

Interrupt List, part 15 of 18

Copyright (c) 1989-1999,2000 Ralf Brown

-----H-51-----

INT 51 - IRQ1 relocated by DESQview

Range: INT 51 to INT F9, selected automatically

Note: this is the default location for older versions; see INT 50"DESQview"

for details of interrupt relocation

SeeAlso: INT 50"DESQview",INT 54"DESQview",INT 58"DESQview"

-----H-51-----

INT 51 - IRQ1 relocated by IBM 3278 emulation control program

SeeAlso: INT 50"IBM 3278",INT 54"IBM 3278"

-----H-51-----

INT 51 - IRQ1 relocated by OS/2 v1.x

SeeAlso: INT 50"OS/2",INT 54"OS/2"

-----51-----

INT 51 - TI Professional PC - RESTART TIMING EVENT

AX = timer count in 25ms intervals

DS:DI -> timing-event table (see #03244)

Note: documented as "for system use only"; intended for multi-tasking software

SeeAlso: INT 50"TI Professional",INT 52"TI Professional"

Format of TI Professional PC timing event table:

Offset Size Description (Table 03244)

00h WORD offset of next event table entry

02h BYTE normally unused (FFh)

03h BYTE flags:

bit 7 set if timing event active

bits 6-0 not used by BIOS (0), but could be used by option ROMs

04h WORD timeout count (decremented every 25ms when active)

06h WORD offset of event handler (in segment F400h) to call on event timeout; the F400h segment allows addressing both system ROMs and the first 16K of memory (due to the 1M memory wraparound)

-----b-51-----

INT 51 - Tandy 2000 - KEYBOARD SERVICES

Note: this interrupt is identical to INT 16 on Tandy 2000

SeeAlso: INT 16/AH=00h,INT 16/AH=01h,INT 16/AH=02h,INT 16/AH=04h"Tandy"

SeeAlso: INT 16/AH=04h,INT 4A"Tandy 2000",INT 4C"Tandy 2000",INT 52"Tandy 2000"

-----H-52-----

INT 52 - IRQ2 relocated by DESQview

Range: INT 52 to INT FA, selected automatically

Note: this is the default location for older versions; see INT 50 "DESQview"

for details of interrupt relocation

SeeAlso: INT 50 "DESQview", INT 54 "DESQview", INT 58 "DESQview"

-----H-52-----

INT 52 - IRQ2 relocated by IBM 3278 emulation control program, OS/2 v1.x

SeeAlso: INT 50 "IBM 3278", INT 51 "OS/2"

-----52-----

INT 52 - TI Professional PC - CANCEL TIMING EVENT

DS:DI -> timing-event table (see #03244)

Note: documented as "for system use only"; intended for multi-tasking software

SeeAlso: INT 51 "TI Professional", INT 53 "TI Professional"

-----b-52-----

INT 52 - Tandy 2000 - VIDEO SERVICES

Note: this interrupt is identical to INT 10

SeeAlso: INT 10/AH=00h, INT 10/AH=01h, INT 10/AH=08h, INT 10/AH=0Eh

SeeAlso: INT 4A "Tandy 2000", INT 51 "Tandy 2000", INT 53 "Tandy 2000"

-----H-53-----

INT 53 - IRQ3 relocated by DESQview

Range: INT 53 to INT FB, selected automatically

Note: this is the default location for older versions; see INT 50 "DESQview" for details of interrupt relocation

SeeAlso: INT 50 "DESQview", INT 54 "DESQview", INT 58 "DESQview"

-----H-53-----

INT 53 - IRQ3 relocated by IBM 3278 emulation control program, OS/2 v1.x

SeeAlso: INT 50 "IBM 3278", INT 51 "OS/2"

-----53-----

INT 53 - TI Professional PC - SVC INTERFACE

Notes: documented as "for system use only"; intended for multi-tasking software

this interrupt is not used by the BIOS; the default handler generates a system error trap (see INT 51 "TI Professional")

SeeAlso: INT 50 "TI Professional", INT 54 "TI Professional"

-----b-53-----

INT 53 - Tandy 2000 - SERIAL COMMUNICATIONS

Note: this interrupt is identical to INT 14 on Tandy 2000

SeeAlso: INT 14/AH=00h "SERIAL", INT 14/AH=01h, INT 14/AH=02h, INT 14/AH=03h

SeeAlso: INT 14/AH=04h "Tandy 2000", INT 52 "Tandy 2000", INT 54 "Tandy 2000"

-----N-53-----

INT 53 - WEB??? - API

BX = function

```
0000h ???
AX = ???
Return: AX = ???
0004h ???
0009h ???
0015h
AX = ???
DX = ???
0017h
Return: ???
InstallCheck: check for the signature "WEBCO" immediately prior to the
interrupt handler
Note: the above calls are made by Show Partner F/X v3.6 (see INT 10/AH=53h)
Index: installation check;unknown|installation check;WEBCO
-----H-54-----
INT 54 - IRQ4 relocated by DESQview
Range: INT 54 to INT FC, selected automatically
Note: this is the default location for older versions; see INT 50"DESQview"
for details of interrupt relocation
SeeAlso: INT 50"DESQview",INT 58"DESQview"
-----H-54-----
INT 54 - IRQ4 relocated by IBM 3278 emulation control program, OS/2 v1.x
SeeAlso: INT 51"IBM 3278",INT 51"OS/2"
-----54-----
INT 54 - TI Professional PC - ACTIVATE TASK SUBROUTINE
Notes: documented as "for system use only"; intended for multi-tasking
software
this interrupt is not used by the BIOS; the default handler generates
a system error trap (see INT 51"TI Professional")
SeeAlso: INT 50"TI Professional",INT 53"TI Professional"
-----b-54-----
INT 54 - Tandy 2000 - LINE PRINTER
Note: this interrupt is identical to INT 17 on Tandy 2000
SeeAlso: INT 17/AH=00h,INT 17/AH=01h,INT 17/AH=02h,INT 4A"Tandy 2000"
SeeAlso: INT 53"Tandy 2000",INT 55"Tandy 2000"
-----X-545400-----
INT 54 U - Toshiba PCMCIA2 - INSTALLATION CHECK
AX = 5400h
Return: AX = 0054h if installed
CX:DX -> INT function handler
-----H-55-----
```

INT 55 - IRQ5 relocated by DESQview

Range: INT 55 to INT FD, selected automatically

Note: this is the default location for older versions; see INT 50"DESQview"

for details of interrupt relocation

SeeAlso: INT 50"DESQview",INT 58"DESQview"

-----H-55-----

INT 55 - IRQ5 relocated by IBM 3278 emulation control program, OS/2 v1.x

SeeAlso: INT 51"IBM 3278",INT 51"OS/2"

-----b-55-----

INT 55 - TI Professional PC - RESERVED FOR FUTURE USE

Notes: documented as "for system use only"; intended for multi-tasking software

this interrupt is not used by the BIOS; the default handler generates a system error trap (see INT 51"TI Professional")

SeeAlso: INT 50"TI Professional",INT 56"TI Professional"

-----b-55-----

INT 55 - Tandy 2000 - SYSTEM CLOCK

Note: this interrupt is identical to INT 1A on Tandy 2000

SeeAlso: INT 1A/AH=00h,INT 1A/AH=01h,INT 1A/AH=02h"Tandy 2000"

SeeAlso: INT 1A/AH=03h"Tandy 2000",INT 54"Tandy 2000",INT 56"Tandy 2000"

-----H-56-----

INT 56 - IRQ6 relocated by DESQview

Range: INT 56 to INT FE, selected automatically

Note: this is the default location for older versions; see INT 50"DESQview"

for details of interrupt relocation

SeeAlso: INT 50"DESQview",INT 58"DESQview"

-----H-56-----

INT 56 - IRQ6 relocated by IBM 3278 emulation control program, OS/2 v1.x

SeeAlso: INT 51"IBM 3278",INT 51"OS/2"

-----b-56-----

INT 56 - TI Professional PC - RESERVED FOR FUTURE USE

Notes: documented as "for system use only"; intended for multi-tasking software

this interrupt is not used by the BIOS; the default handler generates a system error trap (see INT 51"TI Professional")

SeeAlso: INT 50"TI Professional",INT 55"TI Professional"

-----b-56-----

INT 56 - Tandy 2000 - FLOPPY DISK SERVICES

Note: this interrupt is identical to INT 13 on Tandy 2000

SeeAlso: INT 13/AH=00h,INT 13/AH=01h,INT 13/AH=02h,INT 13/AH=03h

SeeAlso: INT 4A"Tandy 2000",INT 51"Tandy 2000",INT 55"Tandy 2000"

-----H-57-----

INT 57 - IRQ7 relocated by DESQview

Range: INT 57 to INT FF, selected automatically

Note: this is the default location for older versions; see INT 50"DESQview"
for details of interrupt relocation

SeeAlso: INT 50"DESQview",INT 58"DESQview"

-----H-57-----

INT 57 - IRQ7 relocated by IBM 3278 emulation control program, OS/2 v1.x

SeeAlso: INT 51"IBM 3278",INT 51"OS/2"

-----b-57-----

INT 57 C - TI Professional PC - CRT MAPPING HOOK

AX/BX/CX/DX/BP/SI/DI same as on entry to CRT subroutine (e.g. INT 49)

DS = BIOS system segment

ES = DE00h

Return: DF/IF flags must be preserved

ES,DS,BP preserved

AX,BX,CX,DX,SI,DI may be changed as necessary to modify the original
call

Desc: hooking this vector permits programs to intercept or modify all
screen output, including both application calls to INT 49 and
calls generated internally by the BIOS which bypass INT 49

Note: by default, this vector points at an IRET instruction

SeeAlso: INT 49/AH=01h"TI",INT 50"TI Professional"

-----H-58-----

INT 58 - IRQ8 relocated by DESQview 2.26+

Range: INT 58 to INT F8, selected automatically

Note: this is the default, but other INTs may be used (see INT 50"DESQview")

SeeAlso: INT 50"DESQview",INT 59"DESQview",INT 70

-----H-58-----

INT 58 - IRQ0 relocated by DoubleDOS

SeeAlso: INT 08

-----b-58-----

INT 58 C - TI Professional PC - SYSTEM TIMER 25ms HOOK

Desc: called from the hardware timer tick interrupt, after executing the
first four BIOS timing events, updating the system clock, invoking
INT 5A if required, saving registers, and switching to a temporary
stack (the one reserved for IRQ3)

Notes: the handler for this interrupt may destroy AX,BX,DI,ES but must
preserve all other registers; 8 WORDs of stack space are available,
of which at most 4 may be used if the handler enables interrupts
if the handler switches stacks (because more than 4/8 WORDs are

required), the original stack must be restored before chaining to
the previous handler

SeeAlso: INT 43"TI Professional",INT 4C"TI Professional"

SeeAlso: INT 5A"TI Professional"

-----H-59-----

INT 59 - IRQ9 relocated by DESQview 2.26+

Range: INT 59 to INT F9, selected automatically

Note: this is the default, but other INTs may be used (see INT 50"DESQview")

SeeAlso: INT 50"DESQview",INT 58"DESQview",INT 5A"DESQview",INT 71

-----H-59-----

INT 59 - IRQ1 relocated by DoubleDOS

SeeAlso: INT 09

-----b-59-----

INT 59 - TI Professional PC - COMMON ROM HARDWARE INTERRUPT EXIT VECTOR

Desc: all hardware interrupts on the TI Pro jump indirectly to the handler
pointed at by this interrupt vector to finish their handling of
the hardware interrupt

Notes: the default handler decrements the interrupt count, restores registers
(including the stack pointer), sends an EOI to the interrupt
controller, and finally does an IRET

can be used by multitaskers which need to get control after every
hardware interrupt

SeeAlso: INT 40"TI Professional",INT 47"TI Professional"

SeeAlso: INT 53"TI Professional"

-----V-59-----

INT 59 - GSS Computer Graphics Interface (GSS*CGI)

DS:DX -> block of 5 array pointers

Return: CF set on error

AX = error code

CF clear if successful

AX = return code

Note: INT 59 is the means by which GSS*CGI language bindings communicate with
GSS*CGI device drivers and the GSS*CGI device driver controller.

also used by the IBM Graphic Development Toolkit

-----H-5A-----

INT 5A - IRQ10 relocated by DESQview 2.26+

Range: INT 5A to INT FA, selected automatically

Note: this is the default, but other INTs may be used (see INT 50"DESQview")

SeeAlso: INT 50"DESQview",INT 59"DESQview",INT 5B"DESQview",INT 72

-----H-5A-----

INT 5A - IRQ2 relocated by DoubleDOS

SeeAlso: INT 0A"IRQ2"

-----N-5A-----

INT 5A - PC Cluster adapter BIOS entry address

???

Return: ???

SeeAlso: INT 5B"PC Cluster"

-----b-5A-----

INT 5A - TI Professional PC - SYSTEM TIMER 100ms HOOK

Desc: called from the hardware timer tick interrupt, after executing the first four BIOS timing events, updating the system clock, saving registers, and switching to a temporary stack (the one reserved for IRQ3), but before calling INT 58

no details available

Notes: this interrupt is invoked on every fourth timer interrupt the handler for this interrupt may destroy AX,BX,DI,ES but must preserve all other registers; 8 WORDs of stack space are available, of which at most 4 may be used if the handler enables interrupts if the handler switches stacks (because more than 4/8 WORDs are required), the original stack must be restored before chaining to the previous handler

SeeAlso: INT 43"TI Professional",INT 4C"TI Professional"

SeeAlso: INT 58"TI Professional"

-----H-5B-----

INT 5B - IRQ11 relocated by DESQview 2.26+

Range: INT 5B to INT FB, selected automatically

Note: this is the default, but other INTs may be used (see INT 50"DESQview")

SeeAlso: INT 50"DESQview",INT 5A"DESQview",INT 5C"DESQview",INT 73

-----H-5B-----

INT 5B - IRQ3 relocated by DoubleDOS

SeeAlso: INT 0B

-----N-5B-----

INT 5B - PC cluster adapter - RELOCATED INT 19

SeeAlso: INT 19,INT 5A"PC Cluster"

-----N-5B-----

INT 5B - AT&T Starlan Extended NetBIOS (variable length names)

ES:BX -> Network Control Block (see #03245)

Return: AL = status (see #03248)

SeeAlso: INT 5C"NetBIOS"

Format of Starlan Network Control Block:

Offset Size Description (Table 03245)

```

00h BYTE ncb_command (see also #03250)
    70h send net Break
01h BYTE ncb_retcode
02h BYTE ncb_lsn
03h BYTE ncb_num
04h DWORD -> ncb_buffer
08h WORD ncb_length
0Ah 16 BYTES ncb_callname
1Ah 16 BYTES ncb_name
2Ah BYTE ncb_rto
2Bh BYTE ncb_sto
2Ch DWORD -> ncb_post /* int (far *ncb_post) (); */
30h BYTE ncb_lana_num
31h BYTE ncb_cmd_cplt
32h DWORD -> ncb_vname
36h BYTE ncb_vnamelen
37h 9 BYTES ncb_reserve

```

Note: fields 00h-31h are the same as for a standard NetBIOS NCB (see #03249)

-----N-5B-----

INT 5B - Microsoft Network Transport Layer Interface

Note: used by MS-NET for executing network commands

SeeAlso: INT 5C"NetBIOS"

-----N-5B-----

INT 5B - used by Alloy NTNX

-----N-5B-----

INT 5B - ISOLAN Multi Protocol Software

ES:BX -> Transfer Control Block (see #03246)

Return: AL = status

Note: this software interface allows multiple protocols/software packages
to access a BICC 411x network card

Format of ISOLAN Transfer Control Block:

Offset Type Description (Table 03246)

```

00h BYTE command code
    B3h Status
    F2h Activate
    F3h Deactivate
    F4h Send Data
01h BYTE command identity
02h BYTE virtual circuit ID
03h WORD buffer length

```



```

05h  DWORD  buffer pointer
09h  BYTE   expedited data flag
0Ah  BYTE   cancelable flag
0Bh  16 BYTES local network address
1Bh  16 BYTES remote network address
2Bh  DWORD  asynchronous notification routine
30h  DWORD  local network number
34h  DWORD  remote network number
38h  BYTE   call timeout
39h  BYTE   not used
3Ah  8 BYTES reserved
42h  BYTE   command code extension
43h  WORD   Blue Book MAC type

```

```
-----b-5B-----
```

INT 5B C - TI Professional PC - KEYBOARD MAPPING HOOK

CF set

AH = shift state (see #03247)

AL = scan code (see #03214)

Return: BX, CX, DI, ES may be destroyed

various return methods are supported:

IRET, AX unchanged: process keystroke normally

IRET, AL = FFh: discard keystroke

IRET, AX changed: process modified keystroke

chain to old INT 5B: allow other handlers to look at (possibly modified) keystroke in AX

RETF 2, CF clear: place returned AX into keyboard buffer without any further processing

Notes: invoked by the keyboard ISR, and used to remap the keyboard if CF is clear on entry, some other handler has processed the keystroke and the current handler should not modify it, instead performing a RETF 2 or IRET (after clearing CF on the stack) when requesting that a value be placed directly into the keyboard buffer, AL and AH may not *both* be nonzero (the TI does not return scan codes as part of the key code for non-extended keys)

SeeAlso: INT 15/AH=4Fh, INT 4A/AH=00h"TI", INT 59"TI Professional"

SeeAlso: INT 5C"TI Professional", INT 5D"TI Professional"

SeeAlso: INT 5E"TI Professional", INT 5F"TI Professional"

Bitfields for TI Professional PC keyboard mapping hook shift states:

Bit(s) Description (Table 03247)

7 CAPS LOCK is on

6-4 reserved (0)
3 repeated key
2 Shift is pressed
1 Alt is pressed
0 Ctrl is pressed

-----U-5B5254DL04-----

INT 5B U - SitBack v3.02R - GET ???

AX = 5254h

DL = 04h

Return: ES:BX -> ??? in resident portion

Program: SitBack is a background file backup utility by SitBack Technologies,
Inc. which initiates backups whenever the system is idle

SeeAlso: AX=8485h/DL=71h,AX=8485h/DL=72h

-----U-5B8485DL70-----

INT 5B U - SitBack v3.02R - INSTALLATION CHECK

AX = 8485h

DL = 70h

Return: CX = 8485h if installed

DX:AX -> ??? (configuration data?)

Program: SitBack is a background file backup utility by SitBack Technologies,
Inc. which initiates backups whenever the system is idle

SeeAlso: AX=5254h/DL=04h,AX=8485h/DL=78h

-----U-5B8485DL71-----

INT 5B U - SitBack v3.02R - SET ??? FLAG AND GET ??? ADDRESS

AX = 8485h

DL = 71h

Return: ES:BX -> FAR entry point to ???

Note: the flag which is modified is located at the address returned by

AX=5254h/DL=04h

SeeAlso: AX=8485h/DL=72h

-----U-5B8485DL72-----

INT 5B U - SitBack v3.02R - CLEAR ??? FLAG

AX = 8485h

DL = 72h

Note: the flag which is modified is located at the address returned by

AX=5254h/DL=04h

SeeAlso: AX=8485h/DL=71h

-----U-5B8485DL73-----

INT 5B U - SitBack v3.02R - ???

AX = 8485h

DL = 73h

???

Return: ???

-----U-5B8485DL74-----

INT 5B U - SitBack v3.02R - ???

AX = 8485h

DL = 74h

???

Return: ???

-----U-5B8485DL75-----

INT 5B U - SitBack v3.02R - ???

AX = 8485h

DL = 75h

CX = ???

Return: ???

SeeAlso: AX=8485h/DL=76h

-----U-5B8485DL76-----

INT 5B U - SitBack v3.02R - ???

AX = 8485h

DL = 76h

CX = ???

Return: ???

Note: conditionally calls the code for AX=8485h/DL=75h

SeeAlso: AX=8485h/DL=75h

-----U-5B8485DL77-----

INT 5B U - SitBack v3.02R - SET ??? FLAG

AX = 8485h

DL = 77h

-----U-5B8485DL78-----

INT 5B U - SitBack v3.02R - GET RESIDENT DATA SEGMENT

AX = 8485h

DL = 78h

Return: CX = 5342h if supported

ES = AX = segment of TSR data

SeeAlso: AX=8485h/DL=70h,AX=8485h/DL=79h

-----U-5B8485DL79-----

INT 5B U - SitBack v3.02R - GET DTA

AX = 8485h

DL = 79h

Return: CX = 5342h if supported

ES:BX -> DTA set by last INT 21/AH=1Ah

Note: this function is provided by SBOS.EXE rather than SB.EXE

SeeAlso: INT 21/AH=1Ah

-----U-5B8485DL7A-----

INT 5B U - SitBack v3.02R - TOGGLE ???

AX = 8485h

DL = 7Ah

Return: CX = 5342h if supported

AL = new value of ??? (00h or 01h)

-----N-5C-----

INT 5C - NetBIOS INTERFACE

ES:BX -> network control block (NCB) (see #03249)

Return: AL = status (see #03248)

Program: NetBIOS was developed by Sytek, Inc. in 1984 as a high-level programming interface to the IBM PC Network; the first implementation was a ROM BIOS extension on Sytek's PCnet LAN adapter card, but many current networks support NetBIOS as the session layer.

Notes: The Sytek PCnet card uses DMA 3.

On some machines this vector is not initialized (e.g. Epson machines).

Hence this interrupt should only be called or hooked if the vector does not point to 0000h:0000h.

SeeAlso: INT 2A/AH=01h, INT 2A/AH=04h, INT 5B"Extended NetBIOS"

(Table 03248)

Values for NetBIOS status:

00h successful

01h bad buffer size

03h invalid NETBIOS command

05h timeout

06h receive buffer too small

07h No-ACK command failed

08h bad session number

09h LAN card out of memory

0Ah session closed

0Bh command has been cancelled

0Dh name already exists

0Eh local name table full

0Fh name still in use, can't delete

11h local session table full

12h remote PC not listening

13h bad NCB_NUM field

14h no answer to CALL or no such remote

15h name not in local name table

16h duplicate name
17h bad delete
18h abnormal end
19h name error, multiple identical names in use
1Ah bad packet
21h network card busy
22h too many commands queued
23h bad LAN card number
24h command finished while cancelling
26h command can't be cancelled
30h name defined by another process (OS/2)
34h NetBIOS environment not defined, must issue reset (OS/2)
35h required operating system resources exhausted (OS/2)
36h maximum applications exceeded (OS/2)
37h no SAPs available for NetBIOS (OS/2)
38h requested resources not available (OS/2)
40h Lana System Error
41h Lana Remote Hot Carrier
42h Lana Local Hot Carrier
43h Lana No Carrier Detected
44h unusual network condition
45h-4Dh hardware error
4Eh token ring is broken
4Fh token ring error
50h adapter malfunction
F7h error in explicit INITIALIZE
F8h error in implicit OPEN
F9h TOKREUI internal error
FAh hardware adapter testing
FBh NetBIOS emulator not found
FCh OPEN or OPEN_SAP failure
FDh unexpected adapter closure
FFh NetBIOS busy (command pending)

Format of NetBIOS Network Control Block:

Offset	Size	Description (Table 03249)
00h	BYTE	command code (see #03250)
01h	BYTE	return code (see #03248)
02h	BYTE	local session number (LSN)
03h	BYTE	"ncb_num" datagram table entry from ADD NAME
04h	DWORD	-> I/O buffer

08h WORD length of data in buffer
0Ah 16 BYTES remote system to call
1Ah 16 BYTES network name of local machine
2Ah BYTE receive timeout in 1/2 seconds
2Bh BYTE send timeout in 1/2 seconds
2Ch DWORD -> FAR post handler /* int (far *ncb_post)(); */
30h BYTE network adapter number on which to execute command
 00h-03h IBM NetBIOS specs
 F0h-FFh Eicon NABios interface (see also INT 7B"Eicon")
31h BYTE command completion code (see #03248)
32h 14 BYTES reserved for network card

(Table 03250)

Values for NetBIOS command code field in NCB:

10h start session with NCB_NAME name (call)
11h listen for call
12h end session with NCB_NAME name (hangu)
14h send data via NCB_LSN
15h receive data from a session
16h receive data from any session
17h send multiple data buffers
20h send unACKed message (datagram)
21h receive datagram
22h send broadcast datagram
23h receive broadcast datagram
30h add name to name table
31h delete name from name table
32h reset adapter card and tables
33h get adapter status (see #03251)
34h status of all sessions for name (see #03253)
35h cancel
36h add group name to name table
48h send data and receive data (LAN Manager NETBEUI.DOS)
70h unlink from IBM remote program (no F0h function)
71h send data without ACK
72h send multiple buffers without ACK
72h UngermannBass Register (conflicts with above function)
73h UngermannBass SendNmc
74h UngermannBass Callniu
75h UngermannBass Calladdr
76h UngermannBass Listenaddr

```

77h UngermannBass SendPkt
78h find name
78h UngermannBass RcvPkt (conflicts with above function)
79h token-ring protocol trace
79h UngermannBass SendAttn (conflicts with above function)
7Ah UngermannBass RcvAttn
7Bh UngermannBass Listenniu
7Ch UngermannBass RcvRaw
7Dh UngermannBass SendNmc2
7Fh Beame&Whiteside BWNB installation check (returns with return code and
    completion code both set to 03h, while invalid functions return only
    return code field set to 03h)

```

Note: OR any of the above except 70h with 80h for non-waiting call

Format of NetBIOS structure "astatus":

Offset Size Description (Table 03251)

```

00h 6 BYTES as_id
06h BYTE as_jumpers
07h BYTE as_post
08h BYTE as_major
09h BYTE as_minor
0Ah WORD as_interval
0Ch WORD as_crcerr
0Eh WORD as_algerr
10h WORD as_colerr
12h WORD as_abterr
14h DWORD as_tcount
18h DWORD as_rcount
1Ch WORD as_retran
1Eh WORD as_xresrc
20h 8 BYTES as_res0
28h WORD as_ncbfree
2Ah WORD as_ncbmax
2Ch WORD as_ncbx
2Eh 4 BYTES as_res1
32h WORD as_sespend
34h WORD as_msp
36h WORD as_sesmax
38h WORD as_bufsize
3Ah WORD as_names
3Ch 16 name structures as_name (see #03252)

```

Note: it has been reported that the first field should be 16 bytes instead of six (shifthing all remaining fields by ten bytes)

Format of NetBIOS structure "name":

Offset	Size	Description (Table 03252)
00h	16 BYTES	"nm_name" symbolic name
10h	BYTE	"nm_num" number associated with name
11h	BYTE	nm_status

Format of NetBIOS structure "sstatus":

Offset	Size	Description (Table 03253)
00h	BYTE	number of sessions being reported
01h	BYTE	number of sessions with this name
02h	BYTE	number of outstanding receive datagrams
03h	BYTE	number of outstanding ReceiveAnys
04h	var	session structures (see #03254)

Format of NetBIOS structure "session":

Offset	Size	Description (Table 03254)
00h	BYTE	local session number
01h	BYTE	state
01h		listen pending
02h		call pending
03h		session established
04h		hangup pending
05h		hangup done
06h		session aborted
02h	16 BYTES	local name
12h	16 BYTES	remote name
22h	BYTE	number of outstanding receives
23h	BYTE	number of outstanding sends/chainsends

-----H-5C-----

INT 5C - IRQ12 relocated by DESQview 2.26+

Range: INT 5C to INT FC, selected automatically

Note: this is the default, but other INTs may be used (see INT 50"DESQview")

SeeAlso: INT 50"DESQview",INT 5B"DESQview",INT 5D"DESQview",INT 74

-----H-5C-----

INT 5C - IRQ4 relocated by DoubleDOS

SeeAlso: INT 0C

-----N-5C-----

INT 5C - TOPS INTERFACE

ES:BX -> Network Control Block

Note: TOPS card uses DMA 1, 3 or none.

-----N-5C-----

INT 5C - ATALK.SYS - AppleTalk INTERFACE

DX:BX -> control block (see #03256)

Return: none

InstallCheck: test for the signature "AppleTalk" 16 bytes prior to the
interrupt handler

Range: INT 5Ch to INT 70h

Index: installation check;ATALK.SYS|installation check;AppleTalk interface

(Table 03255)

Values for ATALK.SYS command code:

01h	"AT_INIT"	initialize the driver
02h	"AT_KILL"	
03h	"AT_GETNETINFO"	get current network info incl init status
04h	"AT_GETCLOCKTICKS"	
05h	"AT_STARTTIMER"	
06h	"AT_RESETTIMER"	
07h	"AT_CANCELTIMER"	
10h	"LAP_INSTALL"	
11h	"LAP_REMOVE"	
12h	"LAP_WRITE"	
13h	"LAP_READ"	
14h	"LAP_CANCEL"	
20h	"DDP_OPENSOCKET"	
21h	"DDP_CLOSESOCKET"	
22h	"DDP_WRITE"	
23h	"DDP_READ"	
24h	"DDP_CANCEL"	
30h	"NBP_REGISTER"	
31h	"NBP_REMOVE"	
32h	"NBP_LOOKUP"	
33h	"NBP_CONFIRM"	
34h	"NBP_CANCEL"	
35h	"ZIP_GETZONELIST"	
36h	"ZIP_GETMYZONE"	
37h	"ZIP_TAKEDOWN"	
38h	"ZIP_BRINGUP"	
40h	"ATP_OPENSOCKET"	
41h	"ATP_CLOSESOCKET"	

```
42h "ATP_SENDREQUEST"
43h "ATP_GETREQUEST"
44h "ATP_SENDRESPONSE"
45h "ATP_ADDRESPONSE"
46h "ATP_CANCELTRANS"
47h "ATP_CANCELRESPONSE"
48h "ATP_CANCELREQUEST"
50h "ASP_GETPARMS"
51h "ASP_CLOSESESSION"
52h "ASP_CANCEL"
53h "ASP_INIT"
54h "ASP_KILL"
55h "ASP_GETSESSION"
56h "ASP_GETREQUEST"
57h "ASP_CMDREPLY"
58h "ASP_WRTCONTINUE"
59h "ASP_WRTREPLY"
5Ah "ASP_CLOSEREPLY"
5Bh "ASP_NEWSTATUS"
5Ch "ASP_ATTENTION"
5Dh "ASP_GETSTATUS"
5Eh "ASP_OPENSESSION"
5Fh "ASP_COMMAND"
60h "ASP_WRITE"
61h "ASP_GETATTENTION"
70h "PAP_OPEN"
71h "PAP_CLOSE"
72h "PAP_READ"
73h "PAP_WRITE"
74h "PAP_STATUS"
75h "PAP_REGNAME"
76h "PAP_REMNAME"
77h "PAP_INIT"
78h "PAP_NEWSTATUS"
79h "PAP_GETNEXTJOB"
7Ah "PAP_KILL"
7Bh "PAP_CANCEL"
```

Format of AppleTalk control block:

Offset Size Description (Table 03256)

00h WORD command code (see #03255)

```
    OR with the following flags
    8000h start command then return
    4000h wait for interrupt service to complete
02h  WORD  returned status
    0000h success (already initialized if func 01h)
04h  DWORD pointer to completion function
08h  WORD  network number
0Ah  BYTE  node ID
---if general func (01h,03h), control block continues:
0Bh  BYTE  "inf_abridge"
0Ch  WORD  "inf_config"
0Eh  DWORD pointer to buffer
12h  WORD  buffer size
---if DDP function (20h-24h), control block continues:
0Bh  BYTE  "ddp_addr_socket"
0Ch  BYTE  "ddp_socket"
0Dh  BYTE  "ddp_type"
0Eh  DWORD pointer to buffer
12h  WORD  buffer size
14h  BYTE  "ddp_chksum"
---if Name Binding Protocol (30h-34h), control block continues:
0Bh  BYTE  "nbp_addr_socket"
0Ch  WORD  "nbp_toget"
0Eh  DWORD pointer to buffer (see #03257)
12h  WORD  buffer size
14h  BYTE  "nbp_interval"
15h  BYTE  "nbp_retry"
16h  DWORD "nbp_entptr"
---if AppleTalk Transaction Protocol (42h), control block continues:
0Bh  BYTE  "atp_addr_socket"
0Ch  WORD  "atp_socket"
0Eh  DWORD pointer to buffer
12h  WORD  buffer size
14h  BYTE  "atp_interval"
15h  BYTE  "atp_retry"
16h  BYTE  ATP flags
    bit 5: exactly one transaction
17h  BYTE  "atp_seqbit"
18h  BYTE  transaction ID
19h  4 BYTES ATP user bytes
1Dh  BYTE  number of BDS buffers
```

1Eh BYTE number of BDS responses
 1Fh DWORD pointer to BDS buffers (see #03258)

Format of Name Binding Protocol Name-to-Address binding entries for NBP_LOOKUP:

Offset	Size	Description (Table 03257)
00h	WORD	"tup_address_network"
02h	BYTE	"tup_address_notid"
03h	BYTE	"tup_address_socket"
04h	BYTE	"tup_enum"
05h	99 BYTES	name

Format of BDS entries:

Offset	Size	Description (Table 03258)
00h	DWORD	pointer to buffer
04h	WORD	size of buffer
06h	WORD	BDS data size
08h	4 BYTES	"bds_userbytes"

-----N-5C-----

INT 5C - IBM 802.2 INTERFACE (LLC)
 ES:BX -> CCB (see #03259)

Return: none

Format of IBM 802.2 CCB:

Offset	Size	Description (Table 03259)
00h	BYTE	adapter
01h	BYTE	command code
02h	BYTE	return code
03h	BYTE	work
04h	DWORD	pointer to ???
08h	DWORD	pointer to completion function???
0Ch	DWORD	pointer to parameters???

