motion, which may cause  
overflow or underflow of the logical position

SeeAlso: AH=00h"HP Vectra",AX=040Ah"SF\_EVENT\_ON",AX=040Eh"SF\_CLIPPING\_ON"

-----b-6F0410BP00AE-----

INT 6F - HP Vectra EX-BIOS - "F\_IO\_CTRL SF\_DISBL\_INTR" - END INTERRUPT SERVICE

AX = 0410h

BP = 00AEh (driver ID for V\_8041)

BH = vector number of driver (vector addr / 6)

Return: AH = status (see #03744)

BP,DS destroyed

Desc: disable 60 Hz interval calls to the specified driver

SeeAlso: AH=00h"HP Vectra",AX=040Ch/BP=00AEh,AX=040Eh/BP=00AEh

SeeAlso: AX=0412h/BP=00AEh,AX=0416h/BP=00AEh,AX=041Ah/BP=00AEh

-----b-6F0410BP0114-----

INT 6F - HP Vectra EX-BIOS - "F\_IO\_CTRL SF\_CRV\_SELF\_TEST" - PERFORM SELFTEST

AX = 0410h

BP = 0114h (driver ID for V\_HPHIL)

BX = device address indicator (see #03760)

DH = HP HIL device major address

ES:SI -> buffer for result

Return: AH = status (see #03744)

CX = number of bytes in buffer

BP,DS destroyed

Note: this function is only available if the driver is in the open state

(see AX=020Eh"SF\_OPEN")

SeeAlso: AH=00h"HP Vectra",AX=040Ch/BP=0114h,AX=0412h/BP=0114h

-----b-6F0412BP00AE-----

INT 6F - HP Vectra EX-BIOS - "F\_IO\_CTRL\_SF\_SET\_RAMSW" - SET RAM SWITCH IN 8041

AX = 0412h

BP = 00AEh (driver ID for V\_8041)

Return: AH = status (see #03744)

BP,DS destroyed

Desc: sets a flag in the 8041 indicating that a second 256K bank of RAM  
on the motherboard is enabled

SeeAlso: AH=00h"HP Vectra",AX=040Ch/BP=00AEh,AX=040Eh/BP=00AEh

SeeAlso: AX=0414h/BP=00AEh,AX=0416h/BP=00AEh,AX=041Ah/BP=00AEh

-----b-6F0412BP0114-----

INT 6F - HP Vectra EX-BIOS - "F\_IO\_CTRL\_SF\_CRV\_REPORT\_STATUS" - GET DEVSTATUS

AX = 0412h

BP = 0114h (driver ID for V\_HPHIL)

BX = device address indicator (see #03760)

DH = HP HIL device major address

ES:SI -> 15-byte buffer for status

Return: AH = status (see #03744)

CX = number of bytes returned in buffer

BP,DS destroyed

Note: this function is only available if the driver is in the open state  
(see AX=020Eh"SF\_OPEN")

SeeAlso: AH=00h"HP Vectra",AX=0410h/BP=0114h,AX=0414h/BP=0114h

-----b-6F0414BP00AE-----

INT 6F - HP Vectra EX-BIOS - "F\_IO\_CTRL\_SF\_CLR\_RAMSW" - CLEAR 8041 RAM SWITCH

AX = 0414h

BP = 00AEh (driver ID for V\_8041)

Return: AH = status (see #03744)

BP,DS destroyed

Desc: clears a flag in the 8041 indicating that a second 256K bank of RAM  
on the motherboard is enabled, causing the second bank to be disabled

Note: this function should normally never be called

SeeAlso: AH=00h"HP Vectra",AX=040Ch/BP=00AEh,AX=040Eh/BP=00AEh

SeeAlso: AX=0412h/BP=00AEh,AX=0418h/BP=00AEh,AX=041Ah/BP=00AEh

-----b-6F0414BP0114-----

INT 6F - HP Vectra EX-BIOS - "F\_IO\_CTRL\_SF\_CRV\_REPORT\_NAME" - GET DEV NAME

AX = 0414h

BP = 0114h (driver ID for V\_HPHIL)

BX = device address indicator (see #03760)

DH = HP HIL device major address

ES:SI -> 15-byte buffer for name

Return: AH = status (see #03744)

CX = number of bytes returned in buffer

BP,DS destroyed

Note: this function is only available if the driver is in the open state

(see AX=020Eh"SF\_OPEN")

SeeAlso: AH=00h"HP Vectra",AX=0412h/BP=0114h

-----b-6F0416BP00AE-----

INT 6F - HP Vectra EX-BIOS - "F\_IO\_CTRL\_SF\_SET\_CRTSW" - SET CRT TO MULTIMODE

AX = 0416h

BP = 00AEh (driver ID for V\_8041)

Return: AH = status (see #03744)

BP,DS destroyed

Desc: sets a flag in the 8041 indicating that the primary display is the

Multimode graphics adapter (default)

SeeAlso: AH=00h"HP Vectra",AX=040Ch/BP=00AEh,AX=040Eh/BP=00AEh

SeeAlso: AX=0412h/BP=00AEh,AX=0418h/BP=00AEh,AX=041Ah/BP=00AEh

-----b-6F0416BP0114-----

INT 6F - HP Vectra EX-BIOS - "F\_IO\_CTRL\_SF\_KEYBOARD\_REPEAT" - SET TYPEMATIC

AX = 0416h

BP = 0114h (driver ID for V\_HPHIL)

BH = what to set (00h typematic rate, 01h delay, 02h both)

BL = flag (00h = non-CCP keypad, 01h = Cursor Control Pad only)

DL = rate

bits 7-4: delay value

bits 3-0: typematic rate

Return: AH = status (see #03744)

BP,DS destroyed

SeeAlso: AH=00h"HP Vectra",AX=040Eh/BP=0114h,AX=0418h/BP=0114h

-----b-6F0418BP00AE-----

INT 6F - HP Vectra EX-BIOS - "F\_IO\_CTRL\_SF\_CLR\_CRTSW" - SET CRT TO MONOCHROME

AX = 0418h

BP = 00AEh (driver ID for V\_8041)

Return: AH = status (see #03744)

BP,DS destroyed

Desc: clears a flag in the 8041, indicating that the primary display is the

monochrome adapter

SeeAlso: AH=00h"HP Vectra",AX=040Ch/BP=00AEh,AX=040Eh/BP=00AEh

SeeAlso: AX=0414h/BP=00AEh,AX=0416h/BP=00AEh,AX=041Ah/BP=00AEh

-----b-6F0418BP0114-----

INT 6F - HP Vectra EX-BIOS - "F\_IO\_CTRL\_SF\_KEYBOARD\_LED" - SET KEYBOARD LEDS

AX = 0418h

BP = 0114h (driver ID for V\_HPHIL)

BL = new LED states  
 bit 0: scroll lock  
 bit 1: NumLock  
 bit 2: CapsLock

Return: AH = status (see #03744)

BP,DS destroyed

SeeAlso: AH=00h"HP Vectra",AX=040Eh/BP=0114h,AX=0416h/BP=0114h

-----b-6F041ABP00AE-----

INT 6F - HP Vectra EX-BIOS - "F\_IO\_CTRL\_SF\_PASS\_THRU" - SEND DATA BYTE TO 8041

AX = 041Ah  
 BP = 00AEh (driver ID for V\_8041)  
 BL = byte to send to 8041

Return: AH = status (see #03744)

BP,DS destroyed

SeeAlso: AH=00h"HP Vectra",AX=040Ah/BP=00AEh,AX=040Eh/BP=00AEh

-----N-6F05-----

INT 6F - 10NET - RECEIVE

AH = 05h  
 CX = number of seconds before timeout  
 DS:DX -> receive buffer (see #03761)

Return: CF clear if successful

AH = FEh if dequeued message is a CB message  
 CF set on error  
 AX = error code (see #03740)

SeeAlso: AH=04h"10NET"

Format of 10NET receive buffer:

Offset Size Description (Table 03761)

00h 12 BYTEs sending node's ID  
 0Ch WORD length of message  
 0Eh N BYTEs message (maximum 1024 bytes)

-----b-6F06-----

INT 6F - HP Vectra EX-BIOS - "F\_PUT\_BYTE" - WRITE A BYTE OF DATA TO DEVICE

AH = 06h  
 BP = driver ID (see #03741) for V\_HPHIL, others  
 AL = value to output  
 ---if driver is V\_HPHIL---  
 BX = device address indicator (see #03760)  
 DH = HP HIL device major address  
 DL = HP-HIL device register (00h-7Fh)

Return: AH = status (see #03744)

BP,DS destroyed

Desc: output a byte to a specific HP-HIL device register

Note: this function is only available if the driver is in the open state

(see AX=020Eh"SF\_OPEN")

SeeAlso: AH=04h"HP Vectra",AH=08h"F\_GET\_BYTE",AH=0Ah"F\_PUT\_BUFFER"

-----b-6F06-----

INT 6F - HP Vectra EX-BIOS - "F\_SAMPLE" - POLL CURRENT ABSOLUTE POSITION

AH = 06h

BP = driver ID (see #03741) for V\_LTOUCH, V\_LPOINTER, V\_LTABLET, etc.

Return: AH = status (see #03744)

BX = current logical X position

CX = current logical Y position

DL = button transitions (bit N = button N state has changed)

DH = current button states (bit N = current button N state)

ES:0000h -> logical device's Describe Record (see #03749)

BP,DS destroyed

SeeAlso: AH=00h"HP Vectra",AX=040Ah"SF\_EVENT\_ON",AX=040Eh"SF\_CLIPPING\_ON"

-----b-6F06--BP0012-----

INT 6F - HP Vectra EX-BIOS - "F\_INS\_XCHGFIX" - EXCH VALUES WITH FIXED VECTOR

AH = 06h

BP = 0012h (driver ID for V\_SYSTEM)

BX = vector address

DX = new value for DS

ES:DI -> new CS:IP entry point

Return: AH = status (see #03744)

DX = old value for DS

ES:DI -> old CS:IP entry point

BP,DS destroyed

Desc: replace an existing EX-BIOS driver at a fixed HPVT location without initialization

SeeAlso: AH=00h"HP Vectra",AH=04h"F\_INS\_BASEHPVT",AH=08h"F\_INS\_XCHGRSVD"

SeeAlso: AH=0Ah"F\_INS\_XCHGFREE",AH=0Ch"F\_INS\_FIXOWNDS"

-----b-6F06--BP002A-----

INT 6F - HP Vectra EX-BIOS - "F\_INQUIRE" - GET DRIVER PHYSICAL DESCRIBE RECORD

AH = 06h

BP = 002Ah (driver ID for V\_SINPUT)

AL = HP-HIL physical device number (01h-07h)

Return: AH = status (see #03744)

ES:SI -> physical describe record for driver

BP,DS destroyed

Note: the returned physical describe record must not be modified

SeeAlso: AH=00h"HP Vectra",AX=0400h/BP=002Ah,AH=08h/BP=002Ah,AH=0Ah/BP=002Ah

SeeAlso: AH=0Ch/BP=002Ah

-----b-6F06--BP005A-----

INT 6F - HP Vectra EX-BIOS - "F\_TRACK\_ON" - ENABLE CURSOR TRACKING

AH = 06h

BP = 005Ah (driver ID for V\_STRACK)

Return: AH = status (see #03744)

BP,DS destroyed

SeeAlso: AH=00h"HP Vectra",AH=04h"F\_TRACK\_INIT",AH=08h"F\_TRACK\_OFF"

SeeAlso: AH=0Ah"F\_DEF\_MASKS",AH=0Ch"F\_SET\_LIMITS\_X"

-----N-6F07-----

INT 6F - 10NET - LOCK HANDLE

AH = 07h

BX = file handle

CX:DX = starting offset in file

SI = record length

Return: CF clear if successful

CF set on error

AX = error code (see also AH=00h)

0002h file not found

SeeAlso: AH=08h"10NET",AH=0Fh,INT 21/AH=5Ch

-----N-6F08-----

INT 6F - 10NET - UNLOCK HANDLE

AH = 08h

BX = file handle

AL = mode

00h unlock all

01h unlock record at CX:DX

Return: CF clear if successful

CF set on error

AX = error code (see also AH=00h)

0002h file not found

SeeAlso: AH=07h,AH=0Fh,INT 21/AH=5Ch

-----b-6F08-----

INT 6F - HP Vectra EX-BIOS - "F\_GET\_BYTE" - READ A BYTE OF DATA FROM DEVICE

AH = 08h

BP = driver ID (see #03741) for V\_HPHIL (0114h), others

---if device is V\_HPHIL---

BX = device address indicator (see #03760)

DH = HP HIL device major address

DL = HP-HIL device register (00h-7Fh)

Return: AH = status (see #03744)

AL = value read from device

BP,DS destroyed

Desc: generic read-byte function; for driver V\_HPHIL, retrieves the value of a specific HP-HIL device register

Note: this function is only available if the driver is in the open state (see AX=020Eh"SF\_OPEN")

SeeAlso: AH=06h"F\_PUT\_BYTE",AH=0Ah"F\_PUT\_BUFFER",AH=0Ch"F\_GET\_BUFFER"

-----b-6F08--BP0012-----

INT 6F - HP Vectra EX-BIOS - "F\_INS\_XCHGRSVD" - EXCH VALUES WITH RESRVED VECTOR

AH = 08h

BP = 0012h (driver ID for V\_SYSTEM)

DX = new value for DS

ES:DI -> new CS:IP entry point

Return: AH = status (00h,F6h) (see also #03744)

00h successful

BX = vector address used

DX = old value for DS

ES:DI -> old CS:IP entry point

BP,DS destroyed

Desc: exchange the supplied entry point and DS with the values in the next reserved entry in the HP\_VECTOR\_TABLE

Note: returns AH=F6h if no more reserved vectors are available

SeeAlso: AH=00h"HP Vectra",AH=04h/BP=0012h,AH=06h/BP=0012h,AH=0Ah/BP=0012h

SeeAlso: AH=0Ch/BP=0012h

-----b-6F08--BP002A-----

INT 6F - HP Vectra EX-BIOS - "F\_INQUIRE\_ALL" - GET DRIVER STATUS TABLE

AH = 08h

BP = 002Ah (driver ID for V\_SINPUT)

ES:SI -> buffer for device status table (see #03762)

Return: AH = status (see #03744)

ES:SI buffer filled

BP,DS destroyed

Note: the returned physical describe record must not be modified

SeeAlso: AH=00h"HP Vectra",AX=0400h/BP=002Ah,AH=06h/BP=002Ah,AH=0Ah/BP=002Ah

SeeAlso: AH=0Ch/BP=002Ah

Format of HP Vectra HP-HIL physical device status table:

Offset Size Description (Table 03762)

00h BYTE status for HP-HIL device #1

bit 0: device present

```

    bits 7-2 reserved
01h BYTE device ID for HP-HIL device #1
...
0Ch BYTE status for HP-HIL device #7
0Dh BYTE device ID for HP-HIL device #7
-----b-6F08--BP005A-----
INT 6F - HP Vectra EX-BIOS - "F_TRACK_OFF" - DISABLE CURSOR TRACKING
    AH = 08h
    BP = 005Ah (driver ID for V_STRACK)
Return: AH = status (see #03744)
    BP,DS destroyed
Note: also removes the graphics cursor sprite from the screen
SeeAlso: AH=00h"HP Vectra",AH=04h"F_TRACK_INIT",AH=06h"F_TRACK_ON"
SeeAlso: AH=0Ah"F_DEF_MASKS",AH=0Ch"F_SET_LIMITS_X"
-----N-6F09-----
INT 6F - 10NET - SUBMIT
    AH = 09h
    DS:BX -> submit record (see #03763)
SeeAlso: AH=04h"10NET"

```

Format of 10NET submit record:

```

Offset Size Description (Table 03763)
00h 12 BYTES destination node ID (must be logged in)
0Ch WORD length+2 of following 'command line' text
0Eh N BYTES command line text (<=100 bytes), system adds CR

```

```

-----N-6F0A-----
INT 6F - 10NET - CHAT
    AH = 0Ah
    DS:BX -> control parameters (see #03764)
    DS:DX -> chat message (see #03765)
SeeAlso: AH=04h"10NET",AH=8Ah

```

Format of 10NET chat control parameters:

```

Offset Size Description (Table 03764)
00h 8 BYTES sender ID, defaults to node's userID if nulls
08h 8 BYTES destination user ID, 'EVERYONE' may be used
10h 12 BYTES destination node ID

```

Format of 10NET chat message:

```

Offset Size Description (Table 03765)
00h WORD length+2 of following text

```



02h N BYTEs text, max 101 bytes

-----b-6F0A-----

INT 6F - HP Vectra EX-BIOS - "F\_PUT\_BUFFER" - WRITE A BUFFER TO DEVICE

AH = 0Ah

BP = driver ID (see #03741) for V\_HPHIL (0114h), others

CX = number of bytes to write

ES:DI -> buffer containing data

---if driver is V\_HPHIL---

BX = device address indicator (see #03760)

DH = HP HIL device major address

DL = HP-HIL device register (00h-7Fh)

Return: AH = status (see #03744)

CX = number of unwritten bytes left in buffer

BP,DS destroyed

SeeAlso: AH=06h"F\_PUT\_BYTE",AH=0Ah"F\_PUT\_BLOCK",AH=0Ch"F\_GET\_BUFFER"

SeeAlso: AH=0Eh"F\_PUT\_WORD"

-----b-6F0A-----

INT 6F - HP Vectra EX-BIOS - "F\_PUT\_BLOCK" - WRITE BLOCKS TO DEVICE

AH = 0Ah

BP = driver ID (see #03741) for block device

DH = device major number

DL = device minor number

ES:DI -> command block (see #03766)

Return: AH = status (see #03744)

BX = operation status

BP,DS destroyed

SeeAlso: AH=06h"F\_PUT\_BYTE",AH=0Ah"F\_PUT\_BUFFER",AH=0Ch"F\_GET\_BUFFER"

SeeAlso: AH=0Eh"F\_PUT\_WORD"

Format of HP Vectra F\_PUT\_BLOCK/F\_GET\_BLOCK command block:

Offset Size Description (Table 03766)

00h DWORD data transfer address

04h WORD number of blocks

06h DWORD block address (some devices only use low word)

-----b-6F0A--BP0012-----

INT 6F - HP Vectra EX-BIOS - "F\_INS\_XCHGFREE" - EXCH VALUES WITH FREE VECTOR

AH = 0Ah

BP = 0012h (driver ID for V\_SYSTEM)

DX = new value for DS

ES:DI -> new CS:IP entry point

Return: AH = status (00h,F6h) (see also #03744)

```
    00h successful
    BX = vector address used
    DX = old value for DS
    ES:DI -> old CS:IP entry point
    BP,DS destroyed
Desc: exchange the supplied entry point and DS with the values in the next
     free entry in the HP_VECTOR_TABLE
Note: returns AH=F6h if no more reserved vectors are available
SeeAlso: AH=00h"HP Vectra",AH=04h/BP=0012h,AH=06h/BP=0012h,AH=08h/BP=0012h
SeeAlso: AH=0Ch/BP=0012h
-----b-6F0A--BP002A-----
INT 6F - HP Vectra EX-BIOS - "F_INQUIRE_FIRST" - GET PHYSICAL DRIVER VECTOR
    AH = 0Ah
    BP = 002Ah (driver ID for V_SINPUT)
Return: AH = status (see #03744)
    BX = vector address for first HP-HIL device
    BP,DS destroyed
Note: all seven HP-HIL devices use consecutive vectors in the
     HP_VECTOR_TABLE
SeeAlso: AH=00h"HP Vectra",AX=0400h/BP=002Ah,AH=06h/BP=002Ah,AH=08h/BP=002Ah
SeeAlso: AH=0Ch/BP=002Ah
-----b-6F0A--BP005A-----
INT 6F - HP Vectra EX-BIOS - "F_DEF_MASKS" - DEFINE GRAPHICS CURSOR MASKS
    AH = 0Ah
    BP = 005Ah (driver ID for V_STRACK)
    BH = width of save area in bytes
    BL = hot-spot X coordinate
    CH = height in scan lines
    CL = hot-spot Y coordinate
    ES:SI -> sprite mask (screen mask followed by XOR sprite mask)
Return: AH = status (see #03744)
    BP,DS destroyed
Note: the save area is one byte wider than the actual sprite mask; the total
     size of the sprite is at most 144 bytes
SeeAlso: AH=00h"HP Vectra",AH=04h"F_TRACK_INIT",AH=08h"F_TRACK_OFF"
SeeAlso: AH=0Ch"F_SET_LIMITS_X",AH=10h"F_PUT_SPRITE",AH=12h"F_REMOVE_SPRITE"
-----N-6F0B-----
INT 6F - 10NET - LOCK SEMAPHORE, RETURN IMMEDIATELY
    AH = 0Bh
    AL = drive number or 0
    ES:SI = Ethernet address or 0
```

DS:BX -> 31-byte ASCIIZ semaphore name  
Return: AL = status (see #03767)  
Note: same as INT 60/AH=12h  
SeeAlso: AH=0Ch"10NET",INT 60/AH=12h

(Table 03767)

Values for 10NET status:

00h successful  
01h semaphore currently locked  
02h server not responding  
03h invalid semaphore name  
04h semaphore list is full  
05h invalid drive ID  
06h invalid Ethernet address  
07h not logged in  
08h write to network failed  
09h semaphore already logged in this CPU

SeeAlso: #03740

Index: error codes;10-Net|10-Net;error codes

-----N-6F0C-----

INT 6F - 10NET - UNLOCK SEMAPHORE

AH = 0Ch  
AL = drive number or 0  
ES:SI = Ethernet address or 0  
DS:BX -> 31-byte ASCIIZ semaphore name

Return: AL = status (see also #03767)

01h semaphore not locked

Note: same as INT 60/AH=13h

SeeAlso: AH=0Bh,INT 60/AH=13h

-----b-6F0C-----

INT 6F - HP Vectra EX-BIOS - "F\_GET\_BUFFER" - READ A BUFFER OF DATA FROM DEVICE

AH = 0Ch  
BP = driver ID (see #03741) for V\_HPHIL (0114h), others  
CX = number of bytes or blocks to read  
DS:SI -> buffer for received data

Return: AH = status (see #03744 at AH=00h"HP Vectra")

BP,DS destroyed

SeeAlso: AH=08h"F\_GET\_BYTE",AH=0Ah"F\_PUT\_BUFFER",AH=0Ch"F\_GET\_BLOCK"

SeeAlso: AH=10h"F\_GET\_WORD"

-----b-6F0C-----

INT 6F - HP Vectra EX-BIOS - "F\_GET\_BLOCK" - READ BLOCKS OF DATA FROM DEVICE

AH = 0Ch  
BP = driver ID (see #03741)  
DH = device major number  
DL = device minor number  
ES:DI -> command block (see #03766)  
Return: AH = status (see #03744 at AH=00h"HP Vectra")  
BX = operation status  
BP,DS destroyed  
SeeAlso: AH=08h"F\_GET\_BYTE",AH=0Ah"F\_PUT\_BLOCK",AH=0Ch"F\_GET\_BUFFER"  
SeeAlso: AH=10h"F\_GET\_WORD"  
-----b-6F0C--BP0012-----  
INT 6F - HP Vectra EX-BIOS - "F\_INS\_FIXOWNDS" - INSTALL VALUES IN FIXED VECTOR  
AH = 0Ch  
BP = 0012h (driver ID for V\_SYSTEM)  
BX = vector address used  
ES:DI -> new CS:IP entry point  
Return: AH = status (00h) (see #03744)  
BP,DS destroyed  
Note: on installation, the given entry point will be invoked with an SF\_INIT  
call (see AX=0200h"SF\_INIT"), which should return the routine's DS  
in BX. If the SF\_INIT call returns with error code FEh, the power-on  
self-test sequence will be called.  
SeeAlso: AH=00h"HP Vectra",AH=04h/BP=0012h,AH=06h/BP=0012h,AH=08h/BP=0012h  
SeeAlso: AH=0Eh/BP=0012h,AH=12h/BP=0012h  
-----b-6F0C--BP002A-----  
INT 6F - HP Vectra EX-BIOS - "F\_INQUIRE\_ENTRY" - GET GID DRIVER ENTRY POINT  
AH = 0Ch  
BP = 002Ah (driver ID for V\_SINPUT)  
Return: AH = status (see #03744)  
ES:BX -> physical GID driver CS:IP  
BP,DS destroyed  
SeeAlso: AH=00h"HP Vectra",AX=0400h/BP=002Ah,AH=06h/BP=002Ah,AH=08h/BP=002Ah  
SeeAlso: AH=0Ah/BP=002Ah  
-----b-6F0C--BP005A-----  
INT 6F - HP Vectra EX-BIOS - "F\_SET\_LIMITS\_X" - SET HORIZONTAL TRACKING LIMITS  
AH = 0Ch  
BP = 005Ah (driver ID for V\_STRACK)  
CX = minimum X coordinate  
DX = maximum X coordinate  
Return: AH = status (see #03744)  
BP,DS destroyed

SeeAlso: AH=00h"HP Vectra",AH=04h/BP=005Ah,AH=08h/BP=005Ah,AH=0Eh/BP=005Ah  
<http://www.foxitsoftware.com> For evaluation only.

SeeAlso: AH=10h/BP=005Ah,AH=12h/BP=005Ah,INT 33/AX=0007h

-----N-6F0D-----

INT 6F - 10NET - "WHO" - ENUMERATE USERS ON NETWORK

AH = 0Dh

AL = type code

01h return superstations only

02h return non-superstations only

otherwise return all

CX = length of data buffer

DS:DX -> array of records to be filled (see #03768)

Return: CL = number of records returned (responding stations)

SeeAlso: AH=16h,AH=8Dh

Format of station record:

Offset Size Description (Table 03768)

00h 12 BYTES node ID

0Ch BYTE station flags (see #03769)

---if AL = 01h---

0Dh BYTE version number

0Eh WORD level number of 10Net software in responding node

---if AL = 02h---

0Dh 8 BYTES user ID

15h BYTE version number

16h WORD level number

Bitfields for station flags:

Bit(s) Description (Table 03769)

1 workstation

2 superstation

3 xgate

4 active gate

-----N-6F0E-----

INT 6F - 10NET - SPOOL/PRINT

AH = 0Eh

DS:DX -> spool/print record (see #03771)

Return: CF clear if successful

CF set on error

AX = error code (see also #03740)

17FFh device not mounted

18FFh already spooling to named device

(Table 03770)

Values for 10NET spooler operation code:

```
0000h initiate spool
0001h abort print
0002h close spool
0003h delete spool
0004h print
0005h get report info
0006h set chat template
0007h queue
0008h return queue
0009h queue non-spooled file for printing
```

Format of 10NET Spool/Print record:

Offset Size Description (Table 03771)

00h WORD operation code (see #03770)

02h 11 BYTES file name in FCB format

---if operation code = 00h or 06h---

0Dh BYTE notification flags (see #03772)

0Eh BYTE days to keep (FFh=forever)

0Fh BYTE bits 0,1: device (1=LPT1)

bits 4-7: remote drive to store spool file (1=A,...)

10h WORD length of following data area

12h N BYTES up to 64 bytes of description

---if operation code = 03h---

0Dh 8 BYTES user ID to associate with filename

---if operation code = 04h---

0Dh WORD block number

0Fh 8 BYTES user ID to associate with filename

---if operation code = 05h---

0Dh BYTE RRN to start retrieve

0Eh BYTE bits 0,1: local print device (LPTx)

bit 3: if set, return entries for all users

0Fh WORD length of following area

11h N BYTES up to 1500 bytes to receive \$SCNTL records returned

---if operation code = 07h---

0Dh BYTE queue number

0Eh BYTE bits 0,1: local print device (LPTx)

0Fh WORD number of bytes of test print to be done

11h BYTE code:

```

01h print device
02h test print count
03h prn
---if operation code = 08h---
0Dh BYTE queue location or $SCNTL location to start access
returns next item for access:
    00h-7Fh queued items
    80h-FEh non-queued, non-printed items
    FFh no more items
0Eh WORD unused
10h WORD length of following area
12h N BYTES up to 64 bytes to receive $SCNTL records (see #03773)
---if operation code = 09h---
0Dh 3 BYTES unused
10h N BYTES path to non-spooled file to be queued for printing

```

## Bitfields for notification flags:

Bit(s) Description (Table 03772)

```

7 queue to top
6 do ID page
5 no form feed
4 reserved
3 explicit queuing only
2 notify at print completion
1 notify server operator, with reply
0 notify at print start

```

## Format of 10NET \$SCNTL record:

Offset Size Description (Table 03773)

```

00h 8 BYTES user ID
08h 11 BYTES filename in FCB format
13h 6 BYTES node ID
19h 3 BYTES creation date
1Ch BYTE notification flags (see #03772)
1Dh BYTE retention time in days
1Eh BYTE printing device (LPTx)
1Fh 3 BYTES date last printed (0 = never)
22h BYTE device containing spoolfile
23h WORD bytes to print for test print
25h WORD block number to start print
27h BYTE reserved

```

-----b-6F0E-----

INT 6F - HP Vectra EX-BIOS - "F\_PUT\_WORD" - WRITE WORD OF DATA TO DEVICE

AH = 0Eh

BP = driver ID (see #03741)

DX = data word

Return: AH = status (see #03744 at AH=00h"HP Vectra")

BP,DS destroyed

SeeAlso: AH=06h"F\_PUT\_BYTE",AH=0Ah"F\_PUT\_BUFFER",AH=10h"F\_GET\_WORD"

-----b-6F0E-----

INT 6F - HP Vectra EX-BIOS - "F\_PUT\_WORD" - WRITE A WORD OF DATA

AH = 0Eh

BP = driver ID (see #03741)

??? details not yet available

Return: ???

SeeAlso: AH=06h"HP Vectra",AH=0Ah"HP Vectra",AH=10h"HP Vectra"

-----b-6F0E--BP0012-----

INT 6F - HP Vectra EX-BIOS - "F\_INS\_FIXGETDS" - INSTALL VALUES IN FIXED VECTOR

AH = 0Eh

BP = 0012h (driver ID for V\_SYSTEM)

BX = vector address used

ES:DI -&gt; new CS:IP entry point

Return: AH = status (00h) (see #03744)

BP,DS destroyed

Note: on installation, the given entry point will be invoked with an SF\_INIT call (see AX=0200h"SF\_INIT"), with the "last used DS" value in BX; the routine should adjust BX and return the new value. If the SF\_INIT call returns with error code FEh, the power-on self-test sequence will be called.

SeeAlso: AH=00h"HP Vectra",AH=04h/BP=0012h,AH=06h/BP=0012h,AH=0Ch/BP=0012h

SeeAlso: AH=10h/BP=0012h,AH=14h/BP=0012h

-----b-6F0E--BP005A-----

INT 6F - HP Vectra EX-BIOS - "F\_SET\_LIMITS\_Y" - SET VERTICAL TRACKING LIMITS

AH = 0Eh

BP = 005Ah (driver ID for V\_STRACK)

CX = minimum Y coordinate

DX = maximum Y coordinate

Return: AH = status (see #03744)

BP,DS destroyed

SeeAlso: AH=00h"HP Vectra",AH=04h/BP=005Ah,AH=08h/BP=005Ah,AH=0Ch/BP=005Ah

SeeAlso: AH=10h/BP=005Ah,AH=12h/BP=005Ah,INT 33/AX=0008h

-----N-6F0F-----



INT 6F - 10NET v5.0 - "RM LOCK" - ???

AH = 0Fh

???

Return: ???

SeeAlso: AH=07h,AH=08h"10NET"

-----N-6F10-----

INT 6F - 10NET - ATTACH/DETACH PRINTER

AH = 10h

AL = subfunction

00h initiate spooling if LPT1 is mounted

01h terminate spooling if LPT1 is mounted

SeeAlso: INT 21/AX=5D08h

-----b-6F10-----

INT 6F - HP Vectra EX-BIOS - "F\_GET\_WORD" - READ WORD OF DATA FROM DEVICE

AH = 10h

BP = driver ID (see #03741)

Return: AH = status (see #03744 at AH=00h"HP Vectra")

DX = data word

BP,DS destroyed

SeeAlso: AH=08h"F\_GET\_BYTE",AH=0Ch"F\_GET\_BUFFER",AH=0Eh"F\_PUT\_WORD"

-----b-6F10--BP0012-----

INT 6F - HP Vectra EX-BIOS - "F\_INS\_FIXGLBDS" - INSTALL VALUES IN FIXED VECTOR

AH = 10h

BP = 0012h (driver ID for V\_SYSTEM)

BX = vector address used

ES:DI -> new CS:IP entry point

Return: AH = status (00h) (see #03744)

BP,DS destroyed

Note: on installation, the given entry point will be invoked with an SF\_INIT call (see AX=0200h"SF\_INIT"), with BX set to the EX-BIOS global data area. If the SF\_INIT call returns with error code FEh, the power-on self-test sequence will be called.

SeeAlso: AH=00h"HP Vectra",AH=04h/BP=0012h,AH=06h/BP=0012h

SeeAlso: AH=0Ch/BP=0012h,AH=0Eh/BP=0012h,AH=12h/BP=0012h,AH=16h/BP=0012h

-----b-6F10--BP005A-----

INT 6F - HP Vectra EX-BIOS - "F\_PUT\_SPRITE" - DISPLAY GRAPHICS CURSOR SPRITE

AH = 10h

BP = 005Ah (driver ID for V\_STRACK)

BX = X coordinate

CX = Y coordinate

Return: AH = status (see #03744)

BP,DS destroyed

SeeAlso: AH=00h"HP Vectra",AH=04h"F\_TRACK\_INIT",AH=08h"F\_TRACK\_OFF"

SeeAlso: AH=12h"F\_REMOVE\_SPRITE",INT 33/AX=0004h

-----N-6F11-----

INT 6F - 10NET - LOCK FCB

AH = 11h

AL = mode

01h sequential

02h random

03h random block

CX = number of records

DS:DX -> FCB (see #01345 at INT 21/AH=0Fh)

Return: CF clear if successful

CF set on error

AX = error code (see also #03740)

0002h file not found

SeeAlso: AH=12h,INT 21/AH=0Fh

-----N-6F12-----

INT 6F - 10NET - UNLOCK FCB

AH = 12h

AL = mode

00h sequential

01h random

02h random block

CX = number of records

DS:DX -> FCB (see #01345 at INT 21/AH=0Fh)

Return: CF clear if successful

CF set on error

AX = error code (see also #03740)

0002h file not found

SeeAlso: AH=11h

-----b-6F12--BP0012-----

INT 6F - HP Vectra EX-BIOS - "F\_INS\_FREEOWNDS" - INSTALL VALUES IN FREE VECTOR

AH = 12h

BP = 0012h (driver ID for V\_SYSTEM)

ES:DI -> new CS:IP entry point

Return: AH = status (00h) (see #03744)

BP,DS destroyed

Note: on installation, the given entry point will be invoked with an SF\_INIT

call (see AX=0200h"SF\_INIT"), which should return the routine's DS

in BX. If the SF\_INIT call returns with error code FEh, the power-on

self-test sequence will be called.

SeeAlso: AH=00h"HP Vectra",AH=04h/BP=0012h,AH=0Ch/BP=0012h,AH=14h/BP=0012h

SeeAlso: AH=16h/BP=0012h,AH=18h/BP=0012h

-----b-6F12--BP005A-----

INT 6F - HP Vectra EX-BIOS - "F\_REMOVE\_SPRITE" - REMOVE GRAPHICS CURSOR SPRITE

AH = 12h

BP = 005Ah (driver ID for V\_STRACK)

Return: AH = status (see #03744)

BP,DS destroyed

SeeAlso: AH=00h"HP Vectra",AH=06h"F\_TRACK\_ON",AH=10h"F\_PUT\_SPRITE"

-----N-6F13-----

INT 6F - 10NET v3.3+ - GET REMOTE CONFIGURATION TABLE ADDRESS

AH = 13h

DS:DX -> node ID, 12 bytes blank-padded

Return: CF clear if successful

ES:BX = configuration table address on given machine

CF set on error

AX = error code (see #03740)

SeeAlso: AH=03h

-----N-6F14-----

INT 6F - 10NET v3.3+ - GET REMOTE MEMORY

AH = 14h

BX:SI = address of remote memory

CX = length (<=1024 bytes)

DS:DX -> node ID, 12 bytes blank-padded

DS:DI -> area to receive remote memory image

Return: CF clear if successful

CX = amount of memory copied to DS:SI

CF set on error

AX = error code (see #03740)

-----b-6F14--BP0012-----

INT 6F - HP Vectra EX-BIOS - "F\_INS\_FREEGETDS" - INSTALL VALUES IN FREE VECTOR

AH = 14h

BP = 0012h (driver ID for V\_SYSTEM)

ES:DI -> new CS:IP entry point

Return: AH = status (00h) (see #03744)

BP,DS destroyed

Note: on installation, the given entry point will be invoked with an SF\_INIT

call (see AX=0200h"SF\_INIT"), with the "last used DS" value in BX;

the routine should adjust BX and return the new value. If the

SF\_INIT call returns with error code FEh, the power-on self-test

sequence will be called.

SeeAlso: AH=00h"HP Vectra",AH=04h/BP=0012h,AH=0Eh/BP=0012h,AH=12h/BP=0012h

SeeAlso: AH=16h/BP=0012h

-----N-6F1501-----

INT 6F - 10NET v3.3+ - GET SHARED DEVICE ENTRY

AX = 1501h

BX = zero-based index

DS:SI -> node ID, 12 bytes blank-padded

ES:DI -> 85-byte buffer for shared device table entry (see #03774)

Return: CF clear if successful

ES:DI buffer contains shared device table entry of BXth device

CF set on error

AX = error code (see #03740)

SeeAlso: AX=1502h,AX=1503h,AX=9501h

Format of 10NET shared device table entry:

Offset Size Description (Table 03774)

00h 8 BYTES device

08h 8 BYTES alias

10h 64 BYTES path

50h 8 BYTES password

58h BYTE access

59h 4 BYTES mask

-----N-6F1502-----

INT 6F - 10NET v3.3+ - SET SHARED DEVICE ENTRY

AX = 1502h

DS:SI -> node ID, 12 bytes blank-padded

ES:DI -> valid shared device table entry

Return: CF clear if successful

CF set on error

AX = error code (see #03740)

SeeAlso: AX=1501h,AX=1503h,AX=9502h

-----N-6F1503-----

INT 6F - 10NET v3.3+ - DELETE SHARED DEVICE ENTRY

AX = 1503h

BX = zero-based index

DS:SI -> node ID, 12 bytes blank-padded

Return: CF clear if successful

CF set on error

AX = error code (see #03740)

SeeAlso: AX=1501h,AX=1502h,AX=9503h

-----N-6F16-----

INT 6F - 10NET v5.0 - "GL WHO" - ???

AH = 16h

???

Return: ???

SeeAlso: AH=0Dh

-----b-6F16--BP0012-----

INT 6F - HP Vectra EX-BIOS - "F\_INS\_FREEGLBDS" - INSTALL VALUES IN FREE VECTOR

AH = 16h

BP = 0012h (driver ID for V\_SYSTEM)

ES:DI -> new CS:IP entry point

Return: AH = status (00h) (see #03744 at INT 6F/AH=00h"HP Vectra")

BP,DS destroyed

Note: on installation, the given entry point will be invoked with an SF\_INIT call (see AX=0200h"SF\_INIT"), with BX set to the EX-BIOS global data area. If the SF\_INIT call returns with error code FEh, the power-on self-test sequence will be called.

SeeAlso: AH=00h"HP Vectra",AH=04h/BP=0012h,AH=10h/BP=0012h,AH=12h/BP=0012h

SeeAlso: AH=14h/BP=0012h,AH=18h/BP=0012h

-----N-6F17-----

INT 6F - 10NET v3.3+ - MOUNT

AH = 17h

AL = local drive number (0=A:)

BL = remote drive letter or '1'..'3' for LPTn or '4' or '5' for COMx

DS:DX -> node ID, 12 bytes blank-padded

Return: CF clear if successful

CF set on error

AX = error code (see #03740)

SeeAlso: AH=18h

-----N-6F18-----

INT 6F - 10NET v3.3+ - UNMOUNT

AH = 18h

AL = local drive number (0=A:)

BL = type

00h disk

01h-03h LPTn

04h,05h COMx

Return: CF clear if successful

CF set on error

AX = error code (see #03740)

SeeAlso: AH=17h

-----b-6F18--BP0012-----

INT 6F - HP Vectra EX-BIOS - "F\_INS\_FIND" - FIND DRIVER BY ATTRIBUTES

AH = 18h  
BP = 0012h (driver ID for V\_SYSTEM)  
AL = sense of test (00h equal, 02h non-equal)  
BX = AND mask  
DX = pattern  
SI = first vector address to search  
DI = offset in HP header of field to test

Return: AH = status (00h,FEh) (see #03744 at INT 6F/AH=00h"HP Vectra")

BP,DS destroyed

---if AH=00h---

SI = first matching vector address

Desc: compare successive drivers' header fields starting at address SI

until the header field AND BX is either equal or non-equal to DX

SeeAlso: AH=00h"HP Vectra",AH=04h/BP=0012h,AH=0Ch/BP=0012h,AH=10h/BP=0012h

SeeAlso: AH=12h/BP=0012h,AH=16h/BP=0012h

-----N-6F19-----

INT 6F U - 10NET v5.0 - AUDIT

AH = 19h

???

Return: ???

SeeAlso: AH=99h

-----N-6F1A-----

INT 6F U - 10NET v5.0 - "BULL" - ???

AH = 1Ah

???

Return: ???

-----N-6F1B-----

INT 6F U - 10NET v5.0 - "GMOUNT" - ???

AH = 1Bh

???

Return: ???

-----N-6F1C-----

INT 6F U - 10NET v5.0 - "GLOGIN" - GET LOGIN LIST

AH = 1Ch

???

Return: ???

-----N-6F1D-----

INT 6F U - 10NET v5.0 - "TABDATA" - ???

AH = 1Dh

```
    ???
Return: ???
-----N-6F1E-----
INT 6F U - 10NET v5.0 - "SCHED" - ???
    AH = 1Eh
    ???
Return: ???
-----b-6F1E--BP0012-----
INT 6F - HP Vectra EX-BIOS - "F_RAM_GET" - GET EX-BIOS FREE RAM AREA
    AH = 1Eh
    BP = 0012h (driver ID for V_SYSTEM)
Return: AH = status (00h) (see #03744 at INT 6F/AH=00h"HP Vectra")
    BX = last-used DS
    DX = maximum DS
    BP,DS destroyed
SeeAlso: AH=00h"HP Vectra",AH=04h/BP=0012h,AH=20h/BP=0012h,AH=22h/BP=0012h
-----N-6F1F-----
INT 6F U - 10NET v5.0 - "WHOAMI" - ???
    AH = 1Fh
    ???
Return: ???
-----N-6F20-----
INT 6F U - 10NET v5.0 - ???
    AH = 20h
    ???
Return: ???
-----b-6F20--BP0012-----
INT 6F - HP Vectra EX-BIOS - "F_RAM_RET" - RESERVE EX-BIOS MEMORY
    AH = 20h
    BP = 0012h (driver ID for V_SYSTEM)
    BX = new value for "last-used DS"
    DX = new value for "maximum DS"
Return: AH = status (00h) (see #03744 at INT 6F/AH=00h"HP Vectra")
    BP,DS destroyed
Note: the F_INS_FIXGETDS and F_INS_FREEGETDS functions
    (see AH=0Eh/BP=0012h,AH=14h/BP=0012h) also modify the values returned
    by this call, requiring care if both methods are used to allocate
    memory
SeeAlso: AH=00h"HP Vectra",AH=0Eh/BP=0012h,AH=14h/BP=0012h,AH=1Eh/BP=0012h
-----N-6F21-----
INT 6F U - 10NET v5.0 - ???
```

```
AH = 21h
???
```

Return: ???

-----N-6F22-----

INT 6F U - 10NET v5.0 - ???

```
AH = 22h
???
```

Return: ???

-----b-6F22--BP0012-----

INT 6F - HP Vectra EX-BIOS - "F\_CMOS\_GET" - READ CMOS MEMORY

```
AH = 22h
BP = 0012h (driver ID of V_SYSTEM)
BL = address of CMOS byte to read
```

Return: AH = status (see #03744 at AH=00h"HP Vectra")

```
AL = byte read
BP,DS destroyed
```

Note: supported by ES, QS, and RS series HP Vectras

SeeAlso: AH=00h"HP Vectra",AH=1Eh/BP=0012h,AH=24h/BP=0012h

-----b-6F24--BP0012-----

INT 6F - HP Vectra EX-BIOS - "F\_CMOS\_RET" - WRITE CMOS MEMORY

```
AH = 24h
BP = 0012h (driver ID of V_SYSTEM)
BL = address of CMOS byte to write
AL = new value
```

Return: AH = status (see #03744)

```
BP,DS destroyed
```

Note: updates the CMOS checksum fields (both IBM-standard and HP checksums)

SeeAlso: AH=00h"HP Vectra",AH=22h/BP=0012h

-----b-6F2A--BP0012-----

INT 6F - HP Vectra EX-BIOS - "F\_YIELD" - GIVE UP CPU TO OTHER TASKS

```
AH = 2Ah
BP = 0012h (driver ID of V_SYSTEM)
```

Return: AH = status (see #03744)

```
BP,DS destroyed
```

Note: this is a hook for multitasking systems

SeeAlso: AH=00h"HP Vectra",INT 15/AX=1000h,INT 2F/AX=1680h

-----b-6F30--BP0012-----

INT 6F - HP Vectra EX-BIOS - "F\_SND\_CLICK\_ENABLE" - TURN ON KEYCLICKS

```
AH = 30h
BP = 0012h (driver ID of V_SYSTEM)
```

Return: AH = status (see #03744 at AH=00h"HP Vectra")



BP,DS destroyed  
Desc: enables keyclicks and flushes any pending keyclicks  
SeeAlso: AH=32h/BP=0012h,AH=34h/BP=0012h,AH=36h/BP=0012h  
-----b-6F32--BP0012-----  
INT 6F - HP Vectra EX-BIOS - "F\_SND\_CLICK\_DISABLE" - TURN OFF KEYCLICKS  
AH = 32h  
BP = 0012h (driver ID of V\_SYSTEM)  
Return: AH = status (see #03744)  
BP,DS destroyed  
Desc: disables keyclicks and flushes any pending keyclicks  
SeeAlso: AH=00h"HP Vectra",AH=30h/BP=0012h,AH=38h/BP=0012h  
-----b-6F34--BP0012-----  
INT 6F - HP Vectra EX-BIOS - "F\_SND\_CLICK" - GENERATE KEYCLICK  
AH = 34h  
BP = 0012h (driver ID of V\_SYSTEM)  
Return: AH = status (see #03744)  
BP,DS destroyed  
Desc: generate a keyclick if fewer than four clicks are already pending  
Note: if keyclicks are already pending, the remaining click count is  
incremented and the function returns immediately  
SeeAlso: AH=00h"HP Vectra",AH=30h/BP=0012h,AH=36h/BP=0012h,AH=3Ah/BP=0012h  
-----b-6F36--BP0012-----  
INT 6F - HP Vectra EX-BIOS - "F\_SND\_BEEP\_ENABLE" - ENABLE BEEPS  
AH = 36h  
BP = 0012h (driver ID of V\_SYSTEM)  
Return: AH = status (see #03744 at AH=00h"HP Vectra")  
BP,DS destroyed  
SeeAlso: AH=00h"HP Vectra",AH=30h/BP=0012h,AH=34h/BP=0012h,AH=38h/BP=0012h  
-----b-6F38--BP0012-----  
INT 6F - HP Vectra EX-BIOS - "F\_SND\_BEEP\_DISABLE" - DISABLE BEEPS  
AH = 38h  
BP = 0012h (driver ID of V\_SYSTEM)  
Return: AH = status (see #03744)  
BP,DS destroyed  
SeeAlso: AH=00h"HP Vectra",AH=32h/BP=0012h,AH=36h/BP=0012h  
-----b-6F3A--BP0012-----  
INT 6F - HP Vectra EX-BIOS - "F\_SND\_BEEP" - SOUND BEEP WITH CURRENT VALUES  
AH = 3Ah  
BP = 0012h (driver ID of V\_SYSTEM)  
Return: AH = status (see #03744 at AH=00h"HP Vectra")  
BP,DS destroyed

Desc: make a sound of the frequency and duration defined in the EX-BIOS

global data area (see #03748)

SeeAlso: AH=00h"HP Vectra",AH=34h/BP=0012h,AH=36h/BP=0012h,AH=38h/BP=0012h

SeeAlso: AH=3Ch/BP=0012h,AH=3Eh/BP=0012h

-----b-6F3C--BP0012-----

INT 6F - HP Vectra EX-BIOS - "F\_SND\_SET\_BEEP" - SET BEEP FREQUENCY AND DURATION

AH = 3Ch

BP = 0012h (driver ID of V\_SYSTEM)

BX = frequency in Hz (1-25000, 0 = off)

DX = duration in 10-microsecond increments

Return: AH = status (see #03744)

BP,DS destroyed

SeeAlso: AH=00h"HP Vectra",AH=3Ah/BP=0012h,AH=3Eh/BP=0012h,#03748

-----b-6F3E--BP0012-----

INT 6F - HP Vectra EX-BIOS - "F\_SND\_TONE" - SOUND TONE WITH FREQ AND DURATION

AH = 3Eh

BP = 0012h (driver ID of V\_SYSTEM)

BX = frequency in Hz (1-25000, 0 = off)

DX = duration in 10-microsecond increments

Return: AH = status (see #03744)

BP,DS destroyed

SeeAlso: AH=00h"HP Vectra",AH=3Ah/BP=0012h,AH=3Ch/BP=0012h,AH=40h/BP=0012h

-----b-6F40--BP0012-----

INT 6F - HP Vectra EX-BIOS - "F\_STR\_GET\_FREE\_INDEX" - FIND FREE STRING INDEX

AH = 40h

BP = 0012h (driver ID of V\_SYSTEM)

Return: AH = status (see #03744)

BX = next free string index

BP,DS destroyed

Desc: get the next string index available to user programs

SeeAlso: AH=00h"HP Vectra",AH=42h/BP=0012h,AH=44h/BP=0012h,AH=46h/BP=0012h

-----b-6F42--BP0012-----

INT 6F - HP Vectra EX-BIOS - "F\_STR\_DEL\_BUCKET" - DEL HEADER FROM BUCKET LIST

AH = 42h

BP = 0012h (driver ID of V\_SYSTEM)

ES:DI -> bucket header (see #03775)

Return: AH = status (00h,FEh) (see #03744 at AH=00h"HP Vectra")

BP,DS destroyed

SeeAlso: AH=00h"HP Vectra",AH=40h/BP=0012h,AH=44h/BP=0012h

Format of HP Vectra EX-BIOS bucket header:

```
Offset  Size  Description (Table 03775)
00h    DWORD pointer to next bucket header (FFFFh:FFFFh if last)
04h    WORD   upper bound on string index
06h    WORD   lower bound on string index
08h    DWORD pointer to array of string offset pointers
0Ch    WORD   segment of strings
```

Note: the strings pointed at by the offset pointers are byte-counted ASCIZ strings

-----b-6F44--BP0012-----

INT 6F - HP Vectra EX-BIOS - "F\_STR\_PUT\_BUCKET" - ADD HEADER TO BUCKET HDR LIST

```
AH = 44h
BP = 0012h (driver ID of V_SYSTEM)
ES:DI -> bucket header (see #03775)
```

Return: AH = status (00h) (see #03744)

BP,DS destroyed

SeeAlso: AH=00h"HP Vectra",AH=42h/BP=0012h,AH=46h/BP=0012h

-----b-6F46--BP0012-----

INT 6F - HP Vectra EX-BIOS - "F\_STR\_GET\_STRING" - FIND STRING IN BUCKET LIST

```
AH = 46h
BP = 0012h (driver ID of V_SYSTEM)
BX = string index
```

Return: AH = status (00h,other) (see #03744)

BP,DS destroyed

---if successful---

```
CX = length of string (excluding count byte and terminating NUL)
DS:SI -> header for string
ES:DI -> found string
```

SeeAlso: AH=00h"HP Vectra",AH=44h/BP=0012h,AH=48h/BP=0012h

-----b-6F48--BP0012-----

INT 6F - HP Vectra EX-BIOS - "F\_STR\_GET\_INDEX" - GET INDEX FOR STRING

```
AH = 48h
BP = 0012h (driver ID of V_SYSTEM)
ES:DI -> ASCIZ string
```

Return: AH = status (00h,other) (see #03744 at AH=00h"HP Vectra")

BP,DS destroyed

---if successful---

BX = index for the given string

SeeAlso: AH=00h"HP Vectra",AH=40h/BP=0012h,AH=46h/BP=0012h

-----A-6F636FBX6D70-----

INT 6F - Clara Empricost - TSR Version - INSTALLATION CHECK

AX = 636Fh ('co')

BX = 6D70h ('mp')  
CX = 696Eh ('in')  
SS:DX -> WORD to set if installed  
Return: SS:DX word set to 636Fh if installed  
Program: Condition List Action Rejection Algorithm for the Evaluation  
of Music at Pseudo-Random Idea Construction Stage  
(CLARA EMPRICOST) is an experimental program by Silas Brown  
SeeAlso: AX=636Fh/BX=6D70h"UNINSTALL"

-----A-6F636FBX6D70-----

INT 6F - Clara Empricost - TSR Version - UNINSTALL

AX = 636Fh ('co')  
BX = 6D70h ('mp')  
CX = 6F73h ('os')

Return: nothing -- call installation check to determine whether successful  
Program: Condition List Action Rejection Algorithm for the Evaluation  
of Music at Pseudo-Random Idea Construction Stage  
(CLARA EMPRICOST) is an experimental program by Silas Brown

Notes: Saves any unfinished task and prevents processing of  
new tasks, even if can't uninstall  
If INT 28h or INT 6Fh have been chained since installation,  
returns without changing interrupts or freeing memory  
do not call while DOS is active (i.e. InDOS flag is nonzero or INT 28  
is active and InDOS > 1)

SeeAlso: AX=636Fh/BX=6D70h"INSTALLATION CHECK"

-----N-6F80-----

INT 6F - 10NET v5.0 - LOGIN

AH = 80h  
DS:DX -> login record (see #03776)

Return: CF clear if successful  
BL = number of days until password expires (00h = never)  
CL = security level  
CF set on error  
AX = status (see #03740)

SeeAlso: AH=00h,AH=81h

Format of 10NET login record:

Offset	Size	Description (Table 03776)
00h	8 BYTES	user name
08h	8 BYTES	password
10h	15 BYTES	server node ID
1Fh	8 BYTES	new password

27h BYTE invoke mode (00h command line, 01h interactive)

-----N-6F81-----

INT 6F - 10NET v5.0 - LOGOUT

AH = 81h

DS:DX -> server node ID (DX=0000h for universal logout)

Return: AX = status (see #03740)

SeeAlso: AH=01h,AH=80h

-----N-6F8A-----

INT 6F - 10NET v5.0 - CHAT

AH = 8Ah

DS:BX -> chat parameters (see #03777)

DS:DX -> chat message (see #03778)

Return: CF clear if successful

CF set on error

AX = status (see #03740)

SeeAlso: AH=0Ah"10NET"

Format of 10NET chat parameters:

Offset Size Description (Table 03777)

00h 8 BYTES sender's user name

08h 8 BYTES destination user name

10h 15 BYTES destination node (0 if broadcast-style chat)

Format of 10NET chat message:

Offset Size Description (Table 03778)

00h WORD message length

02h N BYTES chat message contents

-----N-6F8D-----

INT 6F - 10NET v5.0 - "WHO" - ENUMERATE USERS ON NETWORK

AH = 8Dh

BX = service mask (see #03779)

CX = length of buffer

DS:DX -> buffer for array of Who data structures (see #03780)

Return: CF clear if successful

CX = number of nodes matching service mask

DS:DX buffer filled

CF set on error

AX = status (see #03740)

SeeAlso: AH=0Dh

Bitfields for 10NET service mask:

## Bit(s) Description (Table 03779)

0 workstation  
1 file server  
2 print server  
3 de-spool server

## Format of 10NET Who data structure:

## Offset Size Description (Table 03780)

00h 8 BYTES user name  
08h 15 BYTES node ID  
17h 3 BYTES unique portion of Ethernet address  
1Ah BYTE Who group number  
1Bh WORD service mask (see #03779)  
1Dh DWORD serial number  
21h BYTE maximum concurrent users with same serial number allowed on net  
22h BYTE chat mask (see #03781)  
23h BYTE internal system bits (see #03782)  
24h 9 BYTES version number in format MM.mm.xxx  
2Dh BYTE number of shared directories  
2Eh BYTE number of shared printer queues

## Bitfields for 10NET chat mask:

## Bit(s) Description (Table 03781)

0 chat permitted  
1 bell enabled  
2 chat keyboard initiated  
3 in INT 16 handler  
4 in Get Input  
5 display has timed out  
6 chat is idle

## Bitfields for 10NET internal system bits:

## Bit(s) Description (Table 03782)

0 submit permitted  
1 submit initiated  
2 submit executing  
3 internal client call/chat/spool/autospool  
4 in spool termination  
5 print permitted  
6 waiting for keyboard input

-----N-6F9501-----

```
INT 6F - 10NET v5.0 - GET SHARED DEVICE ENTRY
  AX = 9501h
  BX = zero-based index
  DS:SI -> server's node ID
  ES:DI -> buffer for shared device structure (see #03783)
Return: CF clear if successful
  CF set on error
  AX = status (see #03740)
SeeAlso: AX=1501h,AX=9502h,AX=9503h,AX=9504h
```

Format of 10NET shared device structure:

```
Offset Size Description (Table 03783)
00h 8 BYTES alias
08h BYTE type (02h modem, 03h print queue, 04h directory)
09h BYTE access rights
  bit 0: read
  bit 1: write
  bit 2: create
0Ah 8 BYTES password
12h 32 BYTES comment
---directory---
 32h 64 BYTES pathname of shared directory
---print queue---
 32h BYTE notification bit mask (see #03784)
 33h BYTE job control bit mask (see #03785)
 34h WORD number of days to retain file
 36h WORD test print length
 38h BYTE number of copies to print
 39h BYTE compression algorithm
 3Ah BYTE tab width (00h = don't expand)
 3Bh BYTE priority
 3Ch WORD time to open queue (FFFFh = always)
 3Eh WORD time to close queue
 40h WORD pause following queue switch, in clock ticks
 42h WORD pause between print jobs, in clock ticks
 44h BYTE associate queue file existence mask
  bit 0: queue switch file exists
  bit 1: initiate file exists
  bit 2: abort file exists
 45h 6 BYTES character sequence for page eject
 4Bh BYTE status of print queue
```

bit 0: queue is closed  
4Ch WORD number of jobs on queue  
4Eh WORD offset of next print job to be dispatched (FFFFh = none)  
50h BYTE number of print devices in printer pool  
51h WORD offset of first print device structure (FFFFh if empty)

Bitfields for notification flags:

Bit(s) Description (Table 03784)

0 user at print start  
1 operator at start, with reply  
2 user at print completion  
3 operator at completion, with reply  
4 user on queue switch  
5 operator on queue switch, with reply  
6 user on print error

Bitfields for 10NET job control mask:

Bit(s) Description (Table 03785)

0 print banner page  
1 eject page at end of job  
2 mark as "held" (queue but don't print)  
3 rush job (queue at top)  
4 overwrite file with zeros before deletion  
5 hyperspool if possible

-----N-6F9502-----

INT 6F - 10NET v5.0 - SET SHARED DEVICE ENTRY

AX = 9502h

DS:SI -> server's node ID

ES:DI -> shared device structure (see #03783)

Return: CF clear if successful

CF set on error

AX = status (see #03740)

SeeAlso: AX=1502h,AX=9501h,AX=9503h,AX=9504h

-----N-6F9503-----

INT 6F - 10NET v5.0 - DELETE SHARED DEVICE

AX = 9503h

BX = zero-based index

DS:SI -> server's node ID

Return: CF clear if successful

CF set on error

AX = status (see #03740)



SeeAlso: AX=1503h,AX=9501h,AX=9502h,AX=9504h

-----N-6F9504-----

INT 6F - 10NET v5.0 - ENUMERATE USERS OF SHARED DEVICE

AX = 9504h

BX = zero-based shared device index

CX = zero-based user index

DS:SI -> server's node ID

Return: CF clear if successful

ES:DI -> node ID of the CX'th user of the BX'th device

CF set on error

AX = status (see #03740)

SeeAlso: AX=9501h,AX=9502h,AX=9503h

-----N-6F99-----

INT 6F - 10NET v5.0 - AUDIT

AH = 99h

DS:SI -> server's node ID

ES:DI -> data to be appended to audit trail file (max 106 bytes)

Return: CF clear if successful

CF set on error

AX = status (see #03740)

SeeAlso: AH=19h,AH=9Ch

-----N-6F9C-----

INT 6F - 10NET v5.0 - GET LOGIN LIST

AH = 9Ch

BX = zero-based index

Return: CF clear if successful

DS:DI -> BX'th node ID that caller's machine is logged into

CF set on error

AX = status (see #03740)

-----H-70-----

INT 70 C - IRQ8 - CMOS REAL-TIME CLOCK

Desc: this interrupt is called when the real-time clock chip generates an alarm or periodic interrupt, among others (see CMOS 0Bh). The periodic interrupt by default occurs 1024 times per second.

Notes: many BIOSes turn off the periodic interrupt in the INT 70h handler unless in an event wait (see INT 15/AH=83h,INT 15/AH=86h).

may be masked by setting bit 0 on I/O port A1h

SeeAlso: INT 08,INT 0F"HP 95LX",INT 15/AH=01h"Amstrad",INT 15/AH=83h

SeeAlso: INT 15/AH=86h,INT 1A/AH=02h,INT 58"DESQview",MEM FEE00320h

SeeAlso: MEM 0040h:006Bh

-----v-70-----

INT 70 - VIRUS - "Stupid" - ORIGINAL INT 21h VECTOR

Note: the virus sets this interrupt to be the same as INT 21, and then performs only INT 70 calls; since INT 70 is also generated by the real-time clock on the PC/AT and higher, this could cause random actions on an infected system

SeeAlso: INT 6B"VIRUS",INT 9E"VIRUS",INT E0"VIRUS"

-----H-71-----

INT 71 C - IRQ9 - REDIRECTED TO INT 0A BY BIOS

Notes: may be masked by setting bit 1 on I/O port A1h  
the default BIOS handler invokes INT 0A for compatibility, since the pin for IRQ2 on the PC expansion bus became the pin for IRQ9 on the AT expansion bus.  
under DESQview, only the INT 15h vector and BASIC segment address (the word at 0000h:0510h) may be assumed to be valid for the handler's process

SeeAlso: INT 0A"IRQ2",INT 59"IRQ9",MEM 0040h:006Bh

-----H-72-----

INT 72 C - IRQ10 - RESERVED

Notes: may be masked by setting bit 2 on I/O port A1h  
DOS 3.3+ revector IRQ10 to a stack-switching routine unless STACKS=0 has been set in CONFIG.SYS. MS/PC-DOS use the IBM Interrupt Sharing Protocol (see #02568) when hooking this IRQ

SeeAlso: INT 5A"IRQ10",MEM 0040h:006Bh

-----H-73-----

INT 73 C - IRQ11 - RESERVED

Notes: may be masked by setting bit 3 on I/O port A1h  
DOS 3.3+ revector IRQ11 to a stack-switching routine unless STACKS=0 has been set in CONFIG.SYS. MS/PC-DOS use the IBM Interrupt Sharing Protocol (see #02568) when hooking this IRQ

SeeAlso: INT 5B"IRQ11",MEM 0040h:006Bh

-----H-74-----

INT 74 C - IRQ12 - POINTING DEVICE (PS)

Notes: may be masked by setting bit 4 on I/O port A1h  
DOS 3.3+ revector IRQ12 to a stack-switching routine unless STACKS=0 has been set in CONFIG.SYS. MS/PC-DOS use the IBM Interrupt Sharing Protocol (see #02568) when hooking this IRQ  
under DESQview, only the INT 15h vector and BASIC segment address (the word at 0000h:0510h) may be assumed to be valid for the handler's process

SeeAlso: INT 33/AX=0000h,INT 33/AX=0024h,INT 5C"NetBIOS",MEM 0040h:006Bh

-----H-75-----

INT 75 C - IRQ13 - MATH COPROCESSOR EXCEPTION (AT and up)

Desc: redirected to INT 02 by the BIOS, for compatibility with the PC

Notes: may be masked by setting bit 5 on I/O port Ah

not all clones wire the coprocessor to generate this IRQ; some systems generate an NMI (see INT 02) or assert the -ERROR pin on the CPU (see INT 10"COPROCESSOR")

under DESQview, only the INT 15h vector and BASIC segment address (the word at 0000h:0510h) may be assumed to be valid for the handler's process

SeeAlso: INT 10"COPROCESSOR",INT 5D"IRQ13",MEM 0040h:006Bh

-----H-76-----

INT 76 C - IRQ14 - HARD DISK CONTROLLER OPERATION COMPLETE (AT and later)

Notes: may be masked by setting bit 6 on I/O port Ah

DOS 3.3+ revector IRQ14 to a stack-switching routine unless STACKS=0 has been set in CONFIG.SYS. MS/PC-DOS use the IBM Interrupt Sharing Protocol (see #02568) when hooking this IRQ on the PS/2, this interrupt is designed to be shared with other devices that produce an operation-complete interrupt, though the only current user is the hard disk

SeeAlso: INT 0E"IRQ6",INT 15/AH=91h,INT 5E"IRQ14",MEM 0040h:006Bh

-----H-77-----

INT 77 C - IRQ15 - RESERVED (AT,PS)

Notes: may be masked by setting bit 7 on I/O port Ah

DOS 3.3+ revector IRQ15 to a stack-switching routine unless STACKS=0 has been set in CONFIG.SYS. MS/PC-DOS use the IBM Interrupt Sharing Protocol (see #02568) when hooking this IRQ

SeeAlso: INT 5F"IRQ15",MEM 0040h:006Bh

-----H-77-----

INT 77 C - IRQ15 - SECONDARY IDE CONTROLLER - OPERATION COMPLETE

Notes: may be masked by setting bit 7 on I/O port Ah

DOS 3.3+ revector IRQ15 to a stack-switching routine unless STACKS=0 has been set in CONFIG.SYS. MS/PC-DOS use the IBM Interrupt Sharing Protocol (see #02568) when hooking this IRQ

SeeAlso: INT 5F"IRQ15",MEM 0040h:006Bh

-----H-77-----

INT 77 C - IRQ15 - POWER CONSERVATION (Compaq SLT/286)

Note: may be masked by setting bit 7 on I/O port Ah

SeeAlso: INT 15/AX=4600h,INT 5F"IRQ15"

-----E-78-----

INT 78 - UofSalford DBOS DOS extender - API

AH = function

00h (PB) display 32-bit memory specified by command argument  
01h (PT) display 32-bit instruction(s) specified by command arg  
02h specify offset to subsequent PB and PT commands  
03h switch to protected mode  
DWORD following INT instruction point to map; protected-mode  
entry point is immediately following the DWORD  
Return: never  
Note: if bit 31 of the map address is set, only a stub was  
loaded by DOS and DBOS will load the entire program  
04h specify that subsequent load (AH=03h) should leave program in  
memory  
05h release program from memory  
06h set break point at address and option count from commandline  
07h resume program execution  
08h single-step program  
09h set read/write breakpoint using 386 debug registers  
0Ah set write breakpoint using 386 debug register  
0Bh set memory byte to new value  
0Ch display CPU registers  
0Dh run until specified program address reached  
0Fh print trace from program map  
10h specify an offset using a map symbol  
14h print memory without any offset  
18h switch DBOS into/out of test mode (ON/OFF commandline args)  
1Dh get address of real/protected-mode communication buffer  
Return: ES:BP -> comm buffer  
1Eh set real-mode memory size (specify how much real-mode memory  
to leave free when running FTN77 programs)  
22h uninstall DBOS  
24h force DBOS to emulate coprocessor instructions  
26h set named DBOS switches from commandline  
27h reset named DBOS switched from commandline  
2Ah set list of dynamic link libraries to contents of commandline  
file  
35h specify that subsequent load (AH=03h) should stop at first  
instruction

Return: ???

Notes: DBOS supports functions 00h through 50h; many of these functions  
provide a low-level debugging interface  
command arguments are read from the calling program's PSP

SeeAlso: INT 79"DBOS"

Index: uninstall;DBOS DOS extender

-----E-78-----

INT 78 - DJGPP GO32.EXE DOS EXTENDER - RELOCATED IRQ0

Program: GO32.EXE is a DOS extender included as part of the 80386 port of the GNU C/C++ compiler by DJ Delorie and distributed as DJGPP

Notes: this vector is overwritten when GO32 starts but is not restored by early versions of the extender

the newest versions of GO32 dynamically allocate the vectors used for the relocated IRQs, much as DESQview does (see INT 50"DESQview")

SeeAlso: INT 08,INT 10/AH=FFh"GO32",INT 79"GO32"

-----A-78-----

INT 78 R - AutoCAD Device Interface - PLOTTER - SEND COMMAND

AX = function

0001h begin plot

BX = file level

0000h ASCII file

0001h binary file

0002h AutoCAD DXB file

0003h installed ADI driver

0002h end plot (close plotter)

0003h move (pen up)

BX = new X position

CX = new Y position

0004h move (pen down)

BX = new X position

CX = new Y position

0005h select pen

BX = pen number

0006h select drawing speed

BX = drawing speed N

0007h select line style

BX = line style N

0008h raise pen

0009h abort plot

Return: AX = 0000h (may be used for driver status in future releases)

SeeAlso: INT 79"AutoCAD",INT 7A/AX=0001h"AutoCAD",INT 7B/AX=8001h"AutoCAD"

-----d-78-----

INT 78 - ADP-60 IDE controller - ORIGINAL INT 13

SeeAlso: INT 79"ADP-60"

-----s-78-----

INT 78 - ULTRAMID - API

```
AX = function
--- digital functions ---
    0000h start digital
    ES:DI -> Sound Structure
    Return: AX = voice used
    0001h play next buffer
    ES:DI -> Sound Structure
    0002h set panning
    BX = pan
    CX = voice
    0003h set volume
    BX = volume
    CX = voice
    0004h set rate (not yet implemented)
    0005h pause digital
    CX = voice
    0006h restart digital
    CX = voice
    0007h stop digital
    CX = voice
    0008h get digital position
    CX = voice
    Return: AX:DX -> position in PC buffer
    0009h get voice status (not yet implemented)
--- MIDI functions ---
    000Ah load MIDI patches
    ES:DI -> MIDI track just after the Mtrk header
    BX:DX = length of MIDI track
    000Bh load XMIDI patches
    ES:DI -> event buffer including the EVNT header
    000Ch load patch
    CX = MIDI number of patch
    000Dh unload patch
    CX = MIDI number of patch
    000Eh start sequence
    000Fh unload all patches
    0010h MIDI out byte
    CX = MIDI byte
    0011h MIDI out string
    CX = length
    ES:DI -> string of MIDI data
```

```
0012h all notes off
--- resource functions ---
0013h allocate memory
BX:DX = size of GUS memory needed
Return: AX:DX = address of memory, or 0000:0000 if not available
0014h free memory
BX:DX = address of memory to be freed
0015h uninstall
0016h sem enter
0017h sem leave
0018h add external semaphore
BX:DX -> external semaphore
0019h clear external semaphore
BX:DX -> external semaphore
001Ah application start
001Bh application end
--- v1.12 ---
001Ch ???
001Dh ???
001Eh ???
001Fh ???
```

Program: UltraMid is a TSR from Advanced Gravis to play MIDI and digital data through the Gravis UltraSound (GUS)

InstallCheck: test for the signature "ULTRAMID" at offset 0103h in the interrupt handler's segment

Range: INT 78 to INT 7F

SeeAlso: INT 2F/AX=CD00h"ULTRAMID",INT 7E/AH=00h"SBOS",INT 7E/AX=00FEh"SBOS"

Index: uninstall;UltraMID

-----d-7800-----

INT 78 - TARGA.DEV - CMC International SCSI device driver - SET I/O PORT

AH = 00h

DX = interface board I/O port

Return: CF set on error

AL = error code (see #03786)

InstallCheck: (used by TARGA.DEV) test for the string "SCSI" at offset 03h into the interrupt handler

Note: if this routine is not called, the port is the driver's default (usually 0280h or 0300h)

SeeAlso: AH=01h,AH=02h

Index: installation check;TARGA.DEV

(Table 03786)

Values for TARGA.DEV error code:

- 00h illegal command given to SCSI code
- 01h invalid I/O port specified (must be from 100H to 3F8H, and must be on an 8-port boundary)
- 02h invalid DMA channel specified (must be from 1 to 3)
- 03h invalid SCSI board number specified (must be from 0 to 7)
- 04h error from data register test during self-test
- 05h SCSI input signals not all 0 when SCSI RST activated
- 06h SCSI input signals not all 0 before selecting a SCSI device
- 07h BSY signal is active; SCSI bus is busy
- 08h SCSI board not selected, BSY signal did not come on in response to raising SEL
- 09h time-out waiting for status state, signifying end of DMA transfer

-----E-780000-----

INT 78 - HugeRealMode Driver - INSTALLATION CHECK

AX = 0000h

Return: AX = DBCAh if installed

Program: the HugeRealMode driver was published in the German DOS Extra Nr. 20

(1992), and makes a 4G flat address space (for both data and code)

available to real-mode DOS programs

SeeAlso: AX=0001h

-----E-780001-----

INT 78 - HugeRealMode Driver - GET ENTRY POINT

AX = 0001h

Return: ES:BX -> far call entry point (see #03788,#03803)

SeeAlso: AX=0000h

(Table 03787)

Values for HugeRealMode Server error code:

- 00h successful
- 01h unknown function
- 02h A20 disabled
- 03h A20 enabled
- 04h memory allocation error
- 05h memory not fixeable (LOCK)
- 06h memory not deallocaable
- 07h memory not moveable (UNLOCK)
- 08h change in memory allocation size not possible
- 09h file error (with .XMF-file)
- 0Ah memory error (with .XMF-file)



0Bh unknown procedure (currently Turbo Pascal unit only)

(Table 03788)

Call HugeRealMode Server "Enable A20" function with:

AL = 00h

Return: AX = error code (see #03787)

Note: The HugeRealMode Server requires an XMS-driver like HIMEM.SYS loaded before HUGERREAL (for XMS-memory allocation and A20 control). It utilizes some undocumented 386+ features to enable 4G code and data segments (flat memory model) in the CPUs real-mode. It is not compatible with V86-mode (as with most EMM386). For maximum speed and compatibility with DOS it uses a two stage interrupt model (see #03791)

SeeAlso: #03789,#03790,#03791

(Table 03789)

Call HugeRealMode Server "Disable A20" function with:

AL = 01h

Return: AX = error code (see #03787)

SeeAlso: #03788,#03790

(Table 03790)

Call HugeRealMode Server "Query A20 State" function with:

AL = 02h

Return: AX = A20 state (0000h disabled, 0001h enabled)

SeeAlso: #03788,#03789

(Table 03791)

Call HugeRealMode Server "Enable Two-Stage Interrupt Model" function with:

AL = 03h

Return: AX = error code (see #03787)

Notes: If the two stage interrupt model is disabled, the IP must stay below 1MB, because the high word of EIP is not saved correctly by the standard interrupt management in real mode. Routines in extended memory may only be called with interrupts disabled (only usable for very short routines).

If the two stage interupt model is enabled, these conventions are not required. The outer (transparent) ISRs save the higher word of EIP in the high word of CR3 (possible modification: in memory, if this is not safe) and then invoke the standard ISRs.

Because of this handling, one should use pseudo segmented code even in

extended memory (64K blocks located on 64K boundaries) to avoid the misalignment of the saved EIP (in CR3). Another way is to track and realign CR3 each time when jumping over a 64K boundary in extended memory

SeeAlso: #03788,#03792

(Table 03792)

Call HugeRealMode Server "Disable Two-Stage Interrupt Model" function with:

AL = 04h

Return: AX = error code (see #03787)

SeeAlso: #03788,#03791

(Table 03793)

Call HugeRealMode Server "Get Segment Address Of Server-Management Range" with:

AL = 05h

Return: AX = error code (see #03787)

ES = segment address

SeeAlso: #03803

(Table 03794)

Call HugeRealMode Server "Allocate Memory" function with:

AL = 06h

DX = length in KB

Return: AX = error code (see #03787)

DX = handle

EBX = start address of allocated block

SeeAlso: #03795,#03796

(Table 03795)

Call HugeRealMode Server "Deallocate Memory" function with:

AL = 07h

DX = handle

Return: AX = error code (see #03787)

SeeAlso: #03794,#03796

(Table 03796)

Call HugeRealMode Server "Modify Allocated Memory Size" function with:

AL = 08h

DX = handle

BX = new size in KB

Return: AX = error code (see #03787)

EBX = new address

SeeAlso: #03794,#03795

(Table 03797)

Call HugeRealMode Server "Load .XMF-file" function with:

AL = 09h

ES:BX -> filename (see #03798)

Return: AX = error code (see #03787)

ECX = target address

Notes: memory must be allocated before loading

.XMF file format (eXtended Memory executable File format) is supported  
by the HugeRealMode server for easier handling of XMS-code.