-----N-5C-----

INT 5C - \$25 LAN - INSTALLATION CHECK

Notes: current versions only check whether the vector is 0000h:0000h or not
 future versions are supposed to have the signature "NET" in the three
 bytes preceding the INT 5C handler

-----b-5C0100-----

INT 5C C - TI Professional PC - KEYBOARD PAUSE KEY VECTOR

AX = 0100h
 CF clear

Return: CF clear

AX = keystroke to be placed into keyboard buffer
CF set
AX ignored
Desc: toggle a pause flag which is checked by the CRT Device Service Routine
(see INT 49/AH=01h"TI") and causes it to temporarily halt the
machine on the next video-related function (until a key is pressed)
SeeAlso: INT 09"IRQ1",INT 4A/AH=00h"TI",INT 59"TI Professional"
SeeAlso: INT 5B"TI Professional",INT 5D"TI Professional"
SeeAlso: INT 5E"TI Professional",INT 5F"TI Professional"
-----N-5C04-----
INT 5C - \$25 LAN - CHECK IF CONNECTION ALIVE
AH = 04h
AL = COM port (0 = default)
CX = wait count in character times (should be at least 100)
Return: ZF set if link alive
-----H-5D-----
INT 5D - IRQ13 relocated by DESQview 2.26+
Range: INT 5D to INT FD, selected automatically
Note: this is the default, but other INTs may be used (see INT 50"DESQview")
SeeAlso: INT 50"DESQview",INT 5C"DESQview",INT 5E"DESQview",INT 75
-----H-5D-----
INT 5D - IRQ5 relocated by DoubleDOS
SeeAlso: INT 0D"IRQ5",INT 5C"DoubleDOS"
-----b-5D-----
INT 5D C - TI Professional PC - KEYBOARD BREAK KEY VECTOR
CF clear
AX = 0000h
Return: CF clear
AX = keystroke to place into keyboard buffer
CF set
AX ignored
Desc: invoked by the keyboard ISR when Shift-BrkPause is pressed
Note: the default handler is a simple IRET instruction
SeeAlso: INT 09"IRQ1",INT 4A/AH=00h"TI",INT 59"TI Professional"
SeeAlso: INT 5B"TI Professional",INT 5C"TI Professional"
SeeAlso: INT 5E"TI Professional",INT 5F"TI Professional"
-----H-5E-----
INT 5E - IRQ14 relocated by DESQview 2.26+
Range: INT 5E to INT FE, selected automatically
Note: this is the default, but other INTs may be used (see INT 50"DESQview")
SeeAlso: INT 50"DESQview",INT 5D"DESQview",INT 5F"DESQview",INT 76

-----H-5E-----

INT 5E - IRQ6 relocated by DoubleDOS

SeeAlso: INT 0E,INT 5D"DoubleDOS"

-----b-5E-----

INT 5E C - TI Professional PC - KEYBOARD PRINT-SCREEN VECTOR

CF set

Return: CF clear

AX = keystroke to be placed into keyboard buffer

CF set

AX ignored

Desc: hook to dump the screen to the printer

Notes: hooked by TI MS-DOS, which provides a PRTSCRN character device which can invoke screen prints when a decimal digit (indicating the type of screen dump) is written to it

the TI MS-DOS print-screen routine allows selective dumps of text only, graphics only, or text and graphics superimposed, in either normal or reverse, by pressing the appropriate keystroke combination: Shift-Print, Ctrl-Print, Alt-Print, Shift-Alt-Print, Ctrl-Alt-Print, or Shift-Ctrl-Print.

the BIOS default routine for this vector is an IRET

SeeAlso: INT 05"PRINT SCREEN",INT 4A/AH=00h"TI",INT 59"TI Professional"

SeeAlso: INT 5B"TI Professional",INT 5C"TI Professional"

SeeAlso: INT 5D"TI Professional",INT 5F"TI Professional"

-----H-5F-----

INT 5F - IRQ15 relocated by DESQview 2.26+

Range: INT 5F to INT FF, selected automatically

Note: this is the default, but other INTs may be used (see INT 50"DESQview")

SeeAlso: INT 50"DESQview",INT 5E"DESQview",INT 77

-----H-5F-----

INT 5F - IRQ7 relocated by DoubleDOS

SeeAlso: INT 0F,INT 5E"DoubleDOS"

-----b-5F-----

INT 5F C - TI Professional PC - KEYBOARD QUEUEING VECTOR

Return: all registers preserved

Desc: hook for multitaskers to be informed when a keypress is placed in the keyboard buffer

Note: the default handler is a simple IRET instruction

SeeAlso: INT 09,INT 4A/AH=00h"TI",INT 59"TI Professional"

SeeAlso: INT 5B"TI Professional",INT 5C"TI Professional"

SeeAlso: INT 5D"TI Professional",INT 5E"TI Professional"

-----b-5F00-----

INT 5F - HP 95LX/100LX/200LX GRAPHICS PRIMITIVES - SET VIDEO MODE

AH = 00h

AL = video mode

03h text,CGA color (100LX/200LX)

06h 640x200 CGA graphics (100LX/200LX)

07h text, system manager compliant

20h 240x128 mono graphics, system manager compliant

87h text, not system manager compliant

A0h 240x128 mono graphics, not system manager compliant

Notes: the defaults after setting the mode to graphics are (0,0) logical origin, full-screen clip region, (0,0) pen location, pen color 1, pixel replacement FORCE, line type and fill mask all bits set modes 03h and 06h can also be set with the standard INT 10/AH=00h

SeeAlso: INT 0F"HP 95LX",INT 10/AH=00h,INT 15/AX=4DD4h

-----b-5F01-----

INT 5F - HP 95LX/100LX/200LX GRAPHICS PRIMITIVES - SET FILL MASK

AH = 01h

ES:DI -> 8-byte fill mask

Note: the fill mask represents an 8x8 pixel box and is repeated as necessary when drawing filled rectangles; it is always aligned with the byte boundaries of video memory, regardless of the actual boundaries of the rectangle

SeeAlso: AH=02h

-----b-5F02-----

INT 5F - HP 95LX/100LX/200LX GRAPHICS PRIMITIVES - GET CURRENT GRAPHICS INFO

AH = 02h

ES:DI -> graphics info record (see #03260)

Return: DX:AX -> filled graphics info record (for return to high-level langs)

Format of HP 95LX graphics info record:

Offset Size Description (Table 03260)

00h BYTE current video mode

01h BYTE default video mode

02h WORD display width in pixels

04h WORD display height in pixels

06h WORD current pen column

08h WORD current pen row

0Ah WORD current line type

0Ch WORD current replacement rule

0Eh WORD current pen color

10h WORD current leftmost column of clip region

12h WORD current rightmost column of clip region
14h WORD current topmost row of clip region
16h WORD current bottommost row of clip region
18h WORD current column of logical origin
1Ah WORD current row of logical origin
1Ch 8 BYTES current fill mask

-----b-5F03-----

INT 5F - HP 95LX/100LX/200LX GRAPHICS PRIMITIVES - SET LOGICAL ORIGIN

AH = 03h
CX = column
DX = row

SeeAlso: AH=04h

-----b-5F04-----

INT 5F - HP 95LX/100LX/200LX GRAPHICS PRIMITIVES - SET CLIP REGION

AH = 04h
CX = left-most column
DX = top-most row
SI = right-most column
DI = bottom-most row

SeeAlso: AH=03h

-----b-5F05-----

INT 5F - HP 95LX/100LX/200LX GRAPHICS PRIMITIVES - DRAW RECTANGLE

AH = 05h
AL = fill type
 00h outline, using current line type and color
 01h solid, using current color
 02h pattern, using current fill mask and color
DX,CX = row,column of other corner of rectangle

Note: the rectangle is drawn starting at the current pen position

SeeAlso: AH=01h,AH=06h,AH=07h

-----b-5F06-----

INT 5F - HP 95LX/100LX/200LX GRAPHICS PRIMITIVES - DRAW LINE

AH = 06h
DX,CX = row,column of end point

Note: the line is drawn starting at the current pen position

SeeAlso: AH=05h,AH=07h

-----b-5F07-----

INT 5F - HP 95LX/100LX/200LX GRAPHICS PRIMITIVES - PLOT POINT

AH = 07h
DX,CX = row,column of point

Note: also sets pen position to the specified point

SeeAlso: AH=06h,AH=08h,AH=0Ch

-----b-5F08-----

INT 5F - HP 95LX/100LX/200LX GRAPHICS PRIMITIVES - MOVE PEN

AH = 08h

DX,CX = row,column of new pen position

SeeAlso: AH=07h,AH=09h

-----b-5F09-----

INT 5F - HP 95LX/100LX/200LX GRAPHICS PRIMITIVES - SET PEN COLOR

AH = 09h

AL = new color (00h = white, 01h = black)

SeeAlso: AH=08h,AH=0Ah,AH=0Bh

-----b-5F0A-----

INT 5F - HP 95LX/100LX/200LX GRAPHICS PRIMITIVES - SET REPLACEMENT RULE

AH = 0Ah

AL = new replacement rule

00h force

01h AND

02h OR

03h XOR

---100LX/200LX---

04h InvForce

05h InvAND

06h InvOR

07h InvXOR

08h Txt

SeeAlso: AH=01h,AH=09h,AH=0Bh

-----b-5F0B-----

INT 5F - HP 95LX/100LX/200LX GRAPHICS PRIMITIVES - SET LINE TYPE

AH = 0Bh

CX = new line type

Note: the line type specifies 16 bits which are repeated over and over while
drawing the pixels of a line

SeeAlso: AH=09h,AH=0Ah

-----b-5F0C-----

INT 5F - HP 95LX/100LX/200LX GRAPHICS PRIMITIVES - GET PIXEL

AH = 0Ch

DX,CX = row,column of pixel to read

Return: AX = pixel color

SeeAlso: AH=07h

-----b-5F0D-----

INT 5F - HP 95LX/100LX/200LX GRAPHICS PRIMITIVES - GET IMAGE

AH = 0Dh
 DX,CX = row,column of first corner
 BP,SI = row,column of second corner
 ES:DI -> image buffer (see #03261)

Note: the specified corners are included in the saved image

SeeAlso: AH=0Eh

Format of HP 95LX image buffer:

Offset Size Description (Table 03261)

00h WORD number of planes (always 01h on HP 95LX)
 02h WORD number of bits/pixel (always 01h on HP 95LX)
 04h WORD image width in pixels
 06h WORD image height in pixels
 08h N BYTES image data
 requires (WIDTH+7)/8 * HEIGHT bytes

-----b-5F0E-----

INT 5F - HP 95LX/100LX/200LX GRAPHICS PRIMITIVES - PUT IMAGE

AH = 0Eh
 AL = replacement rule (see #03262)
 DX,CX = row,column of top left corner
 ES:DI -> image buffer (see #03261)

Note: if the specified image does not fit completely on the screen, this call
 does nothing

SeeAlso: AH=0Dh

Bitfields for replacement rule:

Bit(s) Description (Table 03262)

2 invert image before applying rule
 1-0 function (00 force, 01 AND, 10 OR, 11 XOR)

-----b-5F0F-----

INT 5F - HP 95LX/100LX/200LX GRAPHICS PRIMITIVES - WRITE TEXT

AH = 0Fh
 AL = rotate flag (if nonzero, rotate 90 degrees counter-clockwise)
 DX,CX = row,column of first character's top left corner
 ES:DI -> ASCIZ text

-----b-5F10-----

INT 5F - HP 100LX/200LX GRAPHICS PRIMITIVES - GET FONT POINTER

AH = 10h
 CX = font size of desired font
 0808h 8x8 small (80x25 text)
 0A0Bh 11x10 medium (64x18 text)

100Ch 12x16 large (40x16 text)

Return: DX:AX -> ptr to font or 0000h:fontID# if built-in font

SeeAlso: AH=11h

-----b-5F11-----

INT 5F - HP 100LX/200LX GRAPHICS PRIMITIVES - SET CURRENT FONT

AH = 11h

ES:DI -> ptr to font or 0000h:fontID# for built-in font

Note: this function should be called immediately after AH=10h with the
pointer supplied by that call

SeeAlso: AH=10h

-----*-60-----

INT 60 - reserved for user interrupt

-----v-60-----

INT 60 - VIRUS - "Zero Bug" - INSTALLATION CHECK

Desc: The "Zero Bug" virus hooks this vector. It considers itself installed
if offset 103h of the handler's segment contains the bytes "ZE"

SeeAlso: INT 32"VIRUS",INT 44"VIRUS",INT 61"SEMTEX"

-----d-60-----

INT 60 - Adaptec and OMTI controllers - DRIVE 0 DATA

SeeAlso: INT 41"HARD DISK 0",INT 61"Adaptec",INT 62"Adaptec",INT 63"Adaptec"

SeeAlso: INT 64"Adaptec",INT C0"AMI",#00732 at INT 1A/AX=B102h

Notes: this vector stores the first four bytes of the parameter table for
hard disk 0

these vectors are used by the following Adaptec controllers:

ACB 2370 A/B/C, ACB 2372 A/B/C, ACB 2333 A/B, 2322B-8, 2322B-16

these vectors are NOT used by the following Adaptec controllers:

ACB 2310, ACB 2312, ACB 2320D, ACB 2322D

-----b-60-----

INT 60 - TI Professional PC - SYSTEM ROM DATA AREA POINTER (NOT A VECTOR!)

Desc: the low word of this vector contains the segment of the RAM data area
to be used by the system ROM at F400h:A000h, and the high word
contains the length of the data area (see #03263)

SeeAlso: INT 61"TI Professional PC",INT 66"TI Professional PC"

Format of TI Professional System ROM data area:

Offset Size Description (Table 03263)

00h	BYTE	port 00h shadow
01h	BYTE	port 03h shadow
02h	BYTE	port 04h shadow
03h	WORD	system configuration word (see #03266)
05h	BYTE	25ms count

06h WORD offset of timing event 1 (0008h)
08h WORD (event 1) offset of timing event 2 (0010h)
0Ah BYTE (event 1) unused (FFh)
0Bh BYTE (event 1) active flag (bit 7 set if on)
0Ch WORD (event 1) count-down until end of event
0Eh WORD (event 1) event routine (in segment F400h) to call at timeout
10h WORD (event 2) offset of timing event 3 (0018h)
12h BYTE (event 2) unused (FFh)
13h BYTE (event 2) active flag (bit 7 set if on)
14h WORD (event 2) count-down until end of event
16h WORD (event 2) event routine (in segment F400h) to call at timeout
18h WORD (event 3) offset of timing event 3 (0020h)
1Ah BYTE (event 3) unused (FFh)
1Bh BYTE (event 3) active flag (bit 7 set if on)
1Ch WORD (event 3) count-down until end of event
1Eh WORD (event 3) event routine (in segment F400h) to call at timeout
20h WORD (event 4) 0000h - last event
22h BYTE (event 4) unused (FFh)
23h BYTE (event 4) active flag (bit 7 set if on)
24h WORD (event 4) count-down until end of event
26h WORD (event 4) event routine (in segment F400h) to call at timeout
28h WORD offset of start of text display within segment DE00h
2Ah WORD end of display
2Ch WORD current absolute cursor position
2Eh WORD start of current character row
30h WORD current cursor column
32h WORD cursor type and size (see #03264)
34h WORD start of protected status region on screen (0000h = none)
36h BYTE pop flag used by some ROM routines
 00h pop registers before return
 nonzero: do not pop
37h BYTE unused (FFh)
38h BYTE PAUSE flag
 00h off
 FFh on
39h BYTE temp: current attribute while moving characters on screen
3Ah WORD start of keyboard queue (next key at start+2 or 0042h if 0060h)
3Ch WORD end of keyboard queue (next key inserted at end+2 or 0042h)
3Eh BYTE number of characters in buffer
3Fh BYTE keyboard mode flags (see #03265)
40h BYTE Alt-digit-digit-digit accumulator

41h BYTE Alt-digit-digit-digit count of digits (mod 3)
42h 16 WORDs keyboard buffer (circular queue)
62h 8 DWORDs Drive Interface Table pointers for floppies A:-D: + four others
(0000h:0000h if pointer not in use)
82h BYTE disk-error retry count (incremented on each disk error)
83h 2 BYTES BIOS scratch space
85h 8 BYTES current cylinder number for drives 0-7
FFh = unknown
8Dh BYTE BIOS scratch space
8Eh BYTE hard error code for last disk access
00h operation successful
8Fh 8 BYTES save area for disk DSR parameters
97h 6 BYTES current drive status
9Dh BYTE retry error status
9Eh BYTE BIOS scratch space
9Fh 7 BYTES ???
A6h 2 WORDs INT 47 save area for SS,SP
AAh 2 WORDs INT 46 save area for SS,SP
AEh 2 WORDs INT 43 save area for SS,SP
B2h 2 WORDs save area for SS,SP to be restored by common intr. exit routine
B6h 24 WORDs stack for INT 47
E6h 24 WORDs stack for INT 46
116h 17 WORDs stack for INT 43
13Ah 6 BYTES system date and time (hundredths, seconds, minutes, hours, and
WORD days since 01jan1980)

Note: timing event 1 is the disk I/O timeout, event 2 is the floppy disk
motor timeout, event 3 is the floppy disk motor spin-up time, and
event 4 is used to turn off the speaker after a delay

Bitfields for TI Professional PC cursor size and type:

Bit(s) Description (Table 03264)

15 reserved (0)
14-13 cursor type
00 nonblinking
01 off
10 slow blink
11 fast blink
12-8 starting scan line (0-11)
7-5 reserved (0)
4-0 ending scan line (0-11)

SeeAlso: #03263

Bitfields for TI Professional PC keyboard mode flags:

Bit(s) Description (Table 03265)
7 CapsLock was ON at last keypress
6-4 reserved (0)
3 repeat key
2 Shift was down at last keypress
1 Alt was down at last keypress
0 Ctrl was down at last keypress

SeeAlso: #03263

Bitfields for TI Professional System Configuration Word:

Bit(s) Description (Table 03266)
0 drive A: installed
1 drive B: installed
2 drive C: installed
3 drive D: installed
4 drive A: is 80-track
5 drive A: is double-sided
6 60-Hz (USA,etc.) system instead of 50-Hz (Europe)
7 hard disk installed
8 serial port 1 installed
9 serial port 2 installed
10 serial port 3 installed
11 serial port 4 installed
14-12 installed graphics planes
000 none
001 plane A
111 planes A, B, and C
15 clock installed

SeeAlso: #03263

-----b-60-----

INT 60 - Atari Portfolio - USER INTERFACE FUNCTIONS

Desc: supplies a number of subfunctions which perform such functions as
drawing boxes and menus, and provide input line editing

SeeAlso: INT 61/AH=00h"Atari",INT 61"EXTENDED BIOS"

-----V-60-----

INT 60 - Nabbit v2.0 - (NOT A VECTOR!) - INSTALLATION CHECK

Program: Nabbit is a shareware resident screen data grabber by RSE Inc.

Range: INT 60 to INT 66, selected by searching for first free vector

Note: the Nabbit installation check consists of testing whether the

interrupt vector points at the ASCIZ signature string "iG"
(69h 47h 00h)

-----V-60-----

INT 60 - ATI M64VBE.COM - INSTALLATION SIGNATURE

Program: M64VBE is a VESA VBE 2.0 driver TSR for ATI's Mach64 video chip

InstallCheck: scan for an interrupt with the ASCIZ signature "M64VBE" three
bytes past the interrupt handler

Range: INT 60 to INT 66, selected by searching for first free vector

SeeAlso: INT 10/AX=4FDDh"M64VBE",INT 10/AX=4FFFh/BX=364Dh"M64VBE"

-----V-60-----

INT 60 U - Buffit v3.0 - (NOT A VECTOR!) - INSTALLATION CHECK

Program: Buffit is a shareware scrollback utility by D.T. Hamilton

Range: INT 60 to INT 6F, selected by searching for first free vector

Notes: the Buffit installation check consists of testing whether the

interrupt vector points at the ASCII signature "Buffit "

there is a private entry point (see #03267) immediately following the
signature string, i.e. eight bytes beyond the address pointed at
by the interrupt vector

(Table 03267)

Call Buffit private entry point with:

AH = function

00h get information and hotkey state

01h get information and toggle hotkey state

Return: AH = new hotkey state (00h enabled, 01h disabled)

AL = hotkey scan code (see #00006)

BH = hotkey shift states

BL = ??? (01h)

CX = segment of resident code

DH = interrupt number used for signature pointer

DL = ??? (00h)

SI = INT 09 handler offset

DI = INT 21 handler offset

Index: hotkeys;Buffit

-----r-60-----

INT 60 - PC-IPC API

STACK: DWORD pointer to parameter block (see #03268)

Return: STACK: unchanged

Program: PC-IPC is a shareware TSR by Donnelly Software Engineering which
allows communication between independent programs

Range: INT 00 to INT FF, selected by commandline switch

Format of PC-IPC parameter block:

Offset Size Description (Table 03268)

00h WORD caller's ID
 02h WORD to ID
 04h WORD command code (see #03269)
 06h WORD returned status (see #03270)
 08h WORD returned error code (see #03271)
 0Ah WORD size of data
 0Ch DWORD pointer to data buffer

(Table 03269)

Values for PC-IPC command code:

01h "IPC_CMND_INQUIRE" inquire current status
 set status field, writes WORD to data buffer containing free
 message space in bytes, and sets the "size" field to the
 number of messages waiting
 02h "IPC_CMND_ENABLE" reenable PC-IPC
 ignored unless called with the same ID that disabled PC-IPC
 03h "IPC_CMND_DISABLE" disable PC-IPC
 04h "IPC_CMND_INSTALL" reset PC-IPC
 06h "IPC_CMND_RDATA" read data
 returns first message in data buffer, sets "size" to message length
 and "to ID" field to sender's ID
 if no messages available, bit 4 of status is cleared and "size" is
 set to zero
 07h "IPC_CMND_SDATA" send data
 08h "IPC_CMND_REQID" require user ID
 create a new recognized ID and return in "caller's ID" field
 09h "IPC_CMND_DELID" cancel user ID
 delete caller's ID from pool of recognized IDs
 0Ah "IPC_CMND_RDATAW" read data, wait if no messages available
 0Bh "IPC_CMND_VERS" get PC-IPC version
 string representing version returned in data buffer, "size" field
 set to length of string

Bitfields for returned status:

Bit(s) Description (Table 03270)

0 unused
 1 IPC enabled
 2 IPC installed

3 error
4 message(s) available

(Table 03271)

Values for PC-IPC error code:

00h no error
01h invalid command or parameter
02h only process 0 can install/reset IPC
03h process can not install/reset IPC
04h IPC is not enabled
05h process can not disable IPC
06h invalid destination process ID
07h invalid sending process ID
08h invalid data destination
09h no more process IDs available
0Ah can not relinquish that process ID
0Bh message space is full
0Ch IPC is not installed

-----R-60-----

INT 60 - Tangram Arbiter - API

Desc: Arbiter makes a PC disk look like a slow disk over an SNA link to an IBM mainframe

Range: INT 60h to INT 66h, selected by configuration parameter

Notes: identified by string "@ARB_API" immediately following a short jump at the interrupt handler address

-----N-60-----

INT 60 - Excelan LAN Workplace for DOS 3.5 - API

ES:BX -> request packet (see #03272)

Return: request packet updated

InstallCheck: test for the WORD 4142h ('AB') immediately preceding the interrupt handler

Note: this interrupt is also supported by Beame&Whiteside's BWLWP35 shim, which was used in creating this description

BUG: because BWLWP35 range-checks only the low byte of the function number, and has a fencepost error even in that test, functions 000Bh and XX01h-XX0Bh (XX nonzero) branch to random locations

SeeAlso: INT 2F/AX=7A40h

Format of Excelan request packet:

Offset Size Description (Table 03272)

00h 12 BYTES ???

```
0Ch WORD (ret) error code (see #03273)
0Eh DWORD -> FAR function for ???
12h WORD function number
    0001h ???
    0002h NOP
    0003h NOP
    0004h NOP
    0005h ???
    0006h get ??? record
    0007h NOP
    0008h reset ???
    0009h NOP
    000Ah set ???
    ???
---function 01h---
20h BYTE (call) subfunction (32h-3Bh)
    3Bh non-blocking I/O request (will be tested every clock tick)
21h BYTE (ret) error code
    00h successful
    09h invalid connection number
    2Ah bad connection type
    45h ???
---function 01h, subfunction 32h---
3Ah WORD (call) connection type (01h stream, 02h datagram)
---function 01h, subfunction 34h---
26h WORD (call) ???
28h WORD (call) ???
2Ah WORD (call) ???
---function 01h, subfunction 35h---
1Ah WORD (call) connection number???
26h WORD (ret) ???
---function 01h, subfunction 36h---
1Ah WORD (call) connection number???
38h WORD ???
---function 01h, subfunction 37h---
24h WORD (ret) ???
26h WORD (ret) ???
---function 01h, subfunction 38h---
1Ah WORD (call) connection number???
---function 01h, subfunction 3Ah---
22h WORD (call) ???
```

```

667Eh ???
667Fh ???
24h BYTE (call 667Eh) ???
24h WORD (return 667Fh) ???
---function 01h, subfunction 3Bh---
0Eh DWORD (call) -> function to invoke for I/O or 0000h:0000h
    function called with AX = 0000h
        STACK: DWORD -> request packet
        WORD 0000h
        should return STACK unchanged
1Ah WORD (call) connection number???
21h BYTE (ret) set to 01h when I/O becomes possible
22h BYTE (call) direction (00h write, 01h read)
34h DWORD (ret) -> next pending request packet
---function 05h---
1Eh WORD (call) ???
20h WORD (call) ???
34h DWORD (call) -> ???
---function 06h---
16h DWORD (call) -> buffer for ??? record (see #03274)
1Ah WORD (call) number of bytes to copy
22h WORD (ret) number of bytes transferred
---function 08h---
14h WORD (ret) ??? (0001h)
---function 0Ah---
16h DWORD (call) -> WORD ???
1Ch WORD (call) must be 000Ah for BWLWP35

```

(Table 03273)

Values for Excelan error code:

```

0000h successful
002Dh invalid function
0050h ???

```

Format of ??? record:

```

Offset Size Description (Table 03274)
00h WORD offset of ???
02h 4 BYTES ???
06h DWORD IP address (big-endian)
0Ah 6 BYTES physical address (big-endian)
???
```

-----N-60-----

INT 60 - TCPDRV 2.01 - TCP/IP Application Binary Interface (ABI)

Note: The handler for the interrupt will start with a 2-byte NEAR jump instruction, followed by the ASCIZ signature string "TCP DRVR".

To find the interrupt being used by the driver, an application should scan through interrupt vectors 60h to 7Fh until it finds one with the "TCP DRVR" string.

This specification is being proposed by Peter R. Tattam from the University of Tasmania.

Index: installation check;TCPDRV

-----G-60-----

INT 60 U - INTRSPY/CMDSPY v1.0 only - API

Program: INTRSPY is a script-driven debugger included with the book Undocumented_DOS_.

InstallCheck: determine that a) the handler is an IRET instruction, and b) the signature 0Dh "INTRSPY vN.NN" immediately precedes the handler

Notes: INTRSPY will hook the first available interrupt in the range 60h-67h.

If INTRSPY is installed, the DWORD immediately after the IRET stores its entry point (see #03275)

INTRSPY v2.0 (included with the second edition of the book) no longer supports this API

Index: installation check;INTRSPY

(Table 03275)

Call INTRSPY v1.0 entry point with:

AH = function

00h ???

01h set current directory (for use in reporting)

ES:DI -> counted string containing directory name (max 79 char)

02h set name of script file

ES:DI -> counted string containing file name (max 79 chars)

03h set script arguments

ES:DI -> counted string containing arguments (max 79 chars)

04h get directory set with function 01h

ES:DI -> 80-byte buffer for directory name

05h get name of script file

ES:DI -> 80-byte buffer for script filename

06h get script arguments

ES:DI -> 80-byte buffer for script arguments

07h get ???

CL = 00h-15h specifies what to get

```

ES:DI -> WORD to be set with desired value on return
    08h get ???
ES:DI -> WORD to be set with returned value
    09h get ???
ES:DI -> WORD to be set with returned value
    0Bh store code for interrupt handler???
ES:DI -> data
CX = number of bytes
    0Ch ???
ES:DI -> ???
    0Dh get ???
ES:DI -> BYTE to be set with returned value
    0Eh set ??? flag
    0Fh clear ??? flag
    10h ???
Return: AL = 04h or 05h if failed
    11h ???
Return: AL = 05h if failed
    12h get ???
ES:DI -> buffer
Return: CX = number of bytes returned in buffer
    13h ???
Return: AH = 00h
AL = status
    00h successful
    01h invalid function
    02h ???
    03h ???
    04h ???
    05h ???
-----u-60-----
INT 60 U - PC/370 v4.2 - ???
    ???
Return: ???
Program: PC/370 is an IBM 370 emulator by Donald S. Higgins
Range: INT 00 to INT FF, selected by patching the executable
Note: the documentation includes instructions for patching the system for
    another interrupt
SeeAlso: INT 2F/AX=7F24h,INT DC"PC/370"
-----r-60-----
INT 60 - JPI TopSPEED Modula-2 v1 - PROCEDURE ENTRY TRAP

```

SeeAlso: INT 61"JPI"

-----N-60-----

INT 60 - FTP Packet Driver - PC/TCP Packet Driver Specification

Range: INT 20 to INT FF

Notes: The handler for the interrupt will start with a 3-byte jump instruction, followed by the ASCII string "PKT DRVR" (the terminating NUL is significant).

To find the interrupt being used by the driver, an application should scan through interrupt vectors 20h to FFh (60h through 80h for v1.09- of the specification) until it finds one with the "PKT DRVR" string.

AH values of 80h to FFh have been reserved for user-defined additions.

-----I-60-----

INT 60 u - 3270-PC CONTROL PROGRAM - ???

-----b-60----DI0100-----

INT 60 u - HP 95LX System Manager - WAIT FOR EVENT

DI = 0100h

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to event record (see #03277)

Return: event record filled

STACK unchanged

Note: this call will timeout after about 500ms

SeeAlso: INT 15/AX=4DD4h,INT 60/DI=0101h,INT 61"HP 95LX",INT 62"HP 95LX"

(Table 03276)

Values for HP 95LX event type:

- 00h no events
- 01h keystroke available
- 02h Ctrl-Break
- 03h reactivation (always follows deactivation event)
- 04h about to deactivate (sleep)
next get-event call will not return until reactivated
- 05h forced application termination
- 06h 1-2-3 bridge service request (only given to 1-2-3)
- 07h request to grow
- 08h request to shrink
- 09h application's alarm expired
- 0Ah daily chance to set an alarm
- 0Bh system date or time has been changed

Format of HP 95LX event record:

```

Offset  Size  Description (Table 03277)
00h  WORD  event type (see #03276)
02h  WORD  ASCII code page 850 translation of keystroke
      or grow/shrink amount in paragraphs or 0000h if error
      or alarm expiration data
04h  BYTE  scan code from BIOS
05h  BYTE  shift key states at time keystroke is retrieved
06h  WORD  LICS translation of keystroke
08h  BYTE  function key number (1-2-3 only)
09h  DWORD pointer to 1-2-3 bridge record (see #03279)
      or pointer to time change structure (see #03278)
Note: if the System Manager is awaiting the conclusion of a bridge service
      or grow/shrink call and the event type field is set to FFFFh on
      entry, the SysMgr will resume

```

Format of HP 95LX time change structure:

```

Offset  Size  Description (Table 03278)
00h  WORD  old year
02h  BYTE  old month
03h  BYTE  old date
04h  BYTE  old day
05h  BYTE  old hour
06h  BYTE  old minute
07h  BYTE  old second
08h  BYTE  old hundredth of a second
09h  9 BYTES new time in same format as old time

```

-----b-60-----DI0101-----

```

INT 60 u - HP 95LX System Manager - CHECK FOR EVENT
  DI = 0101h
  STACK:  2 WORDs unused dummies (for calls from high level languages)
          DWORD pointer to event record (INT 60/DI=0100h)
Return: event record filled
  STACK unchanged

```

Note: this call returns immediately if no event is available

SeeAlso: INT 60/DI=0100h

-----b-60-----DI0102-----

```

INT 60 u - HP 95LX System Manager - "SH_STATUS"
  DI = 0102h
  STACK:  2 WORDs unused dummies (for calls from high level languages)
Return: ???
  STACK unchanged

```

-----b-60-----DI0104-----

INT 60 u - HP 95LX System Manager - LOTUS 1-2-3 BRIDGE SERVICES

DI = 0104h

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to bridge record (see #03279)

Return: ???

STACK unchanged

Format of HP 95LX bridge record:

Offset Size Description (Table 03279)

00h WORD function code (see #03280)

02h WORD return code from 1-2-3

04h 16 BYTEs ASCII range name

14h WORD start column of range

16h WORD start row of range

18h WORD end column of range

1Ah WORD end row of range

1Ch WORD order in which data is placed in buffer

1Eh WORD buffer size

20h WORD offset within bridge record's segment of buffer for cell data

(Table 03280)

Values for HP 95LX function code:

00h test

01h get range

02h "GETRANGE_ADDR"

03h "SETRANGE_ADDR"

04h "GETRANGE_DATA"

05h "SETRANGE_DATA"

06h recalculate

07h get cursor

08h set cursor

09h redisplay

0Ah cell type

0Bh "CALCTYPE"

-----b-60-----DI0105-----

INT 60 u - HP 95LX System Manager - FLUSH KEYBOARD BUFFER

DI = 0105h

STACK: 2 WORDs unused dummies (for calls from high level languages)

Return: ???

STACK unchanged

-----b-60----DI0106-----

INT 60 u - HP 95LX System Manager - YIELD CPU

DI = 0106h

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to ???

Return: ???

STACK unchanged

SeeAlso: INT 15/AX=1000h, INT 2F/AX=1680h

-----b-60----DI0107-----

INT 60 u - HP 95LX System Manager - "NO_FINI" - REFUSE TERMINATION REQUEST

DI = 0107h

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to ???

Return: ???

STACK unchanged

SeeAlso: INT 15/AX=4DD4h, INT 61"HP 95LX", INT 62"HP 95LX"

-----b-60----DI0200-----

INT 60 u - HP 95LX System Manager - SETUP MENU

DI = 0200h

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to menu data (see #03281)

DWORD pointer to ???

WORD number of items on menu???

WORD ???

DWORD pointer to ???

WORD ???

DWORD pointer to ???

Return: ???

STACK unchanged

SeeAlso: INT 15/AX=4DD4h, INT 60/DI=0201h, INT 60/DI=0203h, INT 60/DI=0205h

Format of HP 95LX menu data:

Offset Size Description (Table 03281)

00h 80 BYTES first line of menu text

50h 80 BYTES second line of menu text

A0h 80 BYTES third line of menu text

F0h WORD number of keywords

F2h WORD index of currently highlighted keyword or FFFFh

F4h WORD single prompt on top line if nonzero

F6h 20 BYTES which line each of 20 keywords is located on

10Ah 20 BYTES offset of each of 20 keywords within its line

11Eh 20 BYTEs length of each of 20 keywords
132h 20 BYTEs first letter of each of 20 keywords
146h 20 WORDs offsets of long prompts for each of 20 keywords
-----b-60----DI0201-----
INT 60 u - HP 95LX System Manager - DISPLAY OR REDISPLAY MENU
DI = 0201h
STACK: 2 WORDs unused dummies (for calls from high level languages)
DWORD pointer to menu data (see #03281)
Return: ???
STACK unchanged
SeeAlso: INT 60/DI=0200h,INT 60/DI=0202h,INT 60/DI=0206h
-----b-60----DI0202-----
INT 60 u - HP 95LX System Manager - "MENU_ON" - ENABLE PROCESSING OF MENU
DI = 0202h
STACK: 2 WORDs unused dummies (for calls from high level languages)
DWORD pointer to menu data (see #03281)
Return: ???
STACK unchanged
SeeAlso: INT 60/DI=0200h,INT 60/DI=0201h,INT 60/DI=0203h
-----b-60----DI0203-----
INT 60 u - HP 95LX System Manager - REMOVE MENU
DI = 0203h
STACK: 2 WORDs unused dummies (for calls from high level languages)
DWORD pointer to menu data (see #03281)
Return: ???
STACK unchanged
SeeAlso: INT 60/DI=0201h,INT 60/DI=0202h,INT 60/DI=0204h,INT 60/DI=0208h
-----b-60----DI0204-----
INT 60 u - HP 95LX System Manager - LET SYSTEM MANAGER HANDLE MENU KEYSTROKE
DI = 0204h
STACK: 2 WORDs unused dummies (for calls from high level languages)
DWORD pointer to menu data (see #03281)
WORD keystroke
DWORD pointer to WORD to receive selection number
Return: buffer for selection number filled with index of selected menu item or
FFFFh if no final selection yet
STACK unchanged
SeeAlso: INT 60/DI=0200h,INT 60/DI=0202h,INT 60/DI=0207h
-----b-60----DI0205-----
INT 60 u - HP 95LX System Manager - INITIALIZE FILE SELECTION MENU
DI = 0205h

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to file menu structure (see #03282)

DWORD pointer to edit record (see #03285 at INT 60/DI=0400h)

DWORD pointer to wildcard filespec for initial file list

WORD row???

WORD column???

Return: ???

STACK unchanged

SeeAlso: INT 60/DI=0200h,INT 60/DI=0206h,INT 60/DI=0208h

Format of HP 95LX file menu structure:

Offset Size Description (Table 03282)

00h DWORD pointer to ASCIZ base directory name

04h DWORD pointer to ASCIZ file pattern (wildcard filespec)

08h DWORD pointer to file list workspace, at least 1024 bytes (see #03283)

0Ch WORD size of file list workspace in bytes

0Eh WORD starting row (-3 is topmost, 0 is first non-"reserved" line)

10h WORD starting column

12h WORD number of lines

14h WORD number of columns

16h WORD number of files displayed on each line

---the remaining fields are initialized by the System Manager---

18h WORD 0000h if first edit character, else multiline

1Ah WORD number of files in file list

1Ch WORD max files workspace has room for

1Eh WORD file at top of list

20h WORD index of file to highlight

22h WORD index of file to unhighlight

24h WORD current focus (01h FMENU, 02h EDIT)

Format of HP 95LX file list workspace entry:

Offset Size Description (Table 03283)

00h BYTE file attributes

01h WORD file time (see #01665 at INT 21/AX=5700h)

03h WORD file date (see #01666 at INT 21/AX=5700h)

05h DWORD file size

09h 13 BYTES ASCIZ filename

-----b-60----DI0206-----

INT 60 u - HP 95LX System Manager - DISPLAY/REDISPLAY FILE SELECTION MENU

DI = 0206h

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to file menu structure (see #03282)
 DWORD pointer to edit record (see #03285 at INT 60/DI=0400h)

Return: ???

STACK unchanged

SeeAlso: INT 60/DI=0205h

-----b-60----DI0207-----

INT 60 u - HP 95LX System Manager - LET SYSMGR PROCESS FILE SEL MENU KEYSTROKE

DI = 0207h

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to file menu structure (see #03282)
 DWORD pointer to edit record (see #03285 at INT 60/DI=0400h)

WORD keystroke

Return: AX = status (see #03284)

STACK unchanged

SeeAlso: INT 60/DI=0205h, INT 60/DI=0208h

(Table 03284)

Values for HP 95LX System Manager status:

0000h keystroke processed, call INT 60/DI=0206h to refresh menu
 0001h redisplay application area before refreshing menu
 0002h user confirmed selection, filename is in edit record's buffer
 0003h user aborted menu
 FFFBh bad filename
 FFFCh bad directory
 FFFDh bad drive
 FFFEh unknown keystroke
 FFFFh keystroke known but invalid in current context

-----b-60----DI0208-----

INT 60 u - HP 95LX System Manager - REMOVE FILE SELECTION MENU

DI = 0208h

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to file menu structure (see #03282)
 DWORD pointer to edit record (see #03285 at INT 60/DI=0400h)

Return: ???

STACK unchanged

SeeAlso: INT 15/AX=4DD4h, INT 60/DI=0205h, INT 60/DI=0206h

-----b-60----DI0300-----

INT 60 u - HP 95LX System Manager - DISPLAY STRING

DI = 0300h

STACK: 2 WORDs unused dummies (for calls from high level languages)

WORD starting row (-3 is topmost, 0 is first user line)

WORD starting column
DWORD pointer to string
WORD length of string
WORD display style: 0000h normal, 0001h reverse video
WORD "OSTYLE"

Return: ???

STACK unchanged

SeeAlso: INT 60/DI=0F03h, INT 60/DI=1005h

-----b-60----DI0301-----

INT 60 u - HP 95LX System Manager - CLEAR PORTION OF SCREEN

DI = 0301h

STACK: 2 WORDs unused dummies (for calls from high level languages)

WORD starting row (-3 is topmost, 0 is first user line)

WORD starting column

WORD number of rows

WORD number of columns

Return: ???

STACK unchanged

SeeAlso: INT 60/DI=0302h, INT 60/DI=1005h

-----b-60----DI0302-----

INT 60 u - HP 95LX System Manager - SCROLL PORTION OF SCREEN

DI = 0302h

STACK: 2 WORDs unused dummies (for calls from high level languages)

WORD starting row???

WORD starting column???

WORD height of scroll region???

WORD width of scroll region???

WORD number of lines to scroll region???

Return: ???