The utility CONFXXMF converts .OBJ-files (.COM 'tiny' model) to  
.XMF-format

SeeAlso: #03794

Format of .XMF program file header:

Offset Size Description (Table 03798)

00h DWORD signature "XMF\_"

04h WORD length of programs code

06h WORD count of 32bit addresses to relocate

08h WORD offset to code start (relative to file start) (see #03799)

0Ah WORD relocation table

Note: each word contains a 16bit address, where a relocation has to be  
performed

SeeAlso: #03797,#03799

Format of .XMF-file code header:

Offset Size Description (Table 03799)

00h WORD count of exported functions

02h N PWORDS 48bit pointers (16bit segment+32bit offset) to each function  
(16bit segment is always zero)

var start of program's code

SeeAlso: #03797,#03798

(Table 03800)

Call HugeRealMode Server "Set Segment Limits" function with:

AL = 0Ah

BL = segment limits (see #03801)

Return: AX = error code (see #03787)

SeeAlso: #03793

Bitfields for HugeRealMode Server segment limits:

Bit(s) Description (Table 03801)

7 CS size: 1=4GB, 0=64KB

6-5 not used

4 SS size: 1=4GB, 0=64KB

3 GS size

2 FS size

1 ES size

0 DS size

SeeAlso: #03800

(Table 03802)

Call HugeRealMode Server "Transfer Memory From DOS To XMS" function with:

AL = 0Bh

ES:SI = source address

EDI = target address

ECX = length

Return: AX = error code

SeeAlso: #03803

(Table 03803)

Call HugeRealMode Server "Transfer Memory From XMS To DOS" function with:

AL = 0Ch

ESI = source address

ES:DI = target address

ECX = length

Return: AX = error code

SeeAlso: #03802

-----d-7801-----

INT 78 - TARGA.DEV - GET I/O PORT

AH = 01h

Return: DX = current interface board I/O port

SeeAlso: AH=00h,AH=03h"TARGA"

-----d-7802-----

INT 78 - TARGA.DEV - SET DMA CHANNEL

AH = 02h

AL = interface board DMA channel

Return: CF set on error

AL = error code (see #03786)

Note: if this routine is not called, the DMA channel is the driver's default

(usually 3)

SeeAlso: AH=00h,AH=03h"TARGA"

-----d-7803-----

INT 78 - TARGA.DEV - GET DMA CHANNEL

AH = 03h

Return: AL = current interface board DMA channel

SeeAlso: AH=01h,AH=02h

-----d-7804-----

INT 78 - TARGA.DEV - SET SCSI DEVICE NUMBER

AH = 04h

AL = SCSI device number

Return: CF set on error

AL = error code (see #03786)

Note: if this routine is not called, the device number used is the driver's

default (usually 0)

SeeAlso: AH=02h,AH=05h

-----d-7805-----

INT 78 - TARGA.DEV - GET SCSI DEVICE NUMBER

AH = 05h

Return: AL = current SCSI device number

SeeAlso: AH=03h"TARGA",AH=04h

-----d-7806-----

INT 78 - TARGA.DEV - SET/CLEAR EARLY RETURN MODE

AH = 06h

AL = new state

00h clear early return mode

01h set early return mode

Note: if early return mode is set then SCSI will return with no errors

when the last DMA transfer is started in a call with AH=13h or AH=14h

if this routine is not called, early return mode is cleared

SeeAlso: AH=13h,AH=14h,AH=15h

-----d-7808-----

INT 78 - TARGA.DEV - INTERFACE BOARD SELF-TEST

AH = 08h

Return: CF set on error

AL = error code (see #03786)

Note: the SCSI bus is also reset

SeeAlso: AH=09h

-----d-7809-----

INT 78 - TARGA.DEV - RESET SCSI BUS

AH = 09h

Return: AL = error code if carry set (see #03786)

SeeAlso: AH=08h

-----d-7810-----

INT 78 - TARGA.DEV - SEND SCSI COMMAND

AH = 10h

DS:SI -> command bytes (see #03804)

Return: AH = SCSI status byte

CF clear if successful

AL = SCSI message byte

CF set on error

AL = error code (see #03786)

SeeAlso: AH=11h,INT 21/AX=4403h"ST-01"

Format of SCSI Command:

Offset Size Description (Table 03804)

00h BYTE length of command

01h ??? command bytes

-----d-7811-----

INT 78 - TARGA.DEV - SEND SCSI COMMAND, RECEIVE DATA (PROGRAMMED I/O)

AH = 11h

DS:SI -> command bytes (see #03805)

ES:BX -> data storage area

CX = number of data bytes to transfer

Return: AH = SCSI status byte

CF clear if successful

AL = SCSI message byte

CF set on error

AL = error code (see #03786)

Note: this command receives data internally one byte at a time

SeeAlso: AH=10h,AH=13h

Format of TARGA.DEV SCSI Command:

Offset Size Description (Table 03805)

00h BYTE length of command

01h ??? command bytes

-----d-7812-----

INT 78 - TARGA.DEV - SEND SCSI COMMAND AND DATA (PROGRAMMED I/O)

AH = 12h

DS:SI -> command bytes (see #03806)

ES:BX -> data storage area

CX = number of data bytes to transfer

Return: AH = SCSI status byte

CF clear if successful

AL = SCSI message byte

CF set on error

AL = error code (see #03786)

Note: this command sends data internally one byte at a time

SeeAlso: AH=14h

Format of TARGA.DEV SCSI Command:

Offset Size Description (Table 03806)

00h BYTE length of command

01h ??? command bytes

-----d-7813-----

INT 78 - TARGA.DEV - SEND SCSI COMMAND, RECEIVE DATA (DMA)

AH = 13h

DS:SI -> command bytes (see #03807)

ES:BX -> data storage area

DX:CX = number of data bytes to transfer

Return: AH = SCSI status byte (if early return mode is clear)

CF clear if successful

AL = SCSI message byte (if early return mode is clear)

CF set on error

AL = error code (see #03786)

Note: this command receives data using DMA

SeeAlso: AH=11h,AH=12h

Format of TARGA.DEV SCSI Command:

Offset Size Description (Table 03807)

00h BYTE length of command

01h ??? command bytes

-----d-7814-----

INT 78 - TARGA.DEV - SEND SCSI COMMAND AND DATA (DMA)

AH = 14h

DS:SI -> command bytes (see #03808)

ES:BX -> data storage area

DX:CX = number of data bytes to transfer

Return: AH = SCSI status byte (if early return mode is clear)

CF clear if successful

AL = SCSI message byte (if early return mode is clear)

CF set on error

AL = error code (see #03786)

Note: this command sends data using DMA

SeeAlso: AH=12h,AH=13h

Format of TARGA.DEV SCSI Command:

Offset Size Description (Table 03808)

00h BYTE length of command

01h ??? command bytes

-----d-7815-----

INT 78 - TARGA.DEV - FINISH DATA TRANSFER (DMA)

AH = 15h

Return: AH = SCSI status byte

CF clear if successful

AL = SCSI message byte

CF set on error

AL = error code (see #03786)

Note: if AH=06h was previously called to set the early return mode, this function finishes a command AH=13h or AH=14h which returned before the last DMA transfer was finished

SeeAlso: AH=06h,AH=13h,AH=14h

-----v-79-----

INT 79 - AVATAR.SYS - FAST GET KEYSTROKE

Return: CF set if no keystroke available

AX = FFFFh

CF clear if key pressed

AX = keystroke

Program: AVATAR.SYS is a CON driver by George Adam Stanislav which interprets AVATAR command codes just as ANSI.SYS interprets ANSI commands

Note: if a keystroke is available, it is removed from the keyboard buffer before being returned

SeeAlso: INT 29,INT 2F/AX=1A00h/BX=4156h

-----E-79-----

INT 79 - DJGPP GO32.EXE DOS EXTENDER - RELOCATED IRQ1

Notes: this vector is overwritten when GO32 starts but is not restored by early versions of the extender

the newest versions of GO32 dynamically allocate the vectors used

for the relocated IRQs, much as DESQview does (see INT 50"DESQview")

SeeAlso: INT 09,INT 78"GO32",INT 7A"GO32"

-----d-79-----

INT 79 U - ADP-60 IDE adapter - ???

SeeAlso: INT 78"ADP-60"

-----E-79-----



INT 79 - DBOS DOS Extender

details not available

Desc: this interrupt is used by an MS Windows virtual device driver with  
which the DBOS extender communicates when run inside a Windows  
Enhanced Mode DOS box

SeeAlso: INT 78"DBOS"

-----A-790001-----

INT 79 R - AutoCAD Device Interface - DIGITIZER - INITIALIZE

AX = 0001h

BX = interface level (0001h)

Return: AX = status (0000h initialization failed, 0001h init successful)

BX = digitizer type

0000h relative pointing device (mouse, etc.)

0001h digitizing tablet

CX = digitizer dataflow type

0000h continuous

0001h pauses between packets

SeeAlso: AX=0002h,AX=0003h,INT 7A/AX=0001h"AutoCAD"

-----A-790002-----

INT 79 R - AutoCAD Device Interface - DIGITIZER - TERMINATE

AX = 0002h

SeeAlso: AX=0001h,AX=0003h

-----A-790003-----

INT 79 R - AutoCAD Device Interface - DIGITIZER - GET DIGITIZER STATUS

AX = 0003h

Return: AX = status

0000h nothing

0002h tracking point (no button pressed)

BX = X coordinate

CX = Y coordinate

0003h picked point (button 0,A,B,C,D pressed)

BX = X coordinate

CX = Y coordinate

0004h button pick

BX = button number

0005h button and coordinates

BX = button number

CX = X coordinate

DX = Y coordinate

SeeAlso: AX=0001h,AX=0002h

-----N-7A-----

INT 7A U - Topware Network Operating System - ???

AL = ???

???

Return: ???

SeeAlso: INT 21/AX=FF00h"Topware",INT 2F/AX=FF00h

-----N-7A-----

INT 7A - X.PC Packet software interface

ES:BX -> parameter block

SeeAlso: INT 60/AX=01FFh

-----E-7A-----

INT 7A - DJGPP GO32.EXE DOS EXTENDER - RELOCATED IRQ2

Program: GO32.EXE is a DOS extender included as part of the 80386 port of the

GNU C/C++ compiler by DJ Delorie and distributed as DJGPP

Notes: this vector is overwritten when GO32 starts but is not restored by

early versions of the extender

the newest versions of GO32 dynamically allocate the vectors used

for the relocated IRQs, much as DESQview does (see INT 50"DESQview")

SeeAlso: INT 0A"IRQ2",INT 79"GO32",INT 7B"GO32"

-----P-7A-----

INT 7A O - PRINDIR - API (moved to INT 7C, then INT 2F/AX=7F00h)

SeeAlso: INT 7C"PRINDIR",INT 2F/AX=7F00h"PRINDIR"

-----N-7A-----

INT 7A - Novell NetWare - LOW-LEVEL API - Notes

Note: this interrupt is used for IPX/SPX access in NetWare versions through

2.0a; in later versions, you should use INT 2F/AX=7A00h to get an

entry point even though INT 7A still exists. For both INT 7A and

the FAR entry point, BX contains the function number; IPX is

sometimes called internally with BX bit 15 set, which causes the

handler to bypass some initial checks and an optional call to the

IPX Windows support handler set with INT 2F/AX=7AFFh/BX=0000h

(see #02926)

SeeAlso: INT 2F/AX=7A00h,INT 64"Novell",INT 7A/BX=0000h

-----N-7A----BX0000-----

INT 7A - Novell NetWare - IPX Driver - OPEN SOCKET

BX = 0000h

AL = socket longevity

00h open until close or terminate

FFh open until close

DX = socket number (high byte in DL)

0000h dynamic allocation

else socket to open (see #03809)

Return: AL = return code

00h success

DX = socket number

FEh socket table full

FFh socket already open

Notes: TSRs which need to use sockets should set AL to FFh, non-resident programs should normally use AL=00h

IPX can be configured to support up to 150 open sockets on a workstation, and defaults to 20

this function is supported by Advanced NetWare 1.02+

SeeAlso: INT 7A/BX=0001h, INT 7A/BX=0004h, INT 7A/BX=0023h

(Table 03809)

Values for IPX socket number:

0451h File Service (NetWare Core Protocol)

0452h Service Advertising Protocol (SAP) (see #03817)

0453h Routing Information Packet (see #03820)

0455h NetBIOS Packet

0456h diagnostics

0457h server serial numbers (labeled "Copy Protection" by Lanalyzer)

2222h NetWare Core Protocol

4000h-7FFFh used for dynamic allocation

4000h \ seem to be used by IPX internally

4001h /

4444h Brightwork Development's SiteLock server

5555h Brightwork Development's SiteLock client (workstation)

8000h-FFFFh assigned by Novell

869Ch ID Software's DOOM

901Fh Lancheck

9023h Server info (NetWare "Slurpee", NetWare Lite, Personal NetWare)

9024h Client info (NetWare "Slurpee", NetWare Lite, Personal NetWare)

9025h Serial socket (NetWare "Slurpee", NetWare Lite, Personal NetWare)

9026h Extension info (NetWare Lite, Personal NetWare???)

Notes: SiteLock is an application metering product using IPX to communicate between the application and the license server

IPX socket numbers should be registered with the Novell Development Products Division

SeeAlso: INT 7A/BX=0001h

-----N-7A----BX0001-----

INT 7A - Novell NetWare - IPX Driver - CLOSE SOCKET

BX = 0001h

DX = socket number (high byte in DL) (see #03809)

Notes: also cancels events set by any Event Control Blocks for the socket  
the program must close all open sockets before terminating  
this function is supported by Advanced NetWare 1.02+

When the NetWare-specific release of the DR DOS command processor does  
a NetWare shutdown sequence on exit, it will close all user sockets  
except for socket numbers 4000 and 4001, which seem to be used by IPX  
internally.

SeeAlso: BX=0000h

-----N-7A----BX0002-----

INT 7A - Novell NetWare - IPX Driver - GET LOCAL TARGET

BX = 0002h

ES:SI -> target internetwork address (see #03824 at INT 7A/BX=000Bh)

ES:DI -> 6-byte buffer for local target

Return: AL = return code

00h success

CX = expected one-way transfer time (clock ticks) for a 576-byte  
packet

ES:DI -> local target

FAh unsuccessful (no path to destination)

Notes: the internetwork address consists of a 4-byte network address followed  
by a 6-byte node address. The local target is only a 6-byte node  
address. If the target is in the same network, the local target is  
just the node address of target; otherwise, the local target is the  
node address of the bridge that leads to the target.

this function may be called from inside IPX and AES Event Service

Routines, but not from other interrupt handlers

this function is supported by Advanced NetWare 1.02+

SeeAlso: BX=0009h

-----N-7A----BX0003-----

INT 7A - Novell NetWare - IPX Driver - SEND PACKET

BX = 0003h

ES:SI -> Event Control Block (see #03810,#03811)

Notes: returns immediately; IPX attempts to send the packet in the background

this function is supported by Advanced NetWare 1.02+

this function is nearly identical to BX=000Fh, except that it always  
copies the source address into the IPX header assumed to be at the  
beginning of the first fragment

SeeAlso: BX=0004h,BX=000Fh,INT 21/AH=EEh"Novell"

Format of IPX Event Control Block:

Offset	Size	Description (Table 03810)
00h	DWORD	Link
04h	DWORD	-> Event Service Routine (00000000h if none) (see #03814)
08h	BYTE	in use flag (see #03812)
09h	BYTE	completion code (see #03813)
0Ah	WORD	(big-endian) socket number (see #03809)
0Ch	4 BYTES	IPX workspace
10h	12 BYTES	driver workspace
1Ch	6 BYTES	immediate local node address
22h	WORD	fragment count
24h	var	fragment descriptors
		Offset Size Description
		00h DWORD -> fragment data
		04h WORD size of fragment in bytes.

Notes: ESR is a far procedure that is called when the ECB has been handled.

On call, the in use flag is zero if the ECB has been handled, non-zero otherwise. If the flag is zero, the completion code holds the result of the event.

the first fragment should start with an IPX header (see #03815)

all fragments are concatenated and sent in one piece

node address FFh FFh FFh FFh FFh FFh broadcasts to all nodes

SeeAlso: #03811

Format of AES-ECB:

Offset	Size	Description (Table 03811)
00h	DWORD	Link
04h	DWORD	ESR (Event Service Routine) address (see #03814)
08h	BYTE	in use flag (see #03812)
09h	5 BYTES	AES workspace

SeeAlso: #03810

(Table 03812)

Values for ECB in use flag:

00h	available
E0h	AES temporary
F6h	\ special IPX/SPX processing for v3.02+
F7h	/
F8h	IPX in critical section
F9h	SPX listening
FAh	processing
FBh	holding

FCh AES waiting  
 FDh AES counting down delay time  
 FEh awaiting packet reception  
 FFh sending packet

SeeAlso: #03810,#03811,#03813

(Table 03813)

Values for ECB completion code:

00h success  
 ECh remote terminated connection without acknowledging packet  
 EDh abnormal connection termination  
 EEh invalid connection ID  
 EFh SPX connection table full  
 F9h event should not be cancelled  
 FAh cannot establish connection with specified destination  
 FCh cancelled  
 FDh malformed packet  
 FEh packet undeliverable  
 FFh physical error

SeeAlso: #03810,#03812

(Table 03814)

Values Event Service Routine is called with:

AL = caller's identity (00h = AES, FFh = IPX)  
 ES:SI -> event control block  
 interrupts disabled

Return: all registers preserved

SeeAlso: #03810,#03815

Format of IPX header:

Offset Size Description (Table 03815)

00h WORD (big-endian) checksum  
 02h WORD (big-endian) length in bytes of total packet  
 04h BYTE transport control  
 05h BYTE packet type (see #03816)  
 06h 10 BYTES destination internetwork address  
 10h WORD (big-endian) destination socket  
 12h 10 BYTES source internetwork address  
 1Ch WORD (big-endian) source socket

Notes: an IPX packet has the identical format as a Xerox Network Standard  
 (XNS) packet

the application must set packet type, destination address, and destination socket; IPX/SPX set the remaining fields

SeeAlso: #03810,#03826

(Table 03816)

Values for IPX packet type:

00h unknown packet type  
01h routing information packet  
02h echo packet  
03h error packet  
04h packet exchange packet (always use this one for data)  
05h SPX packet (see #03826)  
11h NetWare Core Protocol (also used by Service Advertising Protocol)  
14h Propagated Packet (for NetWare), NetBIOS name packet  
15h-1Eh experimental protocols

Note: undocumented packet type 14h will cross up to 16 networks deep in all directions; as Aaron Martin of Origin Systems discovered, the first 64 bytes of the IPX data in such packets should be considered reserved, as IPX places the traversed server nodes there.

Format of Service Advertising Protocol Service Query Packet:

Offset Size Description (Table 03817)

00h 30 BYTES IPX header  
1Eh WORD (big-endian) query type  
    0001h general find service  
    0003h find nearest server  
20h WORD (big-endian) server type  
    (see #01890 at INT 21/AH=E3h"CONNECTION CONTROL")

SeeAlso: #03818

Format of Service Advertising Protocol Server Identification Packet:

Offset Size Description (Table 03818)

00h 30 BYTES IPX header  
1Eh WORD (big-endian) response type  
    0002h general service  
    0004h nearest service  
20h 64N BYTES server entries (1-7) (see #03819)

SeeAlso: #03817,#03820

Format of SAP server entry:

Offset Size Description (Table 03819)

00h WORD (big-endian) server type  
 (see #01890 at INT 21/AH=E3h"CONNECTION CONTROL")  
 02h 48 BYTES ASCII server name  
 32h 2 WORDs (big-endian) network number  
 34h 3 WORDs (big-endian) node number  
 3Ch WORD (big-endian) socket number  
 3Eh WORD (big-endian) number of hops between caller and server  
 10h = Server Shutdown Advertising Packet

SeeAlso: #03818

Format of IPX Routing Information packet:

Offset	Size	Description (Table 03820)
00h	30 BYTES	IPX header
1Eh	WORD	operation (0001h request, 0002h response)
20h	8N BYTES	network entries (1-50) (see #03821)

SeeAlso: #03818

Format of RIP network entry:

Offset	Size	Description (Table 03821)
00h	DWORD	network number (FFFFFFFFh = general request)
04h	WORD	(response) number of hops
06h	WORD	(response) number of clock ticks to reach destination

-----N-7A----BX0004-----

INT 7A - Novell NetWare - IPX Driver - LISTEN FOR PACKET

BX = 0004h

ES:SI -> Event Control Block (see BX=0003h)

Return: AL = status

00h successful

FFh no listening socket for packet

Desc: this function provides IPX with an ECB for receiving an IPX packet, but  
 does not wait for a packet to arrive

Notes: the application must open a socket and initialize the ECB's ESR  
 address, socket number, fragment count, and fragment descriptor  
 fields before invoking this function

there is no limit on the number of ECBs which may simultaneously be  
 listening on a socket

this function is supported by Advanced NetWare 1.02+

SeeAlso: BX=0000h,BX=0003h

-----N-7A----BX0005-----

INT 7A - Novell NetWare - IPX Driver - SCHEDULE IPX EVENT

BX = 0005h



AX = delay time in clock ticks

ES:SI -> Event Control Block (see BX=0003h)

Note: this function is supported by Advanced NetWare 1.02+

SeeAlso: BX=0006h, BX=0007h, BX=0008h

-----N-7A----BX0006-----

INT 7A - Novell NetWare - IPX Driver - CANCEL EVENT

BX = 0006h

ES:SI -> Event Control Block (see BX=0003h)

Return: AL = return code (see #03822)

Notes: cannot cancel packets which the node's driver has already sent

this function is supported by Advanced NetWare 1.02+

SeeAlso: BX=0005h

(Table 03822)

Values for IPX return code:

00h success

F9h event in use

FCh event cancelled

FFh unsuccessful, event not in use, or unrecognized ECB flag

-----N-7A----BX0007-----

INT 7A - Novell NetWare - IPX Driver - SCHEDULE SPECIAL EVENT

BX = 0007h

AX = delay time

ES:SI -> Event Control Block (see BX=0003h)

Note: this function is supported by Advanced NetWare 1.02+

SeeAlso: BX=0006h

-----N-7A----BX0008-----

INT 7A - Novell NetWare - IPX Driver - GET INTERVAL MARKER

BX = 0008h

Return: AX = interval marker in clock ticks (big-endian???)

Notes: may be used to measure the time elapsed between two events, up to one  
hour

this function is supported by Advanced NetWare 1.02+

SeeAlso: BX=0005h

-----N-7A----BX0009-----

INT 7A - Novell NetWare - IPX Driver - GET INTERNETWORK ADDRESS

BX = 0009h

ES:SI -> buffer for own internetwork address (see #03823)

Return: ES:SI buffer filled

SI destroyed

Note: this function is supported by Advanced NetWare 1.02+

SeeAlso: BX=0002h,BX=000Bh

Format of IPX internetwork address:

Offset	Size	Description (Table 03823)
00h	4 BYTES (big-endian)	network number
04h	6 BYTES (big-endian)	node number within network

-----N-7A----BX000A-----

INT 7A - Novell NetWare - IPX Driver - RELINQUISH CONTROL

BX = 000Ah

Desc: this call indicates that the application is idle and permits the IPX driver to do some work

Note: this function is supported by Advanced NetWare 1.02+

SeeAlso: INT 15/AX=1000h,INT 21/AH=89h,INT 2F/AX=1680h

-----N-7A----BX000B-----

INT 7A - Novell NetWare - IPX Driver - DISCONNECT FROM TARGET

BX = 000Bh

ES:SI -> internetwork address (see #03824)

Return: nothing

Notes: this function permits the network software on the remote machine to

remove any virtual connection with the calling machine

only use in point-to-point networks

should never be called from within an Event Service Routine

this function is supported by Advanced NetWare 1.02+

SeeAlso: BX=0002h,BX=0009h

Format of IPX internetwork address:

Offset	Size	Description (Table 03824)
00h	4 BYTES (big-endian)	destination network
04h	6 BYTES (big-endian)	destination node
0Ah	2 BYTES (big-endian)	destination socket

-----N-7A----BX000C-----

INT 7A U - Novell NetWare - IPX Driver - internal - INITIALIZE NETWORK ADDRESS

BX = 000Ch

CX:DX = global network address (see INT 7A/BX=0002h)

ES:DI -> "OSINCRITICALSECTION" flag

DS:SI -> current mode for socket

Note: the address cannot be changed once it has been initialized

SeeAlso: INT 7A/BX=0024h

-----N-7A----BX000D-----

INT 7A U - Novell NetWare - IPX Driver - internal - IPX GET PACKET SIZE

BX = 000Dh

Return: AX = maximum packet size

CX = retry count

SeeAlso: BX=001Ah

-----N-7A----BX000E-----

INT 7A U - Novell NetWare - IPX Driver - internal - TERMINATE SOCKETS

BX = 000Eh

Return: nothing

Notes: this function terminates all sockets opened with the current mode; this may be intended for future enhancements as the socket mode never changes in v2.15

called by the NetWare shell if a program terminates

-----N-7A----BX000F-----

INT 7A - Novell NetWare - IPX Driver - INTERNAL - SEND PACKET

BX = 000Fh

ES:SI -> Event Control Block (see #03810 at BX=0003h)

IPX header's checksum field must be set to FFFFh (see #03815)

IPX header's total-length field must be set

Note: nearly identical to function 0003h, but does not copy address into the first fragment, and bypasses normal error checking

SeeAlso: BX=0003h

-----N-7A----BX0010-----

INT 7A - Novell NetWare - SPX Driver - INSTALLATION CHECK

BX = 0010h

AL = 00h

Return: AL = status

00h if SPX not installed

F0h if IPX not installed

FFh if SPX loaded

BH = SPX major version

BL = SPX minor version

CX = maximum SPX connections

DX = SPX connections available

Notes: this function is supported by Advanced NetWare 2.1+

this interrupt is used for IPX/SPX access in NetWare versions through 2.0a; in later versions, you should use INT 2F/AX=7A00h to get an entry point even though INT 7A still exists. For both INT 7A and the FAR entry point, BX contains the function number

IPX is sometimes called internally with BX bit 15 set, which causes the entry point handler to bypass some checks and an optional call to the IPX Windows support handler set with INT 2F/AX=7AFFh/BX=0000h (see #02926)

SeeAlso: BX=0015h

-----N-7A----BX0011-----

INT 7A - Novell NetWare - SPX Driver - ESTABLISH SPX CONNECTION

BX = 0011h

AL = retry count

AH = watchdog flag

ES:SI -> Event Control Block (see #03810 at BX=0003h)

Return: AL = status (00h,EFh,FDh,FFh) (see #03825)

DX = assigned connection ID number

Desc: attempt to establish a connection with a listening socket

Notes: there should always be at least two SPX ECB's listening to a socket, so

that NetWare can perform its internal packet exchanges

the first fragment should start with a SPX header (see #03826). Fill

in all destination addresses.

this function is supported by Advanced NetWare 2.1+

SeeAlso: BX=0000h,BX=0012h,BX=0013h,BX=0014h,BX=0015h

(Table 03825)

Values for SPX function status:

00h attempting to contact destination socket

EEh no such connection

EFh local connection table full

FDh buffer size not 42 or fragment count not 1

FFh sending socket not open

Format of SPX header:

Offset Size Description (Table 03826)

00h WORD (big-endian) checksum

02h WORD (big-endian) length in bytes of total packet

04h BYTE transport control

05h BYTE packet type (see #03816 at INT 7A/BX=0003h)

06h 10 BYTES destination internet address

10h WORD (big-endian) destination socket

12h 10 BYTES source internet address

1Ch WORD (big-endian) source socket

1Eh BYTE connection control (see #03827)

1Fh BYTE datastream type

FEh terminate connection request packet

FFh terminate connection acknowledgement packet

other user-defined, ignored by SPX

20h WORD (big-endian) source connection ID

22h WORD (big-endian) destination connection ID  
24h WORD (big-endian) sequence number  
26h WORD (big-endian) acknowledge number  
28h WORD (big-endian) allocation number

SeeAlso: #03815

Bitfields for SPX connection control:

Bit(s) Description (Table 03827)

3-0 unused???  
4 end of message  
5 reserved  
6 acknowledgement required  
7 system packet

-----N-7A----BX0012-----

INT 7A - Novell NetWare - SPX Driver - LISTEN FOR SPX CONNECTION

BX = 0012h  
AH = watchdog flag (00h disabled, 01h enabled)  
AL = retry count (00h = default)  
ES:SI -> Event Control Block (see #03810 at BX=0003h)

Return: nothing

Notes: there should always be at least two SPX ECB's listening to a socket, so  
that NetWare can perform its internal packet exchanges  
this function is supported by Advanced NetWare 2.1+

SeeAlso: BX=0011h,BX=0013h,BX=0014h

-----N-7A----BX0013-----

INT 7A - Novell NetWare - SPX Driver - TERMINATE SPX CONNECTION

BX = 0013h  
DX = connection ID to terminate  
ES:SI -> Event Control Block (see #03810 at BX=0003h)

Note: this function is supported by Advanced NetWare 2.1+

SeeAlso: BX=0011h,BX=0012h,BX=0014h

-----N-7A----BX0014-----

INT 7A - Novell NetWare - SPX Driver - ABORT SPX CONNECTION

BX = 0014h  
DX = connection ID to terminate

Return: nothing

Notes: this function is supported by Advanced NetWare 2.1+  
this function does not tell the other side that the connection has been  
terminated

also aborts any outstanding Establish Connection, Terminate Connection,  
and Send Sequenced Packet commands

SeeAlso: BX=0011h,BX=0013h

-----N-7A-----BX0015-----

INT 7A - Novell NetWare - SPX Driver - GET SPX CONNECTION STATUS

BX = 0015h

DX = connection ID

ES:SI -> status buffer (see #03828)

Return: AL = return code (00h,EEh) (see also #03825)

00h connection still valid

ES:SI -> status buffer filled

Note: this function is supported by Advanced NetWare 2.1+

SeeAlso: BX=0010h,BX=0011h

Format of SPX status buffer:

Offset Size Description (Table 03828)

00h BYTE connection state

01h waiting to establish connection

02h starting (attempting to create connection)

03h connection established

04h terminating

01h BYTE watchdog flag

bit 0: used internally by SPX

bit 1: SPX watchdog is monitoring connection

bits 2-7 used internally by SPX

02h WORD (big-endian) source connection ID

04h WORD (big-endian) destination connection ID

06h WORD (big-endian) sequence number of next packet sent

08h WORD (big-endian) acknowledge number, expected sequence number of next received packet

0Ah WORD (big-endian) maximum sequence number remote SPX may send without ACK from local SPX

0Ch WORD (big-endian) remote acknowledge number, next sequence number remote SPX expects to receive

0Eh WORD (big-endian) remote allocation number, maximum sequence number local SPX may send

10h WORD (big-endian) connection socket

12h 6 BYTES immediate node address--bridge on local network to destination

18h 10 BYTES destination internetwork address (see #03824 at INT 7A/BX=000Bh)

22h WORD (big-endian) retransmit count

24h WORD (big-endian) estimated roundtrip delay

26h WORD (big-endian) retransmitted packets

28h WORD (big-endian) suppressed packets

2Ah 12 BYTEs ??? (v2.15)

-----N-7A----BX0016-----

INT 7A - Novell NetWare - SPX Driver - SEND SPX PACKET

BX = 0016h

DX = connection ID

ES:SI -> Event Control Block (see BX=0003h)

Notes: this function is supported by Advanced NetWare 2.1+

CX may need to be 0001h ???

SeeAlso: BX=0011h,BX=0017h

-----N-7A----BX0017-----

INT 7A - Novell NetWare - SPX Driver - LISTEN FOR SPX PACKET

BX = 0017h

DX = connection ID (unused in v2.15)

ES:SI -> Event Control Block (see BX=0003h)

Notes: this function is supported by Advanced NetWare 2.1+

CX may need to be 0001h ???

SeeAlso: BX=0011h,BX=0016h

-----N-7A----BX0018-----

INT 7A U - Novell NetWare - IPX Driver - internal - ADD DIAGNOSTIC ELEMENT

BX = 0018h

ES:SI -> diagnostic element (see #03829) to be added to Diagnostic Queue

Note: this function is supported on file servers only under v2.15; v3.02 also supports it on workstations

SeeAlso: BX=0019h

Format of IPX diagnostic element:

Offset Size Description (Table 03829)

00h DWORD pointer to next diagnostic element

04h DWORD pointer to function for ???

08h DWORD pointer to function for ???

-----N-7A----BX0019-----

INT 7A U - Novell NetWare - IPX Driver - internal - CANCEL DIAGNOSTIC ELEMENT

BX = 0019h

ES:SI -> diagnostic element (see BX=0018h) to be removed

Note: this function is supported on file servers only under v2.15; v3.02 also supports it on workstations

SeeAlso: BX=0018h

-----N-7A----BX001A-----

INT 7A - Novell NetWare - IPX Driver - GET DRIVER MAXIMUM PACKET SIZE

BX = 001Ah

Return: AX = maximum packet size with preamble (at least 576 bytes)

CX = IPX retry count

DS,FLAGS preserved

SeeAlso: BX=000Dh

-----N-7A----BX001B-----

INT 7A U - Novell NetWare - IPX Driver - INTERNAL

BX = 001Bh

???

Return: ???

Notes: this function is supported on file servers only under v2.15; v3.02 also supports it on workstations

used by NetWare Access Server, which may call INT 15/AX=1117h with

BX=0000h in some cases (it uses a modified DESQview)

-----N-7A----BX001C-----

INT 7A U - Novell NetWare - NetWare Access Server - GET ???

BX = 001Ch

Return: AX = length of ???

ES:BX -> ???

Notes: these functions are NOPs for standard IPX drivers such as IPXODI v2.12

SeeAlso: INT 2F/AX=7AF1h, INT 7A/BX=001Dh, INT 7A/BX=001Eh

-----N-7A----BX001D-----

INT 7A U - Novell NetWare - NetWare Access Server - GET ??? ECB AND SOCKET

BX = 001Dh

Return: DX = socket number chosen by IPX

ES:SI -> ECB in IPX's code segment (will be listening)

BX corrupted

Notes: these functions are NOPs for standard IPX drivers such as IPXODI v2.12

SeeAlso: INT 2F/AX=7AF1h, INT 7A/BX=001Ch, INT 7A/BX=001Eh

-----N-7A----BX001E-----

INT 7A U - Novell NetWare - NetWare Access Server - GET ??? ECB

BX = 001Eh

Return: ES:SI -> ECB

BX corrupted

Notes: these functions are NOPs for standard IPX drivers such as IPXODI v2.12

SeeAlso: INT 2F/AX=7AF1h, INT 7A/BX=001Ch, INT 7A/BX=001Dh

-----N-7A----BX001F-----

INT 7A - Novell NetWare - IPXODI v2.12+ - GET IPX INFORMATION

BX = 001Fh

DX = 0000h

Return: AX = ???

BX:CX -> ??? entry point (see #03830) (undocumented)

DX = feature flags



bit 0: IPXODI rather than dedicated IPX  
bit 1: checksumming functions 0020h-0022h supported  
ES:SI -> array of words containing used socket number (undocumented);  
0000h ends array  
all other registers except DS and FLAGS may be destroyed  
Note: IPXODI v2.12 is distributed as part of the Personal NetWare system  
bundled with Novell DOS 7  
SeeAlso: BX=0020h,INT 2F/AX=7A2Fh

(Table 03830)

Call IPXODI entry point with:

ES:SI -> ECB??? (offset 24h is far pointer to ??? data)

-----N-7A----BX0020-----

INT 7A - Novell NetWare - IPXODI v2.12+ - SEND WITH CHECKSUM

BX = 0020h

ES:SI -> ECB (see #03810 at INT 7A/BX=0003h)

Return: DS,FLAGS preserved

BX,BP corrupted (documented as potentially destroying all other regs)

Desc: generate a checksum for the packet data and store it in the IPX

checksum field before transmitting the packet

SeeAlso: BX=001Fh,BX=0021h,BX=0022h,INT 2F/AX=7A2Fh

-----N-7A----BX0021-----

INT 7A - Novell NetWare - IPXODI v2.12+ - IPX GENERATE CHECKSUM

BX = 0021h

ES:SI -> ECB data (see #03810 at INT 7A/BX=0003h)

Return: ES,DS,SI preserved

BX,BP corrupted, all other registers potentially destroyed

Notes: the checksum and TransportControl fields of the IPX packet are updated

this function enables interrupts and is fully reentrant

SeeAlso: BX=001Fh,BX=0020h,BX=0021h,INT 2F/AX=7A2Fh

-----N-7A----BX0022-----

INT 7A - Novell NetWare - IPXODI v2.12+ - IPX VERIFY CHECKSUM

BX = 0022h

ES:SI -> ECB data (see #03810 at INT 7A/BX=0003h)

Return: AX = status (0000h checksum matches)

DS,ES,SI preserved

BX,BP corrupted, all other registers potentially destroyed

Note: this function enables interrupts and is fully reentrant

SeeAlso: BX=001Fh,BX=0020h,BX=0021h,INT 2F/AX=7A2Fh

-----N-7A----BX0023-----

INT 7A - Novell NetWare - IPXODI v2.12+ - OPEN LOOK-AHEAD SOCKET

BX = 0023h  
DX = socket number (0000h for dynamic allocation)  
ES:SI -> Look Ahead handler (see #03831)  
BP = desired lookahead size (0000h-0080h)  
Return: AL = status  
    00h successful  
    DX = assigned socket number (big-endian)  
    FEh maximum number of sockets already open  
    FFh specific socket already opened by another application  
DS,FLAGS preserved  
all other registers may be destroyed  
Notes: the socket will always be long-lived, and must thus be explicitly  
    closed with INT 7A/BX=0001h before the Look Ahead handler code is  
    removed from memory (i.e. the program terminates)  
this function is only supported if INT 2F/AX=7A00h returns ES:BX  
    pointing at an IPX version greater than 3.30  
the desired lookahead size is the smallest-size packet (including IPX  
    header) that should be passed to the lookahead handler when it  
    arrives  
SeeAlso: INT 7A/BX=0000h,INT 7A/BX=0001h

(Table 03831)

Call IPX Look-Ahead handler with:  
    AX = packet's destination socket  
    DS:SI -> look-ahead structure (see #03010 at INT 2F/AX=C000h"LSL")  
    DF clear  
    interrupt disabled (must remain disabled)

Return: AX = packet use  
    0000h application want packet  
    ES:SI -> ODI ECB (see #03011 at INT 2F/AX=C000h"LSL")  
    8001h application does not want packet  
ZF set if AX=0000h  
DS,DI,BP,SS,SP preserved  
interrupts disabled

-----N-7A----BX0024-----

INT 7A U - Novell NetWare - IPXODI v2.20+ - SET INTERNETWORK ADDRESS  
    BX = 0024h  
    ES:SI -> buffer containing internetnetwork address (see #03824)

Return: BX,CX,SI,DI,ES destroyed

Note: this function differs from INT 7A/BX=000Ch in that it unconditionally  
    sets the address

SeeAlso: INT 7A/BX=000Ch

-----A-7A0001-----

INT 7A R - AutoCAD Device Interface - DISPLAY - INITIALIZE (INIT1)

AX = 0001h

BX = configuration

CX = interface level (0001h)

Return: AX = status

0000h initialization failed

0001h pre-v4.0 driver initialized successfully

0003h post-v3.1 driver initialized successfully (driver truncates  
excessivly long packets)

BX = width of graphics area in pixels, or 0000h if using packet  
communication

DX:CX -> packet-mode entry point (see #03832)

SeeAlso: AX=0001h,AX=0200h,AX=07D0h,INT 61/AX=0007h"OPTIMA",INT 78"AutoCAD"

SeeAlso: INT 79/AX=0001h,INT 7B/AX=8001h"AutoCAD",INT 7C"AutoShade"

(Table 03832)

Call packet-mode entry point with:

STACK: DWORD pointer to request packet (see #03833)

WORD length of packet

Format of request packet:

Offset Size Description (Table 03833)

00h WORD function number (00h-45h for Compaq VGADI41.EXE)

-----A-7A0200BX0000-----

INT 7A R - AutoCAD Device Interface - RENDERING DEVICE - INITIALIZE (RDLINKUP)

AX = 0200h

BX = 0000h

CX = interface level (0001h)

Return: BX = 0000h (driver uses packet communication)

DX:CX -> packet-mode entry point (see #03832)

SeeAlso: AX=0001h"AutoCAD",INT 79/AX=0001h,INT 7B/AX=8001h"AutoCAD"

-----I-7A04-----

INT 7A - IBM 3270 Workstation Program API - CREATE A QUEUE

AH = 04h

???

Return: ???

SeeAlso: AH=06h

-----I-7A06-----

INT 7A - IBM 3270 Workstation Program API - DELETE A QUEUE

```
AH = 06h
???
```

Return: ???

SeeAlso: AH=04h

-----7A07D0-----

INT 7A R - AutoCAD Device Interface - Compaq VGADI41.EXE - GET ENTRY POINT

```
AX = 07D0h
```

Return: AX = ??? (0003h)

```
BX = 0000h
DX:DX -> ADI entry point (see AX=0001h)
```

SeeAlso: AX=0001h

-----I-7A09--BX8020-----

INT 7A - IBM 3270 Workstation Program API - SESSION SERVICES

```
AH = 09h
BX = 8020h (synchronous request)
CX = 0000h
DX = ID of session manager (SESSMGR)
AL = service (see #03834)
ES:DI -> control block
```

Return: ???

(Table 03834)

Values for service:

```
01h get session ID
02h get session info
04h dettach from session
05h attach to session
06h get list of windows available
07h get environment of window
08h get 'PIF' (program information file) info
0Ah get base window ID
0Bh get cursor info
```

-----I-7A09--BX8020-----

INT 7A - IBM 3270 Workstation Program API - KEYBOARD SERVICES

```
AH = 09h
BX = 8020h (synchronous request)
CX = 0000h
DX = ID of keyboard manager
AL = service number (see #03835)
ES:DI -> control block
```

Return: ???

(Table 03835)

Values for 3270 Workstation Program keyboard service:

- 01h connect to keyboard
- 02h disconnect from keyboard
- 03h read from keyboard
- 04h send keystroke to session
- 05h disable input
- 06h enable input
- 07h update status code

-----I-7A09--BX8020-----

INT 7A - IBM 3270 Workstation Program API - WINDOW SERVICES

- AH = 09h
- BX = 8020h (synchronous request)
- CX = 00FFh
- DX = ID of window service controller (WSCTRL)
- AL = service number (see #03836)
- ES:DI -> control block

Return: ???

(Table 03836)

Values for 3270 Workstation Program window service:

- 01h connect to WS control
- 02h disconnect from WS control
- 03h add a window
- 04h change window's position on screen
- 05h change window's size
- 06h change window's color
- 07h change window's position in the presentation space
- 08h hide/unhide toggle
- 09h enlarge/reduce toggle
- 0Ah change screen background color
- 0Bh get window's position on screen
- 0Ch get window's size
- 0Dh get window's color
- 0Eh get window's position in the presentation space
- 0Fh determine whether hidden
- 10h determine whether enlarged
- 11h get background color
- 12h get window names
- 13h delete all windows from profile

14h pick active window  
15h redraw screen  
16h redraw window  
17h delete a window from profile  
18h get active window  
19h get active screen  
1Ah get window data  
1Bh change window data  
1Ch select active screen

-----I-7A09--BX8020-----

INT 7A - IBM 3270 Workstation Program API - PRESENTATION SPACE SERVICES

AH = 09h  
BX = 8020h  
CX = 00FFh  
DX = ID of PCPSM  
AL = service  
01h define presentation space  
02h delete presentation space  
03h display presentation space  
04h position cursor in presentation space  
05h change default presentation space

ES:DI -> control block

Return: ???

-----I-7A09--BX8020-----

INT 7A - IBM 3270 Workstation Program API - 3270 EMULATION

AH = 09h  
BX = 8020h  
CX = 00FFh  
DX = ID of 3270EML  
AL = service  
01h connect  
02h disconnect

ES:DI -> control block

Return: ???

-----I-7A09--BX8020-----

INT 7A - IBM 3270 Workstation Program API - OPERATOR INFORMATION AREA

AH = 09h  
BX = 8020h  
CX = 00FFh  
DX = ID of OIAM  
AL = service

```
    01h read Operator Information Area
    02h read OIA subset
ES:DI -> control block
Return: ???
Note: the OIA is the 25th line on the Host session
-----I-7A09--BX8020-----
INT 7A - IBM 3270 Workstation Program API - TRANSLATE DATA
    AH = 09h
    BX = 8020h
    CX = 00FFh
    DX = ID of XLATE
    AL = service
        01h translate from host characters to ASCII and vice versa
        (determined by control block byte 11)
ES:DI -> control block
Return: ???
-----I-7A09--BX8020-----
INT 7A - IBM 3270 Workstation Program API - COPY SERVICE
    AH = 09h
    BX = 8020h
    CX = 00FFh
    DX = ID of copy service
    AL = service
        01h copy string from one presentation space to another
        02h copy block from one presentation space to another
        03h connect to PC session for copy
        04h disconnect PC session from copy
ES:DI -> control block
Return: ???
-----I-7A09--BX8020-----
INT 7A - IBM 3270 Workstation Program API - Multi-DOS
    AH = 09h
    BX = 8020h
    CX = 00FFh
ES:DI -> control block
    DX = ID of INDJQRY
        get environment size
        = ID of INDJASY
        request DOS functions from workstation
        = ID of MEMORY
    AL = function
```

01h allocate memory  
02h deallocate memory  
03h modify allocated size

Return: ???

-----I-7A09-----

INT 7A - IBM 3270 Workstation Program API - HOST SERVICES

AH = 09h  
BX = request type (4000h async, 8028h synchronous)  
CX = 0000h  
DX = ID of MFIC  
AL = service (see #03837)  
ES:DI -> control block

(Table 03837)

Values for 3270 Workstation Program host service:

01h connect to host  
02h disconnect from host  
03h read DFT structured data from host  
04h write DFT structured data to host  
05h create a host buffer

-----I-7A13-----

INT 7A - IBM 3270 Workstation Program API - GET DATA FROM A QUEUE

AH = 13h  
???

Return: ???

-----7A7A-----

INT 7A - Canon IX-30F Image Scanner SI3.SYS - INTERFACE

AH = 7Ah  
???

Return: AX = status

0000h success  
000Eh undefined function code

Range: unknown, selected by device driver commandline arguments

-----I-7A81-----

INT 7A - IBM 3270 Workstation Program API - RESOLVE A GATE NAME

AH = 81h  
ES:DI -> 8-char blank-padded gate name (see #03838)

Return: DX = gate ID

(Table 03838)

Values for 3270 Workstation Program gate name:



"SESSMGR "  
"KEYBOARD"  
"WSCTRL "  
"MFIC "  
"PCPSM "  
"3270EML "  
"COPY "  
"XLATE "  
"OIAM "  
"MEMORY "  
"INDJQRY "  
"INDJASY "

-----I-7A83-----

INT 7A - IBM 3270 Workstation Program API - GET COMPLETION RESULTS

AH = 83h

???

Return: ???

-----I-7AFDCB-----

INT 7A - IBM Personal Communications/3270 - INSTALLATION CHECK

AX = FDCBh

Return: DX:AX -> PCS/3270 signature block if loaded (see #03839)

Format of Personal Communications/3270 signature block:

Offset Size Description (Table 03839)

04h WORD PCS/3270 signature (5741h)

06h WORD version (0501h = PCS/3270 v1.0)

-----I-7AFE01-----

INT 7A - IBM PC3270 EMUL PROG v3 - INTERNAL SEND/RECEIVE FUNCTION

AX = FE01h

???

Return: ???

SeeAlso: AX=FE02h

-----I-7AFE02-----

INT 7A - IBM PC3270 EMUL PROG v3 - INTERNAL SEND/RECEIVE FUNCTION

AX = FE02h

???

Return: ???

SeeAlso: AX=FE01h

-----I-7AFF01-----

INT 7A - IBM PC3270 EMUL PROG v3 - INTERNAL API INITIALIZATION

AX = FF01h

```

ES:DI -> API function handler routine
Return: CX = 1200h
SeeAlso: AX=FF02h,AX=FF03h
-----I-7AFF02-----
INT 7A - IBM PC3270 EMUL PROG v3 - INTERNAL API TERMINATION
  AX = FF02h
Return: CX = 1200h
SeeAlso: AX=FF01h
-----I-7AFF03-----
INT 7A - IBM PC3270 EMUL PROG v3 - INTERNAL API INITIALIZATION
  AX = FF03h
  ES:DI -> send/receive function handler routine
Return: CX = 1200h
SeeAlso: AX=FF01h
-----I-7AFF04-----
INT 7A U - IBM PC3270 EMUL PROG v3 - INTERNAL ???
  AX = FF04h
  ES:DI -> ???
Return: CX = 1200h
-----f-7B-----
INT 7B - Btrieve API (single user)
  DS:DX -> 38-byte parameter record (see #03840)
Return: return code field set
Note: Btrieve sets low byte of vector to 33h; this serves as the installation
      check
SeeAlso: INT 2F/AX=AB01h,INT 7F/AX=0200h

Format of Btrieve parameter record:
Offset Size Description (Table 03840)
00h  DWORD pointer to data buffer
04h  WORD  data buffer length
06h  DWORD pointer to 90-byte record containing positioning info
      (should be same for all calls for same file)
0Ah  DWORD pointer to 38-byte FCB info buffer
      (should be same for all calls for same file)
0Eh  WORD  function code (see #03841)
10h  DWORD pointer to file name/key buffer
14h  BYTE  key length
15h  BYTE  key number
16h  DWORD pointer to status code (see #03842)
1Ah  WORD  interface code (version specific)

```

6176h version 5.10

(Table 03841)

Values for function code:

00h open  
01h close  
02h insert  
03h update  
04h delete  
05h get\_equal  
06h get\_next  
07h get\_prev  
08h get\_greater  
09h get\_gr\_eql  
0Ah get\_less  
0Bh get\_less\_eq  
0Ch get\_first  
0Dh get\_last  
0Eh create  
0Fh stat  
10h extend  
11h set\_dir: set directory information  
12h get\_dir: get directory information  
13h begin\_trans  
14h end\_trans  
15h abort\_trans  
16h get\_pos: get record position number  
17h get\_direct: get data by sending record position  
18h step\_next  
19h stop  
1Ah version  
1Bh unlock  
1Ch reset  
1Dh set owner  
1Eh clear owner  
1Fh create supplemental index  
20h drop supplemental index  
21h step first  
22h step last  
23h step previous  
24h get next extended: get multiple records using a filter

25h get previous extended: get multiple records using a filter  
26h step next extended: get multiple records using a filter  
27h step previous extended: get multiple records using a filter  
28h insert extended: insert one or more records  
31h ???

Notes: add 50 (32h) to any "get" operation to just return the key data  
add 100 (64h) for a single-record wait lock (automatically released  
on next get)  
add 200 (C8h) for a single-record nowait lock (nowait lock returns  
error 54h or 55h if record already locked)  
add 300 (12Ch) for a multiple-record wait lock (not released until  
unlock called)  
add 400 (190h) for a multiple-record nowait lock (nowait lock returns  
error 54h or 55h if record already locked)

(Table 03842)

Values for Btrieve status code:

00h successful  
01h invalid operation  
02h I/O error  
03h file not open  
04h key value not found  
05h duplicate key value  
06h invalid key number  
07h different key number  
08h invalid positioning  
09h end of file  
0Ah modifiable key value error  
0Bh invalid file name  
0Ch file not found  
0Dh extended file error  
0Eh pre-image open error  
0Fh pre-image I/O error  
10h expansion error  
11h close error  
12h disk full  
13h unrecoverable error  
14h record manager inactive  
15h key buffer too short  
16h data buffer length overrun  
17h position block length

18h page size error  
19h create I/O error  
1Ah number of keys  
1Bh invalid key position  
1Ch invalid record length  
1Dh invalid key length  
1Eh not a Btrieve file  
1Fh file already extended  
20h extended I/O error  
22h invalid extension name  
23h directory error  
24h transaction error  
25h transaction is active  
26h transaction control file I/O error  
27h end/abort transaction error  
28h transaction max files  
29h operation not allowed  
2Ah incomplete accelerated access  
2Bh invalid record address  
2Ch null key path  
2Dh inconsistent key flags  
2Eh access to file denied  
2Fh maximum open files  
30h invalid alternate sequence definition  
31h key type error  
32h owner already set  
33h invalid owner  
34h error writing cache  
35h invalid interface  
36h variable page error  
37h autoincrement error  
38h incomplete index  
39h expanded memory error  
3Ah compression buffer too short  
3Bh file already exists  
3Ch reject count reached  
3Dh work space too small  
3Eh incorrect descriptor  
3Fh invalid extended insert  
40h filter limit reached  
41h incorrect field offset

4Ah automatic transaction abort  
4Dh Btrieve engine busy or resource locked by another user  
4Eh deadlock detected  
50h conflict  
51h lock error  
52h lost position  
53h read outside transaction  
54h record in use  
55h file in use  
56h file table full  
57h handle table full  
58h incompatible open mode  
5Ah redirected device table full  
5Bh server error  
5Ch transaction table full  
5Dh incompatible lock type  
5Eh permission error  
5Fh session no longer valid  
60h communications environment error  
61h data message too small  
62h internal transaction error

-----I-7B-----

INT 7B - Eicon Access API (3270/5250 gateways)

SeeAlso: INT 5C"NetBIOS"

-----E-7B-----

INT 7B - DJGPP GO32.EXE DOS EXTENDER - RELOCATED IRQ3

Notes: this vector is overwritten when GO32 starts but is not restored by  
early versions of the extender

the newest versions of GO32 dynamically allocate the vectors used  
for the relocated IRQs, much as DESQview does (see INT 50"DESQview")

SeeAlso: INT 0B,INT 10/AH=FFh"GO32",INT 7A"GO32",INT 7C"GO32"

-----A-7B-----

INT 7B CR - AutoCAD Device Interface - PRINTER/PLOTTER - SEND PLOTTING DATA

AX = data length (0000h-7FFFh)

CX:BX -> data to be plotted

DX = vertical position of line on page

Return: nothing

Notes: This interrupt is called by AutoCAD to perform its output

A data length of 0000h indicates that a blank line is to be output

SeeAlso: AX=8001h,AX=8002h,INT 78"AutoCAD",INT 7A/AX=0001h"AutoCAD"

-----e-7B-----

INT 7B U - MCI Express v2.11+ - DRIVER.EXE - ???

Program: MCI Express is a front end to MCI Mail; DRIVER.EXE is its  
communication driver

-----7B-----

INT 7B - Novell XQL - XQL PRIMITIVES MANAGER API

DS:DX -> parameter block (see #03844)

Return: parameter block updated

(Table 03843)

Values for Novell XQL function number:

00h log in  
01h log out  
02h "tNewFcn"  
03h "tFreeFcn"  
04h "tMoveFldFcn"  
05h "tFieldFcn"  
06h "tDescribeFcn"  
07h "tFetchFcn"  
08h "tComputeFcn"  
09h "tOrderFcn"  
0Ah "tRestrictFcn"  
0Bh "tJoinFcn"  
0Ch "tInsertFcn"  
0Dh "tUpdateFcn"  
0Eh "tUpdallFcn"  
0Fh "tRemoveFcn"  
10h "tRemallFcn"  
11h "tResetFcn"  
12h "tTransFcn"  
13h "tStatFcn"  
14h "tDDCreFcn"  
15h "tStoreFcn"  
16h "tRecallFcn"  
17h "tStopFcn"  
18h "tDDFileFcn"  
19h "tDDFieldFcn"  
1Ah "tDDIndexFcn"  
1Bh "tDDModFcn"  
1Ch "tDDViewFcn"  
1Dh "tDDDrpFcn"  
1Eh "tDDPathFcn"

1Fh "tDDAttrFcn"  
20h "tDDFcn"  
21h "tSecurFcn"  
22h "tUserFcn"  
23h "tAccessFcn"  
24h "tPswdFcn"  
25h "tCharFcn"  
26h "tVersionFcn"  
27h "tCnvFcn"  
28h "tValFcn"  
29h "tMaskFcn"

SeeAlso: #03844

Format of XQL parameter block:

Offset Size Description (Table 03844)

00h 4 BYTES signature "XQLP"  
04h WORD function number (see #03843)  
06h WORD handle  
08h WORD status  
0Ah WORD session ID  
0Ch var data record (varies by function)

---function 00h---

0Ch DWORD -> user name  
10h DWORD -> password  
14h DWORD -> dictionary  
18h DWORD -> filepath  
1Ch DWORD -> machine name  
20h WORD process ID

---function 01h---

no additional fields???

---function 02h---

0Ch DWORD -> filename  
10h DWORD -> owner name  
14h WORD open mode

---function 03h---

no additional fields???

---function 04h---

0Ch WORD "frompos"  
0Eh WORD "topos"

---function 05h---

0Ch WORD subfunction



```
0Eh WORD position
10h WORD count
12h DWORD -> field list
---function 06h---
0Ch WORD subfunction
0Eh WORD length of description
10h WORD position
12h WORD count
14h DWORD pointer to description list
---function 07h---
0Ch WORD data length
0Eh WORD "op"
10h DWORD "select"
14h DWORD "reject"
18h DWORD pointer to data buffer
---function 08h---
0Ch DWORD -> field name
10h WORD field type
12h WORD field size
14h WORD "flddec"
16h WORD "explen"
18h DWORD pointer to "expbuf"
---function 09h---
0Ch WORD count
0Eh DWORD -> field list
---function 0Ah---
0Ch WORD "newexp"
0Eh WORD "explen"
10h DWORD -> "expbuf"
---function 0Bh---
0Ch DWORD -> filename
10h DWORD -> owner name
14h WORD type of join
16h WORD "pCount"
18h DWORD -> "pNames"
1Ch WORD "sCount"
1Eh DWORD -> "sNames"
---functions 0Ch, 0Dh, 0Fh---
0Ch WORD file count
0Eh DWORD -> file list
12h DWORD count
```

```
16h  DWORD -> buffer
---function 0Eh---
0Ch  WORD  file count
0Eh  DWORD -> file list
12h  WORD  "op"
14h  DWORD "select"
18h  DWORD "reject"
1Ch  WORD  count
1Eh  DWORD -> "res"
22h  DWORD -> "comp"
---function 10h---
0Ch  WORD  file count
0Eh  DWORD -> file list
12h  WORD  "op"
14h  DWORD "select"
18h  DWORD "reject"
---function 11h---
0Ch  DWORD machine name
---function 12h---
0Ch  WORD  option
---function 13h---
0Ch  WORD  subfunction
0Eh  WORD  length of buffer
10h  DWORD -> buffer
---function 14h---
0Ch  WORD  subfunction
0Eh  DWORD -> filename
12h  WORD  "create"
14h  DWORD -> pathname
18h  DWORD -> owner name
1Ch  WORD  owner access restrictions
1Eh  WORD  number of fields
20h  DWORD -> field list
24h  WORD  number of keys
28h  DWORD -> key list
2Ch  WORD  "buflen"
2Eh  DWORD -> create parameters
---function 15h---
0Ch  DWORD -> "StoreName"
10h  WORD  length of buffer
12h  DWORD -> buffer
```

```
---function 16h---
0Ch  DWORD -> "RecallName"
10h  WORD  owner number
12h  DWORD -> owner list
16h  WORD  open mode
18h  WORD  length of buffer
1Ah  DWORD -> buffer
---function 17h---
no additional fields???
---function 18h---
0Ch  WORD  "rtype"
0Eh  WORD  count
10h  DWORD -> "filebuf"
14h  WORD  length of following buffer
16h  DWORD -> output buffer
---functions 19h, 1Ah---
0Ch  WORD  "rtype"
0Eh  WORD  count
10h  DWORD -> field buffer
14h  WORD  length of following buffer
16h  DWORD -> output buffer
---function 1Bh---
0Ch  WORD  subfunction
0Eh  DWORD -> filename
12h  WORD  "create"
14h  DWORD -> pathname
18h  DWORD -> owner name
1Ch  WORD  owner access restrictions
1Eh  WORD  field count
20h  DWORD -> field list
24h  WORD  number of keys
26h  DWORD -> list of keys
---function 1Ch---
0Ch  WORD  count
0Eh  DWORD -> view name
12h  WORD  length of following buffer
14h  DWORD -> output buffer
---function 1Dh---
0Ch  DWORD -> filename
10h  WORD  file type
12h  WORD  "deletepath"
```

```
---function 1Eh---
0Ch WORD "dirtytype"
0Eh DWORD -> pathname
---function 1Fh---
0Ch WORD function
0Eh DWORD -> field name
12h WORD attribute
14h WORD size of buffer
16h DWORD -> buffer for attributes
---function 20h---
0Ch DWORD -> directory path
10h WORD function
---function 21h---
0Ch DWORD -> master password
10h WORD "securityflag"
---function 22h---
0Ch DWORD -> master password
10h WORD function (add/modify/remove)
12h DWORD -> user name
16h DWORD -> password
1Ah WORD global rights
1Ch WORD count
1Eh WORD length of following buffer
20h DWORD -> "userbuf"
---function 23h---
0Ch DWORD -> master password
10h DWORD -> user name
14h WORD function (allow/deny/getrights)
16h WORD rights
18h DWORD -> filename
1Ch WORD number of field names in following buffer
1Eh DWORD -> field names
22h WORD maximum buffer length
24h DWORD -> output buffer
---function 24h---
0Ch DWORD -> user name
10h DWORD -> password
---function 25h---
0Ch WORD "charFcn"
0Eh WORD "charType"
10h WORD "charValue"
```

---function 26h---

0Ch DWORD -> buffer

---function 27h---

0Ch WORD option

0Eh WORD type

10h WORD size

12h WORD "dec"

14h WORD "dsize"

16h DWORD -> value

1Ah DWORD -> "retval"

1Eh DWORD -> mask

22h WORD "justify"

---function 28h---

0Ch DWORD -> field name

10h WORD length of buffer

12h DWORD -> buffer for data

---function 29h---

0Ch WORD option

0Eh WORD type

10h WORD size

12h WORD "dec"

14h WORD length of mask

16h DWORD -> mask

-----A-7B8001-----

INT 7B R - AutoCAD Device Interface - PRINTER/PLOTTER - BEGIN PLOT

AX = 8001h

BX = file level

0000h binary file

0001h CAD/camera image file

0002h installed ADI driver

CX = horizontal size in pixels

DX = vertical size in pixels

Return: AX = status (0000h initialization failed, 0001h init successful)

BX = allowed output format

0000h monochrome

0001h color (4 bits per pixel)

SeeAlso: AX=8002h,AX=8003h,INT 7B"AutoCAD",INT 7C"AutoShade"

-----A-7B8002-----

INT 7B R - AutoCAD Device Interface - PRINTER/PLOTTER - END PLOT

AX = 8002h

SeeAlso: AX=8001h,AX=8003h

-----A-7B8003-----

INT 7B R - AutoCAD Device Interface - PRINTER/PLOTTER - ABORT PLOT

AX = 8003h

SeeAlso: AX=8001h,AX=8002h

-----r-7C-----

INT 7C U - IBM REXX88PC command language

???

-----E-7C-----

INT 7C - DJGPP GO32.EXE DOS EXTENDER - RELOCATED IRQ4

Program: GO32.EXE is a DOS extender included as part of the 80386 port of the GNU C/C++ compiler by DJ Delorie and distributed as DJGPP

Notes: this vector is overwritten when GO32 starts but is not restored by early versions of the extender

the newest versions of GO32 dynamically allocate the vectors used

for the relocated IRQs, much as DESQview does (see INT 50"DESQview")

SeeAlso: INT 0C,INT 7B"GO32",INT 7D"GO32"

-----P-7C-----

INT 7C - PRINDIR v8.02-8.xx - API

AH = function

00h set output to Invisible mode

01h set output to LPT1

02h set output to LPT2

03h set output to LPT3

04h set output to COM1

05h set output to COM2

06h set output to COM3

07h set output to COM4

08h set output to screen

09h set output to file

0Ah change file path

ES:DI -&gt; new path

CX = length of path (max 41, but not range-checked)

0Bh change file name

ES:DI -&gt; new filename

CX = length of new name (max 41, but not range-checked)

0Ch set redirected device

(01h = LPT1 ... 03h = LPT3, 04h = COM1 ... 07h = COM4)

0Dh force buffer dump

0Eh set display color

AL = new color attribute (see #00014 at INT 10/AH=08h)

0Fh set popup hotkey

AL = hotkey scancode  
10h toggle byte display  
AL = new state (00h off, 01h on)  
11h change buffer flush delay  
AL = timer ticks before flush

Return: AX = status

0000h successful  
0001h invalid function or nothing in buffer to be dumped  
(documented as "nonzero = error")

Program: PRINDIR is a shareware printer-redirection program by J.M. Allen  
Creations

InstallCheck: test for the ASCIZ signature "PRINDIR N.NN" (where N.NN is the  
version number) two bytes beyond the start of the interrupt handler

Note: prior versions of PRINDIR used INT 7A instead of INT 7C

SeeAlso: INT 7A"PRINDIR",INT 2F/AX=7F00h"PRINDIR"

Index: hotkeys;PRINDIR

-----A-7C-----

INT 7C - AutoShade v2 - RENDERING HARDCOPY DRIVER

Note: this is the default interrupt vector for rendering hardcopy drivers  
instead of INT 7A used by AutoCAD drivers

SeeAlso: INT 7A/AX=0001h"AutoCAD"

-----N-7C-----

INT 7C - SK-UPPS/SK-PASSPORT Data Link Interface (DLI) API

Program: SK-UPPS is the Schneider & Koch Universal Portable Protocol Stack,  
which provides an API on top of it Data Link Interface, allowing the  
use of any of several network transport layers; this product is  
called SK-PASSPORT in the USA.

InstallCheck: test for the ASCII signature string "UPPS\_DLI" immediately  
preceding the interrupt handler

Notes: To find the service interrupt being used by the driver, an application  
should scan through the interrupt table until it finds an interrupt  
vector with the "UPPS\_DLI" string.

There may be more than one DLI loaded, each having its own service  
interrupt.

SeeAlso: AX=0000h,AX=0002h,AX=000Fh

Index: installation check;SK-UPPS

-----N-7C0000-----

INT 7C - SK-UPPS Data Link Interface API - GET DRIVER VERSION

AX = 0000h

Return: AH = major DLI version

AL = minor DLI version

DX = OEM signature (see #03845)  
CL = DLI 2.8+ topology: 1=Ethernet, 2=FDDI, 3=Token-Ring  
SeeAlso: AX=0002h,AX=000Bh,AX=000Dh,AX=000Eh

(Table 03845)

Values for SK-UPPS Data Link Interface OEM signature (defined by SK):

0001h	experimental
3343h ('3C')	3COM
4942h ('IB')	IBM
4943h ('IC')	Intellicom
494Eh ('IN')	Intel
4B4Fh ('KO')	Kodiak
4D41h ('MA')	Madge
4E45h ('NE')	Novell
4F43h ('OC')	Olicom
5349h ('SI')	Siemens
534Bh ('SK')	Schneider & Koch (SysKonnect)
5744h ('WD')	Western Digital

-----N-7C0001-----

INT 7C - SK-UPPS Data Link Interface API - DOWN DRIVER

AX = 0001h

Return: AX = completion code (see #03846)

Note: the DLI will refuse to unload if there are any protocols  
(see AX=0007h) or clients (see AX=0010h) active.

Index: uninstall;SK-UPPS Data Link Interface

(Table 03846)

Values for SK-UPPS completion code:

0000h	SUCCESS
0005h	INVALID_MULTICAST
0006h	BUFFER_TOO_SMALL
0007h	NO_BUFFERS_LEFT
0008h	NO_RESOURCES_LEFT
000Bh	ILLEGAL_PROTOCOL_ID
000Ch	PROTOCOL_IN_USE
000Dh	NO_MULTICAST_ADDRESS
000Fh	PROTOCOLS_ACTIVE
0010h	CLIENTS_ACTIVE
001Bh	INVALID_MODE
001Ch	MODE_NOT_SUPPORTED
FFFFh	UNKNOWN_SERVICE



-----N-7C0002-----

INT 7C - SK-UPPS Data Link Interface API - GET DRIVER STATUS

AX = 0002h

ES:DI -&gt; Status Information Block (SIB) (see #03858)

Return: AX = completion code (see #03846)

Note: GET DRIVER INFO (AX=000Eh) should be used instead.

SeeAlso: AX=0000h,AX=000Bh,AX=000Dh,AX=000Eh

-----N-7C0003-----

INT 7C - SK-UPPS Data Link Interface API - CHANGE DRIVER ADDRESS

AX = 0003h

ES:DI -&gt; new node address (6 bytes)

Return: AX = completion code (see #03846)

Note: the DLI will refuse to change the address if any protocols

(see AX=0007h) are active

SeeAlso: AX=0001h,AX=000Eh

-----N-7C0004-----

INT 7C - SK-UPPS Data Link Interface API - REQUEST BUFFER

AX = 0004h

DX = protocol ID (see AX=0007h,#03849)

Return: AX = completion code (see also #03846)

0000h successful

ES:BX -&gt; Memory Buffer (Mbuf) (see #03847)

Note: if the DLI has no free Mbufs, it will set an internal flag for the protocol and later call the protocol's Event Upcall with

EV\_BUFFER\_AVAILABLE (see #03853,#03854) as soon as an Mbuf becomes available. There will only be one "buffer available" Upcall, even if this function has been called several times before.

SeeAlso: AX=0005h,AX=0006h,AX=0012h,AX=0013h

Format of SK-UPPS Memory Buffer (Mbuf):

Offset Size Description (Table 03847)

00h DWORD -&gt; NEXT Mbuf (linked list)

04h WORD OFFSET of actual data within DATA area

06h 2 BYTES reserved (currently unused)

08h WORD LENGTH of actual data found in DATA area starting at OFFSET

0Ah 2 BYTES reserved (currently unused)

0Ch DWORD -&gt; DATA area (size can be obtained via AX=000Eh call)

10h WORD PROTOCOL ID of protocol currently "owning" the Mbuf

12h BYTE IN USE flag; nonzero if Mbuf still in use by DLI (see AX=0006h)

13h BYTE RECEIVE STATUS (bit 0-7 only; see AX=0007h - Receive Upcall)

-----N-7C0005-----

INT 7C - SK-UPPS Data Link Interface API - RELEASE BUFFER

AX = 0005h

ES:BX -> Mbuf (see #03847)

Return: nothing

SeeAlso: AX=0004h

-----N-7C0006-----

INT 7C - SK-UPPS Data Link Interface API - TRANSMIT FRAME

AX = 0006h

ES:BX -> Mbuf (see #03847)

Return: nothing

Notes: The DLI will send LENGTH bytes found in the Mbuf's DATA area starting at OFFSET. A complete and valid frame must be stored there. The DLI will only copy the current node address (see AX=0003h) to the source node field of the frame's MAC header.

On Ethernet, the DLI will always send a minimum of 60 bytes, regardless of the value found in the LENGTH field.

This function returns immediately; the DLI will call the protocol's Transmit Upcall (see #03848) when the frame has been sent.

The Mbuf's IN USE field will be non-zero until the DLI calls the Transmit Upcall.

SeeAlso: AX=0002h,AX=0004h,AX=0013h

(Table 03848)

Values SK-UPPS Transmit Upcall is called with:

ES:BX -> Mbuf (NEXT field destroyed)

interrupts disabled

Note: On FDDI the bytes of the source and destination node fields in the MAC header have been bit-swapped to physical address format.

-----N-7C0007-----

INT 7C - SK-UPPS Data Link Interface API - REGISTER PROTOCOL

AX = 0007h

ES:BX -> Protocol Control Block (see #03849)

Return: AX = completion code (see #03846)

Program: SK-UPPS is the Schneider & Koch Universal Portable Protocol Stack, which provides an API on top of its Data Link Interface, allowing the use of any of several network transport layers

SeeAlso: AX=0008h,AX=000Bh

Format of SK-UPPS Protocol Control Block (PCB):

Offset Size Description (Table 03849)

00h WORD (big-endian) protocol type

02h WORD protocol mode (see #03850)  
04h DWORD -> Receive Upcall routine (see #03851)  
08h DWORD -> Transmit Upcall routine  
0Ch DWORD -> Event Upcall routine (see #03853)  
10h DWORD -> ASCIZ protocol name (zero if none)  
14h WORD protocol ID will be returned here (always non-zero)

Bitfields for SK-UPPS protocol mode:

Bit(s) Description (Table 03850)

0 promiscuous mode; protocol receives all frames regardless of their destination (PROTOCOL TYPE must be FFFFh)  
1 if set, protocol's Event Upcall will be called on entry and exit of the DLI's Interrupt Service Routine (see #03854)  
2 DLI 2.2+ hook mode; protocol receives all directed frames destined for this station (PROTOCOL TYPE must be FFFFh)  
3 DLI 2.3+ receive error frames (only valid in promiscuous mode (0))  
4-5 DLI 2.5+ frame type:  
00 Ethernet II; specify type code in PROTOCOL TYPE field (high-low)  
01 IEEE 802.2; specify Service Access Point (SAP) in MSB (offset 0) of PROTOCOL TYPE field; LSB must be zero  
10 IEEE 802.2 with SNAP header; specify type code in PROTOCOL TYPE field (high-low)  
11 reserved (currently unused)  
6 DLI 2.5+ if set, protocol's Event Upcall will be called with event code EV\_BUFFER\_WANTED, when DLI runs out of Mbufs  
7 DLI 2.7+ chain mode; if set, client allows the specified frame type to be shared with other clients  
8 DLI 2.8+ receive non-LLC frames rather than LLC frames  
9 DLI 2.8+ receive all multicast frames (see also AX=0009h)  
10-15 reserved (currently unused; must be zero)

Notes: If PROTOCOL TYPE is FFFFh and neither the "promiscuous" nor the "hook" mode is set, the protocol receives all frames that did not match any registered frame type (demultiplexor mode).

Protocols using "promiscuous", "hook" or "chain" mode should always return from the Receive Upcall with the received Mbuf in ES:BX and the Carry Flag set. The DLI will then pass the Mbuf to the FEED BACK BUFFER function (see AX=000Fh).

Some DLIs do not support all or even any of the "promiscuous", "receive non-LLC" and "receive all multicasts" mode bits.

SeeAlso: #03849

(Table 03851)

Values SK-UPPS Receive Upcall is called with:

ES:BX -> Mbuf

AX = receive status (see #03852) (also stored in Mbuf's RECEIVE STATUS field)

interrupts disabled

Return: nothing

SeeAlso: #03853

Bitfields for SK-UPPS receive status:

Bit(s) Description (Table 03852)

0-2 reserved (0), currently unused

3 frame check sequence (FCS) error

4 overflow (frame too long)

5 framing error

6 reserved (0), currently unused

7 non-LLC frame (see #03850 [bit 8])

8-15 reserved (0), currently unused

Notes: The protocol's Receive Upcall routine will be called whenever a frame matching the specified frame type is received. A received frame is passed to the protocol in the Mbuf's DATA area at OFFSET and its size is LENGTH bytes.

The receive status passed to the Receive Upcall in register AL is always zero, except for protocols using "non-LLC" mode (8 of PCB's protocol mode field) or "promiscuous" mode together with "receive error frames" (0 & 3).

Upcalls are FAR routines and must return with a RET FAR. If the Carry Flag is set on return from a Receive or Transmit Upcall, then ES:BX must point to an Mbuf, which the DLI will then pass to the RELEASE BUFFER function (see AX=0005h). See also notes for PROTOCOL MODE.

(Table 03853)

Values SK-UPPS Event Upcall is called with:

AX = event code (see #03854)

Return: nothing

Note: All Event Upcalls may be ignored by the protocol.

SeeAlso: #03851

(Table 03854)

Values for SK-UPPS Event Upcall event code:

00h EV\_BUFFER\_AVAILABLE client may now call REQUEST BUFFER (AX=0004h)

01h EV\_ISR\_START start of Interrupt Service Routine

02h EV\_ISR\_END           end of ISR (see #03850 [bit 1])  
 03h EV\_BUFFER\_WANTED    DLI temporarily out of Mbufs  
 04h EV\_NETWORK\_STATUS   DLI 2.8+ network status -- DX=status (see #03855)  
 05h EV\_IO\_CONTROL        DLI 2.8+ I/O control -- DX=subfunc (see AX=0016h)

SeeAlso: #03853

Bitfields for EV\_NETWORK\_STATUS in register DX:

Bit(s) Description (Table 03855)

Token-Ring	FDDI
15 SIGNAL_LOSS	SIGNAL_LOSS
14 HARD_ERROR	HARD_ERROR
13 SOFT_ERROR	SOFT_ERROR
12 TRANSMIT_BEACON	TRANSMIT_BEACON
11 LOBE_WIRE_FAULT	PATH_TEST_FAILED
10 AUTO_REMOVAL	SELF_TEST_REQUIRED
9 reserved	reserved
8 REMOVE_RECEIVED	REMOTE_DISCONNECT
7 COUNTER_OVERFLOW	reserved
6 SINGLE_STATION	DUPLICATE_ADDRESS
5 RING_RECOVERY	NO_RING_OP_STATUS
4 reserved	VERSION_MISMATCH
3 reserved	STUCK_BYPASS
2 reserved	FDDI_EVENT
1 reserved	RING_OP_CHANGE
0 reserved	reserved

Note: EV\_NETWORK\_STATUS is not supported on Ethernet

-----N-7C0008-----

INT 7C - SK-UPPS Data Link Interface API - DISABLE PROTOCOL

AX = 0008h

BX = protocol ID (see #03849)

Return: AX = completion code (see #03846)

Note: the DLI will handle any Mbufs given to TRANSMIT FRAME (AX=0006h) as if they had been given to TRANSMIT FRAME RELEASE (AX=0013h).

all multicast addresses registered for this protocol will be disabled.

after this function returns, the protocol's Upcall routines will no longer be called.

SeeAlso: AX=0007h,AX=000Bh

-----N-7C0009-----

INT 7C - SK-UPPS Data Link Interface API - REGISTER MULTICAST

AX = 0009h

BX = protocol ID (see #03849)

ES:DI -> multicast address (6 bytes)

Return: AX = completion code (see #03846)

Notes: The broadcast address is always enabled.

Set bit 9 of PCB's protocol mode field in REGISTER PROTOCOL call  
(AX=0007h) to receive all multicast frames.

SeeAlso: AX=000Ah,AX=0014h

-----N-7C000A-----

INT 7C - SK-UPPS Data Link Interface API - DISABLE MULTICAST

AX = 000Ah

BX = protocol ID (see #03849)

ES:DI -> multicast address (6 bytes)

Return: AX = completion code (see #03846)

SeeAlso: AX=0009h,AX=0014h

-----N-7C000B-----

INT 7C - SK-UPPS Data Link Interface API - GET PROTOCOL LIST

AX = 000Bh

ES:DI -> protocol ID buffer

CX = number of 2-byte protocol IDs that may be stored in buffer

Return: AX = completion code (see #03846)

CX = number of active protocols (regardless of value returned in AX)

protocol ID buffer filled with active protocol IDs

Program: SK-UPPS is the Schneider & Koch Universal Portable Protocol Stack,  
which provides an API on top of its Data Link Interface, allowing the  
use of any of several network transport layers

Note: If the buffer supplied is too small, CX still contains the total number  
of active protocols and the buffer has been filled until full.

SeeAlso: AX=000Ch,AX=0014h

-----N-7C000C-----

INT 7C - SK-UPPS Data Link Interface API - GET PROTOCOL STATUS

AX = 000Ch

BX = protocol ID (see #03849,AX=000Bh)

ES:DI -> Protocol Status Block (see #03856)

Return: AX = completion code (see #03846)

SeeAlso: AX=000Bh,AX=0014h

Format of SK-UPPS Protocol Status Block (PSB):

Offset Size Description (Table 03856)

00h DWORD -> ASCIZ protocol name

04h WORD protocol mode (see #03850)

06h WORD protocol type (see AX=0007h)

08h WORD number of registered multicasts (see AX=0009h,AX=0014h)

-----N-7C000D-----

INT 7C - SK-UPPS Data Link Interface API - GET DRIVER STATISTICS

AX = 000Dh

BX = flag: 0=don't clear statistics, 1=clear statistics

ES:DI -> Statistics Data Block (see #03857)

Return: AX = completion code (see #03846)

SeeAlso: AX=0002h,AX=0015h

Format of SK-UPPS Statistics Data Block (SDB):

Offset Size Description (Table 03857)

00h DWORD node uptime in ticks (1/18.21 sec.)

04h DWORD bytes received

08h DWORD bytes transmitted

0Ch DWORD frames received

10h DWORD frames transmitted

14h DWORD multicast bytes received

18h DWORD multicast frames received

1Ch DWORD unrecognized frames

20h WORD missed frames

22h WORD frame check sequence (FCS) errors

24h WORD framing errors

26h WORD babbling errors

28h WORD late collision errors

2Ah WORD loss of carrier errors

2Ch WORD 16 retries failed errors

2Eh WORD overflow errors

Note: fields 26h through 2Ch are always zero, except in some very old versions of the DLI. Use GET MEDIA STATISTICS (AX=0015h) to get more detailed information.

-----N-7C000E-----

INT 7C - SK-UPPS Data Link Interface API - DLI 2.5+ - GET DRIVER INFO

AX = 000Eh

ES:DI -> Driver Information Block (see #03858)

BX = offset of 1st field within DIB to be returned

CX = number of bytes to be returned

Return: AX = completion code (see #03846)

CX = number of bytes copied (regardless of value returned in AX)

Program: SK-UPPS is the Schneider & Koch Universal Portable Protocol Stack, which provides an API on top of its Data Link Interface, allowing the use of any of several network transport layers

SeeAlso: AX=0000h,AX=0002h,AX=000Bh,AX=000Dh,AX=0015h

Format of SK-UPPS Driver Information Block (DIB):

```
Offset  Size  Description (Table 03858)
00h    6 BYTES physical node address (from adapter's ROM)
06h    6 BYTES current node address (see AX=0003h)
0Ch    BYTE   adapter id (defined by OEM)
0Dh    BYTE   IRQ line (FFh if none)
0Eh    DWORD  -> shared memory (zero if none)
12h    DWORD  size of shared memory (zero if none)
16h    WORD   base I/O port (zero if none)
18h    BYTE   DMA line (FFh if none)
19h    BYTE   DLI 2.4+ topology (see AX=0000h)
1Ah    WORD   size of an Mbuf's DATA area (see #03847)
1Ch    2 BYTES reserved (currently unused)
1Eh    WORD   total number of Mbufs the DLI has
20h    WORD   maximum number of Mbufs that can be allocated via REQUEST
        BUFFER (AX=0004h) and SYNC REQUEST BUFFER (AX=0012h)
---- end of Status Information Block returned by GET DRIVER STATUS (AX=0002h)
22h    DWORD  -> ASCIZ adapter name
26h    WORD   number of I/O ports used (zero if none)
28h    DWORD  line speed in bits/sec.
```

-----N-7C000F-----

INT 7C - SK-UPPS Data Link Interface API - FEED BACK BUFFER

AX = 000Fh

ES:BX -> Mbuf (see #03847)

Return: nothing

Notes: A protocol may call this function instead of returning the Mbuf directly from the Receive Upcall if it is running in "promiscuous", "hook" or "chain" mode.

SeeAlso: AX=0007h

-----N-7C0010-----

INT 7C - SK-UPPS Data Link Interface API - CLIENT HOOK

AX = 0010h

Return: nothing

Notes: An application having no active protocols may call this function to prevent the DLI from unloading.

SeeAlso: AX=0001h,AX=0011h

-----N-7C0011-----

INT 7C - SK-UPPS Data Link Interface API - CLIENT UNHOOK

AX = 0011h

Return: nothing



Program: SK-UPPS is the Schneider & Koch Universal Portable Protocol Stack,

which provides an API on top of its Data Link Interface, allowing the  
use of any of several network transport layers

SeeAlso: AX=0010h

-----N-7C0012-----

INT 7C - SK-UPPS Data Link Interface API - SYNC REQUEST BUFFER

AX = 0012h

DX = protocol ID (see #03849)

Return: AX = completion code (see #03846)

if AX=0000h: ES:BX -> Mbuf (see #03847)

Note: If the protocol wants to be informed when an Mbuf becomes available,

REQUEST BUFFER (AX=0004h) should be used instead.