STACK unchanged

SeeAlso: INT 60/DI=0301h

-----b-60----DI0303-----

INT 60 u - HP 95LX System Manager - SCREEN SERVICE "M_XCHG"

DI = 0303h

STACK: 2 WORDs unused dummies (for calls from high level languages)

WORD ???

WORD ???

WORD ???

WORD ???

DWORD pointer to ???

Return: ???

```
STACK unchanged
-----b-60-----DI0304-----
INT 60 u - HP 95LX System Manager - SCREEN SERVICE "M_CHRATTR"
  DI = 0304h
  STACK:  2 WORDs unused dummies (for calls from high level languages)
  DWORD pointer to ???
  WORD   ???
Return: ???
  STACK unchanged
-----b-60-----DI0305-----
INT 60 u - HP 95LX System Manager - SCREEN SERVICE "M_CHRRVRT"
  DI = 0305h
  STACK:  2 WORDs unused dummies (for calls from high level languages)
  WORD   ???
  WORD   ???
  DWORD pointer to ???
  WORD   ???
Return: ???
  STACK unchanged
-----b-60-----DI0307-----
INT 60 u - HP 95LX System Manager - SCREEN SERVICE "M_CHRINV"
  DI = 0307h
  STACK:  2 WORDs unused dummies (for calls from high level languages)
  WORD   ???
  WORD   ???
  WORD   ???
Return: ???
  STACK unchanged
-----b-60-----DI0308-----
INT 60 u - HP 95LX System Manager - SCREEN SERVICE "M_ROWS_COLS"
  DI = 0308h
  STACK:  2 WORDs unused dummies (for calls from high level languages)
Return: ???
  STACK unchanged
-----b-60-----DI0309-----
INT 60 u - HP 95LX System Manager - SET SCREEN (VIDEO???) MODE
  DI = 0309h
  STACK:  2 WORDs unused dummies (for calls from high level languages)
  WORD   new mode
Return: ???
  STACK unchanged
```

-----b-60-----DI030A-----

INT 60 u - HP 95LX System Manager - GET SCREEN (VIDEO???) MODE

DI = 030Ah

STACK: 2 WORDs unused dummies (for calls from high level languages)

Return: ???

STACK unchanged

-----b-60-----DI030B-----

INT 60 u - HP 95LX System Manager - SET CURSOR POSITION

DI = 030Bh

STACK: 2 WORDs unused dummies (for calls from high level languages)

WORD row (-3 is topmost, 0 is first non-reserved line)

WORD column

Return: ???

STACK unchanged

Note: cursor is hidden if the specified position is not on the physical display

SeeAlso: INT 10/AH=02h, INT 15/AX=4DD4h, INT 61"HP 95LX", INT 62"HP 95LX"

-----b-60-----DI0400-----

INT 60 u - HP 95LX System Manager - "EDIT_INIT"

DI = 0400h

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to edit record (see #03285)

DWORD pointer to string to be edited

WORD initial length of string being edited

WORD maximum length of edited string

WORD row of edit field

WORD leftmost column of edit field

Return: ???

STACK unchanged

Format of HP 95LX edit record:

Offset Size Description (Table 03285)

00h WORD current length of edit buffer

02h BYTE flag for special processing on first character

03h BYTE flags

bit 0: tab handling

04h WORD editing in prompt window?

06h DWORD pointer to top line of prompt window message

0Ah WORD length of top line of prompt

0Ch DWORD pointer to second line of prompt window message

10h WORD length of second line of prompt

12h 80 BYTEs workspace for editing
 62h 2 WORDs line array needed for multi-line editing
 66h 36 BYTEs multi-line edit record (see #03286)
 8Ah WORD displayable columns

Format of HP 95LX multi-line edit record:

Offset	Size	Description (Table 03286)
00h	DWORD	pointer to user-supplied edit buffer
04h	WORD	length of edit buffer
06h	WORD	current cursor position
08h	WORD	starting row of edit area (-3 is topmost, 0 is first user line)
0Ah	WORD	starting column of edit area
0Ch	WORD	height of edit area
0Eh	WORD	width of edit area
10h	WORD	current top row (-3 is topmost, 0 is first user line)
12h	WORD	number of rows displayable
14h	BYTE	cursor column
15h	BYTE	01h if buffer has been modified
16h	BYTE	first displayable column (ticker fields only)
17h	BYTE	01h if wordwrap enabled, FFh if ticker field
18h	DWORD	pointer to array of line starts (at least one bigger than edit area is high)
1Ch	BYTE	currently marking?
1Dh	BYTE	flag
1Eh	WORD	offset of mark start
20h	WORD	offset of mark end (inclusive)
22h	WORD	displayable columns

-----b-60----DI0401-----

INT 60 u - HP 95LX System Manager - EDIT ON TOP LINE

DI = 0401h

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to edit record (see #03285)

DWORD pointer to string to edit

WORD initial length of string being edited

WORD maximum length of edited string

DWORD pointer to first line of prompt

WORD length of first line

DWORD pointer to second line of prompt

WORD length of second line

Return: ???

STACK unchanged

-----b-60----DI0402-----

INT 60 u - HP 95LX System Manager - DISPLAY OR REDISPLAY EDIT FIELD

DI = 0402h

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to edit record (see #03285)

Return: ???

STACK unchanged

-----b-60----DI0403-----

INT 60 u - HP 95LX System Manager - LET SYSTEM MANAGER PROCESS EDITING KEYSTROK

DI = 0403h

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to edit record (see #03285)

WORD keystroke

DWORD pointer to WORD buffer for result code

Return: result code buffer filled with 0001h if editing complete

STACK unchanged

-----b-60----DI0404-----

INT 60 u - HP 95LX System Manager - "MDIT_INI"

DI = 0404h

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to ???

WORD ???

WORD ???

WORD ???

WORD ???

DWORD pointer to ???

WORD ???

WORD ???

WORD ???

DWORD pointer to ???

Return: ???

STACK unchanged

-----b-60----DI0405-----

INT 60 u - HP 95LX System Manager - "MDIT_DIS"

DI = 0405h

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to ???

Return: ???

STACK unchanged

-----b-60----DI0406-----

INT 60 u - HP 95LX System Manager - "MDIT_KEY"

```
DI = 0406h
STACK:  2 WORDs unused dummies (for calls from high level languages)
        DWORD pointer to ???
        WORD  ???
Return: ???
        STACK unchanged
-----b-60----DI0407-----
INT 60 u - HP 95LX System Manager - "MDIT_FIL"
        DI = 0407h
        STACK:  2 WORDs unused dummies (for calls from high level languages)
                DWORD pointer to ???
                DWORD pointer to ???
Return: ???
        STACK unchanged
-----b-60----DI0408-----
INT 60 u - HP 95LX System Manager - "MDIT_MARK"
        DI = 0408h
        STACK:  2 WORDs unused dummies (for calls from high level languages)
                DWORD pointer to ???
Return: ???
        STACK unchanged
-----b-60----DI0409-----
INT 60 u - HP 95LX System Manager - "MDIT_UNMARK"
        DI = 0409h
        STACK:  2 WORDs unused dummies (for calls from high level languages)
                DWORD pointer to ???
Return: ???
        STACK unchanged
-----b-60----DI040A-----
INT 60 u - HP 95LX System Manager - "MDIT_CUTMARK"
        DI = 040Ah
        STACK:  2 WORDs unused dummies (for calls from high level languages)
                DWORD pointer to ???
Return: ???
        STACK unchanged
-----b-60----DI040B-----
INT 60 u - HP 95LX System Manager - "MDIT_INS_STR"
        DI = 040Bh
        STACK:  2 WORDs unused dummies (for calls from high level languages)
                DWORD pointer to ???
                DWORD pointer to ???
```

WORD ???

Return: ???

STACK unchanged

SeeAlso: INT 15/AX=4DD4h, INT 61"HP 95LX", INT 62"HP 95LX"

-----b-60-----DI0500-----

INT 60 u - HP 95LX System Manager - OPEN FILE

DI = 0500h

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to file state record (see #03287)

DWORD pointer to filename

WORD length of filename

WORD ???

WORD suppress buffering if nonzero

Return: AX = status

STACK unchanged

SeeAlso: INT 60/DI=0501h, INT 60/DI=0502h, INT 60/DI=0508h

Format of HP 95LX file state record:

Offset Size Description (Table 03287)

00h WORD DOS file handle

02h WORD state flags (see #03288)

04h DWORD current DOS physical file offset (FFFFFFFFh if unknown)

08h DWORD DOS file offset of start of buffer

0Ch DWORD effective file offset as seen by caller

10h WORD number of bytes in file buffer

---buffered I/O only---

12h 512 BYTES file buffer

Bitfields for HP 95LX file state flags:

Bit(s) Description (Table 03288)

0 buffer contents valid

1 buffer is dirty and must be written

2 unbuffered I/O

3 file is a character device

SeeAlso: #03287

-----b-60-----DI0501-----

INT 60 u - HP 95LX System Manager - OPEN FILE IN READ-ONLY MODE

DI = 0501h

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to file state record (see #03287)

DWORD pointer to filename

WORD length of filename
WORD ???
WORD suppress buffering if nonzero

Return: AX = status

STACK unchanged

SeeAlso: INT 60/DI=0500h

-----b-60----DI0502-----

INT 60 u - HP 95LX System Manager - CREATE NEW FILE

DI = 0502h

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to file state record (see #03287)

DWORD pointer to filename

WORD length of filename

WORD ???

WORD suppress buffering if nonzero

Return: AX = status

STACK unchanged

SeeAlso: INT 60/DI=0500h,INT 60/DI=0503h

-----b-60----DI0503-----

INT 60 u - HP 95LX System Manager - CREATE OR TRUNCATE FILE

DI = 0503h

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to file state record (see #03287)

DWORD pointer to filename

WORD length of filename

WORD ???

WORD suppress buffering if nonzero

Return: AX = status

STACK unchanged

SeeAlso: INT 60/DI=0502h

-----b-60----DI0504-----

INT 60 u - HP 95LX System Manager - READ FROM FILE

DI = 0504h

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to file state record (see #03287)

DWORD pointer to data buffer

WORD number of bytes to read

DWORD pointer to WORD in which to return actual bytes read

Return: ???

STACK unchanged

SeeAlso: INT 60/DI=0505h

-----b-60----DI0505-----

INT 60 - HP 95LX System Manager - WRITE TO FILE

DI = 0505h

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to file state record (see #03287)

DWORD pointer to data

WORD length of data

Return: AX = status

STACK unchanged

SeeAlso: INT 60/DI=0504h

-----b-60----DI0506-----

INT 60 u - HP 95LX System Manager - SET FILE POSITION

DI = 0506h

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to file state record (see #03287)

2 WORDs ???

Return: ???

STACK unchanged

SeeAlso: INT 60/DI=0507h

-----b-60----DI0507-----

INT 60 u - HP 95LX System Manager - GET FILE POSITION

DI = 0507h "M_TELL"

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to file state record (see #03287)

DWORD pointer to DWORD buffer for file position???

Return: ???

STACK unchanged

SeeAlso: INT 60/DI=0506h

-----b-60----DI0508-----

INT 60 u - HP 95LX System Manager - CLOSE FILE

DI = 0508h

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to file state record (see #03287)

Return: ???

STACK unchanged

SeeAlso: INT 60/DI=0500h

-----b-60----DI0509-----

INT 60 u - HP 95LX System Manager - FILE SERVICE "M_SETPAT"

DI = 0509h

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to ???

DWORD pointer to ???

WORD ???

WORD ???

Return: ???

STACK unchanged

-----b-60----DI050A-----

INT 60 u - HP 95LX System Manager - FILE SERVICE "M_MATCH"

DI = 050Ah

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to ??? (see #03289)

DWORD pointer to ???

Return: ???

STACK unchanged

Format of HP 95LX pattern match control block:

Offset Size Description (Table 03289)

00h 43 BYTEs FindFirst data block (see #01626 at INT 21/AH=4Eh)

2Bh 80 BYTEs full path name

7Bh BYTE offset of last component of filename

7Ch BYTE DOS function number (4Eh or 4Fh)

-----b-60----DI050B-----

INT 60 u - HP 95LX System Manager - IDENTIFY FILENAME REFERENT

DI = 050Bh

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to ???

WORD ???

WORD ???

DWORD pointer to ???

Return: ??? = result (see #03290)

???

STACK unchanged

(Table 03290)

Values returned by HP 95LX System Manager:

0000h nonexistent

0001h file

0002h directory

0003h character device

-----b-60----DI050C-----

INT 60 u - HP 95LX System Manager - DELETE FILE

DI = 050Ch "M_DELETE"

```
STACK: 2 WORDs unused dummies (for calls from high level languages)
        DWORD pointer to ???
        WORD  ???
        WORD  ???
```

Return: ???

STACK unchanged

-----b-60----DI050D-----

INT 60 u - HP 95LX System Manager - RENAME FILE

DI = 050Dh

```
STACK: 2 WORDs unused dummies (for calls from high level languages)
        DWORD pointer to ???
        WORD  ???
        WORD  ???
        DWORD pointer to ???
        WORD  ???
        WORD  ???
```

Return: ???

STACK unchanged

-----b-60----DI050E-----

INT 60 u - HP 95LX System Manager - FILE SERVICE "M_GETDIR"

DI = 050Eh

```
STACK: 2 WORDs unused dummies (for calls from high level languages)
        WORD  ???
        DWORD pointer to ???
        DWORD pointer to ???
```

Return: ???

STACK unchanged

-----b-60----DI050F-----

INT 60 u - HP 95LX System Manager - FILE SERVICE "M_SETDIR"

DI = 050Fh

```
STACK: 2 WORDs unused dummies (for calls from high level languages)
        DWORD pointer to ???
        WORD  ???
```

Return: ???

STACK unchanged

-----b-60----DI0510-----

INT 60 u - HP 95LX System Manager - FILE SERVICE "M_VOLUME"

DI = 0510h

```
STACK: 2 WORDs unused dummies (for calls from high level languages)
        DWORD pointer to ???
        DWORD pointer to ???
```

```
Return: ???
  STACK unchanged
-----b-60----DI0511-----
INT 60 u - HP 95LX System Manager - MAKE A SUBDIRECTORY
  DI = 0511h
  STACK:  2 WORDs unused dummies (for calls from high level languages)
    DWORD pointer to ???
    WORD  ???
    WORD  ???
Return: ???
  STACK unchanged
SeeAlso: INT 21/AH=39h,INT 60/DI=0512h
-----b-60----DI0512-----
INT 60 u - HP 95LX System Manager - REMOVE A SUBDIRECTORY
  DI = 0512h
  STACK:  2 WORDs unused dummies (for calls from high level languages)
    DWORD pointer to ???
    WORD  ???
    WORD  ???
Return: ???
  STACK unchanged
SeeAlso: INT 21/AH=3Ah,INT 60/DI=0511h
-----b-60----DI0513-----
INT 60 u - HP 95LX System Manager - GET DEFAULT DRIVE
  DI = 0513h
  STACK:  2 WORDs unused dummies (for calls from high level languages)
    DWORD pointer to ??? buffer for current drive
Return: ???
  STACK unchanged
SeeAlso: INT 21/AH=19h,INT 60/DI=0514h
-----b-60----DI0514-----
INT 60 u - HP 95LX System Manager - SET DEFAULT DRIVE
  DI = 0514h
  STACK:  2 WORDs unused dummies (for calls from high level languages)
    WORD  new drive
Return: ???
  STACK unchanged
SeeAlso: INT 21/AH=0Eh"DOS 1+",INT 60/DI=0513h
-----b-60----DI0515-----
INT 60 u - HP 95LX System Manager - FILE SERVICE "M_FDATE"
  DI = 0515h
```



```
STACK: 2 WORDs unused dummies (for calls from high level languages)
        DWORD pointer to ???
        DWORD pointer to ???
Return: ???
        STACK unchanged
-----b-60----DI0516-----
INT 60 u - HP 95LX System Manager - FILE SERVICE "M_GET_SYSDIR"
        DI = 0516h
        STACK: 2 WORDs unused dummies (for calls from high level languages)
                DWORD pointer to ???
Return: ???
        STACK unchanged
-----b-60----DI0517-----
INT 60 u - HP 95LX System Manager - GET FILE ATTRIBUTES
        DI = 0517h
        STACK: 2 WORDs unused dummies (for calls from high level languages)
                DWORD pointer to ???
                WORD ???
                WORD ???
                DWORD pointer to ??? buffer for file's attributes???
Return: ???
        STACK unchanged
SeeAlso: INT 21/AX=4300h,INT 60/DI=0518h
-----b-60----DI0518-----
INT 60 u - HP 95LX System Manager - SET FILE ATTRIBUTES
        DI = 0518h
        STACK: 2 WORDs unused dummies (for calls from high level languages)
                DWORD pointer to ???
                WORD ???
                WORD ???
                WORD new attributes???
Return: ???
        STACK unchanged
SeeAlso: INT 21/AX=4301h,INT 60/DI=0517h
-----b-60----DI0519-----
INT 60 u - HP 95LX System Manager - FILE SERVICE "M_COMMON_OPEN"
        DI = 0519h
        STACK: 2 WORDs unused dummies (for calls from high level languages)
                DWORD pointer to ???
                DWORD pointer to ???
                WORD ???
```

```
WORD ???  
WORD ???  
WORD ???  
WORD ???
```

```
Return: ???
```

```
STACK unchanged
```

```
-----b-60----DI051A-----
```

```
INT 60 u - HP 95LX System Manager - FILE SERVICE "M_COPYDT"
```

```
DI = 051Ah
```

```
STACK: 2 WORDs unused dummies (for calls from high level languages)
```

```
DWORD pointer to ???
```

```
DWORD pointer to ???
```

```
Return: ???
```

```
STACK unchanged
```

```
-----b-60----DI051B-----
```

```
INT 60 u - HP 95LX System Manager - FILE SERVICE "M_GETFDT"
```

```
DI = 051Bh
```

```
STACK: 2 WORDs unused dummies (for calls from high level languages)
```

```
DWORD pointer to ???
```

```
DWORD pointer to ???
```

```
Return: ???
```

```
STACK unchanged
```

```
-----b-60----DI051C-----
```

```
INT 60 u - HP 95LX System Manager - FILE SERVICE "M_PUTFDT"
```

```
DI = 051Ch
```

```
STACK: 2 WORDs unused dummies (for calls from high level languages)
```

```
DWORD pointer to ???
```

```
WORD ???
```

```
Return: ???
```

```
STACK unchanged
```

```
-----b-60----DI0600-----
```

```
INT 60 u - HP 95LX System Manager - PROCESS INITIALIZING
```

```
DI = 0600h
```

```
STACK: 2 WORDs unused dummies (for calls from high level languages)
```

```
Return: ???
```

```
STACK unchanged
```

```
SeeAlso: INT 15/AX=4DD4h, INT 60/DI=0601h, INT 61"HP 95LX"
```

```
-----b-60----DI0601-----
```

```
INT 60 u - HP 95LX System Manager - PROCESS TERMINATION
```

```
DI = 0601h
```

```
STACK: 2 WORDs unused dummies (for calls from high level languages)
```

```
Return: never
    STACK unchanged
SeeAlso: INT 21/AH=4Ch,INT 2F/AX=1122h,INT 60/DI=0600h
-----b-60----DI0602-----
INT 60 u - HP 95LX System Manager - "M_LOCK" - PREVENT TASK SWITCHES
    DI = 0602h
    STACK: 2 WORDs unused dummies (for calls from high level languages)
Return: ???
    STACK unchanged
SeeAlso: INT 15/AX=101Bh,INT 2F/AX=1681h,INT 60/DI=0603h
-----b-60----DI0603-----
INT 60 u - HP 95LX System Manager - "M_UNLOCK" - ALLOW TASK SWITCHES
    DI = 0603h
    STACK: 2 WORDs unused dummies (for calls from high level languages)
Return: ???
    STACK unchanged
SeeAlso: INT 15/AX=101Ch,INT 2F/AX=1682h,INT 60/DI=0602h
-----b-60----DI0604-----
INT 60 u - HP 95LX System Manager - "M_SPAWN"
    DI = 0604h
    STACK: 2 WORDs unused dummies (for calls from high level languages)
        DWORD pointer to ???
        WORD ???
        WORD ???
        DWORD pointer to ???
Return: ???
    STACK unchanged
SeeAlso: INT 21/AH=4Bh
-----b-60----DI0605-----
INT 60 u - HP 95LX System Manager - "M_APPCOUNT"
    DI = 0605h
    STACK: 2 WORDs unused dummies (for calls from high level languages)
Return: ???
    STACK unchanged
-----b-60----DI0606-----
INT 60 u - HP 95LX System Manager - "M_REBOOT"
    DI = 0606h
    STACK: 2 WORDs unused dummies (for calls from high level languages)
Return: ???
    STACK unchanged
SeeAlso: INT 14/AH=17h"FOSSIL",INT 19
```

-----b-60-----DI0607-----

INT 60 u - HP 95LX System Manager - "M_SPAWNARG"

DI = 0607h

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to ???

WORD ???

DWORD pointer to ???

WORD ???

Return: ???

STACK unchanged

-----b-60-----DI0608-----

INT 60 u - HP 95LX System Manager - "M_REG_APP_NAME"

DI = 0608h

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to ???

Return: ???

STACK unchanged

-----b-60-----DI0609-----

INT 60 u - HP 95LX System Manager - "M_APP_NAME"

DI = 0609h

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to ???

Return: DX:AX -> ???

STACK unchanged

SeeAlso: INT 15/AX=4DD4h, INT 61"HP 95LX", INT 62"HP 95LX"

-----b-60-----DI0700-----

INT 60 u - HP 95LX System Manager - OPEN CLIPBOARD

DI = 0700h

STACK: 2 WORDs unused dummies (for calls from high level languages)

Return: ??? = error code (see #03291)

???

STACK unchanged

SeeAlso: INT 60/DI=0701h, INT 60/DI=0702h

(Table 03291)

Values for HP 95LX error code:

0000h successful

FFF8h transfer request out of bounds

FFF9h no such representation

FFFAh no representation open

FFFBh a representation is already open

```
FFFCh  representation already exists
FFFDh  heap allocation failure
FFFEh  clipboard not open
FFFFh  clipboard access denied
-----b-60----DI0701-----
INT 60 u - HP 95LX System Manager - CLOSE CLIPBOARD
  DI = 0701h
  STACK:  2 WORDs unused dummies (for calls from high level languages)
Return: ???
  STACK unchanged
SeeAlso: INT 60/DI=0700h,INT 60/DI=0702h
-----b-60----DI0702-----
INT 60 u - HP 95LX System Manager - RESET CLIPBOARD
  DI = 0702h
  STACK:  2 WORDs unused dummies (for calls from high level languages)
  DWORD pointer to ???
Return: ???
  STACK unchanged
SeeAlso: INT 60/DI=0700h
-----b-60----DI0704-----
INT 60 u - HP 95LX System Manager - "M_NEW_REP" - START A NEW REPRESENTATION???
  DI = 0704h
  STACK:  2 WORDs unused dummies (for calls from high level languages)
  DWORD pointer to ???
Return: ???
  STACK unchanged
SeeAlso: INT 60/DI=0705h,INT 60/DI=0706h,INT 60/DI=0707h
-----b-60----DI0705-----
INT 60 u - HP 95LX System Manager - CLIPBOARD SERVICE "M_FINI_REP"
  DI = 0705h
  STACK:  2 WORDs unused dummies (for calls from high level languages)
Return: ???
  STACK unchanged
SeeAlso: INT 60/DI=0704h
-----b-60----DI0706-----
INT 60 u - HP 95LX System Manager - CLIPBOARD SERVICE "M_REP_NAME"
  DI = 0706h
  STACK:  2 WORDs unused dummies (for calls from high level languages)
  WORD   ???
  DWORD pointer to ???
  DWORD pointer to ???
```

```
Return: ???
  STACK unchanged
SeeAlso: INT 60/DI=0704h,INT 60/DI=0707h
-----b-60----DI0707-----
INT 60 u - HP 95LX System Manager - CLIPBOARD SERVICE "M_REP_INDEX"
  DI = 0707h
  STACK: 2 WORDs unused dummies (for calls from high level languages)
         DWORD pointer to ???
         DWORD pointer to ???
         DWORD pointer to ???
Return: ???
  STACK unchanged
SeeAlso: INT 60/DI=0704h,INT 60/DI=0706h
-----b-60----DI0708-----
INT 60 u - HP 95LX System Manager - WRITE TO CLIPBOARD
  DI = 0708h
  STACK: 2 WORDs unused dummies (for calls from high level languages)
         DWORD pointer to data to be written???
         WORD length of data???
Return: ???
  STACK unchanged
SeeAlso: INT 60/DI=0709h
-----b-60----DI0709-----
INT 60 u - HP 95LX System Manager - READ FROM CLIPBOARD
  DI = 0709h
  STACK: 2 WORDs unused dummies (for calls from high level languages)
         WORD ???
         WORD ???
         DWORD pointer to buffer for data???
         WORD length of buffer???
Return: ???
  STACK unchanged
SeeAlso: INT 60/DI=0708h
-----b-60----DI0800-----
INT 60 u - HP 95LX System Manager - BEEP
  DI = 0800h
  STACK: 2 WORDs unused dummies (for calls from high level languages)
Return: ???
  STACK unchanged
SeeAlso: INT 60/DI=0801h,INT 60/DI=0802h,INT 60/DI=0803h
-----b-60----DI0801-----
```

```
INT 60 u - HP 95LX System Manager - SOUND SERVICE "M_THUD"
  DI = 0801h
  STACK:  2 WORDs unused dummies (for calls from high level languages)
Return: ???
  STACK unchanged
SeeAlso: INT 60/DI=0800h,INT 60/DI=0802h,INT 60/DI=0803h
-----b-60----DI0802-----
INT 60 u - HP 95LX System Manager - MAKE A SOUND PATTERN
  DI = 0802h
  STACK:  2 WORDs unused dummies (for calls from high level languages)
  WORD  pattern number (00h-06h)
Return: ???
  STACK unchanged
SeeAlso: INT 60/DI=0800h,INT 60/DI=0801h,INT 60/DI=0803h
-----b-60----DI0803-----
INT 60 u - HP 95LX System Manager - TURN OFF SOUND
  DI = 0803h
  STACK:  2 WORDs unused dummies (for calls from high level languages)
Return: ???
  STACK unchanged
SeeAlso: INT 60/DI=0800h,INT 60/DI=0801h,INT 60/DI=0802h
-----b-60----DI0900-----
INT 60 - HP 95LX System Manager - ALLOCATE REGULAR MEMORY BLOCK
  DI = 0900h
  STACK:  2 WORDs unused dummies (for calls from high level languages)
  WORD  size of block in bytes
Return: AX -> memory block
  STACK unchanged
Note: System Manager-compliant applications are always small-model (64K code,
      64K data)
SeeAlso: INT 15/AX=4DD4h,INT 60/DI=0902h,INT 60/DI=0903h
-----b-60----DI0902-----
INT 60 u - HP 95LX System Manager - FREE REGULAR MEMORY BLOCK
  DI = 0902h
  STACK:  2 WORDs unused dummies (for calls from high level languages)
  WORD  offset of memory block???
Return: ???
  STACK unchanged
Note: System Manager-compliant applications are always small-model (64K code,
      64K data)
SeeAlso: INT 60/DI=0900h,INT 60/DI=0904h
```

-----b-60----DI0903-----

INT 60 u - HP 95LX System Manager - ALLOCATE LARGE MEMORY BLOCK

DI = 0903h

STACK: 2 WORDs unused dummies (for calls from high level languages)

WORD size of block in bytes???

Return: AX -> memory block???

STACK unchanged

SeeAlso: INT 60/DI=0900h, INT 60/DI=0904h

-----b-60----DI0904-----

INT 60 u - HP 95LX System Manager - FREE LARGE MEMORY BLOCK

DI = 0904h

STACK: 2 WORDs unused dummies (for calls from high level languages)

WORD segment of memory block???

Return: AX -> ???

STACK unchanged

SeeAlso: INT 60/DI=0902h, INT 60/DI=0903h

-----b-60----DI0B00-----

INT 60 u - HP 95LX System Manager - CLOCK/CALENDAR SERVICE "M_DTINFO"

DI = 0B00h

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to ???

Return: ???

STACK unchanged

-----b-60----DI0B01-----

INT 60 u - HP 95LX System Manager - CLOCK/CALENDAR SERVICE "M_GETDTM"

DI = 0B01h

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to ???

Return: ???

STACK unchanged

-----b-60----DI0B02-----

INT 60 u - HP 95LX System Manager - CLOCK/CALENDAR SERVICE "M_SETDTM"

DI = 0B02h

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to ???

Return: ???

STACK unchanged

-----b-60----DI0B03-----

INT 60 u - HP 95LX System Manager - CLOCK/CALENDAR SERVICE "M_XALARM"

DI = 0B03h

STACK: 2 WORDs unused dummies (for calls from high level languages)

WORD ???

Return: ???

STACK unchanged

-----b-60----DI0B04-----

INT 60 u - HP 95LX System Manager - CLOCK/CALENDAR SERVICE "M_ALARM"

DI = 0B04h

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to alarm record??? (see #03292)

WORD ???

Return: ???

STACK unchanged

Format of HP 95LX alarm record:

Offset Size Description (Table 03292)

00h BYTE hour

01h BYTE minute

02h BYTE second

03h BYTE unused padding

04h WORD rescheduling interval, in seconds

06h BYTE are seconds significant?

07h BYTE alarm sound

08h 40 BYTEs message displayed when alarm activates

30h BYTE task ID of owner

31h BYTE application's own use for sub-class

32h 4 BYTEs application's own use for private data

-----b-60----DI0B05-----

INT 60 u - HP 95LX System Manager - CLOCK/CALENDAR SERVICE "M_START_SW"

DI = 0B05h

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to ???

Return: ???

STACK unchanged

-----b-60----DI0B06-----

INT 60 u - HP 95LX System Manager - CLOCK/CALENDAR SERVICE "M_GET_SW"

DI = 0B06h

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to ???

DWORD pointer to ???

DWORD pointer to ???

Return: ???

STACK unchanged

-----b-60----DI0B07-----

INT 60 u - HP 95LX System Manager - CLOCK/CALENDAR SERVICE "M_STOP_SW"

DI = 0B07h

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to ???

Return: ???

STACK unchanged

-----b-60----DI0B08-----

INT 60 u - HP 95LX System Manager - "M_TELLTIME" - DISPLAY TIMESTAMP

DI = 0B08h

STACK: 2 WORDs unused dummies (for calls from high level languages)

WORD timestamp format (see #03293)

WORD row (-3 is topmost, 0 is first non-reserved line)

WORD column

Return: ???

STACK unchanged

Bitfields for HP 95LX timestamp format:

Bit(s) Description (Table 03293)

1-0 timestamp components

00 date only

01 time only

10 date and time

11 day and date

4 supply am/pm

5 supply seconds

6 show year

7 four-digit year

-----b-60----DI0B09-----

INT 60 u - HP 95LX System Manager - CLOCK/CALENDAR SERVICE "M_GET_SETTINGS"

DI = 0B09h

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to ??? (see #03294)

DWORD pointer to ???

Return: ???

STACK unchanged

SeeAlso: INT 60/DI=0B0Ah, INT 60/DI=0B0Fh

Format of HP 95LX system settings:

Offset Size Description (Table 03294)

00h WORD country code

02h WORD speaker volume (00h-03h or FFh for off)
 04h WORD contrast level (00h-0Fh)
 06h WORD week start (00h Sunday, 01h Monday)
 08h WORD punctuation format (see #03295)
 0Ah WORD two-character language code (only 5355h = "US" byte-swapped)
 0Ch WORD current date format (see #03296)
 0Eh WORD current time format (see #03297)
 10h WORD collating sequence
 00h numbers first, 01h letters first, 02h ASCII
 12h 80 BYTES name of picture file
 62h 30 BYTES name
 80h 30 BYTES title
 9Eh 28 BYTES company name
 BAh WORD number of languages
 BCh 6 BYTES available languages
 C2h 66 BYTES language menu
 104h 2 BYTES ASCIZ date separator
 106h 2 BYTES ASCIZ time separator
 108h BYTE date order
 109h BYTE use 24 hour time?
 10Ah 16 BYTES currency string
 11Ah WORD currency string position (00h prefix, 01h suffix)
 11Ch WORD keyboard (see #03298)
 11Eh WORD printer baud rate
 00h 300, 01h 1200, 02h 2400, 03h 4800, 04h 9600, 05h 19200
 120h WORD printer driver code
 00h Epson FX80, 01h HP Laserjet, 02h IBM ProPrinter
 122h WORD printer interface (00h COM1, 01h COM2, 02h IR, 03h LPT1)
 124h WORD system manager interrupt (60h by default)
 126h WORD code page (01h CP850, 02h CP437)
 128h WORD active exit key
 12Ah WORD active menu key
 12Ch WORD active CHAR key toggle
 12Eh 6 BYTES alarm

(Table 03295)

Values for HP 95LX punctuation format:

code decimal arg thousands

00h . , ,

01h , . .

02h . ; ;

03h , ; .
04h . , " "
05h , . " "
06h . ; " "
07h , ; " "

(Table 03296)

Values for HP 95LX current date format:

00h dd-mmm-yy
01h dd-mmm
02h mmm-yy
03h mm/dd/yy
04h dd/mm/yy
05h dd.mm.yy
06h yy-mm-dd
07h mm/dd
08h dd/mm
09h dd.mm
0Ah mm-dd

(Table 03297)

Values for HP 95LX current time format:

00h HH:MM:SS am/pm
01h HH:MM am/pm
02h HH:MM:SS
03h HH.MM.SS
04h HH,MM,SS
05h HHhMMmSSs
06h HH:MM
07h HH.MM
08h HH,MM
09h HHhMMm

(Table 03298)

Values for HP 95LX keyboard layout:

0001h Belgium
0002h French Canadian
0004h Denmark
0008h Finland
0010h French
0020h Finland

0040h Italy
0080h Netherlands
0100h Norway
0200h Portugal
0400h Spain
0800h Sweden
1000h Swiss French
2000h Swiss German
4000h United Kingdom
8000h USA

-----b-60----DI0B0A-----

INT 60 u - HP 95LX System Manager - CLOCK/CALENDAR SERVICE "M_SET_SETTINGS"

DI = 0B0Ah

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to ???

DWORD pointer to ???

Return: ???

STACK unchanged

SeeAlso: INT 60/DI=0B09h

-----b-60----DI0B0B-----

INT 60 u - HP 95LX System Manager - CLOCK/CALENDAR SERVICE "M_START_TIMER"

DI = 0B0Bh

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to ???

Return: ???

STACK unchanged

SeeAlso: INT 60/DI=0B0Ch,INT 60/DI=0B0Dh

-----b-60----DI0B0C-----

INT 60 u - HP 95LX System Manager - CLOCK/CALENDAR SERVICE "M_STOP_TIMER"

DI = 0B0Ch

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to ???

Return: ???

STACK unchanged

SeeAlso: INT 60/DI=0B0Bh,INT 60/DI=0B0Dh

-----b-60----DI0B0D-----

INT 60 u - HP 95LX System Manager - CLOCK/CALENDAR SERVICE "M_GET_TIMER"

DI = 0B0Dh

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to ???

DWORD pointer to ???

```
    DWORD pointer to ???
Return: ???
    STACK unchanged
SeeAlso: INT 60/DI=0B0Bh,INT 60/DI=0B0Ch
-----b-60----DI0B0E-----
INT 60 u - HP 95LX System Manager - CLOCK/CALENDAR SERVICE "M_TELL_ANYTIME"
    DI = 0B0Eh
    STACK:  2 WORDs unused dummies (for calls from high level languages)
    WORD   ???
    WORD   ???
    WORD   ???
    DWORD  pointer to ???
    DWORD  pointer to ???
Return: DX:AX -> ???
    STACK unchanged
-----b-60----DI0B0F-----
INT 60 u - HP 95LX System Manager - CLOCK/CALENDAR SERVICE "M_GET_SETTINGS_ADDR"
    DI = 0B0Fh
    STACK:  2 WORDs unused dummies (for calls from high level languages)
Return: DX:AX -> system settings record (see #03294)
    STACK unchanged
SeeAlso: INT 60/DI=0B09h
-----b-60----DI0B10-----
INT 60 u - HP 95LX System Manager - PARSE DATE SPECIFICATION
    DI = 0B10h
    STACK:  2 WORDs unused dummies (for calls from high level languages)
    WORD   ???
    DWORD  pointer to ???
    DWORD  pointer to ???
Return: ???
    STACK unchanged
-----b-60----DI0B11-----
INT 60 u - HP 95LX System Manager - PARSE TIME SPECIFICATION
    DI = 0B11h
    STACK:  2 WORDs unused dummies (for calls from high level languages)
    WORD   ???
    DWORD  pointer to ???
    DWORD  pointer to ???
Return: ???
    STACK unchanged
-----b-60----DI0B12-----
```

INT 60 u - HP 95LX System Manager - SET DATE PARSING RULE

DI = 0B12h

STACK: 2 WORDs unused dummies (for calls from high level languages)

WORD new parsing rule (see #03299)

Return: ???

STACK unchanged

SeeAlso: INT 60/DI=0B13h

(Table 03299)

Values for HP 95LX date parsing rule:

01h day-month-year

02h month-day-year

03h year-month-day

04h "DMYO"

05h "MDYO"

OR with 08h to get any year

-----b-60-----DI0B13-----

INT 60 u - HP 95LX System Manager - SET TIME PARSING RULE

DI = 0B13h

STACK: 2 WORDs unused dummies (for calls from high level languages)

WORD new parsing rule (see #03300)

Return: ???