SeeAlso: AX=0004h,AX=0005h,AX=0006h,AX=0013h

-----N-7C0013-----

INT 7C - SK-UPPS Data Link Interface API - TRANSMIT FRAME RELEASE

AX = 0013h

ES:BX -> Mbuf (see #03847)

Return: nothing

Notes: Returns immediately; The protocol's Transmit Upcall will NOT be called.

The Mbuf may no more be used by the protocol in any way (i.e. the

Mbuf's IN USE field may not be polled).

SeeAlso: AX=0004h,AX=0013h

-----N-7C0014-----

INT 7C - SK-UPPS Data Link Interface API - GET MULTICAST LIST

AX = 0014h

BX = protocol ID (see #03849,AX=000Bh)

ES:DI -> multicast address buffer

CX = number of 6-byte multicast addresses that may be stored in buffer

Return: AX = completion code (see #03846)

CX = number of registered multicast addresses for this protocol

(regardless of value returned in AX)

multicast address buffer filled with multicast addresses registered for  
this protocol

Note: If the buffer supplied is too small, CX still contains the total number

of registered multicast addresses for this protocol and the buffer

has been completely filled

SeeAlso: AX=000Bh,AX=0009h,AX=000Ch

-----N-7C0015-----

INT 7C - SK-UPPS Data Link Interface API - DLI 2.6+ - GET MEDIA STATISTICS

AX = 0015h

ES:DI -> media-specific statistics buffer (see #03859,#03860,#03861)

BX = offset of 1st field within MSS to be returned

CX = number of bytes to be returned

Return: AX = completion code (see #03846)

CX = number of bytes copied (regardless of value returned in AX)

Note: not supported by all DLIs (will return AX=FFFFh if not supported).

SeeAlso: AX=0000h,AX=000Dh

Format of SK-UPPS media-specific statistics for Ethernet:

Offset Size Description (Table 03859)

00h DWORD alignment errors  
 04h DWORD frame check sequence (FCS) errors  
 08h DWORD single collision frames  
 0Ch DWORD multiple collision frames  
 10h DWORD signal quality error (SQE) test errors  
 14h DWORD deferred transmissions  
 18h DWORD late collisions  
 1Ch DWORD excessive collisions  
 20h DWORD internal MAC transmit errors  
 24h DWORD carrier sense errors  
 28h DWORD excessive deferrals  
 2Ch DWORD frame too longs  
 30h DWORD in range length errors  
 34h DWORD out of range length fields  
 38h DWORD internal MAC receive errors

Format of SK-UPPS media-specific statistics for FDDI:

Offset Size Description (Table 03860)

00h DWORD SMT operating version id (refer to ANSI 7.1.2.2)  
 04h DWORD SMT CF state: 1=Isolated, 2=Wrap\_S, 3=Wrap\_A, 4=Wrap\_B,  
 5=Wrap\_AB, 6=Thru (refer to ANSI SMT 9.7.4.3)  
 08h DWORD SMT frames sent  
 0Ch DWORD SMT frames received  
 10h DWORD SMT ring up count  
 14h 6 BYTES MAC upstream neighbour  
 1Ah 6 BYTES MAC downstream neighbour  
 20h DWORD MAC frame counter (refer to ANSI MAC 2.2.1)  
 24h DWORD MAC error counter ( " )  
 28h DWORD MAC lost counter ( " )  
 2Ch BYTE port 1 link error estimate (ranges from 10\*\*<sup>-4</sup> to 10\*\*<sup>-15</sup> and  
 is reported as the absolute value of the exponent)  
 2Dh BYTE port 2 link error estimate ( " )

```

2Eh  2 BYTEs reserved (currently unused)
30h  DWORD attachment class: 1=single-attachment (S PORT),
      2=dual-attachment (A/B PORT pairs), 3=concentrator (M PORTs)
34h  DWORD attachment optical bypass present: 1=true, 2=false

```

Format of SK-UPPS media-specific statistics for Token-Ring:

Offset Size Description (Table 03861)

```

00h  6 BYTEs upstream neighbour
06h  WORD  local ring number
08h  DWORD ring up count
0Ch  DWORD signal loss errors
10h  DWORD lobe wire faults
14h  DWORD ring recovery count
18h  DWORD line errors
1Ch  DWORD burst errors
20h  DWORD ARI/FCI errors
24h  DWORD lost frame errors
28h  DWORD receive congestion errors
2Ch  DWORD frame copied errors
30h  DWORD token errors
34h  DWORD DMA bus errors
38h  DWORD DMA parity errors
3Ch  DWORD receive overflow errors

```

-----N-7C0016-----

INT 7C - SK-UPPS Data Link Interface API - DLI 2.8+ - DRIVER I/O CONTROL

AX = 0016h

DX = subfunction; bits 14-15 specify direction of I/O:

00 no data at all (ES:BX undefined)

01 driver -> application (GET)

10 application -> driver (SET)

11 both directions (GET/SET)

ES:BX -> I/O buffer (optional)

CX = size of buffer (if ES:BX valid)

Return: AX = completion code; defined by OEM

Program: SK-UPPS is the Schneider & Koch Universal Portable Protocol Stack,  
which provides an API on top of its Data Link Interface, allowing the  
use of any of several network transport layers

Notes: not supported by all DLIs (will return AX=FFFFh if not supported)

subfunction codes are defined by OEMs

SeeAlso: AX=0001h

-----N-7C0017-----

INT 7C - SK-UPPS Data Link Interface API - DLI 2.8+ - INTERRUPT REQUEST

AX = 0017h

Return: AX = completion code (see #03846)

Notes: not supported by all DLIs (will return AX=FFFFh, if not supported).

must be called with interrupts disabled; when the client re-enables  
interrupts, an interrupt will be generated and reported to the client  
through an Event Upcall (see #03853 [EV\_ISR\_START/EV\_ISR\_END])

SeeAlso: AX=0007h

-----t-7D-----

INT 7D O - [obsoleted proposal] - ALTERNATE MULTIPLEX INTERRUPT

Note: this interface has been moved to INT 2D; there are no known  
implementations on INT 7D

SeeAlso: INT 2D"AMIS",INT 2F"NOTES"

-----S-7D-----

INT 7D U - YTERM 1.4 - CLOCK SUPPORT

SeeAlso: INT 7E"YTERM"

-----E-7D-----

INT 7D - DJGPP GO32.EXE DOS EXTENDER - RELOCATED IRQ5

Notes: this vector is overwritten when GO32 starts but is not restored by  
early versions of the extender

the newest versions of GO32 dynamically allocate the vectors used  
for the relocated IRQs, much as DESQview does (see INT 50"DESQview")

SeeAlso: INT 0D"IRQ5",INT 7C"GO32",INT 7E"GO32"

-----U-7D-----

INT 7D - HyperPAD v2.2 - API ACCESS

Note: this vector is hooked but immediately passed on (without checking  
whether the previous handler was 0000h:0000h). The sole purpose of  
this vector is to provide the address of the data area described  
below (see #03862).

Format of HyperPAD data area:

Offset Size Description (Table 03862)

-16h DWORD pointer to ??? FAR function

-12h DWORD pointer to callback setting function

[C calling conventions, (\*callback)(int (\_loads far \*)()) ]

-0Eh 6 BYTES signature "BRC001"

-08h DWORD pointer to previous INT 7D handler

-04h DWORD pointer to ??? data

00h HyperPAD INT 7D handler

-----7D-----

INT 7D - IBM DOS 6.1 E.EXE - ???

Note: E.EXE checks whether this interrupt is in use (not 0000h:0000h), and if it is used, attempts to load E55VGA.EX instead of the default E.EX overlay. However, E55VGA.EX is not included in IBM DOS 6.1.

-----G-7D00-----

INT 7D - SCSILink - RESET SCSI BUS  
AH = 00h

Program: SCSILink is a TSR by Cross Products Ltd which allows its PC-hosted cross assemblers and similar products to communicate with Cross Products hardware debuggers

Desc: reset all devices on the SCSI bus

InstallCheck: test for the signature "SCSILINK" immediately prior to the interrupt handler

Index: installation check;SCSILink

-----G-7D01-----

INT 7D - SCSILink - CONNECT TO TARGET  
AH = 01h  
AL = target ID

Return: CF clear if successful

CF set on error

AL = initiator error

AH = target error

Desc: arbitrate the use of the bus and select the specified target device

-----G-7D02-----

INT 7D - SCSILink - SEND COMMAND  
AH = 02h

ES:BX -> parameter block (see #03863)

Return: CF clear if successful

CF set on error

AL = initiator error

AH = target error

Desc: send the specified command block to the target device and perform any related I/O

Format of SCSILink parameter block:

Offset Size Description (Table 03863)

00h DWORD size of command block

04h DWORD address of command block (see #03864)

08h DWORD size of buffer

0Ch DWORD address of buffer

Format of SCSILink command block:

Offset Size Description (Table 03864)

00h info not yet available

-----G-7D03-----

INT 7D - SCSILink - SET TIMEOUT

AH = 03h

AL = timeout selector (see #03865)

BX = new value in 55ms ticks

Desc: change an internal timeout value to allow communication with very slow targets

(Table 03865)

Values for SCSILink timeout selector:

00h time to wait for bus (default 18)

01h time to wait for new phase (default 5)

02h maximum time to send/receive block (default 18)

03h time to wait for reselect (default 180)

-----G-7D04-----

INT 7D - SCSILink - GET ERROR STRING

AH = 04h

AL = error number

Return: ES:BX -> ASCIZ error string for error number

-----G-7D05-----

INT 7D - SCSILink - GET ADDRESS

AH = 05h

Return: AL = DMA channel

AH = initiator ID

BX = card address

Desc: determine the current hardware configuration

-----G-7D06-----

INT 7D - SCSILink - PUT DATA

AH = 06h

CX = number of bytes to store (0001h-0100h)

ES:BX -> data to be saved

Return: CF clear if successful

CF set on error

AL = error code (01h = too much data)

Note: the specified data is stored in SCSILink's PSP

SeeAlso: AH=07h

-----G-7D07-----

INT 7D - SCSILink - GET DATA

AH = 07h  
CX = number of bytes to retrieve (0001h-0100h)  
ES:BX -> buffer for data

Return: CF clear if successful

CF set on error

AL = error code (01h = too much data)

Note: retrieve data previously stored with AH=06h

SeeAlso: AH=06h

-----G-7D08-----

INT 7D - SCSILink - TERMINATE SESSION

AH = 08h

Desc: indicate to any other programs that intercept INT 7D that the program has finished with the link

Note: the program should call this function even if a SCSI error caused its termination

-----G-7D09-----

INT 7D - SCSILink - POLL REQUEST

AH = 09h

AL = target ID

Return: AL = status

FEh resident driver experienced SCSI error

FFh resident driver handled event

Desc: give any drivers chained onto INT 7D a chance to handle an exception not specifically handled by the calling program

Notes: this service exists so that resident disk servers, etc. can continue running even while debuggers and profilers are active  
the resident driver assumes that a connection has been established and attempts to leave the target connected

-----\*-7E-----

INT 7E - RESERVED FOR DIP, Ltd. ROM LIBRARY

-----S-7E-----

INT 7E U - YTERM 1.4 - ???

SeeAlso: INT 7D"YTERM",INT 7F"YTERM"

-----E-7E-----

INT 7E - DJGPP GO32.EXE DOS EXTENDER - RELOCATED IRQ6

Program: GO32.EXE is a DOS extender included as part of the 80386 port of the GNU C/C++ compiler by DJ Delorie and distributed as DJGPP

Notes: this vector is overwritten when GO32 starts but is not restored by early versions of the extender

the newest versions of GO32 dynamically allocate the vectors used for the relocated IRQs, much as DESQview does (see INT 50"DESQview")

SeeAlso: INT 0E"IRQ6",INT 7D"GO32",INT 7F"GO32"

-----s-7E00-----

INT 7E - MaxSBOS v0.21 - GET ???

AH = 00h

Return: DX:AX -> ???

SeeAlso: AH=01h,AH=02h,AH=03h

-----s-7E0001-----

INT 7E - IWSBDRV v1.41 - SEND MIDI BYTE

AX = 0001h

DX = game device handle

BL = MIDI byte to send???

???

Return: EAX = status

00000000h if successful

FFFFFFFFh on error (function disabled)

Program: IWSBDRV is the resident portion of the InterWave SBOS sound card emulator for the Gravis UltraSound Plug-and-Play

Note: this function is only supported if the MIDISIMPLE device has been opened via the InterWave Game API (see INT 2F/AX=CD21h)

SeeAlso: AX=0002h,INT 2F/AX=CD21h,#03074

-----s-7E0002-----

INT 7E - IWSBDRV v1.41 - SEND MIDI STRING

AX = 0002h

DX = game device handle

ES:EDI -> buffer containing MIDI string to be sent

ECX = length of buffer in bytes

Return: EAX = status

00000000h if successful

FFFFFFFFh on error (function disabled)

Note: this function is only supported if the MIDISIMPLE device has been opened via the InterWave Game API (see INT 2F/AX=CD21h)

SeeAlso: AX=0001h,INT 2F/AX=CD21h,#03074

-----s-7E00F6-----

INT 7E - SBOS v3.82 - SET ???

AX = 00F6h

BL = ??? (max. 20h, value will be set to 20h if BL greater)

Return: AX=00F9h,AX=00FEh

-----s-7E00F7-----

INT 7E - SBOS v3.82 - GET ???

AX = 00F7h

Return: AX = status



```
    0000h failed
    0001h successful
BX = ??? (if nonzero, ??? is hooked) (see AX=00FFh)
CH = ???
CL = ???
SeeAlso: AX=00F6h,AX=00FDh,AX=00FFh
-----s-7E00F9-----
INT 7E - SBOS v3.82 - SET ???
    AX = 00F9h
    BX = ???
SeeAlso: AX=00F6h,AX=00FBh,AX=00FDh
-----s-7E00FB-----
INT 7E - SBOS v3.82 - SET ???
    AX = 00FBh
    BL = ???
SeeAlso: AX=00F6h,AX=00FDh,AX=00FFh
-----s-7E00FD-----
INT 7E - SBOS v3.82 - SET ???
    AX = 00FDh
    BL = ??? (max. 20h, value will be set to 20h if BL greater)
SeeAlso: AX=00F9h,AX=00FEh
-----s-7E00FE-----
INT 7E - SBOS v3.82 - UNINSTALL
    AX = 00FEh
Return: AX = status
    0000h successful
    00FCh unable to unhook INT 21
    00FDh unable to unhook INT 09
    00FEh unable to release memory
Program: SBOS is a SoundBlaster emulator for the Gravis UltraSound
InstallCheck: test for the ASCII signature "SBOS" at offset 0Ah in the
    interrupt handler's segment (similar to one of the possible EMS
    installation checks)
Range: SBOS may use INT 78-INT 7F
Note: this function may not be called if any SBOS vectors have been hooked
SeeAlso: INT 21/AX=FD12h,INT 78"UltraMID"
-----s-7E00FF-----
INT 7E - SBOS v3.82 - SET INTERRUPT FOR ???
    AX = 00FFh
    BL = new interrupt vector to hook
Return: BX = 03CDh if supported
```

Desc: unhooks the current interrupt handler for ???, and reconnects it to the  
specified new interrupt number

SeeAlso: AX=00F7h,AX=00F9h,AX=00FEh

-----s-7E01-----

INT 7E - MaxSBOS v0.21 - ???

AH = 01h

???

Return: ???

Program: MaxSBOS is a SoundBlaster FM synthesis emulator for the Gravis  
UltraSound MAX

Range: INT 78-INT 7F, selected by commandline parameter

Note: the installation check is the same as for 'regular' SBOS (see AX=00FEh)  
although the API in v0.21 does not provide the functions present in  
the older SBOS

SeeAlso: AX=00FEh,AH=02h,AH=03h

-----s-7E02-----

INT 7E - MaxSBOS v0.21 - ???

AH = 02h

Return: AX = FFFFh

SeeAlso: AX=00FEh,AH=01h,AH=03h

-----s-7E03-----

INT 7E - MaxSBOS v0.21 - NOP

AH = 03h

Return: nothing

SeeAlso: AX=00FEh,AH=01h,AH=02h

-----m-7E5857BL57-----

INT 7E - XLOAD - INSTALLATION CHECK

AX = 5857h

BL = 57h

Return: AX = 0000h if installed

Program: XLOAD is the LOADHIGH-equivalent from Helix Software's Netroom

SeeAlso: AX=5857h/BL=5Ah,AX=5857h/BL=5Bh

-----m-7E5857BL5A-----

INT 7E - XLOAD - GET ???

AX = 5857h

BL = 5Ah

Return: AX = 0000h

CX = ???

DS:SI -> ???

SeeAlso: AX=5857h/BL=57h,AX=5857h/BL=5Bh

-----m-7E5857BL5B-----

INT 7E - XLOAD - GET XLOAD MEMORY SIZE

AX = 5857h

BL = 5Bh

Return: AX = 0000h

CX = segment of XLOAD TSR

DX = size of memory block in which XLOAD TSR is located

SeeAlso: AX=5857h/BL=57h,AX=5857h/BL=5Ah

-----s-7E--80-----

INT 7E - IWSBDRV v1.41 - GET ???

AL = 80h

Return: DX:AX -> ???

SeeAlso: AL=81h,AL=82h,AL=83h,AL=84h,AL=85h,AL=86h,AL=87h

-----s-7E--81-----

INT 7E - IWSBDRV v1.41 - ???

AL = 81h

???

Return: ???

SeeAlso: AL=80h,AL=82h,AL=83h,AL=84h,AL=85h,AL=86h,AL=87h

-----s-7E--82-----

INT 7E - IWSBDRV v1.41 - GET ???

AL = 82h

Return: AX = ??? (FFFFh)

SeeAlso: AL=80h,AL=81h,AL=83h,AL=84h,AL=85h,AL=86h,AL=87h

-----s-7E--83-----

INT 7E - IWSBDRV v1.41 - ??? (CALLS VIWD.VXD)

AL = 83h

???

Return: ???

SeeAlso: AL=80h,AL=81h,AL=82h,AL=84h,AL=85h,AL=86h,AL=87h

SeeAlso: INT 2F/AX=1684h/BX=38DAh"VIWD"

-----s-7E--84-----

INT 7E - IWSBDRV v1.41 - WAKE PROGRAM???

AL = 84h

Return: AL = status

00h successful

else failed

SeeAlso: AL=80h,AL=81h,AL=82h,AL=83h,AL=85h,AL=86h,AL=87h

SeeAlso: INT 2F/AX=CD04h"InterWave"

-----s-7E--85-----

INT 7E - IWSBDRV v1.41 - SET ??? FLAG

AL = 85h

Return: nothing

SeeAlso: AL=80h,AL=81h,AL=82h,AL=83h,AL=84h,AL=86h,AL=87h

-----s-7E--86-----

INT 7E - IWSBDRV v1.41 - HOOK INT 21

AL = 86h

Return: AX,BX,DX,ES destroyed

SeeAlso: AL=80h,AL=81h,AL=82h,AL=83h,AL=84h,AL=85h,AL=87h

-----s-7E--87-----

INT 7E - IWSBDRV v1.41 - RESTORE INT 21

AL = 87h

Return: AX,DX destroyed

SeeAlso: AL=80h,AL=81h,AL=82h,AL=83h,AL=84h,AL=85h,AL=86h

-----V-7F-----

INT 7F - Halo88 - API

BX = function number (see #03866)

additional parameters on stack

Return: ???

Program: Halo88 is a suite of graphics routines

Note: according to Stuart Kemp, the code appears to make no provisions for chaining

(Table 03866)

Values for Halo88 API function:

64h arc  
65h bar  
66h box  
67h circle  
68h clr  
69h default hatch style  
6Ah default line style  
6Bh delhcur  
6Ch delln / deltc  
6Dh ellipse  
6Eh fill  
6Fh flood  
70h flood2  
71h init graphics  
72h init hcur  
73h init marker  
74h init tc  
75h inqarc

76h inqbknd  
77h inqclr  
78h inqerr  
79h inqgcur  
7Ah inqhcur  
7Bh inqmarker  
7Dh inqtcurl  
7Eh inqtext  
7Fh lnabs  
80h lnrel  
81h markerabs  
82h markerrel  
83h moveabs  
84h movehcurabs  
85h movehcurrel  
86h moverel  
87h movetcurabs  
88h movetcurrel  
89h movefrom  
8Ah moveto  
8Bh pie  
8Ch polylnabs  
8Dh polylnrel  
8Eh ptabs  
8Fh ptrel  
91h setasp  
92h set color  
93h set font  
94h set hatch style  
95h set line style  
97h setttext  
98h set text color  
99h btext  
9Ah setseg  
9Bh display  
9Ch setscreen  
9Eh close graphics  
9Fh ftinit  
A0h ftlocate  
A1h ftext  
A5h set viewport

A6h set window  
A7h set world  
AAh ftcolor  
ACh initlp  
ADh inqasp  
AEh inqdev  
AFh inqdisplay  
E0h inqft  
B1h inqftcolor  
B2h inqinterlace  
B3h inqlpa  
B4h inqlpg  
B5h inqmode  
B6h inqscreen  
B7h inqverson  
B8h roam  
B9h scroll  
BAh setieeee  
BBh set interlace  
BCh shift  
BDh start graphics  
BEh vpan  
CBh gwrite  
CCh gread  
CDh setxor  
CEh rbox  
CFh rcir  
D0h rlnabs  
D1h rlnrel  
D2h delbox  
D3h delcir  
D5h setseg2  
DCh worldoff  
DDh mapwtod  
DEh mapdtow  
DFh mapwton  
E0h mapntow  
E1h mapdton  
E2h mapntod  
E3h inqworld  
E4h inqviewport

E5h set line width  
E6h lnjoint  
E7h set locator  
E8h read locator  
E9h setdev  
EBh setstext  
ECh setstclr  
EDh setstang  
EEh stext  
EFh inqstext  
F0h setdegree  
F1h inqstsize  
F2h polyfabs  
F3h polyfrel  
F4h inqdrange  
F5h inqstang  
F6h orglocator  
F7h inqlocator  
F8h inqarea  
F9h setipal  
FAh setborder  
FBh inqcrange  
FEh setclip  
FFh fcir  
100h setcrange  
101h setdrange  
102h setlatr  
103h polycabs  
104h polycrel  
108h memcom  
109h memexp  
10Ah memmov  
10Eh movefx  
10Fh movetx  
110h inqrgb  
111h save image  
112h restore image  
113h setapal  
114h setxpal  
118h inqtsize  
12Eh gprint

130h setprn  
131h setpatrr  
133h setbatrr  
135h pexpand  
136h ptnorm  
137h pfnorm  
13Bh inqprn  
13Ch lopen  
13Dh lclose  
13Eh lappend  
13Fh lrecord  
140h lswitch  
142h inqfun  
15Dh lsetup  
15Eh lrest  
15Fh lsave

-----N-7F-----

INT 7F - CONVERGENT TECHNOLOGIES ClusterShare CTOS ACCESS VECTOR

AL = request ID

01h "Request"/"RequestDirect"

ES:BX -> pRq

DX ignored

04h "Wait"

ES:BX -> ppMsgRet

DX = exchange

05h "AllocExch"

ES:BX -> pExchRet

06h "DeAllocExch"

DX = exchange

07h "Check"

ES:BX -> ppMsgRet

DX = exchange

CX = 4354h ('CT')

Return: AX = status

0000h successful

-----S-7F-----

INT 7F - Telebit ACS SERIAL I/O

ES:SI-> parameter block (see #03867)

Return: CF set on error

CF clear on success

InstallCheck: test for the signature "PDGATEWRKSTNIF" just prior to the



interrupt handler

Index: installation check;Telebit ACS Serial I/O

Format of Telebit ACS parameter block:

Offset Size Description (Table 03867)

00h	BYTE	command (see #03868)
01h	BYTE	gateway number
02h	BYTE	reserved
03h	BYTE	port
04h	17 BYTES	auxiliary buffer
15h	BYTE	session
16h	WORD	count of bytes passed to API
18h	DWORD	buffer pointer passed to/from API
1Ch	WORD	count of bytes passed from API
1Eh	BYTE	return code (see #03869)

(Table 03868)

Values for Telebit ACS command:

3Ch	status
3Dh	connect
3Eh	disconnect
3Fh	read
40h	data/command write
41h	clear receive buffer
42h	get configuration
43h	get receiver status
44h	raw write
45h	search servers
46h	set transmit buffer size

(Table 03869)

Values for Telebit ACS return code:

00h	success
01h	invalid session
05h	servername invalid
06h	NetWare fileserver bindery is locked
07h	communication server not active
08h	general failure in NetWare fileserver
09h	not logged into a fileserver
10h	connection table full
11h	no response from communication server

12h connection attempt terminated abnormally  
13h connection refused - no sessions available  
14h gateway number/port already in use  
15h invalid connection response  
16h port invalid  
17h incorrect version in server response  
18h gateway number/port combination not configured  
19h initialization has not been completed  
20h no more sockets are available  
21h no active poolname  
23h FATAL internal interface error  
24h registration of host workstation failed - name is already in use  
25h registration of host workstation failed - workstation name table full  
26h registration of host workstation failed - only one session may be  
registered for dial-in  
FFh Telebit ACS API is busy - retry later

-----N-7F-----

INT 7F - Non-dedicated NetWare 2.x File Server - ENTER CONSOLE MODE  
InstallCheck: test for the signature "Lynn" in the four bytes preceding the  
interrupt handler; if present, the current program is running as a  
DOS task on a non-dedicated NetWare 2.x file server.

Note: Before placing the server into "console" mode, it is recommended that  
NetWare broadcast messages be disabled with INT 21/AH=DEh/DL=00h.

SeeAlso: INT 21/AH=DEh/DL=04h

Index: installation check;non-dedicated NetWare server

-----S-7F-----

INT 7F U - YTERM - ???

SeeAlso: INT 7E"YTERM"

-----E-7F-----

INT 7F - DJGPP GO32.EXE DOS EXTENDER - RELOCATED IRQ7

Notes: this vector is overwritten when GO32 starts but is not restored by  
early versions of the DOS extender

the newest versions of GO32 dynamically allocate the vectors used  
for the relocated IRQs, much as DESQview does (see INT 50"DESQview")

SeeAlso: INT 0F"IRQ7",INT 7E"GO32"

-----7F-----

INT 7F - Canon IXHND2 Scanner Interface

-----U-7F-----

INT 7F - SBS WinRun 1.00 - TRANSMITTER ENTRY POINT

DS:SI -> WinRun control block (WCB) (see #03870)

Return: WinRun receiver status returned in WCB

Program: WinRun is a DOS/Windows utility written by Sven B. Schreiber to start

Windows applications from a virtual DOS machine under Windows

Note: The WinRun transmitter (WINRUN-T.COM) and receiver (WINRUN-R.EXE) need to be up and running

Format of the WinRun control block (WCB):

Offset Size Description (Table 03870)

00h DWORD pointer to ASCIZ command string

04h WORD (ret) WinRun receiver status

0000h-001Fh WinExec() error

0020h-FFFFh WinExec() instance handle

FFFFh Windows not running or WinRun receiver not installed

-----N-7F-----

INT 7F - Alloy 386/MultiWare (MW386), Novell-Type Network Executive (NTNX)

Notes: the words at C800h:0000h and C800h:0002h will both be 584Eh if the MW386 multitasking system is present (i.e. signature "NXNX")

NTNX allows its API to be placed on a different interrupt than 7Fh at load time. To determine the actual vector used, open the device "SPOOLER" with INT 21/AX=3D02h, place it in RAW mode with INT 21/AX=4400h and INT 21/AX=4401h, then read one byte which will be the actual interrupt number being used; the other interrupts may be found with INT 7F/AH=09h/CL=03h

-----N-7F-----

INT 7F - Alloy NetWare Support Kit (ANSK) v2.2+ - INSTALLATION CHECK

Note: a program may determine that it is running on an ANSK Slave by checking the five bytes at F000h:0000h for the ASCIZ signature "ANSK"; this address is RAM, and should not be written. However, the above check will not work on Slaves with <1MB RAM or those using the SLIM.SYS device driver

-----N-7F00-----

INT 7F - Alloy NTNX, MW386 - SEMAPHORE LOCK AND WAIT

AH = 00h

DS:DX -> ASCIZ semaphore name (max 64 bytes)

Return: AL = status (see #03871)

AH = semaphore owner if status=02h

SeeAlso: AH=01h,AH=02h,AH=41h,INT 67/AH=00h

(Table 03871)

Values for Alloy function status:

00h successful

01h invalid function  
02h semaphore already locked  
03h unable to lock/unlock semaphore  
04h semaphore space exhausted  
05h host/target PC did not respond (NTNX)

-----T-7F00-----

INT 7F - MultiLink Advanced v1.0+ - ENQUEUE SYSTEM RESOURCE

AH = 00h  
BX = resource identifier  
AL = wait flag

Return: AL = status

00h successful  
01h resource not available  
02h user error

InstallCheck: ensure that the interrupt vector is not pointing at segment  
0000h, then test whether the byte at offset 0000h in the interrupt  
handler's segment is E9h

Notes: function will not return until the resource is available if AL is  
nonzero on entry

a maximum of 100 resources may be enqueued at once

SeeAlso: AH=01h"MultiLink"

-----N-7F00-----

INT 7F - G8BPQ v4.07+ - GET NODE/SWITCH VERSION AND DESCRIPTION

AH = 00h  
ES:SI -> buffer for "USERS" text string

Return: AX = 4250h ('BP') if installed

BX = 5120h ('Q ') if installed  
DX = version number (DH = major, DL = minor)  
CX = length of returned string

Program: the G8BPQ AX25 Networking Package is amateur packet radio software by  
John Wiseman which allows a PC to act as a node in an AX.25 network

SeeAlso: AH=01h"G8BPQ",AH=09h"G8BPQ"

-----N-7F01-----

INT 7F - Alloy NTNX, MW386 - SEMAPHORE LOCK

AH = 01h  
DS:DX -> ASCIZ semaphore name (max 64 bytes)

Return: AL = status (see #03871)

AH = semaphore owner if status=02h

SeeAlso: AH=00h,AH=02h,AH=41h

-----T-7F01-----

INT 7F - MultiLink Advanced v1.0+ - DEQUEUE SYSTEM RESOURCE

AH = 01h  
BX = resource identifier

Return: AL = status  
00h successful  
02h user error

Note: the indicated resource may be dequeued even if it was enqueued by another task

SeeAlso: AH=00h"MultiLink",AH=02h"MultiLink"

-----V-7F01-----

INT 7F - TIGA Communications Driver v2.05 - GET ENTRY POINTS

AH = 01h  
Return: BX = 1234h if installed  
DX:AX -> array of FAR entry points  
CH = driver major version  
CL = driver minor version

Note: TIGACD 2.05 returns CF set on unrecognized functions in AX

SeeAlso: AX=1234h,AX=4321h

-----N-7F01-----

INT 7F - G8BPQ v4.00+ - HOST MODE - SET APPLICATION FLAGS AND MASK

AH = 01h  
AL = stream number (01h-40h)  
CL = new application flags  
bit 7: monitored frames available via AH=0Bh"G8BPQ"  
DL = new application mask

Program: the G8BPQ AX25 Networking Package is amateur packet radio software by John Wiseman which allows a PC to act as a node in an AX.25 network

Range: INT 00h to INT FFh, set in configuration file BPQCFG.TXT for v4.03+ (earlier versions were hard-wired for INT 7F)

SeeAlso: AH=00h"G8BPQ",AH=02h"G8BPQ",AH=0Bh"G8BPQ"

-----I-7F0104BX0000-----

INT 7F - HLLAPI (IBM 3270 High-Level Language API)/LLAPI (Rabbit Low Level API)

AX = 0104h (HLLAPI gate ID)  
BX = 0000h  
DS:SI -> parameter control block (see #03872)

Return: parameter control block updated

InstallCheck: for the Novell HLLAPI TSR, test for the signature "CXI" (for the company Novell bought) immediately prior to the interrupt handler

SeeAlso: AX=0105h,AX=ABCDh

Format of HLLAPI parameter control block:

Offset Size Description (Table 03872)

00h 3 BYTEs signature = 'PCB'  
03h BYTE function number (see #03873,#03874)  
04h WORD segment of control string  
06h WORD offset of control string  
08h WORD length of control string, unless explicit end-of-str char set  
0Ah BYTE unused (IBM)  
ControlString[0] (Rabbit)  
0Bh WORD return code (see #03876)  
0Dh WORD maximum length of control string (IBM)  
unused (Rabbit)

(Table 03873)

Values for HLLAPI function number:

00h OEM function (Query system for Attachmate implementation)  
01h Connect presentation space  
02h Disconnect presentation space  
03h Send string of keystrokes as if typed from keyboard  
04h Wait ~60s, returns status of presentation space  
05h Copy current presentation space into a user-defined buffer  
06h Search presentation space for first occurrence of a specified string  
07h Query cursor location in current presentation space  
08h Copy part or all of current presentation space into user buffer  
09h Set session parameters; parameters vary by vendor (see #03875)  
0Ah Get info on sessions currently connected  
0Bh Lock current presentation space  
0Ch Unlock previously locked presentation space  
0Dh Return copy of operator info area (OIA) of current presentation space  
0Eh get attribute byte for given position in the current presentation space  
0Fh copy string of characters to the current presentation space  
10h workstation control functions  
11h storage manager functions, intended primarily for BASIC applications  
(not implemented by Rabbit)  
12h set delay period in half-second intervals  
14h get info on level of workstation support used  
15h reset session parameters to default values  
16h get detailed info on the current session  
17h start host notification to application on presentation sp or OIA update  
18h check host update when host notification enabled  
19h stop host notification  
1Eh search field within current presentation space for string  
1Fh get first position of a selected field in the current presentation space

20h get length of specified field  
21h copy string into a specified field  
22h copy specified field into a user-defined buffer  
23h create alternate presentation space (IBM only), don't use with BASIC  
24h switch to alternate presentation space (IBM only), not with BASIC  
25h display cursor in specified area (IBM only), don't use with BASIC  
26h display alternate presentation space (IBM only), don't use with BASIC  
27h delete alternate presentation space (IBM only), don't use with BASIC  
28h set cursor  
29h start Close Intercept  
2Ah query Close Intercept  
2Bh stop Close Intercept  
32h start intercepting keystrokes to allow filtering  
33h get keystrokes after turning on interception  
34h notify operator when keystroke rejected by filter subroutine  
35h stop intercepting keystrokes  
5Ah send file  
5Bh receive file  
5Ch run a program (not implemented by Rabbit)  
5Dh execute DOS command (not implemented by Rabbit)  
63h change presentation space position to PC display row/col or vice versa  
65h connect to Window Services  
66h disconnect from Window Services  
67h set/query window coordinates  
68h set/query window status  
69h change presentation space name  
78h connect Structured Fields  
79h disconnect Structured Fields  
7Ah query size of communications buffer  
7Bh allocate communications buffer  
7Ch free communications buffer  
7Dh get request completion state  
7Eh read Structured Fields  
7Fh write Structured Fields  
FFh Get info on DCA implementation

(Table 03874)

Values for LLAPI function number:

80h initialize LLAPI (internal call)  
83h set Session ID (one-character ID)  
84h read Session ID (one-character ID)

85h lock 327x keyboard  
86h unlock 327x keyboard  
87h wait for Clear to Send  
88h type ASCII character  
89h type 327x key  
8Ah read keyboard lock state  
8Fh force screen update  
90h view session  
91h relinquish (suspend foreground until background becomes idle)  
92h poke screen character  
93h poke translated character  
94h peek screen character  
95h peek translated character  
96h set cursor position  
97h send scan code (Rabbit only)  
98h synchronize (returns after keystroke queue empty)  
99h type PC key (Rabbit only)

(Table 03875)

Values for HLLAPI Function 09h Session Parameters:

ASCII ??? (Rabbit only)  
ATTRIB return attributes in hex  
NOATTRIB return attributes as blanks  
CONPHYS make physical connection  
CONLOG only make logical connection  
EAB copy extended attribute bytes along with data  
NOEAB copy data only  
ESC=n set escape character to "n" (default '@')  
EOT=n set end of string character (default 00h)  
FPAUSE full-duration pause  
FTNOWAIT return immediately from functions 5Ah and 5Bh (Rabbit only)  
FTWAIT wait for file transfer to complete (Rabbit only)  
IPAUSE interruptable pause  
RABESC ??? (Rabbit only)  
NORABESC ??? (Rabbit only)  
SCANCODE ??? (Rabbit only)  
STRLEN use explicit string lengths  
STREOT use terminated strings  
SRCHALL search entire presentation space  
SRCHFROM search from specified offset  
SRCHFRWD search forward from position 1



SRCHBKWD search backward from last position in presentation space  
TIMEOUT=n ??? (Rabbit only)  
TWAIT wait specified time for keyboard ready  
LWAIT wait until keyboard ready  
NWAIT no wait  
TRON enable tracing  
TROFF disable tracing  
AUTORESET send reset before sending keys with function 03h  
NORESET don't send reset  
QUIET don't display messages sent with INT 21/AH=09h  
NOQUIET allow messages to be displayed  
TIMEOUT=n set timeout in 30-second intervals, 0 = wait until ^Break  
XLATE translate extended attribute bytes  
NOXLATE don't translate  
NEWRET use HLLAPI v3.0 return code conventions  
OLDRET use HLLAPI v2.0 return code conventions

(Table 03876)

Values for Windows HLLAPI return code:

00h successful  
01h Presentation Space not connected/requested size unavailable  
02h invalid function or parameter error/invalid block ID  
03h file transfer complete  
04h file transfer complete (segmented)/Presentation Space busy  
05h inhibited or keyboard locked  
06h data truncated  
07h invalid Presentation Space position  
08h operation not available  
09h system error  
0Ah blocking error  
0Bh resource not available  
0Ch session stopped  
14h undefined key combination  
15h OIA updated  
16h Presentation Space updated  
17h both Presentation Space and OIA updated  
18h no such field  
19h no keystrokes available  
1Ah Presentation Space or Operator Information Area changed  
1Bh file transfer aborted  
1Ch zero-length field

1Eh cursor type invalid  
1Fh keystroke overflow  
20h another application is already connected  
22h message sent to host cancelled  
23h transmission from host cancelled  
24h lost contact with host  
25h function successful  
26h function incomplete  
27h a DDM session is already connected  
28h disconnected, but asynchronous requests still pending  
29h buffer already in use  
2Ah no matching request found  
12Dh invalid function number  
12Eh file not found  
131h access denied  
134h out of memory  
136h environment invalid  
137h format invalid  
270Eh (9998) invalid Presentation Space ID  
270Fh (9999) invalid row or column code

---Windows HLLAPI extensions---

F000h asynchronous call already in progress  
F001h invalid asynchronous task ID  
F002h blocking call cancelled  
F003h underlying subsystem not started  
F004h unsupported application version

-----V-7F0105-----

INT 7F - IBM 8514/A Adapter Interface (HDILOAD.EXE)

AX = 0105h

Return: CF set on error

CF clear if successful

CX:DX -> array of FAR pointers to entry points (see #03877)

Note: most functions are invoked by pushing the DWORD parameter block pointer  
and then performing a FAR call via the appropriate vector of the  
entry point array, placing the FAR address of the function's  
parameter block on the top of the stack

SeeAlso: AX=0104h,AX=0106h,AX=ABCDh

(Table 03877)

Values for HDILOAD function number: (do FAR call via entry\_points+4\*function)

08h HOPEN Open Adapter (see #03909)

09h HSMX Set Mix (see #03938)  
10h HINT Interrupt (see #03905)  
13h HLDPAL Load Palette (see #03906)  
15h HBBW BitBLT Write Image Data (see #03892)  
17h HBBR BitBLT Read Image Data (see #03893)  
18h HBBCHN Chained Data Go (see #03891)  
19h HBBC BitBLT Copy Data (see #03890)  
1Dh HQMODE Get Current Mode (see #03915)  
20h HRECT Fill Rectangle (see #03917)  
22h HCLOSE Close Adapter and place in quiescent state (see #03898)  
30h HINIT Initialize State (see #03904)  
31h HSYNC Synchronize Adapter (see #03943)  
39h HSPAL Save Palette (see #03940)  
3Ah HRPAL Restore Palette (see #03920)  
??? HSGQ Set Graphics Quality (see #03929)  
??? HSCoord Set Coordinate Type (see #03926)  
??? HESC Escape, Terminate Processing (see #03903)  
??? HSBCOL Set Background Color (see #03921)  
??? HSBP Set Bit Plane Controls (see #03922)  
??? HSCMP Set Color Comparison Register (see #03924)  
??? HSCOL Set Color (see #03925)  
??? HSHS Set Scissors (clipping rectangle) (see #03930)  
??? HXLATE Assign Text Color (see #03944)  
??? HQMODES Get Adapter Modes (see #03916)  
??? HQDPS Get Drawing Process State Size (see #03914)  
??? HQDFPAL Get Default Palette (see #03913)  
??? HQCOORD Get Coordinate Type (see #03911)  
??? HQCP Get Current Position (see #03912)  
??? HSMODE Change Mode (see #03936)  
??? HLINE Draw Line at Given Position (see #03907)  
??? HCLINE Draw Line at Current Position (see #03897)  
??? HRLINE Draw Line Relative from Given Position (see #03918)  
??? HCRLINE Draw Line Relative at Current Position (see #03900)  
??? HSLT Set Line Type (see #03933)  
??? HSLW Set Line Width (see #03935)  
??? HSLPC Save Line Pattern Count (see #03932)  
??? HRLPC Restore Line Pattern Count (see #03919)  
??? HCBW BitBLT Write Image Data at Current Position (see #03894)  
??? HBAR Begin Area (see #03889)  
??? HEAR End Area (see #03901)  
??? HSPATT Set Pattern (see #03941)

??? HSPATTO Set Pattern Reference Point (see #03942)  
 ??? HEGS Erase Graphics Screen (see #03902)  
 ??? HSCP Set Current Position (see #03927)  
 ??? HMRK Set Marker (see #03908)  
 ??? HCMRK Set Marker at Current Position (see #03899)  
 ??? HSMARK Set Marker Shape (see #03937)  
 ??? HSCS Set Character Set (see #03928)  
 ??? HCHST Write Character String at Given Position (see #03896)  
 ??? HCCHST Write Character String at Current Position (see #03895)  
 ??? HSCELL Set Cell Size for Alphanumeric Text (see #03923)  
 ??? ABLOCKMFI Write Character/Attribute Block MFI (see #03880)  
 ??? ABLOCKCGA Write Character Block CGA (see #03878)  
 ??? AERASE Erase Rectangle (see #03883)  
 ??? ASCROLL Scroll Rectangle (see #03884)  
 ??? ACURSOR Set Alphanumerics Cursor Position (see #03882)  
 ??? ASCUR Set Alphanumeric Cursor Shape (see #03885)  
 ??? ASFONT Set Font (see #03886)  
 ??? AXLATE Assign Alphanumeric Color (see #03888)

Format of ABLOCKCGA parameter block:

Offset	Size	Description (Table 03878)
00h	WORD	000Ah (length of following data)
02h	WORD	top-left coordinate of character block
04h	WORD	width of block
06h	DWORD	-> block of WORDs describing characters (see #03879)
0Ah	BYTE	length of block in characters
0Bh	BYTE	highlight attribute
		bit 4: transparent background
		bit 5: overstrike
		bit 6: reverse video
		bit 7: underscore

SeeAlso: #03880

Bitfields for one character in ABLOCKCGA character block:

Bit(s)	Description (Table 03879)
3-0	foreground attribute
7-4	background attribute
15-8	character code

Format of ABLOCKMFI parameter block:

Offset	Size	Description (Table 03880)
--------	------	---------------------------

00h WORD 0009h (length of following data)  
02h WORD top-left coordinate of character block  
04h WORD width of block  
06h DWORD -> block of DWORDs describing characters (see #03881)  
0Ah BYTE length of block in characters

SeeAlso: #03878

Bitfields for one character in ABLOCKMFI character block:

Bit(s) Description (Table 03881)  
7-0 reserved  
9-8 low two bits of font number  
12 transparent background  
13 overstrike  
14 reverse video  
15 underscore  
19-16 foreground color attribute  
23-20 background color attribute  
31-24 character code

Format of ACURSOR parameter block:

Offset Size Description (Table 03882)  
00h WORD 0002h (length of following data)  
02h BYTE column (0-based)  
03h BYTE row (0-based)

SeeAlso: #03885

Format of AERASE parameter block:

Offset Size Description (Table 03883)  
00h WORD 0005h (length of following data)  
02h BYTE left-most column (0-based)  
03h BYTE top-most row (0-based)  
04h BYTE rectangle's width in character cells  
05h BYTE rectangle's height in character cells  
06h BYTE background color (bits 7-4)

SeeAlso: #03884

Format of ASCROLL parameter block:

Offset Size Description (Table 03884)  
00h WORD 0006h (length of following data)  
02h BYTE left-most column (0-based) of source  
03h BYTE top-most row (0-based) of source

04h BYTE rectangle's width in character cells  
 05h BYTE rectangle's height in character cells  
 06h BYTE left-most column (0-based) of destination  
 07h BYTE top-most row (0-based) of destination

SeeAlso: #03883

Format of ASCUR parameter block:

Offset Size Description (Table 03885)

00h WORD 0003h (length of following data)  
 02h BYTE cursor start line (00h = top of cell, FFh = keep current shape)  
 03h BYTE cursor stop line  
 04h BYTE cursor attribute  
     00h normal  
     01h hidden  
     02h left-arrow (requires start = 2 and stop = bottom of cell)  
     03h right-arrow (requires start = 2 and stop = bottom of cell)

Notes: no cursor is shown if the start line is greater than the stop line  
 the alphanumeric cursor is hidden after each mode change

SeeAlso: #03882, INT 10/AH=01h

Format of ASFONT parameter block:

Offset Size Description (Table 03886)

00h WORD 0005h (length of following data)  
 02h BYTE font number (0-3)  
 03h DWORD -> character set definition block (see #03887)

Format of 8514/A character set definition block:

Offset Size Description (Table 03887)

00h BYTE reserved  
 01h BYTE type of character set  
     00h bitmapped, 01h&02h reserved, 03h short-stroke font  
 02h BYTE reserved  
 03h DWORD reserved  
 07h BYTE cell width in pixels  
 08h BYTE cell height in pixels  
 09h BYTE reserved  
 0Ah WORD cell size in bytes  
 0Ch WORD flags  
     bit 15: reserved (0)  
     bit 14: color bitmap  
     bit 13: proportional spacing

0Eh DWORD -> index table  
12h DWORD -> character width table  
16h BYTE initial code point  
17h BYTE final code point  
18h DWORD -> character definition table  
1Ch WORD reserved  
1Eh DWORD -> second character definition table  
22h WORD reserved  
24h DWORD -> third character definition table

## Format of AXLATE parameter block:

Offset	Size	Description (Table 03888)
00h	WORD	0080h (length of following data)
02h	64 BYTES	character foreground translation table
42h	64 BYTES	character background translation table

## Format of HBAR parameter block:

Offset	Size	Description (Table 03889)
00h	WORD	0000h (no data following)

SeeAlso: #03901

## Format of HBBC parameter block:

Offset	Size	Description (Table 03890)
00h	WORD	0010h (length of following data)
02h	WORD	data format 0000h across-the-plane copy (color expansion) 0008h through-the-plane copy
04h	WORD	data rectangle's width
06h	WORD	data rectangle's height
08h	BYTE	source bit plane number (across-the-plane copies only)
09h	BYTE	reserved
0Ah	2 WORDs	X,Y coordinates of source's upper-left corner in display memory
0Eh	2 WORDs	X,Y coordinates of destination's upper-left corner in display memory

Note: copies data from one location in video memory to another

SeeAlso: #03891

## Format of HBBCHN parameter block:

Offset	Size	Description (Table 03891)
00h	WORD	0006h (length of following data)
02h	DWORD	-> data buffer in system memory

06h WORD number of bytes to transfer

Note: this function performs the actual data transfer for a bitBLT set up

with HBBR, HBBW, or HCBBW

SeeAlso: #03890,#03892,#03893,#03894

Format of HBBR parameter block:

Offset Size Description (Table 03892)

00h WORD 000Ch or 0014h (length of following data)

02h WORD data format

0000h across-the-plane copy (color expansion)

0008h through-the-plane copy

04h WORD data rectangle's width

06h WORD data rectangle's height

08h BYTE source bit plane number

09h BYTE reserved

0Ah 2 WORDs X,Y coordinates of destination's upper-left corner in display memory

---optional---

0Eh WORD sub-rectangle left margin in pixels

10h WORD sub-rectangle top margin in pixels

12h WORD sub-rectangle width

14h WORD sub-rectangle height

SeeAlso: #03891,#03892,#03894

Format of HBBW parameter block:

Offset Size Description (Table 03893)

00h WORD 000Ah or 0012h (length of following data)

02h WORD data format

0000h across-the-plane copy (color expansion)

0008h through-the-plane copy

04h WORD data rectangle's width

06h WORD data rectangle's height

08h 2 WORDs X,Y coordinates of destination in display memory

---optional---

0Ch WORD sub-rectangle left margin in pixels

0Eh WORD sub-rectangle top margin in pixels

10h WORD sub-rectangle width

12h WORD sub-rectangle height

SeeAlso: #03891,#03893,#03894

Format of HCBBW parameter block:



Offset Size Description (Table 03894)  
00h WORD 0006h or 000Eh (length of following data)  
02h WORD data format  
    0000h across-the-plane copy (color expansion)  
    0008h through-the-plane copy  
04h WORD data rectangle's width  
06h WORD data rectangle's height  
---optional---  
08h WORD sub-rectangle left margin in pixels  
0Ah WORD sub-rectangle top margin in pixels  
0Ch WORD sub-rectangle width  
0Eh WORD sub-rectangle height  
SeeAlso: #03891,#03892,#03893

Format of HCCHST parameter block:

Offset Size Description (Table 03895)  
00h WORD length of following data  
02h N BYTES ASCII string to display (length given by 'length' field above)  
SeeAlso: #03896

Format of HCHST parameter block:

Offset Size Description (Table 03896)  
00h WORD length of following data  
02h 2 WORDs X,Y of left-bottom corner of string on screen  
06h N BYTES ASCII string to display (length given by 'length' field above)  
SeeAlso: #03895

Format of HCLINE parameter block:

Offset Size Description (Table 03897)  
00h WORD length of following data (multiple of 4)  
02h 2N WORDs X,Y coordinates for each of N points in polyline  
Notes: the first line segment is drawn beginning at the current position  
    on completion, the current position is set to the last point drawn  
SeeAlso: #03907,#03918,#03900

Format of HCLOSE parameter block:

Offset Size Description (Table 03898)  
00h WORD 0001h (length of following data)  
01h BYTE (ret) return code  
SeeAlso: #03909

Format of HCMRK parameter block:

Offset Size Description (Table 03899)

00h WORD length of following data

02h 2N WORDs X,Y of N points

Note: draws N+1 marker symbols, the first one at the current position

SeeAlso: #03908

Format of HCRLINE parameter block:

Offset Size Description (Table 03900)

00h WORD length of following data (multiple of 4)

02h 2N WORDs X,Y coordinates relative to the position of the previous point

(current position for first point) for each of N points in

polyline

Notes: the first line segment is drawn beginning at the current position

on completion, the current position is set to the last point drawn

SeeAlso: #03907,#03897,#03918

Format of HEAR parameter block:

Offset Size Description (Table 03901)

00h WORD 0001h (length of following data)

02h BYTE area definition flags

bits 7-6: End Area type

00 complete, perform fill

01 suspend definition

10 complete, but don't fill

SeeAlso: #03889

Format of HEGS parameter block:

Offset Size Description (Table 03902)

00h WORD 0000h (no data following)

Format of HESC parameter block:

Offset Size Description (Table 03903)

00h WORD 0000h (no data following)

Format of HINIT parameter block:

Offset Size Description (Table 03904)

00h WORD 0002h (length of following data)

02h WORD segment of task buffer

SeeAlso: #03943

Format of HINT parameter block:

Offset	Size	Description (Table 03905)
00h	WORD	0004h (length of following data)
02h	DWORD	interrupt/event identifier
		bit 31: vertical blanking

Format of HLDAPAL parameter block:

Offset	Size	Description (Table 03906)
00h	WORD	000Ah (length of following data)
02h	BYTE	palette ID (00h = user, 01h = default)
03h	BYTE	reserved
04h	WORD	number of first palette entry
06h	WORD	number of entries
08h	DWORD	-> palette entries

SeeAlso: #03940

Format of HLINE parameter block:

Offset	Size	Description (Table 03907)
00h	WORD	length of following data (multiple of 4)
02h	2N WORDs	X,Y coordinates for each of N points in polyline

Note: on completion, the current position is set to the last point drawn

SeeAlso: #03897,#03918,#03900

Format of HMRK parameter block:

Offset	Size	Description (Table 03908)
00h	WORD	length of following data (multiple of 4)
02h	2N WORDs	X,Y for N points

SeeAlso: #03899

Format of HOPEN parameter block:

Offset	Size	Description (Table 03909)
00h	WORD	0003h (length of following data)
01h	BYTE	initialization flags
		bit 6: don't load default palette
		bit 7: clear bitplanes
02h	BYTE	mode type (see #03910)
03h	BYTE	(ret) return code
		bit 7: no adapter (hardware mismatch)

SeeAlso: #03898

(Table 03910)

Values for 8514/A display mode:

0000h 12x20 characters, 1024x768  
0001h 8x14 characters, 640x480  
0002h 8x14 characters, 1024x768  
0003h 7x15 characters, 1024x768

Format of HQCOORD parameter block:

Offset Size Description (Table 03911)

00h WORD 0004h (length of following data)  
02h BYTE (ret) coordinate format  
bits 7-4: bytes per coordinate  
bits 3-0: fraction bytes in coordinate  
03h BYTE (ret) relative coordinate format  
bits 7-4: bytes per coordinate  
bits 3-0: fraction bytes in coordinate  
04h BYTE (ret) number of dimensions (2-4)  
05h BYTE (ret) test results  
bit 7: coordinate format not supported  
bit 6: relative coordinate format not supported  
bit 5: specified dimension not supported

Format of HQCP parameter block:

Offset Size Description (Table 03912)

00h WORD 0004h (length of following data)  
02h WORD (ret) current X position  
04h WORD (ret) current Y position

Format of HQDFPAL parameter block:

Offset Size Description (Table 03913)

00h WORD 0040h (length of following data)  
02h 16 DWORDS (ret) color index values

Note: the default palette is set to match the default EGA/VGA 16-color palettes

Format of HQDPS parameter block:

Offset Size Description (Table 03914)

00h WORD 0006h (length of following data)  
02h WORD (ret) size of data buffer in bytes  
04h WORD (ret) stack size in bytes  
06h WORD (ret) size of palette save buffer in bytes

Format of HQMODE parameter block:

Offset	Size	Description (Table 03915)
00h	WORD	0012h (length of following data)
02h	BYTE	current video mode (see #03910)
03h	WORD	driver version
		bit 6: 80286/8086 CPU
		bit 5: 8 bit planes instead of 4 planes
		bits 4-0: hardware release number
05h	BYTE	adapter type
		03h 8514/A
		04h XGA
06h	BYTE	reserved (display type)
07h	BYTE	character cell width
08h	BYTE	character cell height
09h	BYTE	number of bit planes
0Ah	WORD	screen width (pixels)
0Ch	WORD	screen height (pixels)
0Eh	WORD	horizontal resolution (pixels/inch)
10h	WORD	vertical resolution (pixels/inch)
12h	BYTE	flag: 00h = monochrome, FFh = color
13h	BYTE	intensity levels

SeeAlso: #03916

Format of HQMODES parameter block:

Offset	Size	Description (Table 03916)
00h	WORD	0021h (length of following data)
02h	BYTE	(ret) adapter type
03h	32 BYTES	(ret) available display modes (FFh byte marks end of data)

SeeAlso: #03915

Format of HRECT parameter block:

Offset	Size	Description (Table 03917)
00h	WORD	0008h (length of following data)
02h	2 WORDs	X,Y coordinates of top left corner or rectangle
06h	WORD	rectangle's width
08h	WORD	rectangle's height

Note: the rectangle is filled using the current pattern, color, and mix

Format of HRLINE parameter block:

Offset	Size	Description (Table 03918)
00h	WORD	length of following data (multiple of 4)

02h 2 WORDs X,Y coordinates of starting point

06h 2N WORDs X,Y coordinates relative to the position of the previous point  
for each of N points in polyline

Note: on completion, the current position is set to the last point drawn

SeeAlso: #03907,#03897,#03900

Format of HRLPC parameter block:

Offset Size Description (Table 03919)

00h WORD 0000h (no data following)

Note: used for continuity of lines crossing scissors boundaries

SeeAlso: #03932

Format of HRPAL parameter block:

Offset Size Description (Table 03920)

00h WORD 0300h (length of following data)

02h 768 BYTEs buffer containing previously-saved palette table

SeeAlso: #03940

Format of HSBCOL parameter block:

Offset Size Description (Table 03921)

00h WORD 0004h (length of following data)

02h DWORD color index for new background color

SeeAlso: #03925

Format of HSBP parameter block:

Offset Size Description (Table 03922)

00h WORD 000Ch (length of following data)

02h DWORD bitmask for graphics updates

06h DWORD bitmask for alphanumeric updates

0Ah DWORD display bitmask

Format of HSCELL parameter block:

Offset Size Description (Table 03923)

00h

Format of HSCMP parameter block:

Offset Size Description (Table 03924)

00h WORD 0005h (length of following data)

02h DWORD comparison color

06h BYTE logical operation

00h True

```
01h pel > testcolor
02h pel == testcolor
03h pel < testcolor
04h False
05h pel >= testcolor
06h pel <> testcolor
07h pel <= testcolor
```

Format of HSCOL parameter block:

```
Offset Size Description (Table 03925)
00h WORD 0004h (length of following data)
02h DWORD color index for new foreground color
```

SeeAlso: #03921

Format of HSCoord parameter block:

```
Offset Size Description (Table 03926)
00h WORD 0003h (length of following data)
02h BYTE coordinate format
    bits 7-4: bytes per coordinate
    bits 3-0: fraction bytes in coordinate
03h BYTE relative coordinates format
    bits 7-4: bytes per coordinate
    bits 3-0: fraction bytes in coordinate
04h BYTE number of dimensions (2-4)
```

Format of HSCP parameter block:

```
Offset Size Description (Table 03927)
00h WORD 0004h (length of following data)
02h 2 WORDs X,Y or coordinate for new current position
```

Format of HSCS parameter block:

```
Offset Size Description (Table 03928)
00h WORD 0004h (length of following data)
02h DWORD -> character set definition
```

Format of HSGQ parameter block:

```
Offset Size Description (Table 03929)
00h WORD 0002h (length of following data)
02h WORD quality settings (see #03931)
```

Format of HSHS parameter block:

Offset Size Description (Table 03930)  
00h WORD 0008h (length of following data)  
02h WORD left edge of clipping rectangle (-2048 to +6143)  
04h WORD right edge  
06h WORD top edge  
08h WORD bottom edget

Bitfields for 8514/A quality settings:

Bit(s) Description (Table 03931)  
15 reserved  
14 high precision  
13 reserved  
12-11 pel code  
00 not drawn  
01 drawn  
02 conditional on overpainting/mixes  
10 don't close areas to be filed  
9-0 reserved

Format of HSLPC parameter block:

Offset Size Description (Table 03932)  
00h WORD 0000h (no data following)  
Note: used for continuity of lines crossing scissors boundaries  
SeeAlso: #03919

Format of HSLT parameter block:

Offset Size Description (Table 03933)  
00h WORD 0006h (length of following data)  
02h BYTE line type (see #03934)  
03h BYTE reserved  
04h DWORD -> user line-type definition (if user type)  
SeeAlso: #03935

(Table 03934)

Values for 8514/A line type:

00h user line type  
01h dotted  
02h short dashes  
03h dash-dot  
04h double dotted  
05h dashed



06h dash double dot  
07h solid  
08h invisible

SeeAlso: #03933

Format of HSLW parameter block:

Offset	Size	Description (Table 03935)
00h	WORD	0001h (length of following data)
02h	BYTE	width of line in pixels

SeeAlso: #03933

Format of HSMODE parameter block:

Offset	Size	Description (Table 03936)
00h	WORD	0001h (length of following data)
02h	BYTE	new display mode number (see #03910)

Format of HSMRK parameter block:

Offset	Size	Description (Table 03937)
00h	WORD	000Eh (length of following data)
02h	BYTE	cell width
03h	BYTE	cell height
04h	BYTE	flags
05h	BYTE	reserved
06h	WORD	length of marker symbol
08h	DWORD	-> image definition data
0Ch	DWORD	-> color definition data

SeeAlso: #03941

Format of HSMX parameter block:

Offset	Size	Description (Table 03938)
00h	WORD	0002h (length of following data)
02h	BYTE	foreground mix (see #03939)
03h	BYTE	background mix (see #03939)

(Table 03939)

Values for 8514/A mix:

00h retain previous mix  
01h source OR destination  
02h source  
04h source XOR destination  
05h leave as-is

06h max(source,destination)  
07h min(source,destination)  
08h source+destination (clipped)  
09h destination-source (clipped to zero)  
0Ah source-destination (clipped to zero)  
0Bh average source and destination  
10h zero destination  
11h source AND destination  
12h source AND NOT destination  
13h source  
14h NOT source AND destination  
15h leave as-is  
16h source XOR destination  
17h source OR destination  
18h NOT source AND NOT destination  
19h NOT (source XOR destination)  
1Ah NOT destination  
1Bh source OR NOT destination  
1Ch NOT source  
1Dh NOT source OR destination  
1Eh NOT source OR NOT destination  
1Fh set all bits of destination

## Format of HSPAL parameter block:

Offset	Size	Description (Table 03940)
00h	WORD	0300h (length of following data)
02h	768 BYTES	buffer for palette table

SeeAlso: #03906,#03920

## Format of HSPATT parameter block:

Offset	Size	Description (Table 03941)
00h	WORD	000Eh (length of following data)
02h	BYTE	cell width
03h	BYTE	cell height
04h	BYTE	flags
05h	BYTE	reserved
06h	WORD	length of marker symbol
08h	DWORD	-> image definition data
0Ch	DWORD	-> color definition data

SeeAlso: #03937,#03942

Format of HSPATTO parameter block:

Offset	Size	Description (Table 03942)
00h	WORD	0004h (length of following data)
02h	2 WORDs	X,Y of pattern reference point (origin)

SeeAlso: #03941

Format of HSYNC parameter block:

Offset	Size	Description (Table 03943)
00h	WORD	0002h (length of following data)
02h	WORD	segment of task state buffer

SeeAlso: #03904

Format of HXLATE parameter block:

Offset	Size	Description (Table 03944)
00h	WORD	0020h (length of following data)
02h	32 BYTES	color index table

-----V-7F0105-----

INT 7F - IBM XGA Adapter Interface (XGAAIDOS.SYS)

AX = 0105h

Return: CF set on error

CF clear if successful

CX:DX -> array of FAR pointers to entry points (see #03877)

Note: this API is a superset of the 8514/A Adapter Interface

(see AX=0105h"HDILOAD")

-----V-7F0106-----

INT 7F - HDILOAD Mach32 Adapter Interface - UNINSTALL

AX = 0106h

Return: AX = 0105h if successfully unloaded

SeeAlso: AX=0105h

-----N-7F02-----

INT 7F - Alloy NTNX, MW386 - RELEASE SEMAPHORE

AH = 02h

DS:DX -> ASCIZ semaphore name (max 64 bytes)

Return: AL = status (see #03871)

AH = semaphore owner if status=02h

SeeAlso: AH=00h,AH=01h,AH=42h

-----T-7F02-----

INT 7F - MultiLink Advanced v1.0+ - RELEASE CPU

AH = 02h

Return: ???

Desc: yields CPU to other tasks

SeeAlso: AH=00h"MultiLink",AH=09h"MultiLink",INT 15/AX=1000h

-----N-7F02-----

INT 7F - G8BPQ v4.00+ - HOST MODE - SEND FRAME

AH = 02h

AL = stream number (01h-40h)

CX = length of frame

ES:SI -> frame to be sent

SeeAlso: AH=00h"G8BPQ",AH=03h"G8BPQ",AH=07h"G8BPQ",AH=0Ah"G8BPQ"

-----f-7F0200-----

INT 7F - Btrieve Multi-User - GIVE UP TIME???

AX = 0200h

SeeAlso: INT 2F/AX=AB01h,INT 2F/AX=AB02h,INT 7B"Btrieve"

-----N-7F03-----

INT 7F - Alloy ANSK, NTN, MW386 - GET USER NUMBER

AH = 03h

Return: AL = user number

AH = machine number (MW386)

Note: this function call is the recommended method for a CPU-bound process to prevent its priority from being lowered

SeeAlso: AH=04h,AH=05h,AH=A1h

-----N-7F03-----

INT 7F - G8BPQ v4.00+ - HOST MODE - RECEIVE FRAME

AH = 03h

AL = stream number (01h-40h)

ES:DI -> buffer for frame (must be large enough for a full frame; 350 bytes is usually sufficient)

Return: BX = number of pending frames (0000h if returned frame was last avail)

CX = length of received frame

SeeAlso: AH=02h"G8BPQ",AH=07h"G8BPQ",AH=0Bh"G8BPQ"

-----N-7F04-----

INT 7F - Alloy NTN, MW386 - GET NUMBER OF USERS

AH = 04h

Return: AL = total number of users on current machine (MW386)

AL = number of slaves on system (NTN)

SeeAlso: AH=03h

-----N-7F04-----

INT 7F - G8BPQ v4.00+ - HOST MODE - GET STREAM STATUS

AH = 04h

AL = stream number (01h-40h)

Return: CX = state (0000h disconnected, 0001h connected)

DX = delta state (0000h no change, 0001h changed since last check)

SeeAlso: AH=00h"G8BPQ",AH=02h"G8BPQ",AH=05h"G8BPQ"

-----N-7F05-----

INT 7F - Alloy NTN (Host) - LOCK/UNLOCK SYSTEM, SPOOLER CONTROL

AH = 05h

AL = function

00h lock system (disable slave services)

01h unlock system

02h enable spooler

03h disable spooler

04h enable slave timer update

05h disable slave timer update

06h enable form feeds

07h disable form feeds

SeeAlso: INT 17/AH=A4h

-----N-7F05-----

INT 7F - Alloy NTN (Slave), MW386 - GET USER PARAMETERS

AH = 05h

DX:DI -> buffer for user information record (see #03945)

Notes: MW386 provides this function for backward compatibility only, and sets

many of the fields to zero because they are meaningless under MW386

this function has no effect when called by the host (user 0)

SeeAlso: AH=03h

Format of Alloy user information record:

Offset Size Description (Table 03945)

00h WORD segment of video RAM

02h WORD segment of secondary copy of video RAM

04h WORD offset of screen update flag (see INT 10/AH=8Bh)

flag nonzero if update needed

06h WORD video NMI enable port

(not used by MW386, set to 0000h)

08h WORD video NMI disable port

(not used by MW386, set to 0000h)

0Ah BYTE processor type

00h 8088

01h V20

02h 8086

03h V30

06h 80386

0Bh WORD multitasking flag (00h = single tasking, 01h = multitasking)

(not used by MW386, set to 0000h)

0Dh WORD offset of terminal driver  
(not used by MW386, set to 0000h)

0Fh BYTE port for console I/O  
(not used by MW386, set to 0000h)

10h WORD offset of processor communication busy flag  
bit 7 set when slave communicating with host

12h WORD pointer to FAR NX system call  
(not used by MW386, set to 0000h)

14h WORD offset of 16-byte user configuration record (see AH=38h)

16h WORD offset of command/status word

18h WORD offset of screen valid flag (see INT 10/AH=93h)  
nonzero if screen must be repainted

1Ah WORD offset of screen repaint flag

1Ch WORD pointer to NEAR NX system call  
(not used by MW386, set to 0000h)

1Eh WORD offset for intercept flags  
(not used by MW386, set to 0000h)  
intercept flag = FFh if MS-DOS intercepts should be disabled

20h WORD offset of terminal lock flag (see INT 10/AH=92h)  
lock flag = FFh if backgrnd screen updates should be suspended

22h 26 BYTES reserved

-----N-7F05-----

INT 7F - G8BPQ v4.00+ - HOST MODE - ACKNOWLEDGE STREAM STATUS CHANGE

AH = 05h

AL = stream number (01h-40h)

Note: this function must be called in order to receive a report of another  
status change

SeeAlso: AH=00h"G8BPQ",AH=04h"G8BPQ"

-----N-7F06-----

INT 7F - Alloy NTN (Host) - GET SHARED DRIVE INFO

AH = 06h

AL = drive number (1=A:, 2=B:, etc)

ES:DI -> drive info record (see #03946)

Return: AX = status

0000h successful

ES:DI buffer filled

0001h not shared drive

Format of Alloy drive info record:

Offset Size Description (Table 03946)

00h WORD segment of drive IO-REQUEST structure (MS-DOS DPB)

02h WORD segment of allocation map (owner table)  
 one byte per FAT entry, containing user ID owning that entry  
 04h WORD segment of master FAT for drive (copy of FAT on disk)  
 06h WORD pointer to configuration file  
 08h WORD total number of clusters  
 0Ah WORD bytes per sector  
 0Ch WORD sectors per cluster  
 0Eh BYTE FAT type (0Ch = 12-bit, 10h = 16-bit)

-----N-7F06-----

INT 7F - Alloy NTNX (Slave) - ALLOCATE FREE CLUSTER ON SHARED DRIVE

AH = 06h  
 DL = drive number (1=A:,2=B:,etc)  
 CX = number of clusters to allocate

Return: AH = status

00h successful  
 CX = number of clusters still free  
 10h invalid shared drive request  
 CL = first and second shared drives  
 11h invalid cluster count (must be 01h-FFh)

-----N-7F06-----

INT 7F - G8BPQ v4.00+ - HOST MODE - SESSION CONTROL

AH = 06h  
 AL = stream number (01h-40h)  
 CX = subfunction  
 0000h connect to node  
 DL bit 0: use BBS callsign instead of Node Call  
 0001h connect to node  
 use BBS Call if APPLMASK=1  
 0002h disconnect  
 0003h return user to node

SeeAlso: AH=01h"G8BPQ",AH=04h"G8BPQ"

-----N-7F07-----

INT 7F - Alloy NTNX, MW386 - GET LIST OF SHARED DRIVES

AH = 07h

Return: ES:DI -> shared drive list (see #03947)

Note: MW386 considers all fixed disks to be shared drives; only C and D will  
 be returned as shared

Format of Alloy shared drive list:

Offset Size Description (Table 03947)

00h BYTE string length

01h BYTE number of shared drives  
02h N BYTES one byte per shared drive  
-----N-7F07-----  
INT 7F - G8BPQ v4.00+ - HOST MODE - GET BUFFER COUNTS FOR STREAM  
AH = 07h  
AL = stream number (01h-40h)  
Return: BX = number of pending receive frames  
CX = number of unacknowledged sent frames  
DX = number of buffers available  
SeeAlso: AH=02h"G8BPQ",AH=03h"G8BPQ"  
-----N-7F08-----  
INT 7F - Alloy NTN (Host) - GET INTERRUPT VECTORS  
AH = 08h  
CL = function  
00h get original interrupt vector  
01h get Network Executive interrupt  
AL = interrupt number  
DX:SI -> DWORD to hold interrupt vector  
Return: AL = status  
00h successful  
01h interrupt vector not used by network executive  
02h invalid subfunction  
Note: the network executive uses interrupts 02h,08h,09h,0Fh,10h,13h,16h-19h,  
1Ch,20h,28h,2Ah,2Fh,5Bh,67h,7Fh,ECh, and F0h-FFh  
SeeAlso: AH=09h/CL=03h,INT 21/AH=35h  
-----N-7F08--CL02-----  
INT 7F - Alloy NTN - SET MESSAGE DISPLAY TIMEOUT  
AH = 08h  
CL = 02h  
DX = timeout in seconds  
Return: AL = status  
00h successful  
02h invalid subfunction  
-----N-7F08-----  
INT 7F - G8BPQ v4.00+ - HOST MODE - PORT CONTROL/INFORMATION  
AH = 08h  
AL = stream number (01h-40h)  
Return: ES:DI -> 10-byte buffer containing blank-padded callsign  
---v4.05+ ---  
AL = radio port to which channel is connected (level 2)  
AH = session type (see #03948)



BX = L2 paclen for session  
CX = maximum frame size  
DX = L4 window size or 0000h if not L4 circuit

Program: the G8BPQ AX25 Networking Package is amateur packet radio software by  
John Wiseman which allows a PC to act as a node in an AX.25 network  
SeeAlso: AH=01h"G8BPQ",AH=02h"G8BPQ",AH=03h"G8BPQ",AH=0Ah"G8BPQ"

Bitfields for G8BPQ session type:

Bit(s) Description (Table 03948)

0 L2LINK  
1 SESSION  
2 UPLINK  
3 DOWNLIND  
5 BPQHOST

-----T-7F09-----

INT 7F - MultiLink Advanced v1.0+ - SET TASK PRIORITY

AH = 09h  
AL = priority (0-7)

Return: nothing

InstallCheck: ensure that the interrupt vector is not pointing at segment  
0000h, then test whether the byte at offset 0000h in the interrupt  
handler's segment is E9h

Index: installation check;MultiLink Advanced

SeeAlso: AH=00h"MultiLink",AH=0Ah"MultiLink"

-----N-7F09-----

INT 7F - G8BPQ - proposed addition - GET NODE/APPLICATION CALLSIGN AND ALIAS

AH = 09h  
AL = application  
00h node  
01h BBS  
02h HOST  
03h SYSOP

BL = what to get (00h callsign, 01h application name)

ES:SI -> buffer for callsign/name string

Return: CX = length of returned string

SeeAlso: AH=00h"G8BPQ",AH=01h"G8BPQ",AH=0Ch"G8BPQ"

-----N-7F09-----

INT 7F - Alloy NTNX - ENABLE/DISABLE MUD FILE CHECKING

AH = 09h  
CL = function  
00h enable checking of RTNX.MUD file

01h disable RTNX.MUD checking

-----N-7F09--CL02-----

INT 7F - Alloy NTNX - SWITCH HOST TO DEDICATED MODE

AH = 09h

CL = 02h

Note: in dedicated mode, the host will only poll for I/O requests from the slave processors, and not provide workstation services

-----N-7F09--CL03-----

INT 7F - Alloy NTNX,MW386 - GET ALTERNATE INTERRUPT

AH = 09h

CL = 03h

AL = default interrupt number (67h,7Fh,etc)

Return: CL = actual interrupt which handles specified interrupt's calls

SeeAlso: AH=08h

-----N-7F0A--CL00-----

INT 7F - Alloy NTNX - GET SYSTEM FLAGS

AH = 0Ah

CL = 00h

ES:DI -> buffer for system flags (see #03949)

Return: ES:DI buffer filled

Notes: on a slave, only the NX\_Busy flag is returned

all three flags are at fixed positions, so this function only needs to be called once

an interrupt handler should only perform DOS or device accesses when

all three flags are 00h

Format of Alloy system flags:

Offset Size Description (Table 03949)

00h DWORD pointer to NX\_Busy flag (nonzero when communicating with users)

04h DWORD pointer to device driver busy flag

08h DWORD pointer to InTimer flag

-----N-7F0A-----

INT 7F - G8BPQ v4.00+ - HOST MODE - TRANSMIT RAW (KISS) FRAME

AH = 0Ah

AL = radio port

ES:SI -> buffer containing data to be sent

CX = number of bytes to send

SeeAlso: AH=02h"G8BPQ",AH=08h"G8BPQ",AH=0Bh"G8BPQ"

-----T-7F0A-----

INT 7F - MultiLink Advanced v1.0+ - SET KEYBOARD TEST STATUS

AH = 0Ah

AL = task-switch status  
 00h normal (disable task when it repeatedly polls keyboard)  
 01h disable task until keyboard input available  
 FFh never disable task

Return: ???