STACK unchanged

SeeAlso: INT 60/DI=0B12h

(Table 03300)

Values for HP 95LX time parsing rule:

01h HH:MM:SS (am/pm)

02h HH:MM:SS (24hr)

03h HHMM:SS (24hr)

04h HH:MM:SS.hh (24hr)

05h HH:MM (am/pm)

06h HH:MM (24hr)

07h HHMM (24hr)

-----b-60-----DI0B14-----

INT 60 u - HP 95LX System Manager - CLOCK/CALENDAR SERVICE "M_POST_TIME"

DI = 0B14h

STACK: 2 WORDs unused dummies (for calls from high level languages)

Return: ???

STACK unchanged

-----b-60-----DI0B15-----

INT 60 u - HP 95LX System Manager - CLOCK/CALENDAR SERVICE "M_DAY_TRIGGER"

DI = 0B15h

STACK: 2 WORDs unused dummies (for calls from high level languages)

WORD ???

Return: ???

STACK unchanged

SeeAlso: INT 15/AX=4DD4h, INT 61"HP 95LX", INT 62"HP 95LX"

-----b-60----DI0C00-----

INT 60 u - HP 95LX System Manager - OPEN PRINTER

DI = 0C00h

STACK: 2 WORDs unused dummies (for calls from high level languages)

Return: ???

STACK unchanged

SeeAlso: INT 60/DI=0C01h, INT 60/DI=0C02h, INT 60/DI=0C03h

-----b-60----DI0C01-----

INT 60 u - HP 95LX System Manager - CLOSE PRINTER

DI = 0C01h

STACK: 2 WORDs unused dummies (for calls from high level languages)

Return: ???

STACK unchanged

Note: relinquishes control of printer

SeeAlso: INT 60/DI=0C00h

-----b-60----DI0C02-----

INT 60 u - HP 95LX System Manager - WRITE TO PRINTER

DI = 0C02h

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to data to be written

WORD length of data

Return: ???

STACK unchanged

SeeAlso: INT 60/DI=0C00h

-----b-60----DI0C03-----

INT 60 u - HP 95LX System Manager - INITIALIZE PRINTER

DI = 0C03h

STACK: 2 WORDs unused dummies (for calls from high level languages)

Return: ???

STACK unchanged

SeeAlso: INT 60/DI=0C00h

-----b-60----DI0C04-----

INT 60 u - HP 95LX System Manager - "M_TRANS_PRINTER"

DI = 0C04h

STACK: 2 WORDs unused dummies (for calls from high level languages)

WORD ???

DWORD pointer to ???

Return: ???

STACK unchanged

-----b-60----DI0C05-----

INT 60 u - HP 95LX System Manager - "M_FALL_PRINTER"

DI = 0C05h

STACK: 2 WORDs unused dummies (for calls from high level languages)

WORD ???

DWORD pointer to ???

Return: ???

STACK unchanged

SeeAlso: INT 15/AX=4DD4h, INT 61"HP 95LX", INT 62"HP 95LX"

-----b-60----DI0E00-----

INT 60 u - HP 95LX System Manager - COMMUNICATIONS SERVICE "M_COMM_INIT"

DI = 0E00h

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to ???

Return: ???

STACK unchanged

SeeAlso: #03301, INT 60/DI=0E01h, INT 60/DI=0E02h

(Table 03301)

Values for HP 95LX error code:

0000h successful

FFF1h "E_BUSY"

FFF2h timeout

FFF3h framing error

FFF4h parity error

FFF5h overrun error

FFF6h "E_EMPTY"

FFF7h "E_CONNECT"

FFF8h not open

FFF9h out of memory

FFFAh buffer overflow

FFFBh "E_NOFIT"

FFFCh unsupported

FFFDh "E_IVOPR"

FFFEh "E_IVCHN"

FFFFh "E_REOPEN"

-----b-60----DI0E01-----

INT 60 u - HP 95LX System Manager - OPEN COMMUNICATIONS CHANNEL

DI = 0E01h

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to WORD buffer for comm channel handle

WORD communications line number (01h-04h)

Return: ???

STACK unchanged

SeeAlso: INT 60/DI=0E00h,INT 60/DI=0E02h

-----b-60----DI0E02-----

INT 60 u - HP 95LX System Manager - CLOSE COMMUNICATIONS CHANNEL

DI = 0E02h

STACK: 2 WORDs unused dummies (for calls from high level languages)

WORD comm channel handle

Return: ???

STACK unchanged

SeeAlso: INT 60/DI=0E00h,INT 60/DI=0E01h

-----b-60----DI0E03-----

INT 60 u - HP 95LX System Manager - "M_COMM_GETTDM"

DI = 0E03h

STACK: 2 WORDs unused dummies (for calls from high level languages)

WORD ???

Return: ???

STACK unchanged

-----b-60----DI0E04-----

INT 60 u - HP 95LX System Manager - "M_COMM_ANSWER"

DI = 0E04h

STACK: 2 WORDs unused dummies (for calls from high level languages)

WORD ???

WORD ???

Return: ???

STACK unchanged

-----b-60----DI0E05-----

INT 60 u - HP 95LX System Manager - "M_COMM_DIAL"

DI = 0E05h

STACK: 2 WORDs unused dummies (for calls from high level languages)

WORD ???

DWORD pointer to ???

Return: ???

STACK unchanged

-----b-60----DI0E06-----

INT 60 u - HP 95LX System Manager - RESET COMMUNICATIONS CHANNEL

DI = 0E06h

STACK: 2 WORDs unused dummies (for calls from high level languages)

WORD comm channel handle

WORD reset options (see #03302)

Return: ???

STACK unchanged

Bitfields for HP 95LX reset options:

Bit(s) Description (Table 03302)

0 reset line

1 flush transmit buffer

2 flush receive buffer

3 reset modem

4 reset receiver's ^S state

5 reset transmitter's ^S state

-----b-60----DI0E07-----

INT 60 u - HP 95LX System Manager - "M_COMM_HANGUP"

DI = 0E07h

STACK: 2 WORDs unused dummies (for calls from high level languages)

WORD ???

Return: ???

STACK unchanged

-----b-60----DI0E08-----

INT 60 u - HP 95LX System Manager - SEND DATA OVER COMM CHANNEL

DI = 0E08h

STACK: 2 WORDs unused dummies (for calls from high level languages)

WORD comm channel handle

DWORD pointer to data to be sent

WORD option flags

bit 0: send partial buffer

bit 1: turn on receiver after sending

DWORD pointer to WORD containing length of data to be sent

Return: length WORD updated to contain number of bytes actually sent???

STACK unchanged

SeeAlso: INT 60/DI=0E09h, INT 60/DI=0E0Bh

-----b-60----DI0E09-----

INT 60 u - HP 95LX System Manager - QUERY COMM CHANNEL TRANSMIT QUEUE

DI = 0E09h

STACK: 2 WORDs unused dummies (for calls from high level languages)

WORD ???

```
    DWORD pointer to ??? WORD
    DWORD pointer to ??? WORD
Return: ???
    STACK unchanged
SeeAlso: INT 60/DI=0E0Ah
-----b-60----DI0E0A-----
INT 60 u - HP 95LX System Manager - QUERY COMM CHANNEL RECEIVE QUEUE
    DI = 0E0Ah
    STACK:  2 WORDs unused dummies (for calls from high level languages)
            WORD comm channel handle
            DWORD pointer to WORD to get receive buffer size
            DWORD pointer to WORD to get free bytes in receive buffer
Return: ???
    STACK unchanged
SeeAlso: INT 60/DI=0E09h,INT 60/DI=0E0Bh
-----b-60----DI0E0B-----
INT 60 u - HP 95LX System Manager - RECEIVE DATA FROM COMM CHANNEL
    DI = 0E0Bh
    STACK:  2 WORDs unused dummies (for calls from high level languages)
            WORD comm channel handle
            DWORD pointer to data buffer
            DWORD pointer to WORD (call) length of data buffer
            (ret) number of bytes received
Return: ???
    STACK unchanged
SeeAlso: INT 60/DI=0E08h,INT 60/DI=0E0Ah
-----b-60----DI0E0C-----
INT 60 u - HP 95LX System Manager - "M_COMM_HAZCMD"
    DI = 0E0Ch
    STACK:  2 WORDs unused dummies (for calls from high level languages)
            WORD ???
            DWORD pointer to ???
            WORD ???
Return: ???
    STACK unchanged
-----b-60----DI0E0D-----
INT 60 u - HP 95LX System Manager - "M_COMM_COMAND"
    DI = 0E0Dh
    STACK:  2 WORDs unused dummies (for calls from high level languages)
            WORD ???
            DWORD pointer to ???
```

```
WORD ???
Return: ???
STACK unchanged
-----b-60----DI0E0E-----
INT 60 u - HP 95LX System Manager - "M_COMM_BREAK"
DI = 0E0Eh
STACK: 2 WORDs unused dummies (for calls from high level languages)
WORD ???
WORD ???
Return: ???
STACK unchanged
-----b-60----DI0E0F-----
INT 60 u - HP 95LX System Manager - "M_COMM_FRCXON"
DI = 0E0Fh
STACK: 2 WORDs unused dummies (for calls from high level languages)
WORD ???
Return: ???
STACK unchanged
-----b-60----DI0E10-----
INT 60 u - HP 95LX System Manager - "M_COMM_FRCXOF"
DI = 0E10h
STACK: 2 WORDs unused dummies (for calls from high level languages)
WORD ???
Return: ???
STACK unchanged
-----b-60----DI0E11-----
INT 60 u - HP 95LX System Manager - "M_COMM_SETDTR"
DI = 0E11h
STACK: 2 WORDs unused dummies (for calls from high level languages)
WORD ???
WORD ???
Return: ???
STACK unchanged
-----b-60----DI0E12-----
INT 60 u - HP 95LX System Manager - "M_COMM_XMITNG"
DI = 0E12h
STACK: 2 WORDs unused dummies (for calls from high level languages)
WORD ???
Return: ???
STACK unchanged
-----b-60----DI0E13-----
```

```
INT 60 u - HP 95LX System Manager - "M_COMM_STATUS"
  DI = 0E13h
  STACK: 2 WORDs unused dummies (for calls from high level languages)
  WORD ???
```

Return: ???

STACK unchanged

-----b-60----DI0E14-----

```
INT 60 u - HP 95LX System Manager - SET COMMUNICATIONS SETTINGS
  DI = 0E14h
  STACK: 2 WORDs unused dummies (for calls from high level languages)
  WORD comm channel handle
  DWORD pointer to communications settings (see #03303)
```

Return: ???

STACK unchanged

SeeAlso: INT 60/DI=0E15h

Format of HP 95LX communications settings:

Offset Size Description (Table 03303)

00h	BYTE	dial type ('T' tone, 'P' pulse)
01h	WORD	baud rate divisor (115200/ baud_rate)
03h	BYTE	parity (00h none, 08h odd, 18h even, 28h mark, 38h space)
04h	BYTE	stop bits (00h one, 04h two)
05h	BYTE	data bits - 5
06h	BYTE	software handshake
		01h none, 02h XOFF/XON, 04h XOFF/any, 08h ENQ/ACK
07h	BYTE	infrared (01h off, 02h on)
08h	BYTE	duplex (01h half, 02h full)
09h	BYTE	echo (01h echo, 02h no echo)

-----b-60----DI0E15-----

```
INT 60 u - HP 95LX System Manager - GET COMMUNICATIONS SETTINGS
  DI = 0E15h
  STACK: 2 WORDs unused dummies (for calls from high level languages)
  WORD ???
  DWORD pointer to buffer for settings (see #03303)
```

Return: ???

STACK unchanged

SeeAlso: INT 60/DI=0E14h

-----b-60----DI0E16-----

```
INT 60 u - HP 95LX System Manager - "M_COMM_CNFGUR"
  DI = 0E16h
  STACK: 2 WORDs unused dummies (for calls from high level languages)
```

WORD ???

WORD ???

WORD ???

WORD ???

WORD ???

Return: ???

STACK unchanged

-----b-60----DI0E17-----

INT 60 u - HP 95LX System Manager - "M_COMM_QRYERR"

DI = 0E17h

STACK: 2 WORDs unused dummies (for calls from high level languages)

WORD ???

Return: ???

STACK unchanged

SeeAlso: INT 15/AX=4DD4h, INT 61"HP 95LX", INT 62"HP 95LX"

-----b-60----DI0F00-----

INT 60 u - HP 95LX System Manager - "M_ERRMSG"

DI = 0F00h

STACK: 2 WORDs unused dummies (for calls from high level languages)

WORD ???

DWORD pointer to ???

WORD ???

DWORD pointer to ???

Return: ???

STACK unchanged

-----b-60----DI0F01-----

INT 60 u - HP 95LX System Manager - DRAW STANDARD TITLE BOX

DI = 0F01h

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to ASCIZ title string

Return: ???

STACK unchanged

-----b-60----DI0F02-----

INT 60 u - HP 95LX System Manager - "SHOWNAME"

DI = 0F02h

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to ???

Return: ???

STACK unchanged

-----b-60----DI0F03-----

INT 60 u - HP 95LX System Manager - DISPLAY TWO-LINE MESSAGE BOX

```
DI = 0F03h
STACK:  2 WORDs unused dummies (for calls from high level languages)
        DWORD pointer to first line of message
        WORD  length of first line
        DWORD pointer to second line of message
        WORD  length of second line
Return: ???
        STACK unchanged
SeeAlso: INT 60/DI=0300h,INT 60/DI=0F04h,INT 60/DI=0F09h
-----b-60----DI0F04-----
INT 60 u - HP 95LX System Manager - REMOVE MESSAGE BOX
        DI = 0F04h
        STACK:  2 WORDs unused dummies (for calls from high level languages)
Return: ???
        STACK unchanged
SeeAlso: INT 60/DI=0F03h,INT 60/DI=0F09h
-----b-60----DI0F05-----
INT 60 u - HP 95LX System Manager - "M_COM_TIMER_ADDR"
        DI = 0F05h
        STACK:  2 WORDs unused dummies (for calls from high level languages)
Return: DX:AX -> ???
        STACK unchanged
-----b-60----DI0F06-----
INT 60 u - HP 95LX System Manager - "M_COM_TIMER_COUNT_ADDR"
        DI = 0F06h
        STACK:  2 WORDs unused dummies (for calls from high level languages)
Return: DX:AX -> ???
        STACK unchanged
-----b-60----DI0F07-----
INT 60 u - HP 95LX System Manager - "M_SYS_RSRC_ADDR"
        DI = 0F07h
        STACK:  2 WORDs unused dummies (for calls from high level languages)
Return: DX:AX -> ???
        STACK unchanged
-----b-60----DI0F08-----
INT 60 u - HP 95LX System Manager - "M_BIOS_OUTSTR"
        DI = 0F08h
        STACK:  2 WORDs unused dummies (for calls from high level languages)
        ???
Return: ???
        STACK unchanged
```


-----b-60----DI0F09-----

INT 60 u - HP 95LX System Manager - DISPLAY THREE-LINE MESSAGE BOX

DI = 0F09h

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to first line of message

WORD length of first line

DWORD pointer to second line of message

WORD length of second line

DWORD pointer to third line of message

WORD length of third line

Return: ???

STACK unchanged

SeeAlso: INT 60/DI=0F03h,INT 60/DI=0F04h

-----b-60----DI0F0A-----

INT 60 u - HP 95LX System Manager - DISABLE MACROS

DI = 0F0Ah

STACK: 2 WORDs unused dummies (for calls from high level languages)

Return: ???

STACK unchanged

SeeAlso: INT 60/DI=0F0Bh

-----b-60----DI0F0B-----

INT 60 u - HP 95LX System Manager - ENABLE MACROS

DI = 0F0Bh

STACK: 2 WORDs unused dummies (for calls from high level languages)

Return: ???

STACK unchanged

SeeAlso: INT 60/DI=0F0Ah

-----b-60----DI0F0C-----

INT 60 u - HP 95LX System Manager - "M_DATE_TIME_SEPS"

DI = 0F0Ch

STACK: 2 WORDs unused dummies (for calls from high level languages)

???

Return: ???

STACK unchanged

-----b-60----DI0F0D-----

INT 60 u - HP 95LX System Manager - "M_FORM_FT"

DI = 0F0Dh

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to ???

Return: DX:AX -> ???

STACK unchanged

-----b-60----DI0F0E-----

INT 60 u - HP 95LX System Manager - "M_RAM_IV_INFO"

DI = 0F0Eh

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to ???

Return: DX:AX -> ???

STACK unchanged

SeeAlso: INT 15/AX=4DD4h, INT 61"HP 95LX", INT 62"HP 95LX"

-----b-60----DI1005-----

INT 60 u - HP 95LX System Manager - "M_DIRTY_SYNC" - FORCE SCREEN UPDATE

DI = 1005h

STACK: 2 WORDs unused dummies (for calls from high level languages)

Return: ???

STACK unchanged

SeeAlso: INT 10/AH=FFh, INT 60/DI=0300h, INT 60/DI=0301h

-----b-60----DI1200-----

INT 60 u - HP 95LX System Manager - RESOURCE SERVICE "MAP_RESOURCE_FILE"

DI = 1200h

STACK: 2 WORDs unused dummies (for calls from high level languages)

DWORD pointer to ???

Return: ???

STACK unchanged

-----b-60----DI1201-----

INT 60 u - HP 95LX System Manager - "GET_RESOURCE_PTR"

DI = 1201h

STACK: 2 WORDs unused dummies (for calls from high level languages)

WORD ???

Return: DX:AX -> ???

STACK unchanged

SeeAlso: INT 15/AX=4DD4h, INT 61"HP 95LX", INT 62"HP 95LX"

-----b-60----DI1202-----

INT 60 u - HP 95LX System Manager - "GET_RSRC_TAB_PTR"

DI = 1202h

STACK: 2 WORDs unused dummies (for calls from high level languages)

Return: DX:AX -> ???

STACK unchanged

-----b-60----DI1203-----

INT 60 u - HP 95LX System Manager - "INIT_SYSMGR_RSRCs"

DI = 1203h

STACK: 2 WORDs unused dummies (for calls from high level languages)

Return: ???

```
STACK unchanged
-----b-60-----DI1300-----
INT 60 u - HP 95LX System Manager - INITIALIZE HELP SYSTEM
DI = 1300h
STACK: 2 WORDs unused dummies (for calls from high level languages)
        DWORD pointer to ???
        DWORD pointer to ???
        WORD  ???
Return: ???
STACK unchanged
-----b-60-----DI1301-----
INT 60 u - HP 95LX System Manager - DISPLAY HELP
DI = 1301h
STACK: 2 WORDs unused dummies (for calls from high level languages)
        DWORD pointer to ???
Return: ???
STACK unchanged
-----b-60-----DI1302-----
INT 60 u - HP 95LX System Manager - "M_HELP_KEY"
DI = 1302h
STACK: 2 WORDs unused dummies (for calls from high level languages)
        DWORD pointer to ???
        WORD  ???
Return: ???
STACK unchanged
-----b-60-----DI1303-----
INT 60 u - HP 95LX System Manager - "M_HELP_TERM"
DI = 1303h
STACK: 2 WORDs unused dummies (for calls from high level languages)
        DWORD pointer to ???
Return: ???
STACK unchanged
SeeAlso: INT 15/AX=4DD4h, INT 61"HP 95LX", INT 62"HP 95LX"
-----b-60-----DI1400-----
INT 60 u - HP 95LX System Manager - "M_ColInit"
DI = 1400h
STACK: 2 WORDs unused dummies (for calls from high level languages)
Return: AX = ???
STACK unchanged
-----b-60-----DI1401-----
INT 60 u - HP 95LX System Manager - "M_ColCpStr"
```

```
DI = 1401h
STACK: 2 WORDs unused dummies (for calls from high level languages)
  DWORD pointer to ???
  WORD ???
  DWORD pointer to ???
  WORD ???
Return: ???
  STACK unchanged
-----b-60----DI1402-----
INT 60 u - HP 95LX System Manager - "M_ColLicsStr"
  DI = 1402h
  STACK: 2 WORDs unused dummies (for calls from high level languages)
  ???
Return: ???
  STACK unchanged
-----b-60----DI1403-----
INT 60 u - HP 95LX System Manager - "M_ColLicsChar"
  DI = 1403h
  STACK: 2 WORDs unused dummies (for calls from high level languages)
  ???
Return: ???
  STACK unchanged
-----b-60----DI1404-----
INT 60 u - HP 95LX System Manager - "M_ColToLower"
  DI = 1404h
  STACK: 2 WORDs unused dummies (for calls from high level languages)
  DWORD pointer to ???
  WORD ???
Return: ???
  STACK unchanged
-----b-60----DI1405-----
INT 60 u - HP 95LX System Manager - "M_ColCpSearch"
  DI = 1405h
  STACK: 2 WORDs unused dummies (for calls from high level languages)
  DWORD pointer to ???
  WORD ???
  DWORD pointer to ???
  WORD ???
  WORD ???
Return: ???
  STACK unchanged
```

```
-----b-60----DI1406-----
INT 60 u - HP 95LX System Manager - "M_ColToUpper"
  DI = 1406h
  STACK: 2 WORDs unused dummies (for calls from high level languages)
  DWORD pointer to ???
  WORD ???
Return: ???
  STACK unchanged
SeeAlso: INT 15/AX=4DD4h,INT 61"HP 95LX",INT 62"HP 95LX"
-----b-60----DI1500-----
INT 60 u - HP 95LX System Manager - "GrDispInit"
  DI = 1500h
  STACK: 2 WORDs unused dummies (for calls from high level languages)
  ???
Return: ???
  STACK unchanged
-----b-60----DI1501-----
INT 60 u - HP 95LX System Manager - "GrDispClear"
  DI = 1501h
  STACK: 2 WORDs unused dummies (for calls from high level languages)
  ???
Return: ???
  STACK unchanged
-----b-60----DI1502-----
INT 60 u - HP 95LX System Manager - "GrDispDot"
  DI = 1502h
  STACK: 2 WORDs unused dummies (for calls from high level languages)
  ???
Return: ???
  STACK unchanged
-----b-60----DI1503-----
INT 60 u - HP 95LX System Manager - "GrDispDraw"
  DI = 1503h
  STACK: 2 WORDs unused dummies (for calls from high level languages)
  ???
Return: ???
  STACK unchanged
-----b-60----DI1504-----
INT 60 u - HP 95LX System Manager - "GrDispFill"
  DI = 1504h
  STACK: 2 WORDs unused dummies (for calls from high level languages)
```

???

Return: ???

STACK unchanged

-----b-60----DI1505-----

INT 60 u - HP 95LX System Manager - "GrDispRead"

DI = 1505h

STACK: 2 WORDs unused dummies (for calls from high level languages)

???

Return: ???

STACK unchanged

-----b-60----DI1506-----

INT 60 u - HP 95LX System Manager - "GrDispString"

DI = 1506h

STACK: 2 WORDs unused dummies (for calls from high level languages)

???

Return: ???

STACK unchanged

-----b-60----DI1507-----

INT 60 u - HP 95LX System Manager - "GrDispPan"

DI = 1507h

STACK: 2 WORDs unused dummies (for calls from high level languages)

???

Return: ???

STACK unchanged

-----b-60----DI1508-----

INT 60 u - HP 95LX System Manager - "GrDispZoom"

DI = 1508h

STACK: 2 WORDs unused dummies (for calls from high level languages)

???

Return: ???

STACK unchanged

-----b-60----DI1509-----

INT 60 u - HP 95LX System Manager - "GrDispSave"

DI = 1509h

STACK: 2 WORDs unused dummies (for calls from high level languages)

???

Return: ???

STACK unchanged

-----b-60----DI150A-----

INT 60 u - HP 95LX System Manager - "GrDispRestore"

DI = 150Ah

STACK: 2 WORDs unused dummies (for calls from high level languages)

???

Return: ???

STACK unchanged

-----b-60----DI150B-----

INT 60 u - HP 95LX System Manager - "GrDispCorner"

DI = 150Bh

STACK: 2 WORDs unused dummies (for calls from high level languages)

???

Return: ???

STACK unchanged

SeeAlso: INT 15/AX=4DD4h, INT 61"HP 95LX", INT 62"HP 95LX"

-----b-60----DI1604-----

INT 60 u - HP 95LX System Manager - "CP_TO_LICS"

DI = 1604h

STACK: 2 WORDs unused dummies (for calls from high level languages)

???

Return: ???

STACK unchanged

SeeAlso: INT 15/AX=4DD4h, INT 61"HP 95LX", INT 62"HP 95LX"

-----G-6000-----

INT 60 - SYS_PROF.EXE - PROFILER STATUS

AH = 00h

Return: AX = 0000h profilng is off

 otherwise profilng is on

Note: SYS_PROF.EXE is the TSR portion of a profiler from Micro Cornucopia

Issue 47

SeeAlso: AH=01h"SYS_PROF", AH=02h"SYS_PROF"

-----G-6000-----

INT 60 - MDEBUG - GET STATUS

AH = 00h

DS:SI -> password or a null byte

Return: AX = return code

 FFFEh password is invalid

 FFFDh display mode is invalid

 else successful

ES = value of the monitor register SE

DI = value of the monitor register OF

CH = monitor color

CL = interpreter color

BH = monitor start line

BL = interpreter start line
 AH = makecode of the hotkey
 AL = ASCII code of the hotkey
 DL = status of special keys (only SHIFT, ALT, CTRL) for the
 hotkey (coded as for the keyboard flag at 0040h:0017h)
 DH = basic process number for the communication with drivers
 process number for the display driver, DH+1 = process
 number for the command driver(s)

DS:SI -> MDEBUG identification table (see #03304)

Program: MDEBUG is a shareware memory-resident debugging tool by Bernd
 Schemmer, including a memory monitor, an interpreter, and a
 disassembler

Notes: MDEBUG uses INT 60 by default, but may be directed to any of INT 60
 through INT 67; the interrupt handler is preceded by the signature
 "USERINT" and is not chained

if DS:SI points at a null byte, MDEBUG will prompt for a password if
 passwords are active; enough stack space must be provided for an
 INT 10h call (which MDEBUG uses while prompting for the password)

SeeAlso: AH=02h"MDEBUG"

Index: hotkeys;MDEBUG

Format of MDEBUG identification table:

Offset Size Description (Table 03304)

-2	WORD	entry offset
00h	WORD	CS of MDEBUG
02h	DWORD	old INT 08h vector
06h	DWORD	old INT 09h vector
0Ah	DWORD	address INT 16h routine used by MDEBUG
0Eh	BYTE	length of version string
0Fh	N BYTES	version string

-----N-600000-----

INT 60 - RIFS - CLIENT - INSTALLATION CHECK

AX = 0000h

Return: AX = 1234h if installed

CF clear

Program: RIFS is the Remote Installable File System by "kyle"

InstallCheck: test for the signature "RIFS" immediately preceding the
 interrupt handler

Range: INT 60 to INT 66 and INT 18, selected by scanning for 0000h:0000h vector

SeeAlso: AX=0001h,AX=0005h,AX=0007h,AX=0008h,INT 2F/AX=5600h

-----N-600001-----

INT 60 - RIFS - CLIENT - UNINSTALL

AX = 0001h

Return: CF clear if successful

SeeAlso: AX=0000h,AX=0009h

-----N-600002-----

INT 60 - RIFS - CLIENT - REMAP DRIVE

AX = 0002h

BH = local drive number

BL = remote drive number

Return: CF clear if successful

CF set on error

SeeAlso: AX=0000h,AX=0003h,AX=0004h,INT 21/AX=5F03h,INT 21/AX=5F05h"STARLITE"

-----N-600003-----

INT 60 - RIFS - CLIENT - UNMAP DRIVE

AX = 0003h

BL = drive to unmap

Return: CF clear if successful

CF set on error

SeeAlso: AX=0000h,AX=0002h,AX=0004h,INT 21/AX=5F04h,INT 21/AX=5F06h"STARLITE"

-----N-600004-----

INT 60 - RIFS - CLIENT - UNMAP ALL DRIVES

AX = 0004h

Return: CF clear if successful

CF set on error

SeeAlso: AX=0000h,AX=0002h,AX=0003h

-----N-600005-----

INT 60 - RIFS - CLIENT - GET TRANSLATION TABLE

AX = 0005h

Return: CF clear if successful

ES:BX -> translation table

CF set on error

SeeAlso: AX=0000h,AX=0006h,AX=0007h

-----N-600006-----

INT 60 - RIFS - CLIENT - GET STATISTICS TABLE

AX = 0006h

Return: CF clear if successful

ES:BX -> statistics table

CF set on error

SeeAlso: AX=0000h,AX=0005h,AX=0007h,AX=000Ah

-----N-600007-----

INT 60 - RIFS - CLIENT - GET PORT TRANSLATION TABLE

```
AX = 0007h
Return: CF clear if successful
      CX = number of entries
      ES:BX -> port mapping table
      CF set on error
SeeAlso: AX=0000h,AX=0005h
-----N-600008-----
INT 60 - RIFS - SERVER - INSTALLATION CHECK
      AX = 0008h
Return: CF clear if successful
      CF set on error
InstallCheck: test for the signature "RIFS" immediately preceding the
      interrupt handler
Range: INT 60 to INT 66 and INT 18,selected by scanning for 0000h:0000h vector
SeeAlso: AX=0000h,AX=0009h,AX=000Ah,AX=000Bh
-----N-600009-----
INT 60 - RIFS - SERVER - UNINSTALL
      AX = 0009h
Return: CF clear if successful
      CF set on error
SeeAlso: AX=0001h,AX=0008h
-----N-60000A-----
INT 60 - RIFS - SERVER - GET STATISTICS TABLE
      AX = 000Ah
Return: CF clear if successful
      ES:BX -> statistics table
      CF set on error
SeeAlso: AX=0006h,AX=0008h,AX=0009h,AX=000Bh
-----N-60000B-----
INT 60 - RIFS - SERVER - RESET
      AX = 000Bh
Return: CF clear if successful
      CF set on error
Note: closes all open files
SeeAlso: AX=0008h,AX=0009h
-----G-6001-----
INT 60 - MDEBUG - GET ADDRESS OF THE HELP REGISTERS
      AH = 01h
      DS:SI -> password or a null byte
Return: AX = return code
      FFFEh password is invalid
```

```

FFFFh display mode is invalid
else successful
ES:DI point to the help registers of MDEBUG
  ES:DI-02h  -> R0 (WORD)
  ES:DI     -> R1 (WORD)
  ES:DI+02h -> R2 (WORD)
  ES:DI+04h -> R3 (WORD)
  ...
  ES:DI+0Eh -> R8 (WORD)

```

```
-----G-6001-----
```

```
INT 60 - SYS_PROF.EXE - TURN PROFILING OFF
```

```
  AH = 01h
```

Note: SYS_PROF.EXE is the TSR portion of a profiler from Micro Cornucopia

Issue 47

```
SeeAlso: AH=00h"SYS_PROF",AH=02h"SYS_PROF"
```

```
-----N-6001FF-----
```

```
INT 60 - FTP Packet Driver - BASIC FUNC - GET DRIVER INFO
```

```
  AX = 01FFh
```

```
  BX = handle returned by function 02h
```

Return: CF set on error

```
  DH = error code (see #03305)
```

CF clear if successful

```
  BX = version
```

```
  CH = network interface class (see #03306)
```

```
  DX = interface type (see #03306)
```

```
  CL = number
```

```
  DS:SI -> name
```

```
  AL = driver functions supported
```

```
01h basic
```

```
02h basic and extended
```

```
05h basic and high-performance
```

```
06h basic, high-performance, and extended
```

```
FFh not installed
```

Note: the handle in BX is optional for drivers written to v1.07 or later of the packet driver specification

(Table 03305)

Values for Packet Driver error code:

```
01h "BAD_HANDLE" invalid handle number
```

```
02h "NO_CLASS"   no interfaces of the specified class found
```

```
03h "NO_TYPE"   no interfaces of the specified type found
```

04h "NO_NUMBER" no interfaces of the specified number found
05h "BAD_TYPE" bad packet type
06h "NO_MULTICAST" interface does not support multicast messages
07h "CANT_TERMINATE" this packet driver cannot terminate
08h "BAD_MODE" invalid receiver mode
09h "NO_SPACE" insufficient space
0Ah "TYPE_INUSE" type accessed but never released
0Bh "BAD_COMMAND" bad command
0Ch "CANT_SEND" packet could not be sent
0Dh "CANT_SET" hardware address could not be changed
0Eh "BAD_ADDRESS" hardware address has a bad length or format
0Fh "CANT_RESET" could not reset interface

(Table 03306)

Values for Packet Driver network interface classes/types:

Class 01h Ethernet/IEEE 802.3
01h 3COM 3C500/3C501
02h 3COM 3C505
03h MICOM-Interlan NI5010
04h BICC Data Networks 4110
05h BICC Data Networks 4117
06h MICOM-Interlan NP600
08h Ungermann-Bass PC-NIC
09h Univation NC-516
0Ah TRW PC-2000
0Bh MICOM-Interlan NI5210
0Ch 3COM 3C503
0Dh 3COM 3C523
0Eh Western Digital WD8003
0Fh Spider Systems S4
10h Torus Frame Level
11h 10Net Communications
12h Gateway PC-bus
13h Gateway AT-bus
14h Gateway MCA-bus
15h IMC PCnic
16h IMC PCnic II
17h IMC PCnic 8-bit
18h Tigan Communications
19h Micromatic Research
1Ah Clarkson "Multiplexor"

1Bh D-Link 8-bit
1Ch D-Link 16-bit
1Dh D-Link PS/2
1Eh Research Machines 8
1Fh Research Machines 16
20h Research Machines MCA
21h Radix Microsystems EXM1 16-bit
22h Interlan Ni9210
23h Interlan Ni6510
24h Vestra LANMASTER 16-bit
25h Vestra LANMASTER 8-bit
26h Allied Telesis PC/XT/AT
27h Allied Telesis NEC PC-98
28h Allied Telesis Fujitsu FMR
29h Ungermann-Bass NIC/PS2
2Ah Tiara LANCard/E AT
2Bh Tiara LANCard/E MC
2Ch Tiara LANCard/E TP
2Dh Spider Communications SpiderComm 8
2Eh Spider Communications SpiderComm 16
2Fh AT&T Starlan NAU
30h AT&T Starlan-10 NAU
31h AT&T Ethernet NAU
32h Intel smart card
33h Xircom Packet Adapter
34h Aquila Ethernet
35h Novell NE1000
36h Novell NE2000
37h SMC PC-510
38h AT&T Fiber NAU
39h NDIS to Packet Driver adapter
3Ah Racal-InterLan ES3210
3Bh General Systems ISDN simulated Ethernet
3Ch Hewlett-Packard
3Dh IMC EtherNic-8
3Eh IMC EtherNic-16
3Fh IMC EtherNic-MCA
40h NetWorth EtherNext
41h Dataco Scanet
42h DEC DEPCA
43h C-Net

44h Gandalf LANLine
45h Apricot built-in
46h David Systems Ether-T
47h ODI to Packet Driver adapter (see also INT 2F/AX=5100h)
48h AMD Am21110-16
49h Intel ICD Network controller family
4Ah Intel ICD PCL2
4Bh Intel ICD PCL2A
4Ch AT&T LANPacer
4Dh AT&T LANPacer+
4Eh AT&T EVB
4Fh AT&T StarStation
50h SLIP simulated ethernet
51h Racal-Interlan NIA310
52h Racal-Interlan NISE
53h Racal-Interlan NISE30
54h Racal-Interlan NI6610
55h Ethernet over IP/UDP
56h ICL EtherTeam 16
57h David Systems
58h NCR WaveLAN
59h Thomas Contrad TC5045
5Ah Russ Nelson's Parallel Port driver
5Bh Intell EtherExpress 16
5Ch IBMTOKEN
5Dh Zenith Z-Note
5Eh 3Com 3C509
5Fh Mylex LNE390
60h Madge Smart Ringnode
61h Novell NE2100
62h Allied Telesis 1500
63h Allied Telesis 1700
64h Fujitsu EtherCoupler
 Class 02h ProNET-10
01h Proteon p1300
02h Proteon p1800
 Class 03h IEEE 802.5/ProNet-4 (without expanded RIFs)
01h IBM Token-Ring Adapter
02h Proteon p1340
03h Proteon p1344
04h Gateway PC-bus

```
05h Gateway AT-bus
06h Gateway MCA-bus
07h Madge board
39h NDIS to Packet Driver adapter
47h ODI to Packet Driver adapter
    Class 04h Omninet
    Class 05h Appletalk
01h ATALK.SYS adapter
    Class 06h Serial Line
01h Clarkson 8250-SLIP
02h Clarkson "Multiplexor"
03h Eicon Technologies
    Class 07h StarLAN (subsumed by Ethernet class)
    Class 08h ARCnet
01h Datapoint RIM
    Class 09h AX.25
01h Ottawa PI card
02h Eicon Technologies
    Class 0Ah KISS
    Class 0Bh IEEE 802.3 with 802.2 headers
types same as for class 01h
    Class 0Ch FDDI with 802.2 headers
01h Western Digital
02h Frontier Technology
    Class 0Dh Internet X.25
01h Western Digital
02h Frontier Technology
03h Emerging Technologies
04h The Software Forge
05h Link Data Intelligent X.25
06h Eicon Technologies
    Class 0Eh N.T. LANSTAR (encapsulating DIX Ethernet)
01h NT LANSTAR/8
02h NT LANSTAR/MC
    Class 0Fh SLFP (MIT serial specification)
01h MERIT
    Class 10h PPP (Point-to-Point Protocol)
01h 8250/16550 UART
02h Niwot Networks synch
03h Eicon Technologies
    Class 11h 802.5 with expanded RIFs
```

types same as for class 3

Class 12h reserved for LCP/NCPs

Note: class and type numbers are cleared through FTP Software

-----G-6002-----

INT 60 - MDEBUG - SET STATUS

AH = 02h

DS:SI -> password or a null byte

ES = new value for the register SE

DI = new value for the register OF

CH = new monitor color if nonzero

CL = new interpreter color if nonzero

BH = new monitor start line if nonzero

BL = new interpreter start line if nonzero

AL = new ASCII code for the hotkey ('A'..'Z', 'a'..'z') if nonzero

DL = new status of the special keys (SHIFT, ALT, CTRL) for the hotkey
if nonzero

DH = if nonzero, new basic process number for communication with the
drivers (DH = multiplex number for the display driver,
DH+1 = multiplex number for the command driver or drivers)

Return: AX = return code

FFFFh call not allowed

FFFEh password is invalid

FFFDh display mode is invalid

0000h successful, status changed

else AL = error reasons (see #03307)

Note: the values of the registers SE and OF are always changed, the other
values are only changed if they are valid

SeeAlso: AH=00h"MDEBUG"

Index: hotkeys;MDEBUG

Bitfields for MDEBUG error reasons:

Bit(s) Description (Table 03307)

0 invalid monitor start line

1 invalid interpreter start line

2 invalid hotkey

3 invalid process number

4-7 reserved

-----N-6002-----

INT 60 - FTP Packet Driver - BASIC FUNC - ACCESS TYPE

AH = 02h

AL = interface class

BX = interface type
 DL = interface number
 DS:SI -> type
 CX = length of type (0000h for all packets)
 ES:DI -> receiver function (see #03308)

Return: CF set on error
 DH = error code (see #03305)
 CF clear if successful
 AX = handle
 SeeAlso: AH=03h"FTP"

(Table 03308)

Values packet driver receiver is called with when a packet is received:

AX = subfunction
 00h get packet buffer
 CX = buffer length
 DX = lookahead length (v1.10+)
 DS:SI -> lookahead buffer if DX nonzero (v1.10+)
 DI = error flags (class dependent) (v1.10+)
 Return: ES:DI -> packet buffer
 0000h:0000h means throw away packet
 CX = size of buffer (v1.10+), may be smaller than
 incoming data
 01h copy completed
 DS:SI -> buffer
 CX = bytes actually copied (v1.10+)
 BX = handle

-----G-6002-----

INT 60 - SYS_PROF.EXE - TURN PROFILING ON

AH = 02h

Note: SYS_PROF.EXE is the TSR portion of a profiler from Micro Cornucopia
 Issue 47

SeeAlso: AH=00h"SYS_PROF",AH=01h"SYS_PROF"

-----G-6003-----

INT 60 - MDEBUB - POP UP

AH = 03h

DS:SI -> password or a null byte

ES -> new value for the register SE

DI -> new value for the register OF

Return: AX = return code (see #03309)

SeeAlso: AH=04h"MDEBUB"

(Table 03309)

Values for MDEBUG return code:

FFFFh call not allowed
FFFEh password is invalid
FFFDh display mode is invalid
else successful

-----N-6003-----

INT 60 - FTP Packet Driver - BASIC FUNC - RELEASE TYPE

AH = 03h
BX = handle

Return: CF set on error

DH = error code (see #03305)

CF clear if successful

SeeAlso: AH=02h"FTP"

-----G-6003-----

INT 60 - SYS_PROF.EXE - GET ADDRESS OF PROFILING TABLE

AH = 03h

Return: ES:BX -> profiling table

Note: SYS_PROF.EXE is the TSR portion of a profiler from Micro Cornucopia

Issue 47

SeeAlso: AH=04h"SYS_PROF"

-----N-6004-----

INT 60 - FTP Packet Driver - BASIC FUNC - SEND PACKET

AH = 04h
DS:SI -> buffer
CX = length

Return: CF set on error

DH = error code (see #03305)

CF clear if successful

Note: the buffer may be modified immediately upon return from this call

SeeAlso: AH=0Bh

-----G-6004-----

INT 60 - MDEBUG - POP UP

AH = 04h
DS:SI -> password or a null byte

Return: AX = return code (see #03309)

SeeAlso: AH=03h"MDEBUG",AH=07h"MDEBUG"

-----G-6004-----

INT 60 - SYS_PROF.EXE - CLEAR PROFILING TABLE

AH = 04h

Note: SYS_PROF.EXE is the TSR portion of a profiler from Micro Cornucopia

Issue 47

SeeAlso: AH=03h"SYS_PROF"

-----N-6005-----

INT 60 - FTP Packet Driver - BASIC FUNC - TERMINATE DRIVER FOR HANDLE

AH = 05h

BX = handle (optional for v1.10+)

Return: CF set on error

DH = error code (see #03305)

CF clear if successful

-----G-6005-----

INT 60 - MDEBUG - GET AND SET MDEBUG FLAGS

AH = 05h

DS:SI -> password or a null byte

BL = new value for the semaphor of MDEBUG

00h enable popup of MDEBUG

else disable popup of MDEBUG

Return: AX = return code

FFFEh password is invalid

FFFDh display mode is invalid

else successful

BL = old value of the semaphor of MDEBUG

BH = old value of the INT 08h semaphor

(this semaphor is always reset after this function)

-----N-6006-----

INT 60 - FTP Packet Driver - BASIC FUNC - GET ADDRESS

AH = 06h

BX = handle (optional for v1.10+)

ES:DI -> buffer

CX = length

Return: CF set on error

DH = error code (see #03305)

CF clear if successful

CX = length

Note: copies the local net address associated with the handle into the buffer

-----G-6006-----

INT 60 - MDEBUG - GET PASSWORD STATUS

AH = 06h

Return: AL = status

00h password inactive

01h password active

-----N-6007-----

INT 60 - FTP Packet Driver - BASIC FUNC - RESET INTERFACE

AH = 07h

BX = handle (optional for v1.10+)

Return: CF set on error

DH = error code (see #03305)

CF clear if successful

-----G-6007-----

INT 60 - MDEBUG v1.70+ - GET ACTIVE PART OF MDEBUG

AH = 07h

Return: AL = active part for the next popup session of MDEBUG:

bit 0: the next popup session will start in the interpreter rather
than in the monitor

bit 1: the next popup session will start in the online-help

SeeAlso: AH=03h"MDEBUG",AH=04h"MDEBUG"

-----G-6008-----

INT 60 - MDEBUG - UNUSED

AH = 08h-FFh

Return: AX = FFFCh

-----N-600A-----

INT 60 - FTP Packet Driver 1.09+ - HIGH-PERF FUNC - GET PARAMETERS

AH = 0Ah

Return: CF set on error

DH = error code (0Bh) (see #03305)

CF clear if successful

ES:DI -> parameter table (see #03310)

Format of packet driver parameter table:

Offset Size Description (Table 03310)

00h	BYTE	major revision of packet driver spec driver conforms to
01h	BYTE	minor revision of packet driver spec
02h	BYTE	length of this structure in bytes
03h	BYTE	length of a MAC-layer address
04h	WORD	maximum transfer unit, including MAC headers
06h	WORD	buffer size for multicast addr
08h	WORD	number of receive buffers (one less than back-to-back MTU rcvs)
0Ah	WORD	number of transmit buffers
0Ch	WORD	interrupt number to hook for post-EOI processing, 00h=none

-----N-600B-----

INT 60 - FTP Packet Driver 1.09 - HIGH-PERF FUNC - ASYNCHRONOUS SEND PACKET

AH = 0Bh

DS:SI -> buffer
 CX = length of buffer
 ES:DI -> FAR function to call when buffer becomes available (see #03311)

Return: CF set on error

DH = error code (0Bh,0Ch) (see #03305)

CF clear if successful

Notes: unlike function 04h, the buffer is not available for modification as soon as the call returns; the buffer may be queued by the driver and not processed until later

this function has been dropped from v1.10+ of the specification and replaced by function 0Ch

SeeAlso: AH=04h"Packet Driver",AH=0Ch"Packet Driver"

(Table 03311)

Values packet driver completion function is called with:

AX = result

00h copy OK

nonzero error

ES:DI -> buffer passed to INT 60/AH=0Bh call

-----N-600C-----

INT 60 - FTP Packet Driver 1.10+ - HIGH-PERF FUNC - ASYNCHRONOUS SEND PACKET

AH = 0Ch

ES:DI -> pointer to IOCB (see #03312)

Return: CF set on error

DH = error code (see #03305)

CF clear if successful

SeeAlso: AH=04h"Packet Driver",AH=0Bh"Packet Driver"

Format of packet driver IOCB:

Offset Size Description (Table 03312)

00h DWORD pointer to buffer

04h WORD length of buffer

06h BYTE flags

bit 0: packet driver is finished with IOCB

bit 1: application requests upcall when driver completes

07h DWORD function address for upcall (see #03313)

0Bh 4 BYTES future gather write

0Fh BYTE ???

10h 8 BYTES private driver workspace

(Table 03313)

Values completion function is called with:

ES:DI -> IOCB passed to INT 60/AH=0Ch

-----N-600C-----

INT 60 - Banyan VINES, 3com - GET STATION ADDRESS

AH = 0Ch

Return: AL = status

00h successful

ES:SI -> 6-byte station address

02h semaphore service is unavailable

-----N-600D-----

INT 60 - FTP Packet Driver 1.10+ - HIGH-PERF FUNC - DROP PACKET FROM QUEUE

AH = 0Dh

ES:DI -> IOCB

Return: CF set on error

DH = error code (see #03305)

CF clear if successful

SeeAlso: AH=0Ch"Packet Driver"

-----N-6011-----

INT 60 - 3com, 10NET, Banyan VINES - LOCK AND WAIT

AH = 11h

AL = drive number or 0

DX = number of seconds to wait

ES:SI = Ethernet address or 0

DS:BX -> 31-byte ASCIZ semaphore name

Return: AL = status (see #03314)

SeeAlso: AH=12h,AH=13h

(Table 03314)

Values for 3com semaphore status:

00h successful

01h timeout

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 for this CPU

-----N-6012-----

INT 60 - 3com, 10NET, Banyan VINES - LOCK

AH = 12h
AL = drive number or 00h
ES:SI = Ethernet address or 0000h:0000h
DS:BX -> 31-byte ASCIZ semaphore name

Return: AL = status (see also #03314)

01h semaphore currently locked by another PC

Note: unlike function 11h, this function returns immediately

SeeAlso: AH=11h,AH=13h

-----N-6013-----

INT 60 - 3com, 10NET, Banyan VINES - UNLOCK

AH = 13h

AL = drive number or 00h

ES:SI = Ethernet address or 0000h:0000h

DS:BX -> 31-byte ASCIZ semaphore name

Return: AL = status (see also #03314)

01h semaphore not locked

SeeAlso: AH=11h,AH=12h

-----N-6014-----

INT 60 - FTP Packet Driver - EXTENDED FUNC - SET RECEIVE MODE

AH = 14h

BX = handle (optional for v1.10+)

CX = mode (see #03315)

Return: CF set on error

DH = error code (01h,08h) (see #03305)

CF clear if successful

SeeAlso: AH=15h

(Table 03315)

Values for packet driver receive mode:

01h turn off receiver

02h receive only packets sent to this interface

03h mode 2 plus broadcast packets

04h mode 3 plus limited multicast packets

05h mode 3 plus all multicast packets

06h all packets

07h raw mode for serial line only (v1.10+)

-----N-6015-----

INT 60 - FTP Packet Driver - EXTENDED FUNC - GET RECEIVE MODE

AH = 15h

BX = handle (optional for v1.10+)

Return: CF set on error

DH = error code (01h) (see #03305)

CF clear if successful

AX = receive mode (see #03315)

SeeAlso: AH=14h

-----N-6016-----

INT 60 - FTP Packet Driver - EXTENDED FUNC - SET MULTICAST LIST

AH = 16h

ES:DI -> multicast list

CX = length of list in bytes

Return: CF set on error

DH = error code (06h,09h,0Eh) (see #03305)

CF clear if successful

SeeAlso: AH=17h

-----N-6017-----

INT 60 - FTP Packet Driver - EXTENDED FUNC - GET MULTICAST LIST

AH = 17h

Return: CF set on error

DH = error code (06h,09h) (see #03305 at AX=01FFh)

CF clear if successful

ES:DI -> multicast addresses (do not modify)

CX = bytes of multicast addresses currently in use

SeeAlso: AH=16h

-----N-6018-----

INT 60 - FTP Packet Driver - EXTENDED FUNC - GET STATISTICS

AH = 18h

BX = handle (optional for v1.10+)

Return: CF set on error

DH = error code (01h) (see #03305)

CF clear if successful

DS:SI -> statistics (see #03316)

Format of packet driver statistics:

Offset Size Description (Table 03316)

00h DWORD packets in

04h DWORD packets out

08h DWORD bytes in

0Ch DWORD bytes out

10h DWORD errors in

14h DWORD errors out

18h DWORD packets dropped

-----N-6019-----

INT 60 - FTP Packet Driver - EXTENDED FUNC - SET NETWORK ADDRESS

AH = 19h

ES:DI -> address

CX = length of address

Return: CF set on error

DH = error code (0Dh,0Eh) (see #03305)

CF clear if successful

CX = length

-----N-601A-----

INT 60 - FTP Packet Driver v1.10+ - EXTENDED FUNC - SEND RAW BYTES

AH = 1Ah

DS:SI -> buffer

CX = length of buffer

Return: CF set on error

DH = error code (see #03305 at AX=01FFh)

CF clear if successful

SeeAlso: AH=1Ch

-----N-601B-----

INT 60 - FTP Packet Driver v1.10+ - EXTENDED FUNC - FLUSH RAW BYTES RECEIVED

AH = 1Bh

Return: CF set on error

DH = error code (see #03305)

CF clear if successful

SeeAlso: AH=1Ch

-----N-601C-----

INT 60 - FTP Packet Driver v1.10+ - EXTENDED FUNC - FETCH RAW BYTES RECEIVED

AH = 1Ch

DS:SI -> buffer

CX = length of buffer

DX = timeout in clock ticks

Return: CF set on error

DH = error code (see #03305 at AX=01FFh)

CF clear if successful

CX = number of bytes transferred to buffer

SeeAlso: AH=1Ah,AH=1Bh

-----60899B-----

INT 60 U - Agfa TTSR.EXE - API

AX = 899Bh

BX = function

0001h installation check

Return: BX = 899Bh if installed

0002h ???

0003h ???

???

Return: ???

Note: TTSR.EXE is found in FotoLook 2.09 (16-bit), which is available from
agfa.com; it appears to provide DOS ASPI support

!!!

-----a-60AD-----

INT 60 - AccessDOS - API

AH = ADh

AL = function

E1h ???

Return: AX = ???

E2h get configuration

Return: BX:AX -> configuration data

Program: AccessDOS is a public domain TSR developed at The Trace Research and
Development Center which provides extensions for keyboard, mouse,
and sound access by the visually, hearing, or motor-control
impaired.

InstallCheck: test for the signature "ACCESSv1.00" beginning three bytes
past the start of the interrupt handler (this same signature is also
present in v1.10)

Range: INT 60 to INT 66, selected by scanning for 0000h:0000h vector

-----N-60E9-----

INT 60 - FTP Packet Driver - Crynwr Software - AUTOSELECT TRANSCEIVER

AH = E9h

???

Return: ???

-----*-61-----

INT 61 - reserved for user interrupt

-----b-61-----

INT 61 - Atari Portfolio - EXTENDED BIOS

Desc: provides subfunctions such as turning off the machine, accessing
internal variables, and mapping memory cards

SeeAlso: INT 60"Atari",AH=00h"Portfolio",AH=08h"Portfolio",AH=0Dh"Portfolio"

SeeAlso: AH=11h"Portfolio",AH=15h"Portfolio",AH=19h"Portfolio"

SeeAlso: AH=1Bh"Portfolio",AH=24h"Portfolio",AH=2Ch"Portfolio"

SeeAlso: AH=30h"Portfolio"

-----b-61-----

INT 61 - TI Professional PC - FACTORY ROM DATA AREA POINTER (NOT A VECTOR!)

Desc: the low word of this vector contains the segment of the RAM data area

to be used by the expansion ROM at F400h:0000h, and the high word

contains the length of the data area; this segment and size are

both set to 0000h if no ROM is installed at F400h:0000h

SeeAlso: INT 60"TI Professional PC",INT 62"TI Professional"

SeeAlso: INT 66"TI Professional PC"

-----b-61-----

INT 61 - HP 95LX System Manager - LOAD DS

SeeAlso: INT 0F"HP 95LX",INT 60/DI=0102h"HP 95LX"

-----r-61-----

INT 61 - JPI TopSPEED Modula-2 v1 - PROCEDURE EXIT TRAP

SeeAlso: INT 60"JPI"

-----d-61-----

INT 61 - Adaptec and OMTI controllers - DRIVE 0 DATA

Desc: this vector stores the second four bytes of the parameter table for
hard disk 0

SeeAlso: INT 60"Adaptec",INT 62"Adaptec",INT 63"Adaptec"

-----I-61-----

INT 61 - Sangoma CCIP (CCPOP 3270 resident module) INTERFACE

BX:DX -> control block

SeeAlso: INT 67"Sangoma"

-----v-61-----

INT 61 - VIRUS - "SEMTEX"/"Screen Trasher" - INT 21h SUBSTITUTE

Note: the virus copies the original INT 21h vector into INT 61h

SeeAlso: INT 21/AH=00h,INT 21/AH=25h,INT 60"VIRUS",INT 6B"VIRUS"

-----U-61-----

INT 61 - PC-Magazin - PAGE, SWAPx

details not yet available

Program: PAGE and SWAPx are utilities from PC-Magazin, the German edition of
PC Magazine, issue 47/85, page 31

SeeAlso: INT 66"INCA"

-----N-61-----

INT 61 - TCPOpen kernel - API

ES:BX -> request packet (see #03317)

Program: TCPOpen is a TCP/IP protocol stack by Lanera

Range: INT 60 to INT 7F, selected by configuration file

Note: there does not appear to be an installation check, though function
FFFFh can be used to verify that the software is indeed functioning

Format of TCPOpen request packet:

Offset Size Description (Table 03317)

00h 8 BYTES ???

08h WORD function number (0000h-001Ah or FFF9h-FFFFh)
0Ah WORD ???
0Ch 2 BYTES ???
0Eh WORD (ret) return code
10h 2 BYTES ???
12h WORD ???
14h WORD (ret) 0001h if requested function complete
16h WORD (ret) ???
18h 6 BYTES ???
1Eh DWORD -> ???
22h WORD ??? or 0000h
24h WORD (ret) ???

SeeAlso: #03318

(Table 03318)

Call TCPOpen Function FFFFh: installation verification

Return: offset 10h = BEEFh

offset 14h = 0001h

offset 24h = ???

Index: installation check;TCPOpen kernel

-----N-61-----

INT 61 - XFS v1.76 - XFSKRNL.EXE - API

AX = function

8000h installation check???

Return: AX = 0000h

8001h ???

ES:DI -> DWORD ???

Return: AX = status???

8002h ???

BX = ???

CX = ???

ES:DI -> ???

Return: AX = status???

8003h ???

BX = ???

CX = ???

ES:DI -> ???

Return: AX = status???

8004h ???

BX = ???

Return: AX = status???

```
8005h wait until ???
Return: DX:AX = ???
8006h get ???
Return: DX:AX -> ??? record
8007h set IP address??? (or htonl/ntohl)
DX:BX = IP address???
Return: DX:AX = byte-swapped IP address
8008h set IP address??? (or htonl/ntohl)
DX:AX = IP address???
Return: DX:AX = byte-swapped IP address
8009h ???
ES:DI -> ???
Return: DX:AX = ???
800Ah ???
800Bh ???
Return: DX:AX = ???
800Ch ???
800Dh ???
800Eh ???
800Fh ???
8010h get ??? record
Return: DX:AX -> ??? record
8011h ???
8012h ???
BX = ???
Return: DX:AX = ???
8013h get ??? record
Return: DX:AX -> ??? record
8014h ???
ES:DI -> ???
Return: nothing
8015h get ??? record
Return: DX:AX -> 8-byte ??? record
8016h get ??? record
Return: DX:AX -> 6-byte ??? record
8017h ???
ES:DI -> ???
Return: DX:AX = ???
8018h get ??? record
Return: DX:AX -> ??? record
8019h get ??? record
```

Return: DX:AX -> 14-byte ??? record
801Ah get ??? record
Return: DX:AX -> ??? record (max 98 bytes)
801Bh get ??? record
Return: DX:AX -> ??? record
801Ch ???
Return: nothing
801Dh get ???
Return: AL = ???
801Eh ???
Return: AX = ???
801Fh uninstall
Return: AX = status
0000h successful
other interrupt which could not be released

AX = other

Return: AX:BX = ??? (0000h:0000h by default)

Program: XFS is a shareware Network File System client by Robert Juhasz
Range: INT 60 to INT 65, selected by scanning for two consecutive free
vectors and hooking the first

SeeAlso: INT 62"XFS"

-----N-61-----

INT 61 - FTP Software PC/TCP - TCP/IP TSR System Call interface

AH = system call number (also see entries below)

01h "pkt_alloc" (v2.05; this is "net_config" in v2.1+)

02h "pkt_free" (v2.05; this is "get_kernel_info" in v2.1+)

Return: CF clear if successful

CF set on error

AL = basic error (see #03319)

AH = suberror number (see #03320,#03321)

InstallCheck: test for the signature "TCPTSR" three bytes beyond the start
of the interrupt handler

Range: INT 20h to INT E0h, selected by configuration

BUG: the SLIP kernel for v2.05 bounds-checks the wrong register, so values
greater than 54h in AH may crash the system. Other kernels may have
this bug as well.

SeeAlso: INT 61/AH=00h"PC/TCP", INT 61/AH=2Ah, INT 61/AH=54h

Index: installation check;PC/TCP

(Table 03319)

Values for PC/TCP error code:

00h "NET_NOERR" successful
01h "NET_ERR_INUSE" protocol or socket already in use
02h "NET_DOS_ERR" MS-DOS error (returned as suberror code in AH)
03h "NET_ERR_NOMEM" out of memory
04h "NET_ERR_NOTNETCONN" not a network descriptor
05h "NET_ERR_ILLEGALOP" invalid operation on given kind of network descriptor
06h "NET_ERR_BADPKT" illegal or corrupted packet
07h "NET_ERR_NOHOST" no host bound to specified connection
08h "NET_ERR_CANTOPEN" unable to open file
09h "NET_ERR_NET_UNREACHABLE" network is unreachable
0Ah "NET_ERR_HOST_UNREACHABLE" host is unreachable (see #03320)
0Bh "NET_ERR_PROT_UNREACHABLE" protocol is unreachable
0Ch "NET_ERR_PORT_UNREACHABLE" port is unreachable
0Dh "NET_ERR_TIMEOUT" operation timed out
0Eh "NET_ERR_HOSTUNKNOWN" unable to resolve host name
0Fh "NET_ERR_NOSERVERS" no name servers configured
10h "NET_ERR_SERVER_ERR" bad reply from name server
Subcodes: 0= no error, 1 = Host unreachable
11h "NET_ERR_BADFORMAT" bad format for IP address or field in IP address
structure is zero
12h "NET_ERR_BADARG" invalid argument
13h "NET_ERR_EOF" foreign host closed its end of connection
14h "NET_ERR_RESET" connection has been reset
15h "NET_ERR_WOULDBLOCK" recv() call was done on a non-blocking connection
with no data available
16h "NET_ERR_UNBOUND" insufficient resources to do operation
17h "NET_ERR_NODESC" could not allocate network descriptor
18h "NET_ERR_BADSYSCALL" invalid/unsupported kernel call
19h "NET_ERR_CANTBROADCAST" unable to broadcast
1Ah "NET_ERR_NOTESTAB" operation illegal because connection not established
1Bh kernel busy, try again later
1Ch "NET_ERR_ICMPMSG" an ICMP message was received (not on streams)
(see #03321)
---v2.1+---
1Dh "NET_ERR_TERMINATING" internal kernel fatal error
1Eh "NET_ERR_TAG_LOCKED" not allowed to set this tag (net_config)
1Fh "NET_ERR_BAD_INTERFACE" non existent interface specified
20h "NET_ERR_BADCONFIG" kernel cannot run - bad configuration
21h "NET_ERR_EMM" expanded memory error
22h "NET_ERR_CANT_SHUTDOWN" cant unload kernel (multitasker running)
23h "NET_ERR_PARKED_IN" unable to unhook DOS interrupt

24h "NET_ERR_NOQIOS" ran out of resources; try again later
25h "NET_ERR_WOULD_TRUNCATE" datagram too large and "don't truncate" was set

(Table 03320)

Values for subcodes of PC/TCP error 0Ah "NET_ERR_HOST_UNREACHABLE":

00h no error
01h host unreachable
02h ARP failed
03h hardware failure
04h link failure
05h no route
06h gateway down

SeeAlso: #03319,#03321

(Table 03321)

Values for subcodes of PC/TCP error 1Ch "NET_ERR_ICMPMSG":

07h unrecognised
08h can't fragment
09h srcr_fail
0Ah source quench
0Bh time exceeded
0Ch parameter problem
0Dh admin_prohib. see also code 0Ah

SeeAlso: #03319,#03320

-----N-6100-----

INT 61 U - PC/TCP kernel v2.05+ - GET DEBUG INFORMATION

AH = 00h

DS:SI -> 216-byte buffer for network debugging information (see #03322)

Return: CF clear

AX = 0000h

buffer filled

Notes: this call is not documented by FTP, Inc. for any version

most of the information returned by this call is available via the
documented get_kernel_info or net_info commands.

this function is not supported by Beame&Whiteside's BWPCTCP v3.0a shim

SeeAlso: INT 61"FTP Software",INT 61/AH=2Ah"PC/TCP"

Format of network debugging information:

Offset Size Description (Table 03322)

00h DWORD number of interrupts

04h DWORD receive buffer low-water mark


```

08h  DWORD transmit buffer low-water mark
0Ch  DWORD number of packets received
10h  DWORD number of packets transmitted
14h  DWORD total receive errors
18h  DWORD total transmit errors
1Ch  4 BYTES ???
20h  DWORD receive resets
24h  DWORD transmit resets
28h  DWORD number of "runts" received
2Ch  DWORD number of alignment errors on received packets
30h  DWORD number of CRC errors on received packets
34h  DWORD number of parity errors on received packets
38h  DWORD number of receive overflow errors
3Ch  DWORD number of oversized packets received
40h  DWORD number of packets lost due to lack of buffers
44h  DWORD receive timeouts
48h  32 BYTES ???
68h  DWORD number of transmit collisions
6Ch  DWORD number of transmit timeouts
70h  DWORD number of transmit underflows
74h  DWORD number of lost "crs" on transmit
78h  DWORD number of times heartbeat failed on transmit
7Ch  24 BYTES ???
94h  WORD  free packet buffers
96h  WORD  total packet buffers
98h  WORD  minimum number of packet buffers free since kernel started
9Ah  24 BYTES ???
B2h  DWORD pointer to TCP connection list???
B6h  DWORD pointer to IP routing table???
BAh  30 BYTES ???

```

```
-----N-6100-----
```

```
INT 61 - Banyan VINES - CALL A SERVICE
```

```
  AH = 00h
```

```
  AL = service ID (1..16)
```

```
  other registers vary by service
```

```
Return: AX = status (see #03325)
```

```
Desc: Calls an adapter driver or redirector service. The service handler
pointers are stored at offset 0010h in the code segment of the
BANV interrupt handler (see #03323).
```

```
Note: Banyan can use any interrupt from 60h through 66h. The Banyan
interrupt handler is identified by the string "BANV" in the four
```

bytes immediately preceding the interrupt handler
SeeAlso: AH=01h,AH=03h

Format of the BANV code segment (revision 5.53):

Offset	Size	Description (Table 03323)
00h	16 BYTES	reserved
10h	16 DWORDs	far pointers to the service handlers (0 = not installed)
50h	DWORD	far pointer to last called service
54h	WORD	VINES revision
56h	DWORD	VINES revision long
5Ah	WORD	VINES revision (lower bound)
5Ch	DWORD	VINES revision (lower bound) long
60h	WORD	VINES revision (upper bound)
62h	DWORD	VINES revision (upper bound) long
66h	WORD	end of PCCONFIG area
68h	WORD	start of PCCONFIG area
6Ah	4 BYTES	BANV interrupt identifier ("BANV")
6Eh	N BYTES	BANV interrupt service

SeeAlso: #03324

-----b-6100-----

INT 61 - Atari Portfolio - SERVICE INITIALIZATION

AH = 00h

Return: nothing

Note: applications using any Portfolio INT 61 calls should call this function exactly once at startup

SeeAlso: AH=07h"Portfolio",AH=09h"Portfolio",AH=0Dh"Portfolio"

SeeAlso: AH=11h"Portfolio",AH=15h"Portfolio",AH=19h"Portfolio"

SeeAlso: AH=1Bh"Portfolio",AH=24h"Portfolio",AH=2Ch"Portfolio"

SeeAlso: AH=30h"Portfolio"

-----V-610000-----

INT 61 - OPTIMA 1024 VGA-Sync,ET-3000 chipset - ZOOM DISPLAY

AX = 0000h

BX = zoom factor (0-7)

Desc: zooms the display based on the given zoom factor

Notes: INT 61h is the default interrupt; the actual interrupt number can be obtained by calling INT 16/AH=FFh

not all vendors include the Tseng TSR which supports these functions

SeeAlso: AX=0001h"OPTIMA",AX=0002h"OPTIMA",AX=0005h"OPTIMA"

SeeAlso: INT 16/AH=FFh"OPTIMA"

-----N-610001DS0001-----

INT 61 - Banyan VINES - INTERNAL VINES SOCKET API (NEAR DATA)

AX = 0001h
 DS = 0001h (to distinguish from alternate AX=0001h socket interface)
 CX:BX -> internal communications control block (see #03324)
 DX = caller's data segment

Return: AX = status

0000h successful
 0099h invalid operation
 00A5h resource not available

Desc: StreetTalk access with session handles

Range: INT 60 to INT 66

Notes: This service is handled by the adapter driver

The request dispatch table contains dummy entries for subfunctions
 06h, 0Dh, 0Eh, 0Fh, 10h, and 11h, which always return error 0099h
 (invalid operation) in the request block's result buffer
 in recent VINES releases, this interface is superseded by the AX=000Fh
 interface, which allows FAR parameter block pointers (refer to that
 function's entries for more details on individual subfunctions)
 some adapter drivers (e.g. the NDIS2 token ring driver NDTOKBAN for
 VINES 5.53) ignore the value in CX and use the value from DX instead

SeeAlso: AX=000Fh

Format of internal communications control block:

Offset Size Description (Table 03324)

00h	WORD	subfunction number
02h	WORD	magic number (FEFEh)
04h	WORD	(ret) result, should be set to 0 before calling
06h	WORD	reserved (will be set to client DS)
08h	WORD	caller's code segment if function 000Bh, 0 else
0Ah	WORD	pointer to parameter block
0Ch	WORD	reserved (segment of parameter block; will be set to client DS)
0Eh	WORD	StreetTalk handle

SeeAlso: #03323, #03444 at INT 61/AX=000Fh

-----N-610001SF0001-----

INT 61 - Banyan VINES - "Sosock" - OPEN COMMUNICATIONS SOCKET

AX = 0001h subfn 0001h

DS:DX -> communications control block (function 0001h) (see #03326)

Return: AX = status (0000h-0002h, 0098h, 009Eh-00A2h) (see #03325)

Range: INT 60h to INT 66h

Note: The Banyan interrupt handler is identified by the string "BANV" in
 the four bytes immediately preceding the interrupt handler

SeeAlso: AX=0001h/SF=0002h, AX=0001h/SF=0008h, INT 15/AX=DE2Eh

(Table 03325)

Values for VINES status:

0000h successful
0001h service not installed
0002h invalid service ID
0003h invalid function
0004h-000Ah reserved for BANV interface errors
0097h invalid socket identifier
0098h resource already in use
009Bh destination node unreachable
009Ch message overflow
009Dh destination socket nonexistent
009Eh address family does not exist
009Fh socket type does not exist
00A0h protocol does not exist
00A1h no more sockets available
00A2h no more buffer space available
00A3h timeout
00A5h resource not available
00A6h internal communication failure
00B1h resource disconnect

Format of VINES control block to open socket:

Offset Size Description (Table 03326)

00h WORD 0001h
02h WORD pointer to argument block
04h WORD error return code
06h 4 BYTES reserved

SeeAlso: #03327

Format of VINES argument block for opening socket:

Offset Size Description (Table 03327)

00h WORD pointer to 2-byte buffer for socket identifier
02h WORD address family
 0003h Banyan
04h WORD socket type
 in address family 0003h
 0001h IPC socket
 0002h SPP socket
06h WORD protocol number

FFFFh default

08h WORD pointer to 16-byte buffer for socket address (see #03328)

0Ah WORD local port number

0000h if service should assign transient port number

0001h to 01FFh well-known port number (assigned by Banyan)

SeeAlso: #03326

Format of Banyan VINES IPC port:

Offset Size Description (Table 03328)

00h WORD address family (always 0003h for Banyan ports)

04h 4 BYTES network number (server's serial number)

06h WORD subnet number (0001h = server, 8000h-FFFEh = PC)

08h WORD port ID (0001h-01FFh for "well-known" ports)

0Ah BYTE hop count

0Bh 5 BYTES filler

SeeAlso: #03333

-----N-610001SF0002-----

INT 61 - Banyan VINES - "Sosend" - INITIATE OUTPUT EVENT

AX = 0001h subfn 0002h

DS:DX -> communications control block (function 0002h) (see #03329)

Return: AX = status (0000h-0002h,0097h,009Bh-009Dh,00A2h,00A3h,00B1h)

(see #03325)

Range: INT 60h to INT 66h

Note: The Banyan interrupt handler is identified by the string "BANV" in the four bytes immediately preceding the interrupt handler

SeeAlso: AX=0001h/SF=0001h,AX=0001h/SF=0005h

Format of VINES control block for subfunction 0002h:

Offset Size Description (Table 03329)

00h WORD 0002h

02h WORD pointer to argument block (see #03330)

04h WORD error return code (0000h,0097h,00A2h,00A3h,00A5h,00A6h,00B1h)

(see #03325)

06h 4 BYTES reserved

SeeAlso: #03334

Format of VINES argument block for subfunction 0002h:

Offset Size Description (Table 03330)

00h WORD routine metric

02h WORD error return code

04h WORD socket identifier

06h WORD pointer to send buffer (see #03332)
 08h WORD length of send buffer
 0Ah WORD message request flags (see #03331)
 0Ch 16 BYTES socket address (see #03333)
 1Ch WORD timeout value in multiples of 200ms
 1Eh WORD connection identifier
 20h WORD type of request
 0001h send message
 0002h establish a virtual connection
 0003h terminate a virtual connection
 SeeAlso: #03329,#03335

Bitfields for message request flags:

Bit(s) Description (Table 03331)

0 async request
 1 reliable message
 2 flush receive buffer on overflow
 3 end of user message received
 4 vectored request (if set, send buffer contains buffer descriptors)
 5 connection-specific receive
 6 change to connection-specific receive mode

SeeAlso: #03330,#03335

Format of buffer descriptor:

Offset Size Description (Table 03332)

00h WORD data segment
 02h WORD buffer pointer
 04h WORD buffer length
 06h WORD character count

SeeAlso: #03330

Format of VINES socket address for unreliable datagrams:

Offset Size Description (Table 03333)

00h WORD 0003h address family
 02h DWORD FFFFFFFFh network number
 06h WORD FFFFh subnet number
 08h WORD local port number
 0Ah BYTE 00h-0Fh hop count
 0Bh 5 BYTES 0000h filler

SeeAlso: #03328

-----N-610001SF0003-----

INT 61 - Banyan VINES - "Sorec" - RECEIVE INPUT EVENT NOTIFICATION

AX = 0001h subfn 0003h

DS:DX -> communications control block (function 0003h) (see #03334)

Return: AX = status (00-0Ah,97h,A2h,A3h) (see #03325)

Range: INT 60h to INT 66h

Note: The Banyan interrupt handler is identified by the string "BANV" in the four bytes immediately preceding the interrupt handler

SeeAlso: AX=0001h/SF=0002h

Format of VINES control block for subfunction 0003h:

Offset Size Description (Table 03334)

00h WORD 0003h

02h WORD pointer to argument block (see #03335)

04h WORD error return code (0000h,0097h,00A2h,00A3h,00A5h,00A6h,00B1h)
(see #03325)

06h 4 BYTES reserved

SeeAlso: #03329,#03335,#03337

Format of VINES argument block for subfunction 0003h:

Offset Size Description (Table 03335)

00h WORD character count

02h WORD error return code

04h WORD socket identifier

06h WORD pointer to receive buffer (see #03336)

08h WORD length of receive buffer

0Ah WORD message request flags (see #03331)

0Ch 16 BYTES socket address (see #03328)

1Ch WORD timeout value in multiples of 200ms

1Eh WORD connection identifier

20h WORD type of response

0001h message received

0002h virtual connection established

0003h virtual connection terminated

SeeAlso: #03334,#03338

Format of buffer descriptor:

Offset Size Description (Table 03336)

00h WORD data segment

02h WORD buffer pointer

04h WORD buffer length

06h WORD character count

SeeAlso: #03335

-----N-610001SF0004-----

INT 61 - Banyan VINES - "Soclose" - CLOSE A SOCKET

AX = 0001h subfn 0004h

DS:DX -> communications control block (function 0004h) (see #03337)

Return: AX = status (0000h-000Ah,0097h) (see #03325)

Range: INT 60h to INT 66h

Note: The Banyan interrupt handler is identified by the string "BANV" in
the four bytes immediately preceding the interrupt handler

SeeAlso: AX=0001h/SF=0001h,INT 15/AX=DE2Eh

Format of VINES control block to close socket:

Offset Size Description (Table 03337)

00h WORD 0004h

02h WORD pointer to argument block (see #03338)

04h WORD error return code (see #03325)

06h 4 BYTES reserved

SeeAlso: #03334,#03340

Format of VINES argument block for closing socket:

Offset Size Description (Table 03338)

00h WORD socket identifier

SeeAlso: #03335,#03337,#03341

-----N-610001SF0005-----

INT 61 - Banyan VINES - "Sowait" - WAIT FOR ASYNCHRONOUS EVENT COMPLETION

AX = 0001h subfn 0005h

DS:DX -> communications control block (see #03340)

Return: AX = status (0000h-000Ah,00A2h,00A3h) (see #03339)

Desc: returns results for all asynchronous operations invoked from the
data segment used for this call

Range: INT 60h to INT 66h

Notes: The Banyan interrupt handler is identified by the string "BANV" in
the four bytes immediately preceding the interrupt handler

SeeAlso: AX=0001h/SF=0002h,AX=0001h/SF=0009h

(Table 03339)

Values for VINES function status:

0000h successful

0001h service not installed

0002h invalid service ID

0003h-000Ah reserved for BANV interface errors

00A2h no more buffer space available
00A3h timeout event

Format of VINES control block for subfunction 0005h:

Offset Size Description (Table 03340)

00h WORD 0005h
02h WORD pointer to argument block (see #03341)
04h WORD error return code
06h 4 BYTES reserved

SeeAlso: #03337,#03342

Format of VINES argument block for subfunction 0005h:

Offset Size Description (Table 03341)

00h WORD pointer to WORD event pointer
02h WORD timeout in multiples of 200ms, FFFFh = infinite

SeeAlso: #03338,#03340

-----N-610001SF0008-----

INT 61 - Banyan VINES - "Sosession" - REGISTER APPLICATION WITH COMM SERVICE

AX = 0001h subfn 0008h

DS:DX -> communications control block (function 0008h) (see #03342)

Return: AX = status

0000h successful
00A2h no more buffer space available

Range: INT 60h to INT 66h

Note: The Banyan interrupt handler is identified by the string "BANV" in
the four bytes immediately preceding the interrupt handler

SeeAlso: AX=0001h/SF=0001h,AX=0001h/SF=0009h

Format of VINES control block for subfunction 0006h:

Offset Size Description (Table 03342)

00h WORD 0008h
02h WORD process type
0001h transient process
0002h resident process
04h WORD error return code
06h 4 BYTES reserved

SeeAlso: #03340,#03343

-----N-610001SF000B-----

INT 61 - Banyan VINES - "Soint" - SET USER COMPLETION FUNCTION

AX = 0001h subfn 000Bh

DS:DX -> communications control block (function 000Bh) (see #03343)

Return: AX = status (00h-0Ah,A2h) (see #03339)

Range: INT 60h to INT 66h

Note: The Banyan interrupt handler is identified by the string "BANV" in the four bytes immediately preceding the interrupt handler

SeeAlso: AX=0001h/SF=0005h,AX=0001h/SF=0008h

Format of VINES control block for subfunction 000Bh:

Offset Size Description (Table 03343)

00h WORD 000Bh

02h WORD pointer to argument block (see #03344)

04h WORD error return code (see #03325)

06h 2 BYTES reserved

08h WORD user CS register

SeeAlso: #03342

Format of VINES argument block for subfunction 000Bh:

Offset Size Description (Table 03344)

00h WORD pointer to user interrupt function (see #03345)

02h WORD pointer to user stack

04h WORD initial timeout value in multiples of 200ms, FFFFh = infinite

SeeAlso: #03344

(Table 03345)

Values FAR user function is invoked with:

SS = DS = ES = segment of control block

STACK: DWORD return address

WORD argument pointer (sosend or sorec argument block)

WORD error return code

0000h argument pointer is valid

00A3h timeout

-----V-610001-----

INT 61 - OPTIMA 1024 VGA-Sync,ET-3000 chipset - CENTER ZOOM WINDOW

AX = 0001h

BX = X coordinate to center

CX = Y coordinate to center

Desc: Positions the zoom window such that the specified window-relative coordinates appear as close as possible to the center of the display. Useful for scrolling and panning.