SeeAlso: AH=09h"MultiLink"

-----N-7F0B--CL02-----

INT 7F - Alloy NTN (Host) - SET/RESET GRAPHICS DOS ON SLAVE

AH = 0Bh  
 CL = 02h  
 AL = slave ID number  
 CH = DOS to activate  
 00h graphics DOS  
 01h character DOS

Return: AL = status

00h successful  
 01h nothing done, proper DOS type already loaded

-----N-7F0B-----

INT 7F - G8BPQ v4.00+ - HOST MODE - RECEIVE TRACE (RAW DATA) FRAME

AH = 0Bh  
 ES:DI -> buffer for received data (see #03950)

Return: CX = number of bytes received

Note: the specified buffer must be large enough to receive a full frame

SeeAlso: AH=03h"G8BPQ",AH=08h"G8BPQ",AH=0Ah"G8BPQ"

Format of G8BPQ received data:

Offset Size Description (Table 03950)

00h WORD internal control information  
 02h BYTE port number (bit 7 set if transmitted frame)  
 03h WORD frame length including this header  
 05h var user data

-----N-7F0C-----

INT 7F - G8BPQ v4.00+ - HOST MODE - UPDATE SWITCH INFORMATION

AH = 0Ch  
 DX = function  
 0001h update beacon text  
 CX = length of data  
 ES:SI -> data to be sent in beacons  
 0002h (v4.07+) initiate NODES broadcast

SeeAlso: AH=09h"G8BPQ"

-----N-7F0D00-----

INT 7F - G8BPQ v4.07+ - HOST MODE - GET AVAILABLE STREAM

AX = 0D00h

Return: AL = first available stream number, or FFh if none free

SeeAlso: AH=00h"G8BPQ",AH=0Dh

-----N-7F0D-----

INT 7F - G8BPQ v4.07+ - HOST MODE - ALLOCATE/DEALLOCATE STREAM

AH = 0Dh

AL = stream number (01h-FFh)

CL = function

01h allocate stream

Return: CX = status (0000h successful, else already in use)

02h deallocate stream

SeeAlso: AX=0D00h

-----N-7F0F-----

INT 7F - G8BPQ v4.00+ - HOST MODE - GET TIME MARKER

AH = 0Fh

Return: AX = time marker (clock ticks modulo 64K)

Program: the G8BPQ AX25 Networking Package is amateur packet radio software by

John Wiseman which allows a PC to act as a node in an AX.25 network

SeeAlso: AH=01h"G8BPQ",AX=0D00h,INT 1A/AH=00h

-----N-7F10--CL00-----

INT 7F - Alloy NTN, MW386 - CHANNEL CONTROL - OPEN CHANNEL

AH = 10h

CL = 00h

AL = channel number

DX:DI -> channel buffer

Return: AL = status (00h-03h,0Dh) (see #03951)

Note: may not be invoked from within a hardware interrupt handler

SeeAlso: AH=10h/CL=01h,AH=10h/CL=04h,AH=14h/CL=02h

(Table 03951)

Values for Alloy function status:

00h successful

01h busy

02h channel range error (not 00h-3Fh)

03h invalid subfunction

0Ah channel not open

0Ch channel already locked

0Dh unable to open

-----N-7F10--CL01-----

INT 7F - Alloy NTN, MW386 - CHANNEL CONTROL - CLOSE CHANNEL

AH = 10h  
CL = 01h  
AL = channel number

Return: AL = status (00h-03h,0Ah) (see #03951)

Note: may not be invoked from within a hardware interrupt handler

SeeAlso: AH=10h/CL=00h,AH=10h/CL=05h

-----N-7F10--CL02-----

INT 7F - Alloy NTN, MW386 - CHANNEL CONTROL - LOCK CHANNEL

AH = 10h  
CL = 02h  
AL = channel number

Return: AL = status (00h-03h,0Ah,0Ch) (see #03951)

Note: may not be invoked from within a hardware interrupt handler

SeeAlso: AH=10h/CL=03h,AH=10h/CL=06h,AH=10h/CL=08h

-----N-7F10--CL03-----

INT 7F - Alloy NTN, MW386 - CHANNEL CONTROL - UNLOCK CHANNEL

AH = 10h  
CL = 03h  
AL = channel number

Return: AL = status (00h-03h,0Ah) (see #03951)

Notes: should only be used on channels locked with AH=10h/CL=02h, not on those  
locked by receipt of a datagram

may not be invoked from within a hardware interrupt handler

SeeAlso: AH=10h/CL=02h,AH=10h/CL=04h,AH=10h/CL=09h

-----N-7F10--CL04-----

INT 7F - Alloy NTN, MW386 - CHANNEL CONTROL - RELEASE BUFFER

AH = 10h  
CL = 04h  
AL = channel number

Return: AL = status (00h-03h) (see #03951)

Notes: unlocks buffer after received datagram has been processed

may not be invoked from within a hardware interrupt handler

SeeAlso: AH=10h/CL=00h

-----N-7F10--CL05-----

INT 7F - Alloy NTN, MW386 - CHANNEL CONTROL - CLOSE ALL CHANNELS

AH = 10h  
CL = 05h

Return: AL = status (00h-03h) (see #03951)

Notes: clears all pending datagrams and clears buffer pointers before closing  
the channels

may not be invoked from within a hardware interrupt handler

SeeAlso: AH=10h/CL=01h

-----N-7F10--CL06-----

INT 7F - Alloy NTNX, MW386 - CHANNEL CONTROL - LOCK ALL OPEN CHANNELS

AH = 10h

CL = 06h

Return: AL = status (00h-03h) (see #03951)

Note: may not be invoked from within a hardware interrupt handler

SeeAlso: AH=10h/CL=02h,AH=10h/CL=08h

-----N-7F10--CL07-----

INT 7F - Alloy NTNX, MW386 - CHANNEL CONTROL - UNLOCK ALL LOCKED IDLE CHANNELS

AH = 10h

CL = 07h

Return: AL = status (00h-03h) (see #03951)

Notes: unlocks all locked channels which have no pending datagrams

may not be invoked from within a hardware interrupt handler

SeeAlso: AH=10h/CL=03h,AH=10h/CL=09h

-----N-7F10--CL08-----

INT 7F - Alloy NTNX, MW386 - CHANNEL CONTROL - LOCK MULTIPLE CHANNELS

AH = 10h

CL = 08h

DX = maximum channel number to lock

Return: AL = status (00h-03h) (see #03951)

Notes: locks channels numbered 00h through the value in DX

may not be invoked from within a hardware interrupt handler

SeeAlso: AH=10h/CL=02h,AH=10h/CL=06h,AH=10h/CL=09h

-----N-7F10--CL09-----

INT 7F - Alloy NTNX, MW386 - CHANNEL CONTROL - UNLOCK MULTIPLE CHANNELS

AH = 10h

CL = 09h

DX = maximum channel number to unlock

Return: AL = status (00h-03h) (see #03951)

Notes: unlocks channels numbered 00h through the value in DX

may not be invoked from within a hardware interrupt handler

SeeAlso: AH=10h/CL=03h,AH=10h/CL=07h,AH=10h/CL=08h

-----N-7F11-----

INT 7F - Alloy NTNX, MW386 - SEND DATAGRAM

AH = 11h

DX:SI -> request block (see #03953)

Return: AL = status (see #03952)

Note: if wildcard channel FFh used, actual channel number will be filled in

SeeAlso: AH=12h

(Table 03952)

Values for Alloy function status:

00h successful  
01h busy  
02h channel range error (not 00h-3Fh)  
03h invalid subfunction  
0Ah packet too large (or <2 bytes if NTNX)  
0Bh can't send packet to itself  
0Ch invalid number of destinations  
0Dh destination channel number out of range  
0Eh destination user is busy  
0Fh destination user has locked channel  
10h channel not open  
11h no datagram server on destination (NTNX)

Format of Alloy request block:

Offset Size Description (Table 03953)

00h DWORD pointer to packet to send  
04h WORD packet size in bytes (1-4096)  
06h BYTE number of destinations for packet (max 1Fh)  
07h 31 BYTES destination user IDs (FFh = broadcast to all except sender)  
26h 31 BYTES destination channels (FFh = first available channel)  
45h 31 BYTES return destination statuses

-----N-7F12-----

INT 7F - Alloy NTNX, MW386 - ACKNOWLEDGE DATAGRAM

AH = 12h

AL = channel number being acknowledged

DI:DX = 32-bit status to return to sender

Return: AL = status (see #03954)

Note: also unlocks the channel, allowing the next datagram to be received

SeeAlso: AH=11h,AH=15h/CL=04h

(Table 03954)

Values for Alloy function status:

00h successful  
01h busy  
02h channel range error (not 00h-3Fh)  
03h invalid subfunction  
0Ah channel not open  
0Bh no message in channel

0Ch destination slave busy--retry (NTNX)  
0Dh destination user not active  
0Eh destination slave not active (NTNX)  
0Fh destination disabled datagram service

-----V-7F1234-----

INT 7F - TIGA Communications Driver v2.05 - UNINSTALL  
AX = 1234h

SeeAlso: AX=4321h

-----N-7F13--CL00-----

INT 7F - Alloy NTNX, MW386 - RESET USER DATAGRAMS  
AH = 13h  
CL = 00h

Note: clears all pending datagrams and removes all channels opened in NTNX  
compatibility mode

-----N-7F14--CL00-----

INT 7F - Alloy NTNX, MW386 - SET RECEIVE ISR  
AH = 14h  
CL = 00h

DX:DI -> application FAR receive service routine (see #03955)

Return: AL = status (00h-03h) (see #03954)

SeeAlso: AH=14h/CL=01h,AH=14h/CL=03h

(Table 03955)

Values Alloy receive service routine is called with:

DH = sender ID

DL = channel with datagram

interrupts disabled

Return: AL = response code

00h leave buffer locked, set channel status, and repeat call later

01h release channel buffer

02h change buffer pointer to DX:DI

AH,CX,DX,DI,SI may be destroyed

-----N-7F14--CL01-----

INT 7F - Alloy NTNX, MW386 - SET ACKNOWLEDGE ISR  
AH = 14h  
CL = 01h

DX:DI -> application FAR acknowledge service routine (see #03956)

Return: AL = status (00h-03h) (see #03954)

Note: the service routine will be called as soon as an acknowledgment arrives

SeeAlso: AH=12h,AH=14h/CL=00h,AH=14h/CL=04h,AH=15h/CL=04h



(Table 03956)

Values Alloy acknowledge service routine is called with:

DS:SI -&gt; acknowledge structure (see #03960)

Return: AL = response code

00h application busy, network executive should call again later

01h acknowledge accepted

AH,DX,SI may be destroyed

-----N-7F14--CL02-----

INT 7F - Alloy NTN, MW386 - SET CHANNEL BUFFER POINTER

AH = 14h

CL = 02h

AL = channel number

DX:DI -&gt; receive buffer

Return: AL = status (00h-03h) (see #03954)

Note: may be called from within a receive ISR or when a datagram is pending

SeeAlso: AH=10h/CL=00h,AH=14h/CL=00h

-----N-7F14--CL03-----

INT 7F - Alloy NTN, MW386 - GET RECEIVE ISR

AH = 14h

CL = 03h

Return: DX:DI -&gt; current receive ISR

SeeAlso: AH=14h/CL=00h,AH=14h/CL=04h

-----N-7F14--CL04-----

INT 7F - Alloy NTN, MW386 - GET ACKNOWLEDGE ISR

AH = 14h

CL = 04h

Return: DX:DI -&gt; current acknowledge ISR

SeeAlso: AH=14h/CL=01h,AH=14h/CL=03h

-----N-7F14--CL05-----

INT 7F - Alloy NTN (Host), MW386 - GET BUSY POINTER

AH = 14h

CL = 05h

DX:DI -&gt; buffer for busy structure (see #03957)

Return: DX:DI buffer filled

Format of Alloy busy structure:

Offset Size Description (Table 03957)

00h DWORD pointer to busy flag byte

04h WORD fixed port address (FF00h)

-----N-7F15--CL00-----

INT 7F - Alloy NTN, MW386 - GET CHANNEL STATUS

AH = 15h  
 CL = 00h  
 AL = channel number  
 DX:DI -> status structure (see #03958)  
 Return: AL = status (00h-03h) (see #03954)  
 SeeAlso: AH=15h/CL=01h

Format of Alloy channel status structure:

Offset Size Description (Table 03958)

00h BYTE channel status  
     bit 0: channel open  
     bit 1: channel buffer contains received data  
     bit 7: channel locked

01h BYTE sender ID

-----N-7F15---CL01-----

INT 7F - Alloy NTN, MW386 - GET NEXT FULL CHANNEL

AH = 15h

CL = 01h

DX:DI -> full-channel structure (see #03959)

Return: AL = status

00h successful

01h busy

0Ah no datagrams available

Note: MW386 v1.0 returns the lowest channel with a datagram; newer versions  
 and NTN return the oldest datagram

SeeAlso: AH=15h/CL=00h

Format of Alloy full-channel structure:

Offset Size Description (Table 03959)

00h BYTE number of channel with oldest datagram

01h BYTE sender ID

-----N-7F15---CL02-----

INT 7F - Alloy NTN, MW386 - GET MAXIMUM NUMBER OF CHANNELS

AH = 15h

CL = 02h

Return: AH = number of channels available (40h for MW386)

Note: the application may always assume at least 32 channels available

SeeAlso: AH=15h/CL=03h

-----N-7F15---CL03-----

INT 7F - Alloy NTN, MW386 - GET MAXIMUM PACKET SIZE

AH = 15h

CL = 03h

DX:DI -> WORD for return value

Return: buffer WORD filled with maximum packet size (4096 for MW386)

SeeAlso: AH=15h/CL=02h

-----N-7F15--CL04-----

INT 7F - Alloy NTN, MW386 - GET AND CLEAR ACKNOWLEDGE STATUS

AH = 15h

CL = 04h

DX:DI -> status structure (see #03960)

Return: AL = status

00h successful

DX:DI structure filled

01h busy

0Ah no acknowledgement has arrived

SeeAlso: AH=12h,AH=14h/CL=01h

Format of Alloy status structure:

Offset Size Description (Table 03960)

00h BYTE sender ID

01h BYTE channel number

02h 4 BYTES receiver status (see #03954)

-----N-7F16-----

INT 7F - Alloy NTN, MW386 - DIRECT MEMORY TRANSFER

AH = 16h

DX:SI -> transfer structure (see #03961)

Return: AL = status

00h successful

0Ah source or destination out of range

0Bh transfer kernel busy--try again

Notes: this call transfers memory contents directly between users; both source  
and destination user IDs may differ from the caller's ID  
no segment wrap is allowed

Format of Alloy transfer structure:

Offset Size Description (Table 03961)

00h WORD bytes to transfer

02h BYTE source ID

FEh = caller

03h DWORD source address

07h BYTE destination ID

FFh = all slaves except caller

FEh = caller  
08h DWORD destination address  
-----N-7F21-----  
INT 7F - Alloy NTNX, MW386 - SEND MESSAGE OR COMMAND TO USER(S)  
AH = 21h  
AL = sender's user ID  
DS:DX -> control packet (see #03962)  
Note: messages or commands are ignored if disabled by the destination user  
SeeAlso: AH=22h

Format of Alloy control packet:

Offset Size Description (Table 03962)

00h BYTE packet type  
00h message  
01h NTNX command  
02h MW386 command  
01h BYTE destination user ID or 'A' for all users  
02h 62 BYTES ASCIZ message (packet type 00h)  
BIOS keycodes terminated by NUL byte (type 01h) or word (02h)  
Note: a maximum of 16 keycodes will be processed for NTNX and MW386 commands

-----N-7F22-----  
INT 7F - Alloy NTNX - GET MESSAGE  
AH = 22h  
Return: pending messages displayed on user's screen  
SeeAlso: AH=21h

-----N-7F24-----  
INT 7F - Alloy NTNX, MW386 - ATTACH OR RELEASE DRIVE FOR LOW-LEVEL WRITE ACCESS  
AH = 24h  
CL = function  
00h attach  
01h release  
CH = drive (0=A:,1=B:,etc)  
Return: AX = status (see #03963)  
Note: only drives on the current machine may be attached

(Table 03963)

Values for Alloy function status:

00h successful  
01h invalid request  
02h already attached  
03h not attached

04h lock table full

-----N-7F24-----

INT 7F - Alloy NTNX - ATTACH/RELEASE HOST PROCESSOR

AH = 24h

CL = function

02h attach host

03h release host

Return: AX = status (see #03963)

Note: the host processor may be attached in order to perform I/O via the host

-----N-7F25--CL00-----

INT 7F - Alloy ANSK, NTNX, MW386 - GET NETWORK EXECUTIVE VERSION

AH = 25h

CL = 00h

Return: AH = version suffix letter

CH = major version number

CL = minor version number

SeeAlso: AH=25h/CL=01h

-----N-7F25--CL01-----

INT 7F - Alloy ANSK, NTNX, MW386 - GET NETWORK EXECUTIVE TYPE

AH = 25h

CL = 01h

Return: CL = executive type (see #03964)

SeeAlso: AH=25h/CL=00h

(Table 03964)

Values for Alloy network executive type:

00h RTNX

01h ATNX

02h NTNX

03h BTNX

04h MW386

05h ANSK

-----V-7F2525-----

INT 7F - TIGA Communications Driver v2.05 - ???

AX = 2525h

BX = ???

Return: ???

SeeAlso: AX=4321h,AX=5555h

-----N-7F26--CL00-----

INT 7F - Alloy NTNX, MW386 - GET NTNX FILE MODE

AH = 26h

CL = 00h

Return: AX = file mode bits (see #03965)

Note: MW386 does not support file modes, and always returns AX=001Fh

SeeAlso: AH=26h,AH=26h/CL=06h

Bitfields for Alloy file mode bits:

Bit(s) Description (Table 03965)

0 directory protection enabled

1 extended open enabled

2 flush on every disk write

3 flush on every disk write in locked interval

4 flush on reads from simultaneously opened file

-----N-7F26-----

INT 7F - Alloy NTNX - SET FILE I/O CHECKING LEVEL

AH = 26h

CL = check type to set/reset

01h directory protection

02h extended open

03h flush on every disk write

04h flush on disk write if any lock set during write

05h flush on all reads if file written

AL = new state (00h off, 01h on)

SeeAlso: AH=26h/CL=00h,AH=26h/CL=06h

-----N-7F26--CL06-----

INT 7F - Alloy NTNX - CANCEL FLUSH ON WRITE

AH = 26h

CL = 06h

Note: cancels flags set by AH=26h/CL=03h and AH=26h/CL=04h

SeeAlso: AH=26h/CL=00h

-----N-7F30-----

INT 7F - Alloy MW386 - GET PORT INFORMATION

AH = 30h

CX = MW386 port number

Return: AL = result

FFh if port not found

else driver unit number

BL = port mode

BH = port type

02h remote

DH = owner's machine ID

DL = owner's user ID

SeeAlso: INT 17/AH=8Bh

-----N-7F31-----

INT 7F - Alloy MW386 v1.x only - CHECK PORT ASSIGNMENT

AH = 31h

???

Return: ???

-----N-7F37-----

INT 7F - Alloy NTN (Host) - GET SEMAPHORE TABLE

AH = 37h

Return: ES:AX -> semaphore table

-----N-7F37-----

INT 7F - Alloy ANSK, NTN (Slave) - DUMP STRING TO TERMINAL

AH = 37h

DS:DX -> ASCIZ string to display

Note: if the string is empty, a terminal update will be forced

-----N-7F38-----

INT 7F - Alloy NTN (Slave), MW386 - SET NEW TERMINAL DRIVER

AH = 38h

AL = new terminal driver number

FFh dummy driver

FEh current driver

FDh load new driver

DS:SI -> new driver

SeeAlso: AH=39h

-----N-7F39-----

INT 7F - Alloy MW386 - SET TERMINAL DRIVER FOR ANOTHER USER

AH = 39h

AL = new terminal driver number

FFh dummy driver

FEh current driver

FDh load new driver

DS:SI -> new driver

DL = user number (FFh = caller)

DH = machine number if DL <> FFh

Return: CF set if invalid user number

CF clear if successful

Notes: only available to supervisors

the new driver number will not take effect until the user is rebooted

SeeAlso: AH=38h

-----N-7F3A-----

INT 7F - Alloy MW386 - GET TERMINAL PARAMETERS

AH = 3Ah  
DL = user number (FFh = caller)  
DH = machine number  
Return: CF clear if successful  
AH = terminal driver number  
AL = baud rate (00h = 38400, 01h = 19200, etc)  
CL = parity (00h none, 01h even, 02h odd)  
CH = handshaking (00h none, 01h XON/XOFF, 02h DTR/DSR, 03h XPC)  
CF set if invalid user number

SeeAlso: AH=3Bh

-----N-7F3B-----

INT 7F - Alloy MW386 - SET TERMINAL PARAMETERS

AH = 3Bh  
AL = baud rate (00h = 38400, 01h = 19200, etc)  
CL = parity (00h none, 01h even, 02h odd)  
CH = handshaking (00h none, 01h XON/XOFF, 02h DTR/DSR, 03h XPC)  
DL = user number (FFh = caller)  
DH = machine number for user

Return: CF set if invalid user number

Notes: only available to supervisors

the new parameters will take effect immediately if the user's terminal  
has not been started, else AH=3Dh must be called to post the changes

SeeAlso: AH=3Ah,AH=3Dh

-----N-7F3C-----

INT 7F - Alloy MW386 - ENABLE/DISABLE AUTOBAUD DETECT

AH = 3Ch  
AL = new state (00h disabled, 01h enabled)  
DL = user number (FFh = caller)  
DH = machine number for user

Return: CF set if invalid user number

Note: only available to supervisors

SeeAlso: AH=3Dh

-----N-7F3D-----

INT 7F - Alloy MW386 - POST TERMINAL CONFIGURATION CHANGES

AH = 3Dh  
Note: should be called whenever a program changes the terminal type or its  
parameters

SeeAlso: AH=3Bh

-----N-7F41-----

INT 7F - Alloy NTN - LOCK FILE FOR USER

AH = 41h



AL = user ID  
DS:DX -> ASCIZ filename  
Return: AL = status (see #03966)  
Note: requests exclusive read/write access to file  
SeeAlso: AH=00h,AH=41h"MW386",AH=42h"NTNX"

(Table 03966)

Values for Alloy function status:

00h successful  
01h invalid function  
02h already locked  
03h unable to lock  
04h lock table full or semaphore space exhausted

-----N-7F41-----

INT 7F - Alloy MW386 - LOCK SEMAPHORE FOR USER

AH = 41h  
AL = user ID  
DS:DX -> ASCIZ semaphore name

Return: AL = status (see #03966)

SeeAlso: AH=00h,AH=42h"MW386"

-----s-7F4150BHC1-----

INT 7F U - Voyetra - AAPISG - API

AX = 4150h ('AP')  
BH = C1h  
BL = function (00h-13h)  
    00h initialize (fails except first time called)  
    ???

Return: AX = status???

0000h successful  
0001h failed

Program: AAPISG is a driver by Voyetra for the Aztech Sound Galaxy sound board

BUG: the function range check uses JL instead of JB, so it will cause a  
crash if BL >= 80h on entry

SeeAlso: AX=4331h,AX=564Dh,AX=5658h

-----N-7F42-----

INT 7F - Alloy NTNX - UNLOCK FILE FOR USER

AH = 42h  
AL = user ID  
DS:DX -> ASCIZ filename

Return: AL = status (see #03966)

SeeAlso: AH=00h,AH=41h"NTNX",AH=42h"MW386"

-----N-7F42-----

INT 7F - Alloy MW386 - UNLOCK SEMAPHORE FOR USER

AH = 42h

AL = user ID

DS:DX -> ASCIZ semaphore name

Return: AL = status

00h successful

01h invalid function

03h unable to unlock semaphore

SeeAlso: AH=02h,AH=41h"MW386",AH=42h"NTNX"

-----V-7F4321-----

INT 7F - TIGA Communications Driver v2.05 - INSTALLATION CHECK

AX = 4321h

Return: AX = 0000h if installed

Note: INT 7F is the default, but may be overridden

SeeAlso: AH=01h"TIGA",AX=1234h,AX=2525h,AX=4321h,AX=5555h

-----s-7F4331BHCl-----

INT 7F U - Voyetra - VAPISG - API

AX = 4331h ('C1')

BH = C1h

BL = function (00h-7Ah)

???

Return: ???

Program: VAPISG is a MIDI driver by Voyetra for the Aztech Sound Galaxy  
sound board

SeeAlso: AX=4150h,AX=564Dh,AX=5658h

-----N-7F4E-----

INT 7F - Alloy MW386 v2+ - SET ERROR MODE

AH = 4Eh

AL = error mode flags

bit 0: display critical disk errors

bit 1: display sharing errors

DX = 4E58h ("NX")

Return: AL = status

00h successful

SeeAlso: AH=4Fh

-----N-7F4F-----

INT 7F - Alloy MW386 v2+ - SET FCB MODE

AH = 4Fh

AL = FCB mode

02h read/write compatibility

```
    42h read/write shared
    DX = 4E58h ("NX")
Return: AL = status
    00h successful
-----V-7F5555-----
INT 7F - TIGA Communications Driver v2.05 - ???
    AX = 5555h
    BX = ???
Return: ???
SeeAlso: AX=4321h
-----s-7F564DBHC1-----
INT 7F U - Voyetra Multimedia Player - VMP.EXE API
    AX = 564Dh ('VM')
    BH = C1h
    BL = function (00h-1Bh)
    00h ???
    Return: CF clear
    AX = 0000h
    ???
Return: AX = FFFFh if invalid function
    ???
SeeAlso: AX=4331h,AX=5658h
-----s-7F5658BHC1-----
INT 7F U - Voyetra - VAPISG - API
    AX = 5658h ('VX')
    BH = C1h
    BL = function (00h-1Bh)
    ???
Return: ???
Program: VAPISG is a MIDI driver by Voyetra for the Aztech Sound Galaxy
    sound board
SeeAlso: AX=4331h,AX=564Dh
-----N-7F81-----
INT 7F - Alloy NTNX - ATTACH DEVICE FOR USER
    AH = 81h
    AL = user ID
    DS:DX -> ASCIZ device name
SeeAlso: AH=82h
-----N-7F82-----
INT 7F - Alloy NTNX - RELEASE DEVICE FOR USER
    AH = 82h
```

AL = user ID  
DS:DX -> ASCIZ device name  
SeeAlso: AH=81h  
-----N-7FA0-----

INT 7F - Alloy MW386 - GET USER NAME  
AH = A0h  
DL = user number (FFh = caller)  
DH = machine number for user  
ES:DI -> 17-byte buffer for ASCIZ user name  
Return: CF set if invalid user number  
SeeAlso: AH=03h,AH=A1h

-----N-7FA1-----  
INT 7F - Alloy MW386 - GET MACHINE, USER, AND PROCESS NUMBER  
AH = A1h  
Return: AL = process number  
DL = user number  
DH = machine number  
SeeAlso: AH=03h,AH=A0h,AH=A2h

-----N-7FA2-----  
INT 7F - Alloy MW386 - GET USER PRIVILEGE LEVEL  
AH = A2h  
DL = user number (FFh = caller)  
DH = machine number for user  
Return: CF clear if successful  
AL = privilege level  
00h supervisor  
01h high  
02h medium  
03h low  
CF set if invalid user number  
SeeAlso: AH=A1h,AH=A3h

-----N-7FA3-----  
INT 7F - Alloy MW386 - GET USER LOGIN STATE  
AH = A3h  
DL = user number  
DH = machine number for user  
Return: CF clear if successful  
AL = login state  
00h never logged in  
01h currently logged out  
03h currently logged in

```
CF set if invalid user number or user not active
SeeAlso: AH=A2h
-----N-7FA4-----
INT 7F - Alloy MW386 - VERIFY USER PASSWORD
  AH = A4h
  DS:DX -> ASCIZ password (null-padded to 16 bytes)
Return: AL = status
  00h  accepted
  else invalid password
-----N-7FA500-----
INT 7F - Alloy MW386 - GET USER STATUS
  AX = A500h
  DI = machine number and user number
Return: CF clear if successful
  BX = user flags
  bit 5: allow messages
  CL = scan code for task manager hotkey
  CH = scan code for spooler hotkey
  DL = scan code for task swapper hotkey
  DH = modifier key status
  CF set if invalid user number
SeeAlso: AX=A501h
Index:  hotkeys;Alloy MW386
-----N-7FA501-----
INT 7F - Alloy MW386 - SET USER STATUS
  AX = A501h
  BX = user flags (see AX=A500h)
  CL = scan code for task manager hotkey
  CH = scan code for spooler hotkey
  DL = scan code for task swapper hotkey
  DH = modifier key status
  DI = machine number and user number
Return: CF set if invalid user number
Note: must have supervisor privilege to set another user's status
SeeAlso: AX=A500h
Index:  hotkeys;Alloy MW386
-----V-7FABCDBX0000-----
INT 7F - IBM 8516 Touch Screen Device Driver - GET API ENTRY
  AX = ABCDh
  BX = 0000h
Return: AX = total number of functions available
```

ES:BX -> entry point array (see #03967)

SeeAlso: AX=0104h,AX=0105h

(Table 03967)

Values for 8516 Touch Screen function number:

00h check initialization and reset (see #03968)

14h set user-defined subroutine (see #03969)

Notes: each driver function takes two stack parameters using Pascal calling conventions: address of parameter block and address of results buffer

all pointers are FAR pointers

on return, AX contains the status of the call:

AX = 0000h successful

0001h invalid input

0002h interface error

0003h unable to perform function

Format of 8516 Touch Screen Function 00h parameter block:

Offset Size Description (Table 03968)

00h WORD 0000h (function number)

Note: this function should be called before any other device driver functions

Format of 8516 Touch Screen Function 00h results buffer:

Offset Size Description (Table 03969)

00h WORD touch screen status

0000h unavailable

0001h uncalibrated

FFFFh available

02h WORD aux mouse status (0000h not present, FFFFh present)

Notes: the following driver parameters will have been reset to zero:

touchdown counter, liftoff counter, position at last touch, position at last lift, int call mask, select on count, select off count, pos select on count, pos select off count.

the following driver parameters will have been reset as listed:

mouse emulation mode: left on

thresholds: 46 on screen, 96 push harder, 80 push release

x, y hysteresis: 400

data repeat rate: 40/sec

select mechanism: push-harder - first-touch

coordinate origin: upper left corner

filter frequency: medium

data block mask: all enabled

click lock: on

-----N-7FB0-----

INT 7F - Alloy NTN, MW386 - RELEASE ALL SEMAPHORES FOR USER

AH = B0h

AL = user number

DS = code segment

Note: MW386 ignores AL and DS; it releases all semaphores locked using INT 67  
or INT 7F locking functions

SeeAlso: AH=B1h,AH=B2h,AH=B3h,AH=B4h

-----N-7FB1--SF00-----

INT 7F - Alloy NTN, MW386 - RELEASE NORMAL SEMAPHORES FOR USER

AH = B1h subfn 00h

AL = (bits 7-5) 000

(bits 4-0) user ID

Note: MW386 ignores AL; it releases all semaphores locked using INT 67 or  
INT 7F locking functions

SeeAlso: AH=B0h,AH=B2h,AH=B3h,AH=B4h

-----N-7FB2--SF01-----

INT 7F - Alloy NTN - RELEASE MESSAGES FOR USER

AH = B2h subfn 01h

AL = (bits 7-5) 001

(bits 4-0) user ID

SeeAlso: AH=B0h,AH=B1h,AH=B3h,AH=B4h

-----N-7FB3--SF02-----

INT 7F - Alloy NTN - RELEASE FILES FOR USER

AH = B3h subfn 02h

AL = (bits 7-5) 010

(bits 4-0) user ID

SeeAlso: AH=B0h,AH=B1h,AH=B2h,AH=B4h

-----N-7FB4-----

INT 7F - Alloy NTN - RELEASE DEVICES FOR USER

AH = B4h

AL = user ID

SeeAlso: AH=B0h,AH=B1h,AH=B2h,AH=B3h

-----N-7FC3-----

INT 7F - Alloy MW386 - WRITE BYTE TO TERMINAL AUX PORT

AH = C3h

AL = byte to write

Return: CF clear if successful

CF set on error

SeeAlso: AH=C6h

-----N-7FC5-----

INT 7F - Alloy MW386 - CHANGE CONSOLE MODE

AH = C5h

AL = new console mode

00h keyboard indirect

01h keyboard direct

02h data handshake enforced

03h no data handshake

Return: CF clear if successful

AL = prior console mode

CF set on error (caller is not remote user)

Note: modes 2 and 3 may be used for input through the console port; no video

output should be performed in these modes

-----N-7FC6-----

INT 7F - Alloy MW386 - WRITE BYTE TO CONSOLE PORT

AH = C6h

AL = byte to write

Return: CF clear if successful

CF set on error (caller is not remote user)

Note: any terminal driver data translation will be bypassed

SeeAlso: AH=C3h,AH=C7h

-----N-7FC7-----

INT 7F - Alloy MW386 - READ CONSOLE DATA BYTE

AH = C7h

Return: CF clear if successful

AL = byte read

CF set on error (no data available or caller is not remote user)

Note: used to read data after placing console in mode 2 or 3 (see AH=C5h)

SeeAlso: AH=C5h,AH=C6h,AH=C8h

-----N-7FC8-----

INT 7F - Alloy MW386 - READ CONSOLE DATA INTO BUFFER

AH = C8h

AL = maximum bytes to read

ES:DI -> buffer for console data

Return: CF clear if successful

CX = number of bytes read

CF set on error (caller is not remote user)

SeeAlso: AH=C7h

-----N-7FCF-----

INT 7F - Alloy NTNK - REBOOT USER PROCESSOR

AH = CFh



DS:DX -> ASCIZ string containing user number to be reset  
SeeAlso: AH=D6h  
-----N-7FD6-----  
INT 7F - Alloy MW386 - RESET NETWORK EXECUTIVE  
AH = D6h  
DS:DX -> reset packet (see #03970)  
Return: never if successful  
Note: all users will be shut down immediately if successful  
SeeAlso: AH=CFh

Format of Alloy MW386 reset packet:

Offset	Size	Description (Table 03970)
00h	DWORD	reset code (60606060h)
04h	16 BYTES	ASCIZ supervisor password padded with nulls

-----N-7FD7-----

INT 7F - Alloy MW386 - POST EVENT

AH = D7h  
AL = user number (if local event)  
DX = event number

-----N-7FD8-----

INT 7F - Alloy MW386 - FLUSH DISK BUFFERS

AH = D8h  
Return: CF set on error

Note: forces all disk buffers to be written out immediately  
SeeAlso: INT 21/AH=0Dh, INT 21/AX=5D01h, INT 2F/AX=1120h

-----N-7FDB-----

INT 7F - Alloy MW386 v2+ - GET MW386 INVOCATION DRIVE

AH = DBh  
Return: AL = drive from which MW386 was started (2=C:,3=D:,etc)

-----N-7FE0-----

INT 7F - Alloy MW386 - CREATE DOS TASK

AH = E0h  
AL = memory size (00h=128K, 01h=256K, 02h=384K, 03h=512K, 04h=640K)  
DS:DX -> ASCIZ task name (max 16 bytes)

Return: CF clear if successful

AL = task create ID

CF set on error

Note: only foreground DOS tasks can use this function

SeeAlso: AH=E1h, AH=E2h, AH=E3h, AH=E6h, AH=E7h

-----N-7FE1-----

INT 7F - Alloy MW386 - GET DOS TASK PID FROM CREATE ID

AH = E1h  
AL = create ID (from AH=E0h)  
Return: AL = DOS process number  
CL = memory size (00h=128K, 01h=256K, 02h=384K, 03h=512K, 04h=640K)  
Note: this function should not be called immediately after creating a new  
DOS task, since the new task is being initialized by a concurrent  
process  
SeeAlso: AH=E0h,AH=E2h

-----N-7FE2-----

INT 7F - Alloy MW386 - SWITCH TO NEW DOS TASK

AH = E2h  
AL = DOS process number (from AH=E1h)  
Return: CF set on error (invalid process number or caller not foreground task)  
Notes: specified task becomes the foreground task and current task is placed  
in the background  
may only be called by a foreground task  
SeeAlso: AH=E0h,AH=E1h

-----N-7FE3-----

INT 7F - Alloy MW386 - CHANGE NAME OF DOS TASK

AH = E3h  
DS:DX -> ASCIZ task name

---v1.x---

AL = user number

---v2+---

BH = user number

BL = task number

Return: CF clear if successful  
CF set on error (invalid process number)

SeeAlso: AH=E0h,AH=E4h,AH=E5h

-----N-7FE4-----

INT 7F - Alloy MW386 - GET TASK NAME FROM PROCESS NUMBER

AH = E4h  
ES:DI -> buffer for task name

---v1.x---

AL = user number

---v2+---

BH = user number

BL = task number

Return: CF clear if successful  
CL = memory size (00h=128K, 01h=256K, 02h=384K, 03h=512K, 04h=640K)  
DX = task flags

```
    bit 7: MS-DOS process
    ES:DI buffer filled
    CF set on error (invalid process number)
SeeAlso: AH=E3h,AH=E5h
-----N-7FE5-----
INT 7F - Alloy MW386 - GET PROCESS NUMBER FROM TASK NAME
    AH = E5h
    DS:DX -> ASCIZ task name
    BH = user number
Return: CF clear if successful
    AL = DOS process number
    CL = memory size (00h=128K, 01h=256K, 02h=384K, 03h=512K, 04h=640K)
    CF set on error (no match for name)
SeeAlso: AH=E3h,AH=E4h
-----N-7FE6-----
INT 7F - Alloy MW386 - GET NUMBER OF AVAILABLE USER TASKS
    AH = E6h
Return: AX = number of processes available to current user
SeeAlso: AH=E0h
-----N-7FE7-----
INT 7F - Alloy MW386 - REMOVE DOS TASK
    AH = E7h
    AL = DOS process number
Return: CF clear if successful
    CF set on error (invalid process number or first process)
Note: can only be called by a foreground task
SeeAlso: AH=E0h
-----N-7FE8-----
INT 7F - Alloy MW386 - DOS TASK DELAY
    AH = E8h
    CX = delay time in milliseconds
Note: a delay of 0 may be used to surrender the current time slice
SeeAlso: INT 15/AX=1000h,INT 1A/AX=FF01h,INT 21/AH=EEh"DoubleDOS"
SeeAlso: INT 2F/AX=1680h
-----N-7FF0-----
INT 7F - Alloy MW386 - RESTRICT DIRECTORY TO GROUP
    AH = F0h
    AL = group number
    DS:DX -> ASCIZ directory name
Return: CF clear if successful
    AX = status
```

0002h directory not found  
0003h directory not found  
0005h directory in use, cannot be restricted  
02xxh restricted to group xxh

CF set on error

Note: the restriction on the directory may be removed by calling this function with group 0, then using AH=F1h to assign the directory to group 0

SeeAlso: AH=F1h,AH=F2h,AH=F3h

-----N-7FF1-----  
INT 7F - Alloy MW386 - ASSIGN DIRECTORY TO GROUP

AH = F1h  
AL = group number  
DS:DX -> ASCIZ directory name

Notes: performs permanent assignment to a group; no immediate action is taken unless the directory has been restricted with AH=F0h  
may be used to restrict a nonexistent directory

SeeAlso: AH=F0h

-----N-7FF2-----  
INT 7F - Alloy MW386 - READ RESTRICTED DIRECTORY ENTRY

AH = F2h  
CX = entry number  
ES:DI -> 64-byte buffer

Return: CF clear if successful  
buffer filled with 63-byte directory info and 1-byte group number  
CF set on error (invalid entry)

SeeAlso: AH=F0h,AH=F3h

-----N-7FF3-----  
INT 7F - Alloy MW386 - READ RESTRICTED DIRECTORY ENTRY FOR GROUP

AH = F3h  
AL = group number  
CX = entry number  
ES:DI -> 64-byte buffer

Return: CF clear if successful  
CX = next entry number  
buffer filled with 63-byte directory info and 1-byte group number  
CF set on error (no more matching entries)

Note: like AH=F2h, but only returns directories belonging to the specified group

SeeAlso: AH=F2h

-----N-7FF8-----

INT 7F - Alloy MW386 - ASSIGN USER TO GROUP

AH = F8h  
AL = group number  
DL = user number  
DH = machine number (currently 00h)

Return: CF clear if successful

CF set on error (user already in maximum number of groups)

Note: each user is allowed eight group assignments

SeeAlso: AH=F9h,AH=FAh

-----N-7FF9-----

INT 7F - Alloy MW386 - REMOVE USER FROM GROUP

AH = F9h  
AL = group number  
DL = user number  
DH = machine number (currently 00h)

Return: CF clear if successful

CF set if failed

SeeAlso: AH=F8h,AH=FAh

-----N-7FFA-----

INT 7F - Alloy MW386 - GET USER GROUP LIST

AH = FAh  
DL = user number  
DH = machine number (currently 00h)  
ES:DI -> 16-byte buffer for group list

Return: CX = number of groups

ES:DI buffer filled with group numbers

SeeAlso: AH=F8h,AH=F9h

-----N-7FFB-----

INT 7F - Alloy MW386 - ASSIGN GROUP NAME

AH = FBh  
CL = group number  
ES:DI -> ASCIZ group name (max 17 bytes)

SeeAlso: AH=FCh

-----N-7FFC-----

INT 7F - Alloy MW386 - GET GROUP NAME

AH = FCh  
CL = group number  
ES:DI -> 17-byte buffer for ASCIZ name

Return: ES:DI buffer filled

Note: if the group has not been named, "(unnamed)" is returned

SeeAlso: AH=FBh

-----80-----

INT 80 - Q-PRO4 - ???

-----r-80-----

INT 80 - reserved for BASIC

Note: this vector and INT 81 through INT ED are modified but not restored by Direct Access v4.0, and may be left dangling by other programs written with the same version of compiled BASIC

SeeAlso: INT 81"BASIC",INT 86"BASIC",INT EF"BASIC"

-----E-80-----

INT 80 - Phar Lap 386|DOS-Extender - RELOCATED PRINT-SCREEN

Note: the extender relocates INT 05 to here by default, but can be told to leave INT 05 alone with the commandline (or DOSX= environment variable) flag -PRIVEC 5

SeeAlso: INT 05"PRINT SCREEN"

-----d-80-----

INT 80 - BusLogic BT-946C PCI SCSI Adapter - SCRATCHPAD RAM (NOT A VECTOR!)

Note: the factory-default location for the eight bytes of scratchpad RAM needed by the SCSI adapter is 0000h:0200h, which is interrupt vectors 80h and 81h

SeeAlso: INT 81"BusLogic"

-----O-80-----

INT 80 - Linux - SYSTEM CALLS  
parameters in registers

(Table 04086)

Values for system call number:

00h	???
01h	exit
02h	fork
03h	read
04h	write
05h	open
06h	close
07h	waitpid
08h	creat
09h	link
0Ah	unlink
0Bh	execve
0Ch	chdir
0Dh	time
0Eh	mknod

0Fh chmod  
10h lchown  
11h break  
12h oldstat (obsolete)  
13h lseek  
14h getpid  
15h mount  
16h umount  
17h setuid  
18h getuid  
19h stime  
1Ah ptrace  
1Bh alarm  
1Ch oldfstat  
1Dh pause  
1Eh utime  
1Fh stty  
20h gtty  
21h access  
22h nice  
23h ftime  
24h sync  
25h kill  
26h rename  
27h mkdir  
28h rmdir  
29h dup  
2Ah pipe  
2Bh times  
2Ch prof  
2Dh brk  
2Eh setgid  
2Fh getgid  
30h signal  
31h geteuid  
32h getegid  
33h acct  
34h umount2  
35h lock  
36h ioctl  
37h fcntl

38h mpx  
39h setpgid  
3Ah ulimit  
3Bh oldolduname (obsolete)  
3Ch umask  
3Dh chroot  
3Eh ustat  
3Fh dup2  
40h getppid  
41h getpgrp  
42h setsid  
43h sigaction  
44h sgetmask  
45h ssetmask  
46h setreuid  
47h setregid  
48h sigsuspend  
49h sigpending  
4Ah sethostname  
4Bh setrlimit  
4Ch getrlimit  
4Dh getrusage  
4Eh gettimeofday  
4Fh settimeofday  
50h getgroups  
51h setgroups  
52h select  
53h symlink  
54h oldlstat (obsolete)  
55h readlink  
56h uselib  
57h swapon  
58h reboot  
59h readdir  
5Ah mmap  
5Bh munmap  
5Ch truncate  
5Dh ftruncate  
5Eh fchmod  
5Fh fchown  
60h getpriority



61h setpriority  
62h profil  
63h statfs  
64h fstatfs  
65h ioperm  
66h socketcall  
67h syslog  
68h setitimer  
69h getitimer  
6Ah stat  
6Bh lstat  
6Ch fstat  
6Dh olduname (obsolete)  
6Eh iopl  
6Fh vhangup  
70h idle  
71h vm86old  
72h wait4  
73h swapoff  
74h sysinfo  
75h ipc  
76h fsunc  
77h sigreturn  
78h clone  
79h setdomainname  
7Ah uname  
7Bh modify\_ldt  
7Ch adjtimex  
7Dh mprotect  
7Eh sigprocmask  
7Fh create\_module  
80h init\_module  
81h delete\_module  
82h get\_kernel\_syms  
83h quotactl  
84h getpgid  
85h fchdir  
86h bdflush  
87h sysfs  
88h personality  
89h afs\_syscall

```
8Ah  setfsuid
8Bh  setfsgid
8Ch  llseek
8Dh  getdents
8Eh  newselect
8Fh  flock
90h  msync
91h  readv
92h  writev
93h  getsid
94h  fdatsync
95h  sysctl
96h  mlock
97h  munlock
98h  mlockall
99h  munlockall
```

!!! <http://home.snafu.de/phpr/lhpsysc0.html>

-----O-80-----

INT 80 - FreeBSD - SYSTEM CALLS

parameters on stack

-----b-80-----

INT 80 U - AMI BIOS v1.00.12.AX1T - internal - BIOS SUBSYSTEM SELECTION

AH = function

00h install and initialize BIOS subsystem

AL = ??? (00h,01h,03h)

CX:BX = subsystem ID (see #03971)

ESI = address from which to load, or 00000000h for default for  
subsystem

EDI = physical address at which to install, or 0 for default

Note: if CX=0000h on entry, this call is applied to all

subsystems whose ID has low word BX

01h get BIOS subsystem information

CX:BX = subsystem ID (see #03971)

Return: AL = ???