Range: unknown

Notes: INT 61h is the default interrupt; the actual interrupt number can be obtained by calling INT 16/AH=FFh

not all vendors include the Tseng TSR which supports these functions

SeeAlso: AX=0000h"OPTIMA",AX=0002h"OPTIMA",AX=0005h"OPTIMA"

SeeAlso: INT 16/AH=FFh"OPTIMA"

-----I-610002-----

INT 61 - Banyan VINES - 3270 INTERFACE

AX = 0002h

BH = function number (see #03346)

DS:CX -> argument block (except BH=00h,1Ah) (see #03348)

Return: AX = status (see #03347)

Notes: Either 3270/SNA or 3270/BSC interface may use AX=0002h, depending on which is loaded first. The other interface will use AX=000Ah

Status codes greater than 63h indicate an inconsistency in the 3270/SNA or 3270/BSC resident driver, which must be reloaded by the user

(Table 03346)

Values for VINES 3270 interface function number:

00h "pi2reset" reset 3270/SNA or 3270/BSC driver
 02h "pi2bsc" (3270/BSC only)
 03h "pi2get" get information stored in 3270 resident driver
 04h "pi2put" store information in 3270 resident driver
 05h "pi2gcur" get current screen position
 07h "pi2sdat" send data keystroke
 08h "pi2scom" send command keystroke
 0Ah "pi2field" get field info for arbitrary screen positions
 0Fh "pi2stat" get logical unit/device status
 12h "pi2nlus" determine logical unit/device assignment
 13h "pi2gate" specifies comm port address to gateway service
 14h "pi2attach" attach a logical unit/device
 15h "pi2sdev" save logical unit/device info in resident driver
 (not supported in >3.0)
 16h "pi2gdev" get device information (not supported in >3.0)
 17h "pi2luinfo" get info about specific logical unit/device
 18h "pi2gerr" get finer error detail
 19h "pi2dhold" (3270/SNA only) holds a 3270 device
 1Ah "pi2shut" release memory-resident module
 1Ch "pi2sprof" save profile info in res driver (not supp in >3.0)
 1Dh "pi2gprof" get prevsly stored profile info (not supp in >3.0)

(Table 03347)

Values for VINES 3270 status:

0000h successful

000Bh invalid parameter or data does not fit data area
000Ch another code path currently active in resident driver
000Dh operation currently not allowed
0032h encountered connection disconnect error
0033h encountered "sosend" completion error
0034h encountered "sosend" communication error
0035h attach request refused. extended error info via "pi2gerr":
 01h resource unavailable
 02h invalid type
 03h version mismatch
 04h invalid logical unit number
 05h error during ARL processing
 06h no access for user
0071h encountered "sosock" error
0072h encountered unrecognizable error
0073h encountered "sowait" error (extended info via "pi2gerr")
0074h encountered invalid type-of-request on "sowait"
0075h encountered "sorec" error (extended info via "pi2gerr")
0076h encountered "sorec" completion error (ext info via "pi2gerr")
0077h encountered connection request
0078h encountered unrecognizable data
0079h encountered unknown connection ID (ext info via "pi2gerr")

SeeAlso: #03325

Format of argument block for BH=03h,04h:

Offset	Size	Description (Table 03348)
00h	WORD	size of data area (max 256)
02h	N BYTES	data area

SeeAlso: #03349,#03350,#03351

Format of argument block for BH=05h:

Offset	Size	Description (Table 03349)
00h	WORD	logical unit/device number
02h	WORD	pointer to WORD buffer for cursor index
04h	WORD	pointer to BYTE buffer for current field attribute

SeeAlso: #03348,#03350,#03351

Format of argument block for BH=07h:

Offset	Size	Description (Table 03350)
00h	WORD	logical unit/device number
02h	WORD	ASCII data byte

04h WORD pointer to WORD count of characters which will need updating

SeeAlso: #03348,#03349,#03351

Format of argument block for BH=08h:

Offset Size Description (Table 03351)

00h WORD logical unit/device number

02h WORD 3270 keystroke (see #03352)

SeeAlso: #03348,#03349,#03350,#03353

(Table 03352)

Values for 3270 keystroke:

0000h Enter

0001h Clear

0002h PA1

0003h PA2

0004h PA3

0005h PF1

...

001Ch PF24

001Dh CSELECT (cursor select)

001Eh Insert

001Fh Delete

0020h EOFfield

0021h EINPUT (erase input)

0022h Reset

0023h Attention

0024h SysReq

0025h Duplicate

0026h Fieldmark

0027h Home

0028h NextLine

0029h Tab

002Ah BackTab

002Bh cursor up

002Ch cursor down

002Dh cursor right

002Eh cursor left

002Fh double cursor right

0030h double cursor left

0031h PRINT

0032h CANCEL

0033h Backspace

Format of argument block for BH=0Ah:

Offset	Size	Description (Table 03353)
00h	WORD	logical unit/device number
02h	WORD	screen index
04h	WORD	pointer to WORD buffer for field length
06h	WORD	pointer to WORD buffer for offset in screen of field start

SeeAlso: #03351,#03354

Format of argument block for BH=0Fh:

Offset	Size	Description (Table 03354)
00h	WORD	logical unit/device number
02h	WORD	clear mask (clear these bits of status after returning status)
04h	WORD	pointer to WORD buffer for device status (see #03355)

SeeAlso: #03353,#03356

Bitfields for Banyan 3270 device status:

Bit(s)	Description (Table 03355)
10	status modified
9	buffer modified
8	set cursor
5	sound alarm
1-0	size of print line for printer logical units
00	unformatted line
01	40-character line
10	64-character line
11	80-character line

Format of argument block for BH=12h:

Offset	Size	Description (Table 03356)
00h	WORD	pointer to WORD buffer for number of logical units or devices
02h	WORD	pointer to WORD buffer for version number
04h	WORD	pointer to 64-byte buffer for logical unit/device list

SeeAlso: #03354,#03357

Format of argument block for BH=13h:

Offset	Size	Description (Table 03357)
00h	16 BYTES	communications port address (see #03328)

SeeAlso: #03356,#03358

Format of argument block for BH=14h:

Offset Size Description (Table 03358)

00h WORD logical unit/device number
0000h attach any free device of the specified type
02h WORD logical unit/device type
(3270/SNA) 01h, 02h, or 03h
(3270/BSC) 02h display
(3270/BSC) 03h printer

04h WORD pointer to WORD buffer for attached logical unit/device number

SeeAlso: #03357,#03359

Format of argument block for BH=16h:

Offset Size Description (Table 03359)

00h WORD pointer to 18-byte buffer for device block (see #03364)
first WORD must be set to desired logical unit/device number

SeeAlso: #03358,#03360

Format of argument block for BH=17h:

Offset Size Description (Table 03360)

00h WORD logical unit/device number
02h WORD pointer to information block in caller's DS (see #03365)

SeeAlso: #03359,#03361

Format of argument block for BH=18h:

Offset Size Description (Table 03361)

00h WORD pointer to WORD buffer for major error code
02h WORD pointer to WORD buffer for minor error code

SeeAlso: #03360,#03362

Format of argument block for BH=19h:

Offset Size Description (Table 03362)

00h WORD logical unit/device number

SeeAlso: #03361,#03363

Format of argument block for BH=1Ch,1Dh:

Offset Size Description (Table 03363)

00h WORD pointer to profile block in caller's DS (see #03368)

SeeAlso: #03348,#03362

Format of device block, argument block for BH=15h:

Offset Size Description (Table 03364)

00h WORD logical unit/device number
02h WORD logical unit/device type
04h WORD display model number
06h WORD numeric checking
08h WORD status line
0Ah BYTE unprotected normal field attribute
0Bh BYTE unprotected intensified field attribute
0Ch BYTE protected normal field attribute
0Dh BYTE protected intensified field attribute
0Eh WORD reserved
10h WORD printer port number

Format of information block:

Offset Size Description (Table 03365)

00h WORD device model number
02h DWORD screen buffer pointer
06h DWORD status line pointer (see #03366)
0Ah DWORD reserved

Format of Banyan VINES 3270 status line:

Offset Size Description (Table 03366)

00h BYTE comm line status
 00h inactive
 01h active
01h BYTE activation level
 01h physical unit activated
 02h logical unit also activated
 03h session is bound
02h BYTE data traffic state
 00h inactive
 01h active
03h BYTE screen ownership
 00h SLU->PLU session owns screen
 01h SLU->SSCP session owns screen
04h BYTE keyboard status (see #03367)
05h BYTE insert mode
 01h if in insert mode
06h BYTE numeric
 01h if current screen buffer is numeric only
07h BYTE printer status
 00h printer not assigned


```

01h printer is inactive
02h printer error
03h currently printing
04h printer is busy
05h printer is very busy
08h BYTE printer assignment
09h BYTE maximum size of network name
0Ah N BYTES ASCIZ network name
BYTE maximum size of message window
    M BYTES null-terminated message window
BYTE code set
    00h EBCDIC
    01h ASCII
    M BYTES extended attributes
01h extended attributes are in effect (stored at screen+1920)
    each extended attribute specifies
    bits 0,1: 00=normal, 01=blink, 10=reverse, 11=underscor
    bits 2-4: 000=default,001=blue,010=red,011=pink,
        100=green,101=turquoise,110=yellow,111=white
BYTE extended color
    01h other than base color is in effect

```

(Table 03367)

Values for Banyan VINES 3270 keyboard status:

```

00h UNLOCK - ready to accept data
01h TIME - aid was struck
02h SYSTEM - received response no restore
03h FUNCTION - unavailable keyboard function
04h INPUT - not currently used
05h ENDFIELD - field filled in insert mode
06h PROTECTED - attempt to enter in protected field
07h NUMERIC - attempt to enter in numeric field
08h PROGRAM - error in outbound data stream

```

Format of Banyan VINES 3270 profile block:

Offset Size Description (Table 03368)

```

00h 64 BYTES gateway service name
40h 16 BYTES gateway comm port address
50h WORD primary logical unit number
52h WORD secondary logical unit type
54h WORD secondary logical unit number

```

56h WORD printer assignment
58h 50 BYTES keyboard definitions filename
-----V-610002-----
INT 61 - OPTIMA 1024 VGA-Sync,ET-3000 chipset - END ZOOM
AX = 0002h
Desc: switches off zoom and returns window to its original state
Range: unknown
Notes: INT 61h is the default interrupt; the actual interrupt number can be
obtained by calling INT 16/AH=FFh
not all vendors include the Tseng TSR which supports these functions
SeeAlso: AX=0000h"OPTIMA",INT 16/AH=FFh"OPTIMA"
-----N-610003SF00-----
INT 61 - Banyan VINES - ASYNC TERMINAL EMULATION - INIT USER BUFFER PTR INFO
AX = 0003h subfn 00h
DS:BX -> argument block (see #03370)
Return: AX = status (see #03369)
SeeAlso: AX=0003h/SF=06h,AX=0003h/SF=08h,AX=0003h/SF=0Dh

(Table 03369)

Values for VINES function status:

0000h successful
000Bh invalid session ID
000Ch session not active
000Dh invalid request type
000Eh invalid parameters
000Fh out of heap space
0010h timeout on send
0011h Banyan communications error
0012h session not waiting for host
0013h session is active
0014h duplicate suspend session request
0015h no session suspended
0016h ring data buffer full
0017h printer error encountered
0018h Banyan communications error
0019h unable to make connection
001Ah no ring buffer specified at startup
001Bh service is down
001Ch invalid service name
001Dh service is closed
001Eh invalid connection name

001Fh max session limit reached for service
0020h access rights list for connection/dialout does not include this user
0021h service not responding
0022h missing telephone number
SeeAlso: #03325,#03373

Format of VINES argument block:

Offset	Size	Description (Table 03370)
00h	BYTE	session ID (00h)
01h	BYTE	00h (func "initialize user buffer pointer information area")
02h	WORD	-> user buffer ptr info area in caller's current DS (see #03371)

Format of user buffer ptr info area:

Offset	Size	Description (Table 03371)
00h	WORD	flags 0000h don't read interface's data buffer 0001h read data buffer
02h	DWORD	pointer to ring buffer
06h	WORD	length of ring buffer
08h	WORD	ring buffer offset to last byte read by caller
0Ah	DWORD	pointer to WORD containing offset of last byte in ring buffer filled
0Eh	DWORD	pointer to screen buffer
12h	DWORD	pointer to field containing cursor position
16h	DWORD	pointer to terminal status area (see #03372)

Format of VINES terminal status area:

Offset	Size	Description (Table 03372)
00h	BYTE	status of session: 4Eh=online, 46h=offline, 57h=waiting
01h	BYTE	terminal type (00h=VT100, 01h=TTY, 02h=VT52, 03h=IBM3101)
02h	BYTE	current keypad mode (VT100,VT52 only) 4Eh ("N") numeric mode 41h ("A") application mode
03h	4 BYTES	current state of LEDs (VT100 only) 00h off 01h on
07h	WORD	line error count
09h	WORD	primary error code (see #03373)
0Bh	WORD	secondary error code

(Table 03373)

Values for VINES primary error code:

0000h no error
 0001h unable to make connection
 0002h communications error, restart session
 0003h async terminal emulation service unavailable
 0004h lost carrier
 0005h all matching lines busy
 0006h no lines defined for connection name
 0007h no dial lines available on server
 0008h no matching dial lines available
 0009h out of heap space
 000Ah service error encountered
 000Bh timed out waiting to connect
 000Ch communications error
 000Dh communications error
 000Eh host wants file transferred to/from PC
 000Fh host software changed session parameter
 0010h host software changed tap settings
 0011h host software changed LED indicator
 0012h host software changed display background (secondary error code 00h for white on black, 01h for black on white)
 0013h host software changed display option (secondary error code 00h for off, 01h for on)
 0014h communications error
 0015h communications error
 0016h unable to make connection
 0017h unable to make connection

SeeAlso: #03369

-----N-610003SF01-----

INT 61 - Banyan VINES - ASYNC TERMINAL EMULATION - SEND TO HOST

AX = 0003h subfn 01h

DS:BX -> argument block (see #03374)

Return: AX = status (see #03369)

SeeAlso: AX=0003h/SF=00h,AX=0003h/SF=02h,AX=0003h/SF=14h

Format of VINES argument block for Function 0003h subfunction 01h:

Offset Size Description (Table 03374)

00h BYTE session ID (00h)
 01h BYTE 01h (function "send to host")
 02h BYTE type
 00h ASCII byte

01h ASCII string
 02h terminal function code
 03h up arrow
 04h down arrow
 05h left arrow
 06h right arrow
 07h break
 03h N BYTES type-specific info (see #03375)

Format of type-specific info:

Offset Size Description (Table 03375)

---ASCII byte---

03h BYTE byte to send to host

---ASCII string---

03h WORD length of string

05h WORD pointer to string

---terminal function code (VT52/VT100)---

03h BYTE function code

00h keypad 0

01h keypad 1

...

09h keypad 9

0Ah keypad -

0Bh keypad ,

0Ch keypad .

0Dh keypad ENTER

0Eh PF1

0Fh PF2

10h PF3

11h PF4

---terminal function code (IBM3101)---

03h BYTE function code

00h PF1

...

07h PF8

08h Home

SeeAlso: #03374

-----N-610003SF02-----

INT 61 - Banyan VINES - ASYNC TERMINAL EMULATION - "CONTROL MONITOR"

AX = 0003h subfn 02h

DS:BX -> argument block (see #03376)

Return: AX = status (see #03369)

SeeAlso: AX=0003h/SF=03h,AX=0003h/SF=05h

Format of VINES argument block for Function 0003h subfunction 02h:

Offset Size Description (Table 03376)

00h BYTE session ID (00h)

01h BYTE 02h (function "control monitor")

02h BYTE display flag

00h don't display data received from host

01h display data

-----N-610003SF03-----

INT 61 - Banyan VINES - ASYNC TERMINAL EMULATION - "FLOW CONTROL DATA"

AX = 0003h subfn 03h

DS:BX -> argument block (see #03377)

Return: AX = status (see #03369)

Desc: this function permits the caller to freeze/unfreeze the display and
the ring buffer

SeeAlso: AX=0003h/SF=02h,AX=0003h/SF=06h

Format of VINES argument block for Function 0003h subfunction 03h:

Offset Size Description (Table 03377)

00h BYTE session ID (00h)

01h BYTE 03h (function "flow control data")

02h BYTE flow control flag

00h allow characters to be put into display or ring buffer

01h don't place any more characters into display or ring buffer

-----N-610003SF04-----

INT 61 - Banyan VINES - ASYNC TERMINAL EMULATION - END ACTIVE SESSION

AX = 0003h subfn 04h

DS:BX -> argument block (see #03378)

Return: AX = status (see #03369)

SeeAlso: AX=0003h/SF=0Ah,AX=0003h/SF=0Dh,AX=0003h/SF=0Fh

Format of VINES argument block for ending active session:

Offset Size Description (Table 03378)

00h BYTE session ID (00h)

01h BYTE 04h (function "end active session")

-----N-610003SF05-----

INT 61 - Banyan VINES - ASYNC TERMINAL EMULATION - SET SESSION PARAMETER

AX = 0003h subfn 05h

DS:BX -> argument block (see #03379)

Return: AX = status (see #03369)

SeeAlso: AX=0003h/SF=00h,AX=0003h/SF=06h,AX=0003h/SF=08h

Format of VINES argument block for setting session parameters:

Offset Size Description (Table 03379)

00h BYTE session ID (00h)
 01h BYTE 05h (function "set session parameter")
 02h BYTE parameter number (see #03380)
 03h BYTE new parameter value

(Table 03380)

Values for parameter number:

00h line speed (00h=any, 01h=50, 02h=110, 03h=134.5, 04h=150,
 05h=300, 06h=600, 07h=1200, 08h=2400, 09h=4800, 0Ah=9600)
 01h parity (00h=none, 01h=odd, 02h=even)
 02h duplex (00h=full, 01h=half)
 03h character size (00h=7 bits, 01h=8 bits)
 04h stop bits (00h=1, 01h=2)
 05h XON/XOFF flow control (00h=no, 01h=yes)
 07h intercharacter delay in tenths of a second
 08h interline delay in tenths of a second
 09h auto linefeed (00h=no, 01h=yes)
 0Ah filter control characters (00h=no, 01h=yes)
 0Bh terminal type (00h=VT100,01h=glassTTY,02h=VT52,03h=IBM3101)
 0Ch auto wrap (00h=no, 01h=yes)
 0Dh cursor shape (00h=underscore, 01h=block)
 0Eh character set (00h=UK, 01h=US ASCII)
 0Fh printer port (00h=LPT1, 01h=LPT2, 02h=LPT3)

SeeAlso: #03379

-----N-610003SF06-----

INT 61 - Banyan VINES - ASYNC TERMINAL EMULATION - GET SESSION PARAMETER

AX = 0003h subfn 06h

DS:BX -> argument block (see #03381)

Return: AX = status (see #03369)

SeeAlso: AX=0003h/SF=00h,AX=0003h/SF=05h,AX=0003h/SF=07h

Format of VINES argument block for getting session parameters:

Offset Size Description (Table 03381)

00h BYTE session ID (00h)
 01h BYTE 06h (function "get session parameter")
 02h BYTE parameter number (see #03380)

03h BYTE (ret) current parameter value

-----N-610003SF07-----

INT 61 - Banyan VINES - ASYNC TERMINAL EMULATION - SET TAB SETTINGS

AX = 0003h subfn 07h

DS:BX -> argument block (see #03382)

Return: AX = status (see #03369)

SeeAlso: AX=0003h/SF=05h,AX=0003h/SF=08h

Format of VINES argument block for setting tab settings:

Offset Size Description (Table 03382)

00h BYTE session ID (00h)

01h BYTE 07h (function "set tab settings")

02h WORD pointer to 80-byte buffer in caller's current DS
each byte = 00h if no tab, 01h if tab at that position

-----N-610003SF08-----

INT 61 - Banyan VINES - ASYNC TERMINAL EMULATION - GET TAB SETTINGS

AX = 0003h subfn 08h

DS:BX -> argument block (see #03383)

Return: AX = status (see #03369)

SeeAlso: AX=0003h/SF=06h,AX=0003h/SF=07h

Format of VINES argument block for getting tab settings:

Offset Size Description (Table 03383)

00h BYTE session ID (00h)

01h BYTE 08h (function "get tab settings")

02h WORD pointer to 80-byte buffer in caller's current DS
each byte set to 00h if no tab, 01h if tab at that position

-----N-610003SF09-----

INT 61 - Banyan VINES - ASYNC TERMINAL EMULATION - REFRESH EMULATION SCREEN

AX = 0003h subfn 09h

DS:BX -> argument block (see #03384)

Return: AX = status (see #03369)

SeeAlso: AX=0003h/SF=00h,AX=0003h/SF=0Bh

Format of VINES argument block for Function 0003h subfunction 09h:

Offset Size Description (Table 03384)

00h BYTE session ID (00h)

01h BYTE 09h (function "refresh emulation screen")

-----N-610003SF0A-----

INT 61 - Banyan VINES - ASYNC TERMINAL EMULATION - SUSPEND SESSION TEMPORARILY

AX = 0003h subfn 0Ah

DS:BX -> argument block (see #03385)
 Return: AX = status (see #03369)
 SeeAlso: AX=0003h/SF=04h,AX=0003h/SF=09h,AX=0003h/SF=0Bh,AX=0003h/SF=0Dh

Format of VINES argument block:

Offset	Size	Description (Table 03385)
00h	BYTE	session ID (00h)
01h	BYTE	0Ah (function "suspend session temporarily")
02h	WORD	size of session information to be saved
04h	WORD	pointer to buffer in caller's DS

-----N-610003SF0B-----

INT 61 - Banyan VINES - ASYNC TERMINAL EMULATION - RESUME SUSPENDED SESSION

AX = 0003h subfn 0Bh
 DS:BX -> argument block (see #03386)
 Return: AX = status (see #03369)
 SeeAlso: AX=0003h/SF=09h,AX=0003h/SF=0Ah

Format of VINES argument block:

Offset	Size	Description (Table 03386)
00h	BYTE	session ID (00h)
01h	BYTE	0Bh (function "restore previously suspended session")
02h	WORD	size of buffer into which session info is restored
04h	WORD	pointer to buffer in caller's DS

-----N-610003SF0C-----

INT 61 - Banyan VINES - ASYNC TERMINAL EMULATION - SET SCROLL LOCK CHECKING

AX = 0003h subfn 0Ch
 DS:BX -> argument block (see #03387)
 Return: AX = status (see #03369)
 SeeAlso: AX=0003h/SF=00h

Format of VINES argument block:

Offset	Size	Description (Table 03387)
00h	BYTE	session ID (00h)
01h	BYTE	0Ch (function "set state of scroll lock checking")
02h	BYTE	check_scroll_lock flag
		00h off
		01h on (display of host data stopped while ScrollLock on)

-----N-610003SF0D-----

INT 61 - Banyan VINES - ASYNC TERMINAL EMULATION - EXIT EMULATION

AX = 0003h subfn 0Dh
 DS:BX -> argument block (see #03388)

Return: AX = status (see #03369)

SeeAlso: AX=0003h/SF=04h,AX=0003h/SF=0Ah

Format of VINES argument block:

Offset Size Description (Table 03388)

00h BYTE session ID (00h)

01h BYTE 0Dh (function "exit emulation")

-----N-610003SF0E-----

INT 61 - Banyan VINES - ASYNC TERMINAL EMULATION - INTERRUPT ON CHAR FROM HOST

AX = 0003h subfn 0Eh

DS:BX -> argument block (see #03389)

Return: AX = status (see #03369)

SeeAlso: AX=0003h/SF=00h,AX=0003h/SF=14h

Format of VINES argument block:

Offset Size Description (Table 03389)

00h BYTE session ID (00h)

01h BYTE 0Eh (function "interrupt on character from host")

02h DWORD pointer to routine to be called (0000h:0000h = don't call)

06h DWORD stack pointer to use when call is made

-----N-610003SF0F-----

INT 61 - Banyan VINES - ASYNC TERMINAL EMULATION - START A SESSION

AX = 0003h subfn 0Fh

DS:BX -> argument block (see #03390)

Return: AX = status (see #03369)

SeeAlso: AX=0003h/SF=04h,AX=0003h/SF=0Bh

Format of VINES argument block:

Offset Size Description (Table 03390)

00h BYTE session ID (00h)

01h BYTE 0Fh (function "start a session")

02h WORD pointer to information area in caller's current DS (see #03391)

Format of VINES information area:

Offset Size Description (Table 03391)

00h WORD length of service name

02h WORD pointer to service name in caller's DS

04h BYTE type of connection (00h=connection name, 01h=dialout)

05h WORD length of connection name/telephone number

07h WORD pointer to connection name/telephone number

-----N-610003SF10-----

INT 61 - Banyan VINES - ASYNC TERMINAL EMULATION - START/STOP PRINTING

AX = 0003h subfn 10h

DS:BX -> argument block (see #03392)

Return: AX = status (see #03369)

SeeAlso: AX=0003h/SF=06h,AX=0003h/SF=13h

Format of VINES argument block:

Offset Size Description (Table 03392)

00h BYTE session ID (00h)

01h BYTE 10h (function "start/stop printing of data received from host")

02h WORD print capture flag (00h=off, 01h=on)

-----N-610003SF11-----

INT 61 - Banyan VINES - ASYNC TERMINAL EMULATION - GET FILE TRANSFER PARAMETERS

AX = 0003h subfn 11h

DS:BX -> argument block (see #03393)

Return: AX = status (see #03369)

SeeAlso: AX=0003h/SF=00h,AX=0003h/SF=12h

Format of VINES argument block:

Offset Size Description (Table 03393)

00h BYTE session ID (00h)

01h BYTE 11h (function "get file transfer parameters")

02h WORD pointer to info area in caller's current DS (see #03394)

Format of VINES info area:

Offset Size Description (Table 03394)

00h BYTE protocol flag (00h none, 01h Kermit)

01h BYTE direction flag (00h send, 01h receive)

02h BYTE length of null-terminated PC filename

03h DWORD pointer to null-terminated PC filename

07h BYTE length of null-terminated host filename

08h DWORD pointer to null-terminated host filename

-----N-610003SF12-----

INT 61 - Banyan VINES - ASYNC TERMINAL EMULATION - GET CONNECTION INFORMATION

AX = 0003h subfn 12h

DS:BX -> argument block (see #03395)

Return: AX = status (see #03369)

SeeAlso: AX=0003h/SF=11h,AX=0003h/SF=15h

Format of VINES argument block:

Offset Size Description (Table 03395)

```

00h BYTE session ID (00h)
01h BYTE 12h (function "get connection information")
02h WORD offset of buffer for connection information (see #03396)

```

Format of VINES connection information:

```

Offset Size Description (Table 03396)
00h WORD length of service name (returned)
02h WORD pointer to 64-byte buffer for service name
04h BYTE type of connection
      00h connection name
      01h dialout
05h WORD length of connection name/telephone number
07h WORD pointer to 64-byte buffer for name/telno
09h BYTE server line number being used (returned)

```

-----N-610003SF13-----

```

INT 61 - Banyan VINES - ASYNC TERMINAL EMULATION - START/STOP TRACING TRAFFIC
      AX = 0003h subfn 13h
      DS:BX -> argument block (see #03397)
Return: AX = status (see #03369)
SeeAlso: AX=0003h/SF=00h,AX=0003h/SF=0Eh

```

Format of VINES argument block:

```

Offset Size Description (Table 03397)
00h BYTE session ID (00h)
01h BYTE 13h (function "start/stop tracing data traffic in session")
02h BYTE trace flag (00h=off, 01h=on)

```

-----N-610003SF14-----

```

INT 61 - Banyan VINES - ASYNC TERMINAL EMULATION - INTERRUPT ON HOST MESSAGE
      AX = 0003h subfn 14h
      DS:BX -> argument block (see #03398)
Return: AX = status (see #03369)
SeeAlso: AX=0003h/SF=0Eh

```

Format of VINES argument block:

```

Offset Size Description (Table 03398)
00h BYTE session ID (00h)
01h BYTE 14h (function "interrupt on message from host")
02h DWORD pointer to routine to be called (0000h:0000h = don't call)
06h DWORD stack pointer to use when call is made

```

-----N-610003SF15-----

```

INT 61 - Banyan VINES - ASYNC TERMINAL EMULATION - RESET ERROR

```

AX = 0003h subfn 15h
DS:BX -> argument block (see #03399)
Return: AX = status (see #03369)
SeeAlso: AX=0003h/SF=00h,AX=0003h/SF=12h

Format of VINES argument block:

Offset Size Description (Table 03399)

00h BYTE session ID (00h)

01h BYTE 15h (function "reset error")

-----V-610003-----

INT 61 - OPTIMA 1024 VGA-Sync,ET-3000 chipset - REPORT ZOOM FACTOR

AX = 0003h

Return: AX = zoom factor

Desc: returns the current zoom factor

Range: unknown

Notes: INT 61h is the default interrupt; the actual interrupt number can be
obtained by calling INT 16/AH=FFh

not all vendors include the Tseng TSR which supports these functions

SeeAlso: AX=0000h"OPTIMA",AX=0005h"OPTIMA",INT 16/AH=FFh"OPTIMA"

-----N-610004-----

INT 61 - Banyan VINES - FILE SERVICE CONTROL

AX = 0004h

DS:DX -> request block (see #03401)

Return: AX = status (see #03400)

Notes: This service is handled by the redirector

The request dispatch table contains dummy entries for requests 0Ah,
0Dh, 15h, 19h, and 1Ah, which always return error 0033h (invalid
request ID)

(Table 03400)

Values for Banyan VINES file service control status code:

0000h successful

0002h file not found

0003h path not found

000Fh invalid drive

0015h drive not ready

0033h invalid request ID

0034h drive not connected

0037h drive number out of range

0038h drive number not available

003Ah invalid operation

0055h drive already connected
0057h invalid number of entries

Format of Banyan VINES file service request block:

Offset Size Description (Table 03401)
00h WORD request ID (1..28)
02h n BYTES additional parameters and result buffer

-----N-610004SF0001-----

INT 61 - Banyan VINES - SET DRIVE CONNECTION

AX = 0004h subfn 0001h

DS:DX -> request block (see #03402)

Return: AX = status (0000h,0037h,0038h,0055h) (see #03400)

Desc: Maps the specified file service to a DOS drive letter

Note: If an empty service name is specified, an existing connection will
be removed

SeeAlso: AX=0004h/SF=0002h

Format of VINES Function 0004h/Subfunction 0001h request block:

Offset Size Description (Table 03402)
00h WORD 0001h
02h BYTE drive number (0=default, 1=A, ...)
03h 16 BYTES reserved
13h 64 BYTES null-terminated file service name
53h 5 BYTES reserved

-----N-610004SF0002-----

INT 61 - Banyan VINES - GET DRIVE CONNECTION

AX = 0004h subfn 0002h

DS:DX -> request block (see #03403)

Return: AX = status (0000h,0034h,0037h) (see #03400)

Desc: Returns the StreetTalk name of the associated file service

SeeAlso: AX=0004h/SF=0001h,AX=0004h/SF=0003h

Format of VINES Function 0004h/Subfunction 0002h request block:

Offset Size Description (Table 03403)
00h WORD 0002h
02h BYTE drive number (0=default, 1=A, ...)
03h 16 BYTES reserved
13h 64 BYTES buffer for null-terminated file service name
53h 5 BYTES reserved

-----N-610004SF0003-----

INT 61 - Banyan VINES - GET NETWORK DRIVE MAP

AX = 0004h subfn 0003h
 DS:DX -> request block (see #03404)
 Return: AX = status
 0000h drive map returned in request block
 Desc: Returns a bit map indicating which drive numbers are connected to a
 file service
 Note: Only bits 0-25 are significant
 SeeAlso: AX=0004h/SF=0002h

Format of VINES Function 0004h/Subfunction 0003h request block:

Offset	Size	Description (Table 03404)
00h	WORD	0003h
02h	DWORD	buffer for drive map (bit0=A, bit1=B, ...)

-----N-610004SF0004-----

INT 61 - Banyan VINES - SET USER NAME

AX = 0004h subfn 0004h
 DS:DX -> request block (see #03405)
 Return: AX = status (0000h) (see #03400)
 Notes: Other side effects, if any, not known
 SeeAlso: AX=0004h/SF=0005h,#03431 [ID 0004h],AX=0007h/BX=0005h

Format of VINES Function 0004h/Subfunction 0004h request block:

Offset	Size	Description (Table 03405)
00h	WORD	0004h
02h	64 BYTE	null-terminated user name

-----N-610004SF0005-----

INT 61 - Banyan VINES - GET USER NAME

AX = 0004h subfn 0005h
 DS:DX -> request block (see #03406)
 Return: AX = status (see also #03400)
 0000h user name returned in request block
 Desc: Returns the StreetTalk name of the currently logged in user
 SeeAlso: AX=0004h/SF=0004h,#03431 [ID 0004h],AX=0007h/BX=0005h
 SeeAlso: AX=0004h/SF=0004h

Format of VINES Function 0004h/Subfunction 0005h request block:

Offset	Size	Description (Table 03406)
00h	WORD	0005h (subfunction "Get User Name")
02h	64 BYTE	buffer for null-terminated user name

-----N-610004SF0006-----

INT 61 - Banyan VINES - TEST DRIVE CONNECTION

AX = 0004h subfn 0006h
DS:DX -> request block (see #03407)
Return: AX = status (see also #03400)
0000h connection flag returned in request block
Desc: Returns a flag indicating if the drive is connected to a file service
SeeAlso: AX=0004h/SF=0007h

Format of VINES Function 0004h/Subfunction 0006h request block:

Offset	Size	Description (Table 03407)
00h	WORD	0006h (subfunction "Test Drive Connection")
02h	BYTE	buffer for connection flag (1 = not connected, 2 = connected)

-----N-610004SF0007-----

INT 61 - Banyan VINES - GET CONNECTION DATA STRUCTURES

AX = 0004h subfn 0007h
DS:DX -> request block (see #03408)
Return: AX = status (see also #03400)
0000h pointer returned in request block
0033h invalid request ID
Desc: Returns pointers to various internal data structures

Format of VINES Function 0004h/Subfunction 0007h request block:

Offset	Size	Description (Table 03408)
00h	WORD	0007h
02h	BYTE	request ID (0..7) (see #03409)
03h	WORD	buffer for pointer segment
05h	WORD	buffer for pointer offset

SeeAlso: #03410,#03411

(Table 03409)

Values for VINES pointer ID:

00h	job file table ???
01h	drive handle table
02h	drive connection table
03h	IPC port of drive Z:
04h	end of drive connection table
05h	???
06h	???
07h	redirector data segment (offset=0)

SeeAlso: #03408

Format of VINES drive handle table:

Offset Size Description (Table 03410)
00h BYTE handle to default drive (0 = not connected)
01h BYTE handle to drive A: (0 = not connected)
...
1Ah BYTE handle to drive Z: (0 = not connected)
SeeAlso: #03411

Format of VINES drive connection table:

Offset Size Description (Table 03411)
00h WORD file service type (?) (0001h if drive Z:, 0000h else)
02h 16 BYTES IPC port (contains the server serial number)
12h 46 BYTES ???
40h 64 BYTES null-terminated StreetTalk name of associated file service
80h 64 BYTES null-terminated root directory (/ROOT: parameter of SETDRIVE)
SeeAlso: #03410

-----N-610004SF0008-----

INT 61 - Banyan VINES - GET SERVER SERIAL NUMBER

AX = 0004h subfn 0008h

DS:DX -> request block (see #03412)

Return: AX = status (0000h,000Fh,0015h) (see #03400)

Notes: The server ID is extracted from the IPC port address of the associated
file service stored in the drive connection table
The server serial number and the server subnet number are returned in
big-endian (Motorola) format (i.e. MSB first)

SeeAlso: AX=0004h/SF=0005h

Format of VINES request block for getting server serial number:

Offset Size Description (Table 03412)
00h WORD 0008h
02h WORD drive number (0=default, 1=A, ...)
04h DWORD buffer for server serial number
08h WORD buffer for server subnet number (always 0001h)

-----N-610004SF0009-----

INT 61 - Banyan VINES - SET ??? PARAMETER

AX = 0004h subfn 0009h

DS:DX -> request block (see #03413)

Return: AX = status (0000h) (see #03400)

Notes: The specified WORD parameter is copied to offset 1702h of the
redirector code segment (revision 5.53)

SeeAlso: AX=0004h/SF=000Bh

Format of VINES Function 0004h/Subfunction 0009h request block:

Offset Size Description (Table 03413)

00h WORD 0009h

02h WORD parameter

-----N-610004SF000B-----

INT 61 - Banyan VINES - GET ??? DRIVE PARAMETER

AX = 0004h subfn 000Bh

DS:DX -> request block (see #03414)

Return: AX = status (0000h,0034h,0037h) (see #03400)

Notes: This request returns a WORD value from the drive connection table
read from offset 0012h of the specified drive's entry (usually 0037h)

SeeAlso: AX=0004h/SF=0009h

Format of VINES Function 0004h/Subfunction 000Bh request block:

Offset Size Description (Table 03414)

00h WORD 000Bh

02h BYTE drive number (0=default, 1=A, ...)

03h WORD buffer for parameter

-----N-610004SF000C-----

INT 61 - Banyan VINES - LOGOUT FROM VINES

AX = 0004h subfn 000Ch

DS:DX -> request block (see #03415)

Return: AX = status 0000h (see #03400)

Desc: Terminates a running VINES session and clears all session specific
buffers

SeeAlso: AX=0004h/SF=0008h

Format of VINES Function 0004h/Subfunction 000Ch request block:

Offset Size Description (Table 03415)

00h WORD 000Ch

-----N-610004SF000E-----

INT 61 - Banyan VINES - GET ACCESS RIGHTS LIST (ARL)

AX = 0004h subfn 000Eh

DS:DX -> request block (see #03416)

Return: AX = status (see also #03400)

0000h request has been processed

Desc: Returns the owner, group, and extended access rights for the specified
file or directory

Note: A status of 0000h does not necessarily indicate a successful request;
check the status word at offset 000Eh of the request block too

SeeAlso: AX=0004h/SF=000Fh

Format of VINES Function 0004h/Subfunction 000Eh request block:

Offset	Size	Description (Table 03416)
00h	WORD	000Eh (subfunction "Get Access Rights List")
02h	BYTE	drive number (0=default, 1=A, ...)
03h	BYTE	reserved
04h	DWORD	pointer to path specification (drive letter ignored)
08h	DWORD	pointer to ARL control block (see #03417)
0Ch	WORD	buffer for number of ARL entries (1..5)
0Eh	WORD	buffer for status (see #03418)

Format of ARL control block:

Offset	Size	Description (Table 03417)
00h	64 BYTES	1st StreetTalk name (user, list, or service)
40h	WORD	1st ARL
42h	64 BYTES	2nd StreetTalk name (user, list, or service)
82h	WORD	2nd ARL
84h	64 BYTES	3rd StreetTalk name (user, list, or service)
C4h	WORD	3rd ARL
C6h	64 BYTES	4th StreetTalk name (user, list, or service)
106h	WORD	4th ARL
108h	64 BYTES	5th StreetTalk name (user, list, or service)
148h	WORD	5th ARL

(Table 03418)

Values for VINES Get/Set Access Rights List status:

0000h successful
0002h file not found
0003h path not found
003Ah invalid operation
0057h invalid number of entries

-----N-610004SF000F-----

INT 61 - Banyan VINES - SET ACCESS RIGHTS LIST (ARL)

AX = 0004h subfn 000Fh

DS:DX -> request block (see #03419)

Return: AX = status (see also #03400)

0000h request has been processed

Desc: Sets the owner, group, and extended access rights for the specified file or directory

Note: A status of 0000h does not necessarily indicate a successful request; check the status word at offset 000Eh of the request block too

If the number of ARL entries is less than 1 or greater than 5, error

0057h (invalid number of entries) is returned

SeeAlso: AX=0004h/SF=000Eh

Format of VINES Function 0004h/Subfunction 000Fh request block:

Offset Size Description (Table 03419)

00h	WORD	000Fh
02h	BYTE	drive number (0=default, 1=A, ...)
03h	BYTE	reserved
04h	DWORD	pointer to path specification (drive letter ignored)
08h	DWORD	pointer to ARL control block (see #03417)
0Ch	WORD	number of ARL entries (1..5)
0Eh	WORD	buffer for status (see #03418)

-----N-610004SF0010-----

INT 61 - Banyan VINES - FILE SERVICE CONTROL - ???

AX = 0004h subfn 0010h

DS:DX -> request block (see #03420)

Return: AX = status (see also #03400)

0000h request has been processed

SeeAlso: AX=0004h/SF=000Fh

Format of VINES Function 0004h/Subfunction 0010h request block:

Offset Size Description (Table 03420)

00h	WORD	0010h
02h	BYTE	drive number (0=default, 1=A, ...)
03h	BYTE	reserved
04h	64 BYTES	???
44h	16 BYTES	IPC port (see #03328 at AX=0001h"Sosock")

-----N-610004SF0013-----

INT 61 - Banyan VINES - GET VALUE OF GLOBAL VARIABLE

AX = 0004h subfn 0013h

DS:DX -> request block (see #03421)

Return: AX = status (see also #03400)

0000h successful

0033h invalid request ID (returned if variable index out of range)

Desc: Returns the value of the specified global variable

Note: The data will be truncated if the specified buffer is not large enough.

Check the length word in the request block to find out how many

bytes should have been copied.

SeeAlso: AX=0004h/SF=0014h,AX=0006h

Format of request block:

Offset Size Description (Table 03421)

00h WORD 0013h
 02h WORD variable index (0000h-001Fh) (see #03422)
 04h WORD buffer length (on exit set to # of bytes actually needed)
 06h DWORD far pointer to result buffer

SeeAlso: #03423

(Table 03422)

Values for variable indices:

Index	Length	Description
0000h	1	???
0001h	16	StreetTalk port
0002h	4	StreetTalk session handle
0003h	6	colors
0004h	64	StreetTalk name of currently logged in user
0005h	0	old clear text password (not applicable)
0006h	0	default group (not applicable)
0007h	1	number of messages in message queue
0008h	1	message flag
0009h	16	TalkPort
000Ah	2	TalkPort socket
000Bh	64	StreetTalk name of mail service
000Ch	2	message timeout
000Dh	2	next message place
000Eh	246	message queue (3 times 80 characters + 2 flag bytes)
000Fh	14	system nonce
0010h	2	failed login count
0011h	1	new mail message received
0012h	8	mail service capabilities
0013h	3	printer port diverted
0014h	32	international location
0015h	16	IPC port of drive Z:
0016h	1024	pointer to current directory table
0017h	2	maximum number of file volume mappings
0018h	24	routine to call back for messages (6 far pointers)
0019h	4	current virtual machine
001Ah	20	DOS redirector message bell tones
001Bh	2	various flags used for internal communication
001Ch	0	OS/2 client configuration (not applicable)
001Dh	8	redirector/IFS file version

001Eh 0 reserved (not applicable)
 001Fh 4 address of VVINESD.386 dispatch

-----N-610004SF0014-----

INT 61 - Banyan VINES - SET VALUE OF GLOBAL VARIABLE

AX = 0004h subfn 0014h

DS:DX -> request block (see #03423)

Return: AX = status

0000h successful

0033h invalid request ID (returned if variable index out of range)

Desc: Assigns a new value to the specified global variable

Note: If the specified number of bytes to copy exceeds the size of the destination variable, nothing will be copied.

SeeAlso: AX=0004h/SF=0013h,AX=0006h

Format of request block:

Offset Size Description (Table 03423)

00h WORD 0014h

02h WORD variable index (0000h-001Fh) (see #03422)

04h WORD number of bytes to copy

06h DWORD far pointer to new value

SeeAlso: #03421

-----N-610004SF0016-----

INT 61 - Banyan VINES - GET NUMBER OF OPEN FILES ON REMOTE DRIVE

AX = 0004h subfn 0016h

DS:DX -> request block (see #03424)

Return: AX = status

0000h open file count returned in request block

Desc: Counts the number of open files on the specified remote drive.

Note: No error is returned if the specified drive number is invalid or points to a local drive. Use AX=0004h/SF=0006h to test if a given drive number is assigned to a remote drive.

SeeAlso: AX=0004h/SF=0006h

Format of request block:

Offset Size Description (Table 03424)

00h WORD 0016h

02h BYTE drive number (0=default, 1=A:, ...)

03h BYTE reserved

04h WORD buffer for open file count

SeeAlso: #03426

-----N-610004SF0018-----

INT 61 - Banyan VINES - SET UNIDENTIFIED INTERNAL FLAG

AX = 0004h subfn 0018h

DS:DX -> request block (see #03425)

Return: AX = status

0000h success

Desc: Sets an unidentified BYTE variable in the redirector data segment to 1.

Note: In the VINES 5.53 "REDIR4" redirector, this variable is located at

DS:0B53.

Format of request block:

Offset Size Description (Table 03425)

00h WORD 0018h

SeeAlso: #03426

-----N-610004SF001B-----

INT 61 - Banyan VINES - UNIDENTIFIED DRIVE SPECIFIC FUNCTION

AX = 0004h subfn 001Bh

DS:DX -> request block (see #03426)

Return: AX = status

0000h success

0034h drive not connected

0037h drive number out of range

Desc: The purpose of this function is not known.

Format of request block:

Offset Size Description (Table 03426)

00h WORD 001Bh

02h BYTE drive number (0=default, 1=A:, ...)

03h BYTE ???

04h WORD ???

06h DWORD far pointer to data area

0Ah WORD ???

0Ch DWORD far pointer to data area

10h WORD buffer for result

SeeAlso: #03424,#03426

-----N-610004SF001C-----

INT 61 - Banyan VINES - UNIDENTIFIED FUNCTION

AX = 0004h subfn 001Ch

DS:DX -> request block (see #03427)

Return: AX = status

0000h success

0057h invalid number of entries

Desc: The purpose of this function is not known.

Format of request block:

Offset Size Description (Table 03427)

```
00h WORD 001Ch
02h WORD ???
04h WORD ???
06h WORD ???
08h WORD ???
0Ah DWORD far pointer to data area (14 bytes)
```

-----V-610004-----

INT 61 - OPTIMA 1024 VGA-Sync,ET-3000 chipset - ENTER SPECIFY MODE

AX = 0004h

Range: unknown

Notes: Specify Mode is enabled by hot key (seeAlso below), and allows panning and zooming via the numeric keypad.

INT 61 is the default interrupt; the actual interrupt number can be obtained by calling INT 16/AH=FFh

not all vendors include the Tseng TSR which supports these functions

SeeAlso: AX=0000h"OPTIMA",AX=0007h"OPTIMA",INT 16/AH=FFh"OPTIMA"

-----N-610005-----

INT 61 - Banyan VINES - PRINT SERVICE CONTROL

AX = 0005h

DS:DX -> request block (see #03428)

Return: AX = status

0000h successful

0001h network software not installed or incompatible

Notes: This service is handled by the redirector

The size of the request/result block depends on the selected function and the value at offset 04h.

SeeAlso: INT 2F/AX=D702h

Format of VINES request/result block for printer control:

Offset Size Description (Table 03428)

```
00h WORD function
0201h "endspool" all data for a print job has been sent
      (no bytes returned)
0202h "deassign" the virtual printer port
      Before the printer port is deassigned all data sent
      to it is endspooled.
      (no bytes returned)
```


0205h "get_active" get currently active printer port
(1 byte required for result)
Returns: - the currently active printer port (1-3) (0 if
there is none)

0207h ??? purpose unknown; called periodically every second

020Ch "get_bannertitle" gets the title which is printed on the
banner page of a print job sent to the virtual port
(16 bytes required for result)
Returns: - the null-terminated banner title

020Eh "get_servicename" gets the name of the print service
assigned to the virtual port
(64 bytes required for result)
Returns: - the null-terminated print service name assigned
to the virtual port

0210h "set_bannertitle" sets the title which is printed on the
banner page of a print job sent to the virtual port
(16 bytes required for input)
Input: - the null-terminated banner title (max 16 chars)

0212h "set_servicename" sets the name of the print service
assigned to the virtual port
(64 bytes required for input)
Input: - the null-terminated print service name
(max 64 chars)

0216h "get_job_paper_format" gets the paper format for
subsequent print jobs
(16 bytes required for result)
Returns: - the null-terminated paper format

0217h "set_job_paper_format" sets the paper format for
subsequent print jobs
(16 bytes required for input)
Input: - the null-terminated paper format (max 16 chars)

02h WORD number of active virtual port (1-3) (0 for function 0205h)

04h WORD Offset into the request block from which additional input
is to be read and/or at which the result is to be copied
(if there is any). Therefore the smallest block size for a
function which returns a result is
this offset + the maximum number of returned bytes - 1

06h WORD 0000h

-----V-610005-----

INT 61 - OPTIMA 1024 VGA-Sync - QUERY ZOOM WINDOW
AX = 0005h

BX:CX -> buffer for window parameters (see #03429)

Range: unknown

Notes: INT 61 is the default interrupt; the actual interrupt number can be obtained by calling INT 16/AH=FFh

not all vendors include the Tseng TSR which supports these functions

SeeAlso: AX=0000h"OPTIMA",AX=0003h"OPTIMA",AX=0006h"OPTIMA"

SeeAlso: INT 16/AH=FFh"OPTIMA"

Format of Optima 1024 window parameters:

Offset Size Description (Table 03429)

00h	WORD	X start of zoom window
02h	WORD	Y start of zoom window
04h	WORD	X end of zoom window
06h	WORD	Y end of zoom window
08h	WORD	current zoom factor
0Ah	WORD	zoom offset start X
0Ch	WORD	zoom offset start Y

-----V-610006-----

INT 61 - OPTIMA 1024 VGA-Sync,ET-3000 chipset - SET ZOOM WINDOW

AX = 0006h

BX:CX -> zoom window description (see #03430)

Range: unknown

Notes: width of zoom window must be a multiple of the pixel replication (zoom) factor

INT 61 is the default interrupt; the actual interrupt number can be obtained by calling INT 16/AH=FFh

not all vendors include the Tseng TSR which supports these functions

SeeAlso: AX=0000h"OPTIMA",AX=0001h"OPTIMA",AX=0005h"OPTIMA",AX=0008h"OPTIMA"

SeeAlso: INT 16/AH=FFh"OPTIMA"

Format of Optima 1024 zoom window description:

Offset Size Description (Table 03430)

00h	WORD	X start of zoom window
02h	WORD	Y start of zoom window
04h	WORD	X end of zoom window
06h	WORD	Y end of zoom window

-----N-610006-----

INT 61 - Banyan VINES - GET ADDRESS OF GLOBAL VARIABLE

AX = 0006h

BX = bits 14..0: request ID (see #03431)

bit 15: result destination flag (0 for ES:BX, 1 for AX:BX)

Return: AX = status
 0000h successful
 000Bh invalid request
 ES:BX -> internal variable (bit 15 of BX was not set)
 AX:BX -> internal variable (bit 15 of BX was set)
 Note: This service is handled by the redirector
 SeeAlso: AX=0004h/SF=0013h,AX=0004h/SF=0014h

(Table 03431)

Values for request IDs:

Request	Length	Description
0000h	1	???
0001h	16	StreetTalk port
0002h	4	StreetTalk session handle
0003h	6	colors
0004h	64	StreetTalk name of currently logged in user
0005h	0	old clear text password (always returns error code 000Bh)
0006h	0	default group (not applicable)
0007h	1	message queue count
0008h	1	message flag
0009h	16	TalkPort
000Ah	2	TalkPort socket
000Bh	64	StreetTalk name of mail service
000Ch	2	message timeout
000Dh	2	next message place
000Eh	246	message queue
000Fh	14	system nonce
0010h	2	failed login count
0011h	1	new mail message received
0012h	8	mail capabilities
0013h	3	printer port diverted
0014h	32	international location
0015h	16	IPC port of drive Z:
0016h	1024	pointer to current directory table
0017h	2	maximum number of file volume mappings
0018h	24	routine to call back for messages
0019h	4	current virtual machine
001Ah	20	DOS redirector message bell tones
001Bh	2	various flags used for internal communication
001Ch	0	OS/2 client configuration (always returns error code 000Bh)
001Dh	8	redirector/IFS file version

001Eh 0 reserved (always returns error code 000Bh)
001Fh 4 address of VVINESD.386 dispatch
-----V-610007-----
INT 61 - OPTIMA 1024 VGA-Sync,ET-3000 chipset - QUERY APPLICATION KEY
AX = 0007h

Return: AX = current state (0000h/0001h)

Desc: returns the current toggle state of the application (END) key in
specify mode. E.g. in the OPTIMA AutoCAD driver, 0 means AutoCAD
calls INT 61/AX=0001h every time the crosshair cursor moves.

Notes: In specify mode, the END key has been reserved for applications.

INT 61 is the default interrupt; the actual interrupt number can be
obtained by calling INT 16/AH=FFh

not all vendors include the Tseng TSR which supports these functions

SeeAlso: AX=0000h"OPTIMA",AX=0001h"OPTIMA",AX=0004h"OPTIMA"

SeeAlso: INT 16/AH=FFh"OPTIMA",INT 7A/AX=0001h"AutoCAD"

-----N-610007BX0000-----
INT 61 - Banyan VINES - RECONNECT TO STREETTALK

AX = 0007h

BX = 0000h

Return: AX = status (see #03432)

SeeAlso: AX=0007h/BX=0002h

(Table 03432)

Values for VINES function status:

0000h successful
0001h requested handler not installed or incompatible
000Bh service busy
000Ch invalid request ID
03E9h incorrect name syntax
03EAh organization name too long
03EBh group name too long
03ECh item name too long
03EDh StreetTalk name too long
03F3h organization not found
03F4h group not found
03F5h StreetTalk name not found
03F8h not a StreetTalk name
0409h modify access denied
040Dh appropriate StreetTalk name unavailable
0411h all matching names have been returned
0412h some groups unavailable, all available matches returned

-----N-610007BX0001-----

INT 61 - Banyan VINES - GET MM PORT FOR A SERVICE

AX = 0007h

BX = 0001h

DS:DX -> StreetTalk service name (64 bytes)

DS:DI -> 16-byte buffer for IPC port (see #03328 at AX=0001h"Sosock")

Return: AX = status (see #03432)

Notes: This function calls AX=0007/BX=0002h with the port count set to 1 and
the port type set to 00h

SeeAlso: AX=0007h/BX=0002h,AX=0007h/BX=0004h,AX=0007h/BX=000Ah

-----N-610007BX0002-----

INT 61 - Banyan VINES - GET PORTS FOR A SERVICE

AX = 0007h

BX = 0002h

DS:DX -> StreetTalk service name

DS:DI -> port record block (see #03433)

Return: AX = status (see #03432)

Note: if the requested number of ports is greater than five, it will be reset
to five

SeeAlso: AX=0007h/BX=0001h,AX=0007h/BX=0003h,AX=0007h/BX=0004h

SeeAlso: AX=0007h/BX=000Ah

Format of VINES port record block:

Offset Size Description (Table 03433)

00h WORD (call) number of 17-byte elements (01h-05h) to be filled

02h BYTE (call) port type for first port

03h 16 BYTES (ret) port (see #03328)

13h BYTE (call) port type for second port

14h 16 BYTES (ret) port (see #03328)

24h BYTE (call) port type for third port

25h 16 BYTES (ret) port (see #03328)

35h BYTE (call) port type for fourth port

36h 16 BYTES (ret) port (see #03328)

46h BYTE (call) port type for fifth port

47h 16 BYTES (ret) port (see #03328)

57h BYTE reserved

SeeAlso: #03435

-----N-610007BX0003-----

INT 61 - Banyan VINES - FIND A SERVICE

AX = 0007h

BX = 0003h

DS:DX -> port control block (see #03434)
Return: AX = status (see AX=0007h/BX=0000h)
SeeAlso: AX=0007h/BX=0002h,AX=0007h/BX=0004h,AX=0007h/BX=0005h

Format of port control block:

Offset	Size	Description (Table 03434)
00h	WORD	???
02h	WORD	???
04h	16 BYTES	IPC port (see #03328 at AX=0001h"Sosock")
14h	5 WORDs	???

SeeAlso: #03440

-----N-610007BX0004-----

INT 61 - Banyan VINES - SET PORTS FOR A SERVICE

AX = 0007h

BX = 0004h

DS:DX -> StreetTalk name of service (max 64 bytes)

DS:DI -> port record block (see #03435)

Return: AX = status (see #03432)

SeeAlso: AX=0007h/BX=0002h,AX=0007h/BX=0003h,AX=0007h/BX=000Ah

Format of VINES port record block:

Offset	Size	Description (Table 03435)
00h	WORD	number of 17-byte elements
02h	BYTE	port type for first port
03h	16 BYTES	first IPC port (see #03328)
13h	BYTE	port type for second port
14h	16 BYTES	second IPC port (see #03328)
24h	BYTE	port type for third port
25h	16 BYTES	third IPC port (see #03328)
35h	BYTE	port type for fourth port
36h	16 BYTES	fourth IPC port (see #03328)
46h	BYTE	port type for fifth port
47h	16 BYTES	fifth IPC port (see #03328)
57h	BYTE	reserved

SeeAlso: #03433

-----N-610007BX0005-----

INT 61 - Banyan VINES - GET USER NAME

AX = 0007h

BX = 0005h

DS:DX -> 64-byte buffer for user's StreetTalk name

Return: AX = status (see also #03432)

0000h successful

0001h network software not installed or incompatible

Note: if no user logged in, first byte of returned name will be 00h

SeeAlso: AX=0007h/BX=0007h

-----N-610007BX0006-----

INT 61 - Banyan VINES - TRANSLATE ERROR INTO ASCII STRING

AX = 0007h

BX = 0006h

SI = error code (>100)

DS:DX -> 80-byte buffer for error text

Return: AX = status (see also #03432)

0000h successful

0001h network software not installed or incompatible

-----N-610007BX0007-----

INT 61 - Banyan VINES - VERIFY EXISTENCE OF NAME AND RETURN CANONICAL FORM

AX = 0007h

BX = 0007h

DS:DX -> NiceName block (see #03436)

Return: AX = status (see #03432)

SeeAlso: AX=0007h/BX=0005h,AX=0007h/BX=0008h

Format of VINES NiceName block:

Offset Size Description (Table 03436)

00h WORD type of name

0064h (100) organization

00C8h (200) group

012Ch (300) item

02h WORD pointer to ASCIZ input name

04h WORD pointer to 64-byte buffer for output name

SeeAlso: #03437

-----N-610007BX0008-----

INT 61 - Banyan VINES - ENUMERATE StreetTalk NAMES

AX = 0007h

BX = 0008h

DS:DX -> enumerate block (see #03437)

Return: AX = status (see also #03432)

0000h successful

0411h all matching names have been returned

0412h some groups unavailable, all available matches returned

Note: each program using this call should continue until a nonzero status

is returned; otherwise, some resources will not be freed for several

hours

SeeAlso: AX=0007h/BX=0005h,AX=0007h/BX=0007h,AX=0007h/BX=0009h

Format of VINES enumerate block:

Offset Size Description (Table 03437)

00h WORD return code
02h WORD pointer to pattern string
04h WORD enumerate type
 0064h (100) organization
 00C8h (200) group
 012Ch (300) item
06h WORD enumerate class
 0000h unspecified (return all matching items)
 0001h user names
 0002h service names
 0003h list names
 0004h nicknames
08h WORD pointer to category criteria block (see #03438) or 0
0Ah WORD pointer to array of 64-byte returned names
0Ch WORD number of names returned
0Eh 6 BYTES reserved for subsequent enumerated calls (set to zeros on first call)

SeeAlso: #03436

Format of category criteria block:

Offset Size Description (Table 03438)

00h WORD exclude flag
 0000h return only items with the specified categories
 0001h return all items except those with the given categories
02h WORD number of categories (max 5, 0000h = match all categories)
04h WORD category 1 value (see #03439)
06h WORD category 2 value
08h WORD category 3 value
0Ah WORD category 4 value
0Ch WORD category 5 value

SeeAlso: #03437

(Table 03439)

Values for common VINES service categories:

0001h server service
0002h file service

0003h print service
0004h mail service
0005h StreetTalk
0006h time service
0007h FTP service
0008h semaphore service
0009h 3270/SNA service
000Ah terminal emulation service
000Bh asynchronous terminal service
000Ch NETBIOS service
000Dh PC-based service
000Eh BSC service
000Fh APPC service
0010h NM service
0011h VS service
0012h POP service
0013h AFP service
0014h VCS service
0015h EMS service
0016h LN service
0017h BS service
0023h RIPL service
0064h ADM service

SeeAlso: #03438

-----N-610007BX0009-----

INT 61 - Banyan VINES - TRANSLATE ERROR INTO ASCII STRING (SELECTED PORT)

AX = 0007h

BX = 0009h

SI = error code

DS:DX -> 80-byte buffer for error text

DS:DI -> IPC port (16 bytes; see AX=0001h#"Sosock" for port format)

Return: AX = status (see AX=0007h/BX=0000h)

Note: Same as AX=0007h/BX=0006h, except IPC port is specified explicitly

SeeAlso: AX=0007h/BX=0006h

-----N-610007BX000A-----

INT 61 - Banyan VINES - internal, OS/2 only - GET MANY PORTS FOR SERVICE

AX = 0007h

BX = 000Ah

DS:DX -> port control block (see #03440)

Return: AX = status (see AX=0007h/BX=0000h)

Note: This function cannot be called from applications, as the DS register

is checked against the redirector data segment address (status code

0001h will be returned if DS doesn't match)

SeeAlso: AX=0007h/BX=0001h,AX=0007h/BX=0002h,AX=0007h/BX=0004h

Format of port control block:

Offset Size Description (Table 03440)

00h WORD ???

02h WORD ???

04h 16 BYTES IPC port (see #03328 at AX=0001h"Sosock")

14h 14 WORDs ???

SeeAlso: #03434

-----V-610008-----

INT 61 - OPTIMA 1024 VGA-Sync,ET-3000 chipset - SET ZOOM OFFSET

AX = 0008h

BX = X start of zoom offset

CX = Y start of zoom offset

Desc: specifies the first byte of video memory to appear in the zoom window

Range: unknown

Notes: INT 61 is the default interrupt; the actual interrupt number can be
obtained by calling INT 16/AH=FFh

not all vendors include the Tseng TSR which supports these functions

SeeAlso: AX=0000h"OPTIMA",AX=0006h"OPTIMA",INT 16/AH=FFh"OPTIMA"

-----N-610008BX0000-----

INT 61 - Banyan VINES - TALKPORT SERVICE - GET HANDLER ADDRESS

AX = 0008h

BX = 0000h

Return: AX = status

0000h successful

ES:DX = far pointer to handler

Desc: The TalkPort is a facility to display event messages (e.g. mail
received, printing on a local printer, ...) on the screen

Note: This service is handled by the redirector

SeeAlso: AX=0008h/BX=0001h,AX=0008h/BX=0002h

-----N-610008BX0001-----

INT 61 - Banyan VINES - TALKPORT SERVICE - SET HANDLER ADDRESS

AX = 0008h

BX = 0001h

ES:DX -> new handler

Return: AX = status

0000h successful

SeeAlso: AX=0008h/BX=0000h

-----N-610008BX0002-----

INT 61 - Banyan VINES - TALKPORT SERVICE - POST MESSAGE ON LOCAL DISPLAY

AX = 0008h

BX = 0002h

CX = message display flags (see #03441)

DS:DX -> ASCIZ string to display (only first 80 chars used)

Return: AX = status

0000h successful

000Bh message display function currently busy

000Ch message queue full

Notes: queues up to three messages to be displayed on the bottom line

This service is handled by the redirector

In a DOS session, TalkPort messages are displayed on display line 25;

in a Windows session, TalkPort messages are displayed in dialog boxes

SeeAlso: AX=0008h/BX=0000h,AX=0008h/BX=0003h

Bitfields for VINES message display flags:

Bit(s) Description (Table 03441)

0 message will remain on screen until user presses ^X

1 ring bell after displaying message

2 blink

-----N-610008BX0003-----

INT 61 - Banyan VINES - TALKPORT SERVICE - INTERCEPT VINES 25th-LINE MESSAGES

AX = 0008h

BX = 0003h

DS:DX -> request block (see #03442)

Return: AX = status

0000h successful

0001h network software not installed or incompatible

Notes: message handler should not call BIOS or DOS functions, and should

either call next handler or simply return

to stop intercepting messages, set prev and next request blocks to

point at each other

SeeAlso: AX=0008h/BX=0002h

Format of VINES request block for 25-th line messages:

Offset Size Description (Table 03442)

00h DWORD pointer to user-written message handler

04h DWORD pointer to next request block (filled in by VINES)

08h DWORD pointer to previous request block (filled in by VINES)

0Ch DWORD pointer to message storage area (filled by VINES) (see #03443)

Format of message storage area:

Offset	Size	Description (Table 03443)
00h	16 BYTES	IPC port of message sender (see #03328)
10h	BYTE	message flags
11h	WORD	reserved
13h	BYTE	length of message
14h	80 BYTES	message text

SeeAlso: #03442

-----N-610009-----
INT 61 - Banyan VINES - NETBIOS CLIENT SERVICE
AX = 0009h
other parameters, if any, unknown

-----N-61000A-----
INT 61 - Banyan VINES - SECONDARY 3270 INTERFACE
AX = 000Ah

Note: either 3270/SNA or 3270/BSC interface will use AX=000Ah, depending on
which is loaded second. The first interface loaded will use AX=0002h

SeeAlso: INT 61/AX=0002h

-----N-61000B-----
INT 61 - Banyan VINES - SEMAPHORE SERVICE
AX = 000Bh
other parameters, if any, unknown

-----N-61000C-----
INT 61 - Banyan VINES - 3270 KEYBOARD HANDLER
AX = 000Ch
other parameters, if any, unknown

-----N-61000D-----
INT 61 - Banyan VINES - T3270 SERVICE FUNCTION
AX = 000Dh
other parameters, if any, unknown

-----N-61000E-----
INT 61 - Banyan VINES - ENHANCED 3270 SERVICE
AX = 000Eh
other parameters, if any, unknown

-----N-61000F-----
INT 61 - Banyan VINES - VINES SOCKET API (FAR DATA)
AX = 000Fh
CX:BX -> request block (see #03444)
DX = application's data segment (should be = CX)
ES = DS

Return: AX = status
 0000h successful
 0099h invalid operation
 00A5h resource not available

Desc: StreetTalk access with session handles

Notes: This service is handled by the adapter driver
 The request dispatch table contains dummy entries for subfunctions
 06h, 0Dh, 0Eh, 0Fh, 10h, and 11h, which always return error 0099h
 (invalid operation) in the request block's result buffer
 This interface does the same as the corresponding AX=0001h interface,
 but with a slightly modified request block structure (i.e it allows
 FAR parameter block pointers) and without expecting DS=0001h
 Some adapter drivers (e.g. the NDIS2 token ring driver NDTOKBAN for
 VINES 5.53) ignore the value in CX and use the value from DX instead
 SeeAlso: AX=0001h/SF=0001h,AX=000Fh/SF=0001h

Format of VINES Socket request block:

Offset	Size	Description (Table 03444)
00h	WORD	subfunction number (01h-16h)
02h	WORD	magic number (FEFEh)
04h	WORD	buffer for return code (set to 0 before calling)
06h	WORD	reserved (will be set to client DS)
08h	WORD	reserved (obviously not used; set to 0 before calling)
0Ah	DWORD	far pointer to parameter block
0Eh	WORD	StreetTalk handle

SeeAlso: #03324 at INT 61/AX=0001h

-----N-61000FSF0001-----

INT 61 - Banyan VINES - OPEN SOCKET
 AX = 000Fh subfn 0001h
 CX:BX -> request block (see #03445)
 DX = application's data segment (should be = CX)
 ES = DS

Return: AX = status
 0000h successful
 00A5h resource not available

Desc: Opens a socket and returns a socket handle

Format of request block:

Offset	Size	Description (Table 03445)
00h	WORD	0001h
02h	WORD	magic number (FEFEh)

04h WORD buffer for return code (set to 0 before calling)
 06h WORD reserved (will be set to client DS)
 08h WORD reserved (obviously not used; set to 0 before calling)
 0Ah DWORD far pointer to parameter block (see #03446)
 0Eh WORD process handle (see AX=000Fh/SF=0008h)

Format of parameter block:

Offset	Size	Description (Table 03446)
00h	WORD	reserved
02h	WORD	address family (0003h = VINES)
04h	WORD	socket type (0001h = IPC, 0002h = SPP)
06h	WORD	protocol number (FFFFh = default protocol)
08h	WORD	reserved
0Ah	WORD	local port number (0 = transient, 1..1FFh = well-known port)
0Ch	DWORD	far pointer to socket handle buffer (2 bytes)
10h	DWORD	far pointer to IPC port buffer (16 bytes)

SeeAlso: #03445

-----N-61000FSF0002-----

INT 61 - Banyan VINES - SEND MESSAGE ON A SOCKET

AX = 000Fh subfn 0002h
 CX:BX -> request block (see #03447)
 DX = application's data segment (should be = CX)
 ES = DS

Return: AX = status

0000h successful
 00A5h resource not available

Desc: send data to a socket identified by its socket handle

Format of request block:

Offset	Size	Description (Table 03447)
00h	WORD	0002h
02h	WORD	magic number (FEFEh)
04h	WORD	buffer for return code (set to 0 before calling)
06h	WORD	reserved (will be set to client DS)
08h	WORD	reserved (obviously not used; set to 0 before calling)
0Ah	DWORD	far pointer to parameter block (see #03448)
0Eh	WORD	process handle (see AX=000Fh/SF=0008h)

Format of parameter block:

Offset	Size	Description (Table 03448)
00h	WORD	buffer for estimated communication cost in 220 msec ticks

02h WORD buffer for return code (set to 0 before calling)
 04h WORD socket handle
 06h DWORD far pointer to IPC/SPP message buffer
 0Ah WORD length of IPC/SPP message buffer
 0Ch WORD option flags
 bit 0: asynchronous communication
 bit 1: reliable message
 bit 3: end of message
 bit 4: vectored I/O
 0Eh 16 BYTES socket address (IPC port of source)
 1Eh WORD not used (set to 0 before calling)
 20h WORD SPP connection id (0 for IPC messages)
 22h WORD request type (1 = data, 2 = connect SPP, 3 = disconnect SPP)