AH = ???

EDX = uncompressed size of subsystem in bytes

SI = offset within subsystem of initialization  
routine, or FFFFh if none

EDI = physical address of default location or 0

02h get matching subsystem identifier

AL = index into subsystem list (return ALth occurrence)

matching BX)

BX = low word of subsystem identifier  
(0001h,0002h,0004h,0005h,000Bh,FFFFh)

Return: CF clear if successful

CX = high word of ALth matching subsystem

CF set on error

Note: the system is halted if AL=00h on entry

03h set up "big real" mode (4G segment limits)

04h turn off "big real" mode (restore 64K segment limits)

05h remove BIOS subsystem

CX:BX = subsystem ID to leave out

Note: the system is halted if an invalid (not installed)

subsystem ID is specified

06h get installed subsystem info

CX:BX = subsystem ID (see #03971)

Return: CF clear

EDX = length of ???

EDI = linear address of start of ???

Note: the system is halted if an invalid (not installed)

subsystem ID is specified

Return: CF clear if successful

CF set on non-fatal error

Note: this interrupt vector is cleared to 0000h:0000h near the end of the

BIOS startup sequence

(Table 03971)

Values for AMI BIOS subsystem ID:

00010001h	ROM BIOS @F000-FFFF
00010002h	setup??? (loaded @6000-68FF)
00010005h	APM code
0001000Bh	language-specific error message set (English)
00020002h	PnP/PCI ACFG code (loaded @F000-F1FF)
0001FFFFh	
0002FFFFh	recovery code
00030004h	
0003FFFFh	BIOS decompression code
0004000Bh	
00060004h	
0100FFFFh	ACFG data
10000000h	ROM @C000 (64K)
10000001h	HMA (48K)

10000002h RAM @7A00 (24K)  
10000003h RAM @8000-BFFF  
10000004h  
10000005h real-mode address space (0-1M)  
1000000Bh installed language-specific message set???

-----s-80----BL00-----

INT 80 - SBSIM - "STARTSND" - START SOUND ON SPECIFIED DRIVER

BL = 00h

BH = driver number (01h = FM, 02h=DDEV, 03h=memvoice, 05h=MIDI)

Return: AX = initialization result (see #03972)

Program: SBSIM is Creative Labs' SoundBlaster Simplified Interface Module,  
which provides access to multiple drivers for the SoundBlaster  
board through a single interface

Range: INT 80h to INT Bfh, selected automatically

Note: the SBSIM installation check consists of testing for the signature  
"SBSIM" at offset 103h in the interrupt handler's segment.

SeeAlso: INT 80/BL=01h"SBSIM",INT 80/BL=02h"SBSIM",INT 80/BL=03h"SBSIM"

SeeAlso: INT 80/BX=0000h"SBSIM"

(Table 03972)

Values for SBSIM error code:

01h busy--currently in use  
02h bad driver specified  
03h invalid function  
04h voice process already active  
05h couldn't start CT-VOICE  
06h couldn't start CTVDSK  
07h invalid SBSIM handle  
08h buffer not initialized yet  
09h bad filename  
0Ah bad file handle  
0Bh driver not started yet  
0Ch XMS driver not installed  
0Dh no free SBSIM handles  
0Eh bad file type  
0Fh couldn't free XMS block  
10h invalid source selected  
11h get pan position failed  
12h set pan position failed  
13h set volume failed  
14h couldn't start fade/pan

15h couldn't stop fade/pan  
16h couldn't pause fade/pan  
17h not a fade/pan operation  
18h bad mode for fade/pan  
19h couldn't start fade/pan  
1Ah source not fading/panning  
1Bh FM or MIDI already playing  
1Ch bad MIDI mapper format

-----s-80----BL01-----

INT 80 - SBSIM - "PLAYSND" - PLAY MUSIC/VOICE ON SELECTED DRIVER

BL = 01h

BH = driver number (01h = FM, 02h=DDBV, 03h=memvoice, 05h=MIDI)

Return: AX = result (see #03972)

SeeAlso: INT 80/BL=00h"SBSIM",INT 80/BL=02h"SBSIM",INT 80/BL=04h"SBSIM"

-----s-80----BL02-----

INT 80 - SBSIM - "STOPSND" - STOP MUSIC/VOICE ON SELECTED DRIVER

BL = 02h

BH = driver number (01h = FM, 02h=DDBV, 03h=memvoice, 05h=MIDI)

Return: nothing

SeeAlso: INT 80/BL=00h"SBSIM",INT 80/BL=01h"SBSIM",INT 80/BL=03h"SBSIM"

-----s-80----BL03-----

INT 80 - SBSIM - "PAUSESND" - TEMPORARILY PAUSE PLAYBACK ON SELECTED DRIVER

BL = 03h

BH = driver number (01h = FM, 02h=DDBV, 03h=memvoice, 05h=MIDI)

Return: nothing

SeeAlso: INT 80/BL=00h"SBSIM",INT 80/BL=02h"SBSIM",INT 80/BL=04h"SBSIM"

SeeAlso: INT 80/BL=05h

-----s-80----BL04-----

INT 80 - SBSIM - "RESUMESND" - RESTART PLAYBACK ON SELECTED DRIVER

BL = 04h

BH = driver number (01h = FM, 02h=DDBV, 03h=memvoice, 05h=MIDI)

Return: nothing

SeeAlso: INT 80/BL=00h"SBSIM",INT 80/BL=03h"SBSIM",INT 80/BL=05h"SBSIM"

-----s-80----BL05-----

INT 80 - SBSIM - "GETSNDSTAT" - GET DRIVER'S STATUS

BL = 05h

BH = driver number (01h = FM, 02h=DDBV, 03h=memvoice, 05h=MIDI)

Return: AX = status

SeeAlso: INT 80/BL=00h"SBSIM",INT 80/BL=01h"SBSIM",INT 80/BL=03h"SBSIM"

-----s-80----BX0000-----

INT 80 - SoundBlaster SBFM driver - GET VERSION

```

BX = 0000h
Return: ???
Note: SBFM installs at a free interrupt in the range 80h through BFh
SeeAlso: BX=0008h"SBFM",INT 2F/AX=FBBh/ES=0000h
-----s-80----BX0000-----
INT 80 - SBSIM - "QUERYVERSION" - GET VERSION
    BX = 0000h
Return: AX = version (AH = major, AL = minor)
Program: SBSIM is Creative Labs' SoundBlaster Simplified Interface Module,
    which provides access to multiple drivers for the SoundBlaster
    board through a single interface
Range: INT 80h to INT BFh, selected automatically
Note: the SBSIM installation check consists of testing for the signature
    "SBSIM" at offset 103h in the interrupt handler's segment.
SeeAlso: BX=0001h"SBSIM",BX=0005h"SBSIM",INT 21/AX=4402h"CTMMSYS"
SeeAlso: INT 80/BL=00h"SBSIM"
Index: installation check;SBSIM|installation check;SoundBlaster
-----s-80----BX0001-----
INT 80 - SoundBlaster SBFM driver - SET MUSIC STATUS BYTE ADDRESS
    BX = 0001h
    DX:AX -> music status byte
SeeAlso: BX=0000h"SBFM",BX=0002h"SBFM",BX=0003h"SBFM"
-----s-80----BX0001-----
INT 80 - SBSIM - "QUERYDRIVERS" - CHECK DRIVERS INSTALLED
    BX = 0001h
Return: AX = bit flags for loaded drivers (see #03973)
SeeAlso: BX=0000h"SBSIM",BX=0002h"SBSIM",BX=0005h"SBSIM",INT 80/BL=00h"SBSIM"

Bitfields for SBSIM loaded drivers:
Bit(s) Description (Table 03973)
0  FM
1  double disk-buffered voice driver (DDBV)
2  memory voice driver
3  auxiliary driver (mixer)
4  MIDI
-----s-80----BX0002-----
INT 80 - SoundBlaster SBFM driver - SET INSTRUMENT TABLE
    BX = 0002h
    CX = number of instruments
    DX:AX -> instrument table
SeeAlso: BX=0000h"SBFM",BX=0001h"SBFM",BX=0005h"SBFM"
```

-----s-80----BX0002-----

INT 80 - SBSIM - GETADDRESS" - GET SELECTED DRIVER'S ENTRY POINT

BX = 0002h

AX = driver (00h = FM, 01h = DDBV, 02h = memvoice, 03h = mixer, 04h = MIDI)

Return: CF clear if successful

DX:AX -> entry point

CF set on error

SeeAlso: BX=0000h"SBSIM",BX=0001h"SBSIM",BX=0005h"SBSIM",INT 80/BL=00h"SBSIM"

-----s-80----BX0003-----

INT 80 - SoundBlaster SBFM driver - SET SYSTEM CLOCK RATE

BX = 0003h

AX = clock rate divisor (1193180 / desired frequency in Hertz)

FFFFh to restore to 18.2 Hz

SeeAlso: BX=0000h"SBFM",BX=0001h"SBFM",BX=0004h"SBFM"

-----s-80----BX0004-----

INT 80 - SoundBlaster SBFM driver - SET DRIVER CLOCK RATE

BX = 0004h

AX = driver clock rate divisor (1193180 / frequency in Hertz)

Note: default frequency is 96 Hz

SeeAlso: BX=0000h"SBFM",BX=0003h"SBFM"

-----s-80----BX0005-----

INT 80 - SoundBlaster SBFM driver - TRANSPOSE MUSIC

BX = 0005h

AX = semi-tone offset

SeeAlso: BX=0000h"SBFM",BX=0002h"SBFM",BX=0006h"SBFM"

-----s-80----BX0005-----

INT 80 - SBSIM - "GETBUFFERINFO" - GET DRIVER'S BUFFER ADDRESS

BX = 0005h

AX = driver (00h = FM, 01h = DDB Voice, 04h = MIDI)

Return: CF clear if successful

DX:AX -> buffer

CX = buffer size in K

CF set on error

Program: SBSIM is Creative Labs' SoundBlaster Simplified Interface Module,  
which provides access to multiple drivers for the SoundBlaster  
board through a single interface

Range: INT 80h to INT BFh, selected automatically

Note: the SBSIM installation check consists of testing for the signature

"SBSIM" at offset 103h in the interrupt handler's segment.

SeeAlso: BX=0000h"SBSIM",BX=0001h"SBSIM",BX=0002h"SBSIM",INT 80/BL=00h"SBSIM"

-----s-80----BX0006-----

```
INT 80 - SoundBlaster SBFM driver - PLAY MUSIC
  BX = 0006h
  DX:AX -> music block
Return: AX = status
  0000h successful
  0001h music already active
SeeAlso: BX=0000h"SBFM",BX=0007h"SBFM",BX=000Ah"SBFM",INT 1A/AX=FF04h
-----s-80----BX0007-----
INT 80 - SoundBlaster SBFM driver - STOP MUSIC
  BX = 0007h
Return: AX = status
  0000h successful
  0001h music not active
SeeAlso: BX=0000h"SBFM",BX=0006h"SBFM",BX=0009h"SBFM",INT 1A/AX=FF05h
-----s-80----BX0008-----
INT 80 - SoundBlaster SBFM driver - RESET DRIVER
  BX = 0008h
Return: AX = status
  0000h successful
  0001h music is active
SeeAlso: BX=0000h"SBFM"
-----s-80----BX0009-----
INT 80 - SoundBlaster SBFM driver - PAUSE MUSIC
  BX = 0009h
Return: AX = status
  0000h successful
  0001h no music active
SeeAlso: BX=0000h"SBFM",BX=0007h"SBFM",BX=000Ah"SBFM",INT 1A/AX=FF01h
-----s-80----BX000A-----
INT 80 - SoundBlaster SBFM driver - RESUME MUSIC
  BX = 000Ah
Return: AX = status
  0000h successful
  0001h no music paused
SeeAlso: BX=0000h"SBFM",BX=0006h"SBFM",BX=0009h"SBFM"
-----s-80----BX000B-----
INT 80 - SoundBlaster SBFM driver - SET USER-DEF TRAP FOR SYSTEM-EXCLUSIVE CMDS
  BX = 000Bh
  DX:AX -> trap routine
SeeAlso: BX=0000h"SBFM"
-----s-80----BX0400-----
```



INT 80 - SBSIM - "GETVOLUME" - GET SOURCE'S VOLUME  
BX = 0400h  
AX = sound source (see #03974)  
Return: CF clear if successful  
AX = volume  
CF set on error  
AX = error code (see #03972 at INT 80/BL=00h)  
SeeAlso: BX=0401h"SBSIM"

(Table 03974)

Values for SBSIM sound source:

00h master volume  
01h voice  
02h FM  
03h CD  
04h line in  
05h microphone

-----s-80----BX0401-----

INT 80 - SBSIM - "SETVOLUME" - SET SOURCE'S VOLUME  
BX = 0401h  
AX = sound source (see #03974)  
DX = new volume  
Return: AX = result (0000h = success) (see also INT 80/BL=00h)  
SeeAlso: BX=0400h"SBSIM"

-----N-8001-----

INT 80 - QPC Software PKTINT.COM - INITIALIZE  
AH = 01h  
Return: AX = 0000h  
CX = FFFFh  
DX = FFFFh

Notes: this interrupt is the WinQVTNet protected mode interface to Windows 3.0  
all buffer pointers are reset back to 0

-----N-8002-----

INT 80 - QPC Software PKTINT.COM - GET BUFFER ADDRESSES  
AH = 02h  
BX = extra bytes to allocate per packet  
Return: AX = segment address of 10K buffer (for receives???)  
BX = segment address of 2K buffer (for sends???)  
SeeAlso: AH=05h

-----N-8003-----

INT 80 - QPC Software PKTINT.COM - GET ENTRY POINT

AH = 03h  
Return: CX:DX -> receive call address  
Note: the returned address can be used in the packet driver calls since it  
will be a valid address in all DOS boxes  
SeeAlso: AH=06h  
-----N-8004-----  
INT 80 - QPC Software PKTINT.COM - ENABLE???  
AH = 04h  
BX = ???  
Return: ???  
SeeAlso: AH=01h  
-----N-8005-----  
INT 80 - QPC Software PKTINT.COM - GET RECEIVE STATISTICS  
AH = 05h  
Return: AX = amount of buffer currently in use  
BX = current offset in buffer  
CX = number of times receive has been called  
SeeAlso: AH=02h  
-----N-8006-----  
INT 80 - QPC Software PKTINT.COM - REMOVE RECEIVED PACKET  
AH = 06h  
Return: BX = next packet offset  
CX = number of bytes still buffered  
DX = size of packet released back into buffer pool  
SeeAlso: AH=03h  
-----r-81-----  
INT 81 - reserved for BASIC  
Note: this vector is modified but not restored by Direct Access v4.0, and  
may be left dangling by other programs written with the same version  
of compiled BASIC  
SeeAlso: INT 80"BASIC",INT 82"BASIC"  
-----N-81-----  
INT 81 - IBM TOKEN RING ADAPTER - ???  
SeeAlso: INT 82"TOKEN RING",INT 91"TOKEN RING"  
-----d-81-----  
INT 81 - BusLogic BT-946C PCI SCSI Adapter - SCRATCHPAD RAM (NOT A VECTOR!)  
Note: the factory-default location for the eight bytes of scratchpad RAM  
needed by the SCSI adapter is 0000h:0200h, which is interrupt  
vectors 80h and 81h  
SeeAlso: INT 80"BusLogic"  
-----b-81-----

INT 81 - AMI WinBIOS - SECOND HARD DRIVE AUTODETECTION

Notes: used by WinBIOS with core version of July 1994 or later  
QEMM 7.5 began using this interrupt internally after the QPAT3  
maintenance release when searching for "ROM holes", causing a  
lengthy delay during bootup. Adding the QEMM parameter RH:N will  
avoid the QEMM call to INT 81 and speed up the boot process

-----O-81-----

INT 81 - Paragon Technology PTS/DOS 6.5x, S/DOS 1.0 - API

-----s-810200-----

INT 81 - Gravis UltraSound - MegaEm - PROCESS COMMAND LINE

AX = 0200h

BX = PSP segment

SI = ??? (data area)

Return: AX = status (see #03975)

BL = emulation state

bit 1 Emulation is on

bit 2 Emulation is off

Program: MegaEm is a protected-mode SoundBlaster, SoundCanvas, and MT-32  
emulator for the Gravis UltraSound

Range: INT 81 to INT FF (see INT 21/AX=FD12h), selected by scanning for an  
interrupt with vector 0000h:0000h

SeeAlso: AX=0202h,AX=0300h,AX=0400h,INT 21/AX=FD12h/BX=3457h

(Table 03975)

Values for MegaEm status:

0000h successful

0001h invalid command line optie

0002h display options

0003h ???

0004h invalid number of voices specified

0005h could not load enough patches to provide acceptable emulation

0006h to many warnings on patch loading

0007h ??? (not used anymore)

0008h /CO and SCSI don't work together

0009h invalid music volume

000Ah invalid master volume

000Bh /SC switch is no longer valid

0100h ??? (some error on patch loading)

-----s-810202-----

INT 81 - Gravis UltraSound - MegaEm - ???

AX = 0202h

Return: AX = ???

SeeAlso: AX=0200h,AX=0300h,AX=0400h,INT 21/AX=FD12h/BX=3457h

-----s-810300-----

INT 81 - Gravis UltraSound - MegaEm - LOAD SAMPLE ???

AX = 0300h

Return: AX = ???

SeeAlso: AX=0200h,AX=0202h,AX=0400h,INT 21/AX=FD12h/BX=3457h

-----s-810400-----

INT 81 - Gravis UltraSound - MegaEm - ???

AX = 0400h

BX = ??? (segment)

Return: AX = ???

BX = ???

Program: MegaEm is a protected-mode SoundBlaster, SoundCanvas, and MT-32

emulator for the Gravis UltraSound

SeeAlso: AX=0200h,AX=0202h,AX=0300h,INT 21/AX=FD12h/BX=3457h

-----s-812010-----

INT 81 - Gravis UltraSound - MEGA\_EM v3.0+ - INT 78 REDIRECT

AX = 2010h

???

Return: ???

Note: MegaEm calls this function from its INT 78 handler, and then immediately returns (via RETF in v3.10, thus leaving the flags on the stack); this permits it to operate while hooking only one interrupt through the memory manager in protected mode

SeeAlso: AX=2011h,AX=2012h

-----s-812011-----

INT 81 - Gravis UltraSound - MEGA\_EM v3.0+ - NMI REDIRECT

AX = 2011h

???

Return: ???

Note: MegaEm calls this function from its NMI (INT 02) handler, and then immediately returns; this permits it to operate while hooking only one interrupt through the memory manager in protected mode

SeeAlso: AX=2010h,AX=2012h

-----s-812012-----

INT 81 - Gravis UltraSound - MEGA\_EM v3.0+ - SOUND CARD IRQ REDIRECT

AX = 2012h

???

Return: ???

Note: MegaEm calls this function from its GUS IRQ handler, and then

immediately returns; this permits it to operate while hooking only

one interrupt through the memory manager in protected mode

SeeAlso: AX=2010h,AX=2011h

-----s-812015-----

INT 81 - Gravis UltraSound - MEGA\_EM v3.0+ - ???

AX = 2015h

???

Return: ???

-----r-82-----

INT 82 - reserved for BASIC

SeeAlso: INT 81"BASIC",INT 83"BASIC"

-----N-82-----

INT 82 - IBM TOKEN RING ADAPTER - ???

AH = function

00h display message???

DS:BX -> string

???

Return: ???

SeeAlso: INT 81"TOKEN RING",INT 91"TOKEN RING"

-----r-83-----

INT 83 - reserved for BASIC

SeeAlso: INT 82"BASIC",INT 84"BASIC"

-----s-8300-----

INT 83 - JM Pro Tracker v5.0 - ???

AH = 00h

???

Return: CF clear (successful)

AX = ??? (0302h)

Program: JM Pro Tracker is a public-domain resident .MOD (digital music)

player by Josha Munnik

(Table 03976)

Values for JM Pro Tracker error code:

000Ah ???

000Bh ???

0014h ???

001Eh some required parameters have not yet been set

001Fh already playing???

FExxh busy (API call already in progress)

FFxxh invalid function number

-----s-8301-----

INT 83 - JM Pro Tracker v5.0 - ???

AH = 01h

DX = ??? or FFFFh

???

Return: CF clear if successful

CF set on error

AX = error code (see #03976)

-----s-8302-----

INT 83 - JM Pro Tracker v5.0 - ???

AH = 02h

BX = ???

???

Return: CF clear if successful

AX = ???

BX = ???

CX = ???

DX = ???

SI = ???

CF set on error

AX = error code (000Ah,000Bh,other) (see #03976)

-----s-8303-----

INT 83 - JM Pro Tracker v5.0 - ???

AH = 03h

BX = ??? (only low four bits, must be nonzero)

CX = ???

???

Return: CF clear if successful

CF set on error

AX = error code (0014h) (see #03976)

-----s-8304-----

INT 83 - JM Pro Tracker v5.0 - SET DMA BUFFER???

AH = 04h

CX = number of bytes in suggested buffer

ES:DI -> suggested buffer for DMA transfers???

Return: CF clear if successful

ES:AX -> actual buffer (filled with bytes of 80h)

CX = actual length (multiple of 4)

CF set on error

AX = error code (see #03976)

Note: the buffer must not cross a 64K DMA page boundary

-----s-8305-----

```
INT 83 - JM Pro Tracker v5.0 - SET ??? BUFFER
  AH = 05h
  CX = length of buffer
  ES:DI -> buffer for/containing ???
Return: CF clear (successful)
SeeAlso: AH=06h,AH=07h,AH=18h
-----s-8306-----
INT 83 - JM Pro Tracker v5.0 - SET ??? BUFFER
  AH = 06h
  CX = length of buffer
  ES:DI -> buffer for/containing ???
Return: CF clear (successful)
SeeAlso: AH=05h,AH=07h,AH=18h
-----s-8307-----
INT 83 - JM Pro Tracker v5.0 - SET ??? BUFFER
  AH = 07h
  BX = ???
  CX = length of buffer
  ES:DI -> buffer for/containing ???
Return: CF clear (successful)
SeeAlso: AH=05h,AH=06h,AH=18h
-----s-8308-----
INT 83 - JM Pro Tracker v5.0 - START PLAYING???
  AH = 08h
  ???
Return: CF clear if successful
  ???
  CF set on error
  AX = error code (001Eh,001Fh) (see #03976)
SeeAlso: AH=09h,AH=0Ah
-----s-8309-----
INT 83 - JM Pro Tracker v5.0 - PAUSE???
  AH = 09h
  ???
Return: CF clear if successful
  CF set on error
  AX = error code (see #03976)
SeeAlso: AH=08h,AH=0Ah
-----s-830A-----
INT 83 - JM Pro Tracker v5.0 - RESUME???
  AH = 0Ah
```

```
    ???
Return: CF clear if successful
    CF set on error
        AX = error code (001Fh) (see #03976)
SeeAlso: AH=08h,AH=09h
-----s-830B-----
INT 83 - JM Pro Tracker v5.0 - ???
    AH = 0Bh
    BX = ???
    ???
Return: CF clear if successful
    CF set on error
-----s-830C-----
INT 83 - JM Pro Tracker v5.0 - ???
    AH = 0Ch
    DX bit 5 = ???
    ???
Return: CF clear if successful
    BX = ??? (FFFFh)
    CX = ??? (0000h)
    DX = ???
    CF set on error
        AX = error code (see #03976)
-----s-830D-----
INT 83 - JM Pro Tracker v5.0 - ???
    AH = 0Dh
    BX = ???
    ???
Return: CF clear (successful)
-----s-830E-----
INT 83 - JM Pro Tracker v5.0 - ???
    AH = 0Eh
    BX = ??? (ignored if ES:DI = 0000h:0000h)
    ES:DI -> ??? or 0000h:0000h
Return: CF clear (successful)
-----s-830F-----
INT 83 - JM Pro Tracker v5.0 - ???
    AH = 0Fh
    ???
Return: CF clear if successful
    AX = ???
```



```
CF set on error
    AX = error code (see #03976)
-----s-8310-----
INT 83 - JM Pro Tracker v5.0 - ???
    AH = 10h
    BL = subfunction???
    CX = ???
    DX = ???
    ???
Return: CF clear if successful
    CX = DX = ???
    CF set on error
    AX = error code (see #03976)
SeeAlso: AH=18h
-----s-8311-----
INT 83 - JM Pro Tracker v5.0 - ???
    AH = 11h
    BL = ???
    CX = ???
Return: CF clear (successful)
-----s-8312-----
INT 83 - JM Pro Tracker v5.0 - CLEAR ???
    AH = 12h
Return: nothing
-----s-8313-----
INT 83 - JM Pro Tracker v5.0 - ???
    AH = 13h
    BX = ???
Return: CF clear (successful)
-----s-8314-----
INT 83 - JM Pro Tracker v5.0 - ???
    AH = 14h
    ???
Return: CF clear (successful)
    BX = ???
-----s-8315-----
INT 83 - JM Pro Tracker v5.0 - GET ???
    AH = 15h
    BX = what to get (zero/nonzero)
Return: CF clear if successful
    BX = ???
```

```

    DX = ???
    CF set on error
    AX = error code (see #03976)
-----s-8316-----
INT 83 - JM Pro Tracker v5.0 - ???
    AH = 16h
    AL = subfunction
    00h ???
    Return: BX = ???
    CX = ???
    DX = ???
    ES:DI -> ???
    CF indicates ???
    01h ???
    Return: BX = ???
    CX = ???
    DX = ???
Return: CF clear if successful
    CF set on error
    AX = error code (FFxxh) (see #03976)
-----s-8317-----
INT 83 - JM Pro Tracker v5.0 - UNHOOK API INTERRUPT
    AH = 17h
Return: CF clear if successful
    CF set on error (hooked by another program)
-----s-8318-----
INT 83 - JM Pro Tracker v5.0 - SET BUFFERS
    AH = 18h
    ES:DI -> buffer-pointer structure (see #03977)
Return: nothing
SeeAlso: AH=05h,AH=06h,AH=07h,AH=10h

```

Format of JM Pro Tracker buffer-pointer structure:

Offset	Size	Description (Table 03977)
00h	WORD	length of ??? buffer (see AH=05h)
02h	DWORD	-> ??? buffer (see AH=05h)
06h	WORD	length of ??? buffer (see AH=06h)
08h	DWORD	-> ??? buffer (see AH=06h)
0Ch	WORD	length of ??? buffer (see AH=07h)
0Eh	DWORD	-> ??? buffer (see AH=07h)
12h	WORD	??? (see AH=07h)

14h WORD ??? (see AH=10h) (subfunction 01h)  
16h WORD ??? (see AH=10h) (subfunction 01h)  
18h WORD ??? (see AH=10h) (subfunction 00h)  
1Ah WORD ??? (see AH=10h) (subfunction 01h)

-----r-84-----

INT 84 - reserved for BASIC

SeeAlso: INT 83"BASIC",INT 85"BASIC"

-----r-85-----

INT 85 - reserved for BASIC

Note: INT 80 through INT ED are modified but not restored by Direct Access  
v4.0, and may be left dangling by other programs written with the  
same version of compiled BASIC

SeeAlso: INT 84"BASIC",INT 86"BASIC"

-----N-86-----

INT 86 - NetBIOS - ORIGINAL INT 18

Note: some implementations of NetBIOS reportedly relocate INT 18 here

SeeAlso: INT 18"BOOT HOOK"

-----r-86-----

INT 86 - IBM ROM BASIC - used while in interpreter

Notes: called by ROM BASIC, but pointed at IRET by BASIC.COM/BASICA.COM

BASIC.COM/BASICA.COM do not restore vector on termination

SeeAlso: INT 85"BASIC",INT 87"BASIC"

-----r-86-----

INT 86 - APL\*PLUS/PC - Terminate APL session and return to DOS

SeeAlso: INT 21/AH=4Ch,INT 87"APL"

-----r-87-----

INT 87 - IBM ROM BASIC - used while in interpreter

Notes: called by ROM BASIC

BASIC.COM/BASICA.COM do not restore vector on termination

SeeAlso: INT 86"BASIC",INT 88"BASIC"

-----r-87-----

INT 87 - APL\*PLUS/PC - ???

SeeAlso: INT 86"APL",INT 88/AL=00h

-----v-87-----

INT 87 - VIRUS - "ZeroHunt" - VIRAL CODE (NOT A VECTOR!)

Note: the ZeroHunt virus copies its resident code down to 0000h:021Ch and  
following

SeeAlso: INT 8B"VIRUS"

-----r-88-----

INT 88 - IBM ROM BASIC - used while in interpreter

Notes: called by ROM BASIC, but pointed at IRET by BASIC.COM/BASICA.COM

BASIC.COM/BASICA.COM do not restore vector on termination

SeeAlso: INT 87"BASIC",INT 89"BASIC"

-----88-----

INT 88h - WANG PC - GET CONFIGURATION

AL = get info type

01h Get System Console

Return: ES:BX -> Configuration structure (see #04089)

else: ???

SeeAlso: PORT 1010h"Wang",MEM F000h:0000h,MEM FC00h:3FC2h,INT 21/AX=4402h

Format of Wang PC System Console configuration structure:

Offset Size Description (Table 04089)

00h WORD version

02h WORD memory size

04h 2 WORDs reserved

08h WORD number of screens (1..4)

0Ah 4 WORDs offset of console screen info structure 1..4 within  
configuration structure's segment (ES) (see #04090)

Format of Wang PC Screen Info structure:

Offset Size Description (Table 04090)

00h BYTE state

(bit 7 set appears to indicate the active console that is  
currently attached to the screen)

01h BYTE scanoff

02h WORD buffer segment

04h BYTE colors

05h BYTE row

06h BYTE column

07h BYTE attribute

00h = ???

02h = ???

0Fh = white on black

08h BYTE auxiliary mode

09h BYTE auxiliary mode 2

SeeAlso: #04089

-----r-88--00-----

INT 88 - APL\*PLUS/PC - CREATE OBJECT OF ARBITRARY RANK OR SHAPE

AL = 00h

BX = STPTR of the variable to be assigned

ES:SI -> model of type, rank, and shape (see #03978)

Return: ES:DI -> first data byte of object  
 DX:CX = number of elements in the object  
 SeeAlso: INT 88/AL=01h,INT 88/AL=08h,INT C8"APL"

Format of APL\*PLUS/PC shape model:

Offset Size Description (Table 03978)

00h BYTE type  
 01h character (2-byte dimension sizes)  
 02h integer (2-byte dimension sizes)  
 08h floating point (2-byte dimension sizes)  
 11h character (4-byte dimension sizes)  
 12h integer (4-byte dimension sizes)  
 18h floating point (4-byte dimension sizes)  
 01h BYTE rank  
 02h (D)WORD first dimension of shape  
 N (D)WORD second dimension of shape  
 ...

-----r-88--01-----

INT 88 - APL\*PLUS/PC - CREATE CHARACTER SCALAR/VECTOR/MATRIX <64K IN SIZE

AL = 01h  
 AH = rank  
 BX = STPTR of the variable to be assigned  
 CX = first dimension (if any)  
 DX = second dimension (if any)

Return: ES:DI -> object

CX = number of elements in the object

Note: each dimension must be 32767 or smaller

SeeAlso: AL=00h,AL=02h,AL=08h,INT C8"APL"

-----r-88--02-----

INT 88 - APL\*PLUS/PC - CREATE INTEGER SCALAR/VECTOR/MATRIX <64K IN SIZE

AL = 02h  
 AH = rank  
 BX = STPTR of the variable to be assigned  
 CX = first dimension (if any)  
 DX = second dimension (if any)

Return: ES:DI -> object

CX = number of elements in the object

Note: each dimension must be 32767 or smaller

SeeAlso: AL=01h,AL=08h,INT C8"APL"

-----r-88--08-----

INT 88 - APL\*PLUS/PC - CREATE FLOATING POINT SCALAR/VECTOR/MATRIX <64K IN SIZE

AL = 08h  
AH = rank  
BX = STPTR of the variable to be assigned  
CX = first dimension (if any)  
DX = second dimension (if any)

Return: ES:DI -> object

CX = number of elements in the object

Note: each dimension must be 32767 or smaller

SeeAlso: AL=01h,AL=02h,INT C8"APL"

-----r-88--F5-----

INT 88 - APL\*PLUS/PC - FORCE OBJECT INTO REAL WORKSPACE FROM VIRTUAL

AL = F5h

BX = STPTR of object

SeeAlso: INT C8"APL"

-----r-88--F6-----

INT 88 - APL\*PLUS/PC - MAKE NAME IMMUNE FROM OUTSWAPPING

AL = F6h

BX = STPTR of object

SeeAlso: AL=F7h,AL=F8h,INT C8"APL"

-----r-88--F7-----

INT 88 - APL\*PLUS/PC - MAKE NAME ELIGIBLE FOR OUTSWAPPING

AL = F7h

BX = STPTR of object

SeeAlso: AL=F6h,AL=F8h,INT C8"APL"

-----r-88--F8-----

INT 88 - APL\*PLUS/PC - REPORT WHETHER NAME IS ELIGIBLE FOR OUTSWAPPING

AL = F8h

BX = STPTR of object

Return: BX = name's outswapping status

0000h eligible

0001h not eligible

SeeAlso: AL=F6h,AL=F7h,INT C8"APL"

-----r-88--F9-----

INT 88 - APL\*PLUS/PC - DETERMINE NAME STATUS

AL = F9h

ES:SI -> name

CX = length of name

Return: CF set if name ill-formed or already in use

BX = STPTR if already in symbol table

CF clear if name is available for use

BX = 0000h

Note: does not force the name into the workspace

SeeAlso: AL=FEh,AL=FFh,INT C8"APL"

-----r-88--FC-----

INT 88 - APL\*PLUS/PC - DETERMINE IF MEMORY AVAIL WITHOUT GARBAGE COLLECTION

AL = FCh

BX = amount of memory needed (paragraphs)

Return: CF clear if memory available

CF set if a workspace compaction is required

SeeAlso: AL=FDh,INT C8"APL"

-----r-88--FD-----

INT 88 - APL\*PLUS/PC - PERFORM GARBAGE COLLECTION AND RETURN AVAILABLE MEMORY

AL = FDh

Return: BX = number of paragraphs available in workspace

SeeAlso: AL=FCh,INT C8"APL"

-----r-88--FE-----

INT 88 - APL\*PLUS/PC - CREATE NAME

AL = FEh

ES:SI -> name

CX = length of name

Return: BX = STPTR of name

DX = interpreter's data segment

SeeAlso: AL=F9h,AL=FFh,INT C8"APL"

-----r-88--FF-----

INT 88 - APL\*PLUS/PC - DETERMINE NAME STATUS

AL = FFh

ES:SI -> name

CX = length of name

Return: CF set if name ill-formed or already in use

BX = STPTR if already in symbol table

CF clear if name is available for use

BX = 0000h

Note: forces the name into the workspace and makes it immune from outswapping

SeeAlso: AL=F9h,AL=FEh,INT C8"APL"

-----r-89-----

INT 89 - IBM ROM BASIC - used while in interpreter

Notes: called by ROM BASIC

BASIC.COM/BASICA.COM do not restore vector on termination

SeeAlso: INT 88"BASIC",INT 8A"BASIC"

-----r-89-----

INT 89 - APL\*PLUS/PC - ???

Note: STSC moved its interrupts from 86h-8Ch to C6h-CCh, but did not delete

the older interrupts

SeeAlso: INT C9"APL"

-----r-8A-----

INT 8A - IBM ROM BASIC - used while in interpreter

Notes: called by ROM BASIC

BASIC.COM/BASICA.COM do not restore vector on termination

SeeAlso: INT 89"BASIC",INT 8B"BASIC"

-----r-8A-----

INT 8A - APL\*PLUS/PC - PRINT SCREEN

Note: same as INT 05

SeeAlso: INT 05"PRINT SCREEN",INT 8C"APL",INT CA"APL"

-----r-8B-----

INT 8B - IBM ROM BASIC - used while in interpreter

Notes: called by ROM BASIC, but pointed at IRET by BASIC.COM/BASICA.COM

BASIC.COM/BASICA.COM do not restore vector on termination

SeeAlso: INT 8A"BASIC",INT 8C"BASIC"

-----r-8B-----

INT 8B - APL\*PLUS/PC - BEEP

Note: same as printing a ^G via INT 21/AH=02h

SeeAlso: INT 21/AH=02h,INT CB"APL"

-----v-8B-----

INT 8B - VIRUS - "ZeroHunt" - INSTALLATION CHECK (NOT A VECTOR!)

Note: if the ZeroHunt virus is resident, this vector will contain either

EE83h:019Bh (ZH-411) or EE83h:019Fh (ZH-415)

SeeAlso: INT 70"VIRUS",INT 87"VIRUS",INT 9C"VIRUS"

-----r-8C-----

INT 8C - IBM ROM BASIC - used while in interpreter

Notes: called by ROM BASIC, but pointed at IRET by BASIC.COM/BASICA.COM

BASIC.COM/BASICA.COM do not restore vector on termination

SeeAlso: INT 8B"BASIC",INT 8D

-----r-8C-----

INT 8C - APL\*PLUS/PC - CLEAR SCREEN MEMORY

AX = flag

0000h do not save display attributes

0001h save attributes

SeeAlso: INT CC"APL"

-----r-8D-----

INT 8D - IBM ROM BASIC - used while in interpreter

Notes: called by ROM BASIC, but pointed at IRET by BASIC.COM/BASICA.COM

BASIC.COM/BASICA.COM do not restore vector on termination

SeeAlso: INT 8C,INT 8E



-----r-8E-----

INT 8E - IBM ROM BASIC - used while in interpreter

Notes: called by ROM BASIC, but pointed at IRET by BASIC.COM/BASICA.COM

BASIC.COM/BASICA.COM do not restore vector on termination

SeeAlso: INT 8D,INT 8F

-----r-8F-----

INT 8F - IBM ROM BASIC - used while in interpreter

Notes: called by ROM BASIC, but pointed at IRET by BASIC.COM/BASICA.COM

BASIC.COM/BASICA.COM do not restore vector on termination

SeeAlso: INT 8E,INT 90

-----r-90-----

INT 90 - IBM ROM BASIC - used while in interpreter

Notes: called by ROM BASIC, but pointed at IRET by BASIC.COM/BASICA.COM

BASIC.COM/BASICA.COM do not restore vector on termination

SeeAlso: INT 8F,INT 91

-----r-90-----

INT 90 - APL\*PLUS/PC - USED BY PORT 10 PRINTER DRIVER

-----r-91-----

INT 91 - IBM ROM BASIC - used while in interpreter

Notes: called by ROM BASIC, but pointed at IRET by BASIC.COM/BASICA.COM

BASIC.COM/BASICA.COM do not restore vector on termination

SeeAlso: INT 90,INT 92

-----N-91-----

INT 91 - IBM TOKEN RING ADAPTER - ???

SeeAlso: INT 81"TOKEN RING",INT 82"TOKEN RING",INT 93"TOKEN RING"

-----!---Section-----