SeeAlso: #03447

-----N-61000FSF0004-----

INT 61 - Banyan VINES - CLOSE SOCKET

AX = 000Fh subfn 0004h

CX:BX -> request block (see #03449)

DX = application's data segment (should be = CX)

ES = DS

Return: AX = status

 0000h successful

 00A5h resource not available

Desc: Closes a socket identified by its socket handle

Format of request block:

Offset Size Description (Table 03449)

00h WORD 0004h
 02h WORD magic number (FEFEh)
 04h WORD buffer for return code (set to 0 before calling)
 06h WORD reserved (will be set to client DS)
 08h WORD reserved (obviously not used; set to 0 before calling)
 0Ah DWORD far pointer to socket handle (2 bytes)
 0Eh WORD process handle (see AX=000Fh/SF=0008h)

-----N-61000FSF0008-----

INT 61 - Banyan VINES - GET PROCESS HANDLE

AX = 000Fh subfn 0008h

CX:BX -> request block (see #03450)

DX = application's data segment (should be = CX)

ES = DS

Return: AX = status

0000h successful
00A5h resource not available

Desc: Returns a process handle to be used in other request blocks

Notes: This function is a prerequisite to all AX=000Fh subfunctions which expect a process handle in the request block

Although subsequent calls from the same application return different process handles, there seem to be no restrictions on how many times an application may call this function (in fact, the code in the VINES toolkit library always calls it before issuing other AX=000Fh calls)

Format of request block:

Offset Size Description (Table 03450)

00h WORD 0008h
02h WORD magic number (FEFEh)
04h WORD buffer for return code (set to 0 before calling)
06h WORD reserved
08h WORD reserved (obviously not used; set to 0 before calling)
0Ah DWORD process type (1 = transient, 2 = resident; destroyed on exit)
0Eh WORD buffer for process handle

-----N-610010-----

INT 61 - Banyan VINES - INTERFACE TO NBNS

AX = 0010h
other parameters, if any, unknown

-----N-6101-----

INT 61 - Banyan VINES - CHECK SERVICE

AH = 01h
AL = service ID (see #03451)

Return: AX = status

0000h installed
0001h not installed
0002h invalid ID
0003h invalid function

Note: Only the offset part of the service pointer is checked for a nonzero value

SeeAlso: AH=00h,AH=02h"VINES",AH=03h"VINES"

(Table 03451)

Values for VINES service ID:

01h communications
02h primary 3270 emulation
03h async terminal emulation

04h file deflection
05h print service control
06h get address of something
07h StreetTalk access service
08h environment
09h NETBIOS client service
0Ah secondary 3270 emulation
0Bh semaphore service
0Ch 3270 emulation active status
0Dh T3270 keyboard interrupt simulator
0Eh enhanced 3270 service
0Fh comm Windows entry point
10h interface to NBNS

-----N-6101--BX0000-----

INT 61 u - PC/TCP kernel v2.1+ - "net_config" - CONFIGURE RUNNING KERNEL

AH = 01h

BX = 0000h

DH = tag number

DL = device number

DS:SI -> buffer to send to kernel

ES:DI -> integer containing size of buffer

Return: CF clear if successful

CF set on error

AX = error code (see #03319 at INT 61"FTP Software")

Range: INT 20 to INT E0, selected by configuration

Notes: there are a large number of tags available; the items returned
all refer to local kernel configuration, and are not needed in
normal use.

this function is not supported by Beame&Whiteside's BWPCTCP v3.0a shim

SeeAlso: INT 61"FTP Software",INT 61/AH=00h"PC/TCP",INT 61/AH=02h"PC/TCP"

-----N-6102-----

INT 61 - Banyan VINES - GET REVISION NUMBER

AH = 02h

DS:DX -> 2-byte buffer for result

Return: AX = 0000h installed

DS:DX buffer contains revision number as

10000d * major_ver + 100d * minor_ver + patch_revision

SeeAlso: AH=01h"VINES",AH=04h"VINES",AH=07h"VINES"

-----N-6102--BX0000-----

INT 61 u - PC/TCP kernel v2.1+ - "get_kernel_info" - GET MISCELLAN LOCAL INFO

AH = 02h

BX = 0000h
DH = tag number
DL = device number
DS:SI -> buffer for result (up to 48 bytes for version <= 2.2)
ES:DI -> integer containing size of buffer
Return: CF clear if successful
Data loaded into specified buffer, and size value altered
CF set on error
AX = error code (see #03319 at INT 61"FTP Software")
Range: INT 20 to INT E0, selected by configuration
Notes: there are a large number of tags available; the items returned
all refer to local kernel configuration, and are not needed in
normal use.
this function is not supported by Beame&Whiteside's BWPCTCP v3.0a shim
SeeAlso: INT 61"FTP Software",INT 61/AH=00h"PC/TCP",INT 61/AH=01h"PC/TCP"

-----N-6103-----
INT 61 - Banyan VINES - INSTALL SERVICE HANDLER

AH = 03h
AL = service ID (see #03451)
DS:DX -> new service handler
Return: AX = 0000h successful
SeeAlso: AH=00h,AH=01h"VINES",AH=02h"VINES",AH=05h"VINES"

-----N-6104-----
INT 61 - Banyan VINES - CHECK VINES REVISION NUMBER IN RANGE

AH = 04h
DS:DX -> 2-byte buffer containing the VINES revision number
DS:DX buffer contains revision number as
1000d * major_ver + 100d * minor_ver + patch_revision
Return: AX = status

0000h valid version
0004h out of range
Note: For Revision 5.53, the valid range is 4.10(0) to 6.19(99)
SeeAlso: AH=02h"VINES",AH=07h"VINES",AH=08h"VINES"

-----N-6105-----
INT 61 - PC/TCP kernel v2.05+ - "get_addr" - GET INTERNET ADDRESS OF NET DESCR

AH = 05h
BX = network descriptor
Return: CF clear if successful
DX:AX = Internet address of ND
CF set on error
AX = error code (see #03319 at INT 61"FTP Software")

InstallCheck: test for the signature "TCPTSR" three bytes beyond the start

of the interrupt handler

Range: INT 20 to INT E0, selected by configuration

SeeAlso: INT 61"FTP Software",INT 61/AH=00h"PC/TCP",INT 61/AH=16h"PC/TCP"

SeeAlso: INT 63/AH=25h"BW-TCP"

-----N-6105-----

INT 61 - Banyan VINES - READ PCCONFIG BYTES

AH = 05h

CX = number of bytes to copy

(max. 0175h for VINES 4.00, max. 017Fh for VINES 4.10+)

SI = offset of area to be copied (see #03452,#03453)

Return: AX = status

0000h successful

0004h out of range

Notes: The PCCONFIG area is located in the code segment of the BANV handler

To find out which PCCONFIG structure applies, call this function first

with CX=0006h to get the VINES revision number

SeeAlso: AH=02h"VINES",AH=04h"VINES",AH=06h"VINES",AH=0Bh"VINES"

Format of PCCONFIG area (VINES 4.00):

Offset Size Description (Table 03452)

00h 6 BYTES VINES revision (ASCII, zero-padded)

06h 2 BYTES NEWREV-number (ASCII, two digits)

08h WORD hardware interrupt level

0Ah WORD DMA channel

0Ch WORD I/O address

0Eh DWORD adapter ROM address (linear)

12h DWORD adapter RAM address (linear)

16h WORD BANV interrupt

18h 32 BYTES default login group #1

38h 32 BYTES default login group #2

58h 32 BYTES default login group #3

78h WORD semaphore locking support flag

7Ah WORD extended StreetTalk flag

7Ch WORD maximum number of file volumes

7Eh WORD NetBIOS flag

80h 60 BYTES default network adapter

BCh 60 BYTES default communications driver

F8h WORD maximum number of SPP connections

FAh WORD maximum number of open sockets

FCh WORD communications buffer size (10..60 KB)

FEh 20 BYTES location directory on drive Z:
112h 33 BYTES auxiliary configuration area #1 (see #03454)
133h 33 BYTES auxiliary configuration area #2
154h 33 BYTES auxiliary configuration area #3
SeeAlso: #03453

Format of PCCONFIG area (VINES 4.10+):

Offset	Size	Description (Table 03453)
00h	6 BYTES	VINES revision (ASCII, zero-padded)
06h	2 BYTES	NEWREV-number (ASCII, two digits)
08h	WORD	hardware interrupt level
0Ah	WORD	DMA channel
0Ch	WORD	I/O address
0Eh	DWORD	adapter ROM address (linear)
12h	DWORD	adapter RAM address (linear)
16h	WORD	BANV interrupt
18h	32 BYTES	default login group #1
38h	32 BYTES	default login group #2
58h	32 BYTES	default login group #3
78h	WORD	semaphore locking support flag
7Ah	WORD	extended StreetTalk flag
7Ch	WORD	named pipes support flag
7Eh	WORD	maximum number of file volumes
80h	WORD	NetBIOS flag
82h	60 BYTES	default network adapter
BEh	60 BYTES	default communications driver
FAh	WORD	maximum number of SPP connections
FCh	WORD	maximum number of open sockets
FEh	WORD	communications buffer size (10..60 KB)
100h	WORD	number of mailslots
102h	WORD	maximum number of open files
104h	WORD	maximum number of NetBIOS sessions
106h	WORD	maximum number of NetBIOS commands
108h	20 BYTES	location directory on drive Z:
11Ch	33 BYTES	auxiliary configuration area #1 (see #03454)
13Dh	33 BYTES	auxiliary configuration area #2
15Eh	33 BYTES	auxiliary configuration area #3

Format of auxiliary configuration area:

Offset	Size	Description (Table 03454)
00h	16 BYTES	item name (no zero terminator if length = 16)

10h BYTE item value type (1=text, else numeric)
 11h 16 BYTES item value (string if type 1, else DWORD)
 SeeAlso: #03452,#03453
 -----N-6106-----

INT 61 - PC/TCP kernel v2.05 - "net_info" - GET INTERFACE STATISTICS

AH = 06h
 BX = network descriptor (must be allocated and open)
 DS:SI -> 38-byte buffer for interface statistics (see #03455)

Return: CF clear if successful

buffer filled
 CF set on error
 AX = error code (see #03319)

Desc: returns the statistics relevant to the particular network interface
 used by the specified network descriptor

Range: INT 20 to INT E0, selected by configuration

SeeAlso: INT 61"FTP Software",INT 61/AH=00h"PC/TCP",INT 61/AH=05h"PC/TCP"

Format of PC/TCP interface statistics:

Offset	Size	Description (Table 03455)
00h	WORD	interface class (e.g. 802.3)
02h	WORD	type (manufacturer) of interface
04h	WORD	interface number
06h	DWORD	(big-endian) IP address of interface
0Ah	DWORD	subnet mask
0Eh	WORD	0001h if interface is up
10h	DWORD	total packets received
14h	DWORD	total packets sent
18h	DWORD	receive errors
1Ch	DWORD	send errors
20h	WORD	length of local net address (e.g. 0006h for Ethernet)
22h	DWORD	pointer to local net address

-----N-6106-----
 INT 61 - Banyan VINES - GET PCCONFIG BASE ADDRESS
 AH = 06h

DS:DX -> 4-byte buffer for far pointer to PCCONFIG area

Return: AX = 0000h successful

Note: The PCCONFIG area is located in the code segment of the BANV handler

SeeAlso: AH=05h"VINES",AH=0Bh"VINES"

-----N-6107-----
 INT 61 - PC/TCP kernel v2.05+ - "net_globalize" - MAKE NET DESCRIPTOR GLOBAL
 AH = 07h

BX = local network descriptor
Return: CF clear if successful
AX = global network descriptor
CF set on error
AX = error code (see #03319)
Notes: the new network descriptor can be accessed from all processes and is independent of DOS
this function is not supported by Beame&Whiteside's BWPCTCP v3.0a shim
SeeAlso: INT 61"FTP Software",INT 61/AH=00h"PC/TCP",INT 61/AH=08h"PC/TCP"
SeeAlso: INT 61/AH=29h

-----N-6107-----
INT 61 - Banyan VINES - GET VINES REVISION NUMBER (LONG)
AH = 07h
DS:DX -> 4-byte buffer for result
Return: AX = 0000h successful
DS:DX buffer contains revision number as
10000d * major_ver + 100d * minor_ver + patch_revision
SeeAlso: AH=02h"VINES",AH=04h"VINES",AH=08h"VINES"

-----b-6107-----
INT 61 - Atari Portfolio - FORMAT CREDIT CARD MEMORY
AH = 07h
AL = drive number (00h = A:,01h = B:)
Return: CF clear if successful
CF set on error
AH = error code (see #00234 at INT 13/AH=01h)
Note: do not use this function to format the internal disk (drive number 02h)
SeeAlso: AH=00h"Portfolio",AH=08h"Portfolio",AH=09h"Portfolio"

-----N-6108-----
INT 61 - PC/TCP kernel v2.05+ - "net_release" - CLOSE A NETWORK DESCRIPTOR
AH = 08h
BX = network descriptor
Return: CF clear if successful
AX = 0000h (BWPCTCP shim)
CF set on error
AX = error code (see #03319)
Range: INT 20 to INT E0, selected by configuration
Note: the descriptor will be closed and resources released. If a stream descriptor, the protocol (FIN etc) is completed unless the non-blocking option has been set.
SeeAlso: INT 61"FTP Software",INT 61/AH=00h"PC/TCP",INT 61/AH=09h"PC/TCP"
SeeAlso: INT 63/AH=0Eh"BW-TCP"

-----N-6108-----

INT 61 - Banyan VINES - CHECK VINES REVISION NUMBER IN RANGE (LONG)

AH = 08h

DS:DX -> 4-byte buffer containing the VINES revision number

DS:DX buffer contains revision number as

10000d * major_ver + 100d * minor_ver + patch_revision

Return: AX = status

0000h valid version

0004h out of range

Note: For Revision 5.53, the valid range is 4.10(0) to 6.19(99)

SeeAlso: AH=04h"VINES",AH=07h"VINES"

-----b-6108-----

INT 61 - Atari Portfolio - GET SIZE OF INTERNAL DISK

AH = 08h

Return: AX = segment address of RAMdisk

BX = size in KB

SeeAlso: AH=00h"Portfolio",AH=07h"Portfolio",AH=09h"Portfolio"

-----N-6109-----

INT 61 - PC/TCP kernel v2.05+ - "net_releaseall" - CLOSE ALL NON-GLOBAL DESCRS

AH = 09h

Return: CF clear

Range: INT 20 to INT E0, selected by configuration

Notes: this call performs function 08h on every non-global network descriptor.

Global descriptors must be released individually.

SeeAlso: INT 61"FTP Software",INT 61/AH=00h"PC/TCP",INT 61/AH=08h"PC/TCP"

SeeAlso: INT 63/AH=0Eh"BW-TCP"

-----N-6109-----

INT 61 - Banyan VINES - GET VALUE OF AN OS/2 BACKEND VARIABLE

AH = 09h

other parameters, if any, unknown

SeeAlso: AH=0Ah"VINES"

-----b-6109-----

INT 61 - Atari Portfolio - FORMAT INTERNAL DISK

AH = 09h

BX = desired size of disk in KB

Return: CF clear if successful

CF set on error

BX = maximum possible size in KB

Desc: resize and erase the internal RAMdisk, then reboot the system

SeeAlso: AH=00h"Portfolio",AH=07h"Portfolio",AH=08h"Portfolio"

SeeAlso: AH=0Bh"Portfolio"

-----N-610A-----

INT 61 - PC/TCP kernel v2.05 - "net_send" - ???

AH = 0Ah

???

Return: ???

Notes: this function is described as "unused" in the v2.05 and v2.2
documentation

this function is not supported by Beame&Whiteside's BWPCTCP v3.0a shim

SeeAlso: AH=0Bh

-----N-610A-----

INT 61 - Banyan VINES - SET VALUE OF AN OS/2 BACKEND VARIABLE

AH = 0Ah

other parameters, if any, unknown

SeeAlso: AH=09h"VINES"

-----N-610B-----

INT 61 - PC/TCP kernel v2.05 - "net_sendto" - ???

AH = 0Bh

???

Return: ???

Range: INT 20 to INT E0, selected by configuration

Notes: this function is described as "unused" in the v2.05 and v2.2
documentation

this function is not supported by Beame&Whiteside's BWPCTCP v3.0a shim

SeeAlso: AH=0Ah

-----N-610B-----

INT 61 - Banyan VINES - SET PCCONFIG IN BACKEND

AH = 0Bh

other parameters, if any, unknown

SeeAlso: AH=05h"VINES",AH=06h"VINES"

-----b-610B-----

INT 61 - Atari Portfolio - DETERMINE CCM VALIDITY

AH = 0Bh

AL = drive number (00h = A:,01h = B:)

Return: CF clear if successful (card present and correct)

CF set on error

AH = error code (see #00234 at INT 13/AH=01h)

SeeAlso: AH=00h"Portfolio",AH=07h"Portfolio",AH=0Dh"Portfolio"

-----N-610C-----

INT 61 u - PC/TCP kernel v2.05+ - "net_stat" - GET CONNECTION STATISTICS

AH = 0Ch

BX = network descriptor or one of the following:

FFFCh for kernel ICMP statistics (see #03459)
FFFDh for kernel UDP statistics (see #03458)
FFFEh for kernel IP statistics (see #03457)
FFFFh for kernel TCP statistics (see #03456)

DS:DX -> 64-byte buffer

Return: CF clear if successful
buffer filled

CF set on error

AX = error code (see #03319)

Notes: the fields filled in for a network descriptor depend on the protocol
family used by that descriptor's connection

this function is not supported by Beame&Whiteside's BWPCTCP v3.0a shim

SeeAlso: INT 61"FTP Software",INT 61/AH=00h"PC/TCP"

Format of PC/TCP kernel TCP statistics:

Offset Size Description (Table 03456)

00h 16 BYTES unused
10h DWORD bytes sent
14h DWORD bytes received
18h 8 BYTES unused
20h DWORD packets sent
24h DWORD packets received
28h DWORD bad checksums
2Ch DWORD count of window ignored by remote
30h DWORD timeouts
34h DWORD resets
38h DWORD duplicate packets
3Ch DWORD retransmits

SeeAlso: #03457,#03458,#03459

Format of PC/TCP kernel IP statistics:

Offset Size Description (Table 03457)

00h 8 BYTES unused
08h DWORD invalid IP header length errors
0Ch DWORD protocol errors (unwanted packets)
10h DWORD duplicate fragments received
14h DWORD bad fragments received
18h DWORD security errors
1Ch DWORD count of bad IP addresses received
20h DWORD packets sent
24h DWORD packets received

28h DWORD bad checksums received
2Ch DWORD total IP protocol errors
30h DWORD fragmentation errors
34h DWORD IP packets discarded + bad security + bad fragments
38h DWORD fragments received
3Ch 4 BYTES unused
SeeAlso: #03456,#03458,#03459

Format of PC/TCP kernel UDP statistics:

Offset	Size	Description (Table 03458)
00h	28 BYTES	unused
1Ch	DWORD	packets dropped for lack of buffers
20h	DWORD	packets sent
24h	DWORD	packets received
28h	DWORD	bad checksums
2Ch	DWORD	port not listening errors
30h	4 BYTES	unused
34h	DWORD	truncated receives
38h	8 BYTES	unused

SeeAlso: #03456,#03457,#03459

Format of PC/TCP kernel ICMP statistics:

Offset	Size	Description (Table 03459)
00h	DWORD	"TimeEx" sent
04h	DWORD	"TimeEx" received
08h	DWORD	"ParamProb" sent
0Ch	DWORD	"ParamProb" received
10h	DWORD	redirects received
14h	DWORD	source quenches received
18h	DWORD	ICMP Echo Requests ("ping") sent
1Ch	DWORD	ICMP Echo Requests received
20h	DWORD	packets sent
24h	DWORD	packets received
28h	DWORD	bad packets received
2Ch	DWORD	"DestUn" received
30h	DWORD	packet send errors
34h	DWORD	"DestUn" sent
38h	DWORD	ICMP Echo replies received
3Ch	DWORD	ICMP Echo replies sent

SeeAlso: #03456,#03457,#03458

-----N-610D-----

INT 61 - PC/TCP kernel v2.05+ - "is_netnd" - CHECK IF NETWORK DESCRIPTOR VALID

AH = 0Dh

BX = possible network descriptor

Return: CF clear if valid

CF set on error

AX = error code (see #03319 at INT 61"FTP Software")

SeeAlso: INT 61"FTP Software",INT 61/AH=00h"PC/TCP",INT 61/AH=08h"PC/TCP"

SeeAlso: INT 61/AH=22h

-----b-610D-----

INT 61 - Atari Portfolio - GET SCREEN SIZE

AH = 0Dh

Return: AX = physical screen size (AH=rows, AL=columns)

DX = logical screen size (DH=rows, DL=columns)

SeeAlso: AH=00h"Portfolio",AH=0Bh"Portfolio",AH=0Eh"Portfolio"

-----N-610E-----

INT 61 - PC/TCP kernel v2.05+ - "net_select" - DETECT READINESS OF NETWORK

AH = 0Eh

BX = maximum value of network descriptor for which to return info

DS:DX -> 32-bit (max) array of bit flags for read readiness

ES:DI -> 32-bit (max) array of bit flags for write readiness

Return: CF clear

InstallCheck: test for the signature "TCPTSR" three bytes beyond the start
of the interrupt handler

Range: INT 20 to INT E0, selected by configuration

Notes: bits in the DS:DX buffer are set if the corresponding network
descriptor may be read without blocking; bits in the ES:DI buffer
are set if the corresponding network descriptor may be written
without blocking. This implies that the network descriptor has
opened correctly and the protocol initialized.

SeeAlso: INT 61"FTP Software",INT 61/AH=00h"PC/TCP"

-----b-610E-----

INT 61 - Atari Portfolio - GET/SET SCREEN MODE

AH = 0Eh

AL = function

00h get mode

01h set mode

DL = new mode (see #03460)

Return: DL = current/old mode

Note: if changing the mode places the cursor position or virtual screen
origin outside the displayed area, the screen origin will be reset
to (0,0), the screen cleared, and the cursor position set to (0,0)

SeeAlso: AH=00h"Portfolio",AH=0Dh"Portfolio",AH=0Fh"Portfolio",INT 10/AH=00h
<http://www.foxitsoftware.com> For evaluation only.

Bitfields for Atari Portfolio screen mode:

Bit(s) Description (Table 03460)

0 40x8 mode instead of 80x25
1 tracked mode
7 graphics

Note: the above bits are mutually exclusive

-----N-610F-----

INT 61 - PC/TCP kernel v2.05+ - "get_netversion" - GET SOFTWARE VERSION

AH = 0Fh

Return: CF clear

AX = version (AH = major, AL = minor)

BX = patch level

Notes: patch levels are no longer used starting with version 2.10; instead,
the minor version level is incremented.

Beame&Whiteside's BWPCTCP v3.0a shim reports version 2.05, but does not
set BX

SeeAlso: INT 61"FTP Software",INT 61/AH=00h"PC/TCP"

-----b-610F-----

INT 61 - Atari Portfolio - GET/SET CURSOR MODE

AH = 0Fh

AL = function

00h get mode

01h set mode

BL = new cursor mode

00h cursor off

01h underline

02h block

02h force mode (set size to reflect NumLock state)

Return: BL = current/old cursor mode

SeeAlso: AH=00h"Portfolio",AH=0Eh"Portfolio",AH=10h"Portfolio",INT 10/AH=01h

-----N-6110-----

INT 61 - PC/TCP kernel v2.05+ - "net_shutdown" - UNINSTALL

AH = 10h

Return: CF clear if successful

CF set on error

AX = error code (see #03319 at INT 61"FTP Software")

Range: INT 20 to INT E0, selected by configuration

Note: this function is not supported by Beame&Whiteside's BWPCTCP v3.0a shim

SeeAlso: INT 61"FTP Software",INT 61/AH=00h"PC/TCP"

-----b-6110-----

INT 61 - Atari Portfolio - GET/SET VIRTUAL SCREEN POSITION

AH = 10h

AL = function

00h get position

Return: DH,DL = current row,column

01h set position

DH = row

DL = column

Desc: get or specify the upper-left corner of the 40x8 window into the
logical screen

SeeAlso: AH=00h"Portfolio",AH=0Eh"Portfolio",AH=0Fh"Portfolio"

SeeAlso: AH=11h"Portfolio"

-----N-6111-----

INT 61 - PC/TCP kernel v2.05+ - "disable_async" - DISABLE ASYNCHRONOUS HANDLERS

AH = 11h

Return: CF clear

AX = previous state

0000h async calls were already disabled

else async calls were enabled

SeeAlso: INT 61"FTP Software",INT 61/AH=12h

-----b-6111-----

INT 61 - Atari Portfolio - MOVE VIRTUAL SCREEN ORIGIN

AH = 11h

AL = number of lines to move

DL = direction of move (1=up, 2=down, 3=left, 4=right)

SeeAlso: AH=00h"Portfolio",AH=10h"Portfolio",AH=12h"Portfolio"

-----N-6112-----

INT 61 - PC/TCP kernel v2.05+ - "enable_async" - ENABLE ASYNCHRONOUS HANDLERS

AH = 12h

Return: CF clear

AX = previous state

0000h async calls were disabled

else async calls were already enabled

SeeAlso: INT 61"FTP Software",INT 61/AH=11h

-----b-6112-----

INT 61 - Atari Portfolio - SCREEN REFRESH

AH = 12h

Return: nothing

Desc: copy contents of video RAM to LCD controller

SeeAlso: AH=10h"Portfolio",AH=11h"Portfolio",AH=15h"Portfolio"

-----N-6113-----

INT 61 - PC/TCP kernel v2.05 - "net_connect" - OPEN A NETWORK CONNECTION

AH = 13h

BX = network descriptor (FFFFh for automatic net_getdesc)

DX = protocol (see #03461)

DS:SI -> buffer for "addr" structure (see #03462)

Return: CF clear if successful

AX = network descriptor used or allocated

CF set on error

AX = error code (see #03319 at INT 61"FTP Software")

Notes: resets the connection with AH=19h in various cases

will wait for protocol on stream connections unless non-blocking was

set with AH=20h

SeeAlso: INT 61"FTP Software",INT 61/AH=00h"PC/TCP",INT 61/AH=18h,INT 61/AH=23h

SeeAlso: INT 63/AH=14h"BW-TCP"

(Table 03461)

Values for PC/TCP protocol:

0001h raw net (undocumented)

0002h raw IP

0003h datagram (UDP)

0004h stream (TCP)

0005h raw ICMP

Format of structure "addr":

Offset Size Description (Table 03462)

00h DWORD Internet address (network order)

04h WORD remote socket number (network order)

06h WORD local socket number (network order) 0000h means "you choose"

08h BYTE protocol (see #03461)

-----N-6114-----

INT 61 Ou - PC/TCP kernel v2.05 - "net_recv" - NO LONGER SUPPORTED

AH = 14h

BX = network descriptor

Return: CF set

AX = 0018h (see #03319 at INT 61"FTP Software")

Notes: displays error message "Illegal system call! Please upgrade your

PCserver software" to standard output

this function is not supported by Beame&Whiteside's BWPCTCP v3.0a shim

SeeAlso: INT 61"FTP Software",INT 61/AH=00h"PC/TCP",INT 61/AH=0Ah,INT 61/AH=15h

-----N-6115-----

INT 61 - PC/TCP kernel v2.05 - "net_recvfrom" - ???

AH = 15h

???

Return: ???

Notes: this function is described as "unused" in the v2.2 documentation

this function is not supported by Beame&Whiteside's BWPCTCP v3.0a shim

SeeAlso: INT 61/AH=0Bh,INT 61/AH=14h

-----b-6115-----

INT 61 - Atari Portfolio - SOUND GENERATION

AH = 15h

AL = function

00h key click

01h beep

02h alarm

Return: nothing

SeeAlso: AH=00h"Portfolio",AH=12h"Portfolio",AH=16h"Portfolio"

-----N-6116-----

INT 61 - PC/TCP kernel v2.05+ - "net_peer" - GET DATA ON REMOTE PEER

AH = 16h

BX = network descriptor

DS:DX -> 9-byte buffer for "addr" structure (see #03462)

Return: CF clear if successful

buffer filled

CF set on error

AX = error code (see #03319 at INT 61"FTP Software")

Range: INT 20 to INT E0, selected by configuration

SeeAlso: INT 61"FTP Software",INT 61/AH=00h"PC/TCP"

-----b-6116-----

INT 61 - Atari Portfolio - MELODY TONE GENERATOR

AH = 16h

CX = length of tones in 1/100 sec

DL = tone code (see #03463)

SeeAlso: AH=15h"Portfolio",AH=17h"Portfolio"

(Table 03463)

Values for Atari Portfolio tone code:

04h A6#01864.7 Hz

05h B6 1975.5 Hz

06h D7 2348.3 Hz

07h D7# 2489.0 Hz

0Eh F6# 1480.0 Hz

25h C7 2093.0 Hz
29h D6 1174.7 Hz
2Ch G6# 1661.2 Hz
2Fh C7# 2217.5 Hz
30h D5# 622.3 Hz
31h E5 659.3 Hz
32h F5 698.5 Hz
33h F5# 740.0 Hz
34h G5 784.0 Hz
35h G5# 830.6 Hz
36h A5 880.6 Hz
37h A5# 932.3 Hz
38h B5 987.8 Hz
39h C6 1046.5 Hz
3Ah C6# 1108.7 Hz
3Bh D6# 1244.5 Hz
3Ch E6 1318.5 Hz
3Dh F6 1396.9 Hz
3Eh G6 1568.9 Hz
3Fh A6 1760.0 Hz

-----N-6117-----

INT 61 - PC/TCP kernel v2.05+ - "net_reconfig" - RE-READ KERNEL CONFIGURATION

AH = 17h

Return: CF clear if successful

CF set on error

AX = error code (see #03319 at INT 61"FTP Software")

Notes: this routine is deprecated in v2.1+ and will eventually be withdrawn;

in v2.1+, this function calls AH=01h which should be used instead

this function is not supported by Beame&Whiteside's BWPCTCP v3.0a shim

SeeAlso: INT 61"FTP Software",INT 61/AH=00h"PC/TCP",INT 61/AH=01h"PC/TCP"

-----b-6117-----

INT 61 - Atari Portfolio - DIAL NUMBER

AH = 17h

DS:SI -> ASCII string containing number to dial

CX = length of string

Note: valid characters for string are "0123456789ABCD*#"

SeeAlso: AH=00h"Portfolio",AH=15h"Portfolio",AH=16h"Portfolio"

SeeAlso: AH=18h"Portfolio"

-----N-6118-----

INT 61 - PC/TCP kernel v2.05+ - "net_eof" - CLOSE TRANSMIT SIDE OF CONNECTION

AH = 18h

BX = network descriptor
Return: CF clear if successful
CF set on error
AX = error code (see #03319 at INT 61"FTP Software")
Note: a TCP "FIN" command is sent and no further data may be transmitted,
although the connection remains open
SeeAlso: INT 61"FTP Software",INT 61/AH=00h"PC/TCP",INT 61/AH=13h"PC/TCP"
SeeAlso: INT 63/AH=0Eh"EW-TCP"

-----b-6118-----

INT 61 - Atari Portfolio - MUTE STATES

AH = 18h
AL = function
00h get mute state
01h set mute state
02h get key click state
03h set key click state
04h get beep state
05h set beep state
06h get alarm state
07h set alarm state
08h get DTMF duration
09h set DTMF duration
DL = new state ("set" functions) (00h off [muted], 01h on)
Return: DL = current state ("get" functions) (00h off [muted], 01h on)
SeeAlso: AH=15h"Portfolio",AH=16h"Portfolio",AH=17h"Portfolio"
SeeAlso: AH=19h"Portfolio"

-----N-6119-----

INT 61 - PC/TCP kernel v2.05+ - "net_abort" - RESET A NETWORK CONNECTION

AH = 19h
BX = network descriptor
Return: CF clear if successful
CF set on error
AX = error code (see #03319 at INT 61"FTP Software")
Desc: immediately destroys the specified connection
Note: send a TCP "RST" command if a stream connection is open
SeeAlso: INT 61"FTP Software",INT 61/AH=00h"PC/TCP",INT 61/AH=24h

-----b-6119-----

INT 61 - Atari Portfolio - GET SERIAL PORT PARAMETERS

AH = 19h
DX = serial port number
Return: AH = status

00h successful

AL = composite parameters (see #00300 at INT 14/AH=00h"SERIAL")

other error

SeeAlso: AH=00h"Portfolio",AH=15h"Portfolio",AH=1Ah"Portfolio"

-----N-611A-----

INT 61 - PC/TCP kernel v2.05+ - "net_write" - WRITE TO THE NETWORK

AH = 1Ah

BX = network descriptor

CX = number of bytes to transmit (0000h allowed)

DX = send options (see #03464)

DS:SI -> data to be written

Return: CF clear if successful

AX = number of bytes actually written

DX = ???

CF set on error

AX = error code (see #03319 at INT 61"FTP Software")

InstallCheck: test for the signature "TCPTSR" three bytes beyond the start
of the interrupt handler

Range: INT 20h to INT E0h, selected by configuration

SeeAlso: INT 61"FTP Software",INT 61/AH=00h"PC/TCP",INT 61/AH=1Bh,INT 61/AH=1Ch

SeeAlso: INT 63/AH=19h"BW-TCP",INT 63/AH=1Bh"BW-TCP"

Bitfields for PC/TCP send options:

Bit(s) Description (Table 03464)

0 signal "URG"ent data

3 attempt rerouting on non-stream calls if first attempt fails

4 send data with PUSH flag (no override of Nagle)

5 fail rather than truncating datagram

6 fail rather than blocking

7 broadcast packet

-----b-611A-----

INT 61 - Atari Portfolio - GET PERIPHERAL ID BYTE

AH = 1Ah

Return: AH = peripheral ID byte

AL = 00h if no peripheral installed

Desc: get the ID code for the current terminating peripheral

SeeAlso: AH=00h"Portfolio",AH=18h"Portfolio",AH=1Bh"Portfolio"

-----N-611B-----

INT 61 - PC/TCP kernel v2.05+ - "net_read" - READ FROM THE NETWORK

AH = 1Bh

BX = network descriptor

CX = maximum number of bytes to read
DX = receive options (see #03465)
DS:SI -> buffer for data
ES:DI -> "addr" structure (see #03462) for remote from which to read
 0000h:0000h for any
Return: CF clear if successful
 AX = number of bytes actually read
 DX = ???
CF set on error
 AX = error code (see #03319 at INT 61"FTP Software")
SeeAlso: INT 61"FTP Software",INT 61/AH=00h"PC/TCP",INT 61/AH=1Ah,INT 61/AH=1Dh

Bitfields for PC/TCP receive options:

Bit(s) Description (Table 03465)

- 1 do not remove data from queue, just copy it
- 2 do not copy data, just remove it from queue
- 5 fail if datagram would be truncated
- 6 do not block, return error if no data available

Note: special case for UDP: if both 1 and 2 set, return number of datagrams

-----b-611B-----

INT 61 - Atari Portfolio - SET PERIPHERAL ID BYTE

AH = 1Bh
AL = which peripheral to set
 00h serial
 01h parallel
DL = peripheral ID

Return: nothing

Desc: specify that a peripheral is compatible with the BIOS serial or
 parallel services

SeeAlso: AH=1Ah"Portfolio",AH=1Ch"Portfolio"

-----N-611C-----

INT 61 - PC/TCP kernel v2.05+ - "net_writeto" - WRITE A DATAGRAM

AH = 1Ch
BX = network descriptor
CX = number of bytes to transmit (0000h allowed)
DX = send options (see #03464)
DS:SI -> data to be written
ES:DI -> "addr" structure (see #03462)

Return: CF clear if successful

 AX = number of bytes actually written
 DX = ???

CF set on error

AX = error code (see #03319 at INT 61"FTP Software")

Note: this function differs from AH=1Ah in that the address and socket numbers can be overridden

SeeAlso: INT 61"FTP Software",INT 61/AH=00h"PC/TCP",INT 61/AH=1Ah,INT 61/AH=1Dh

SeeAlso: INT 63/AH=21h"BW-TCP"

-----b-611C-----

INT 61 - Atari Portfolio - PRESET/GET PERIPHERAL DATA

AH = 1Ch

AL = function

00h preset data values

BL = data value

DX = I/O address

01h get data values

BH = table entry number (04h-09h)

Return: if function 01h

BL = data value

DX = I/O address

Desc: preset peripheral I/O data in a table mapping I/O addresses to data values; function 00h additionally outputs the data value to the I/O port

SeeAlso: AH=1Ah"Portfolio",AH=1Bh"Portfolio",AH=1Eh"Portfolio"

-----N-611D-----

INT 61 - PC/TCP kernel v2.05+ - "net_readfrom" - READ A DATAGRAM

AH = 1Dh

BX = network descriptor

CX = maximum number of bytes to read

DX = receive options (see #03464)

DS:SI -> buffer for received data

ES:DI -> 9-byte buffer containing "addr" structure (see #03462)

Return: CF clear if successful

AX = number of bytes read

DX = ???

CF set on error

AX = error code (see #03319 at INT 61"FTP Software")

Notes: this function can read from any host or a designated host depending on settings in the "addr" structure

only for use with datagram or Raw descriptors

SeeAlso: INT 61"FTP Software",INT 61/AH=00h"PC/TCP",INT 61/AH=1Bh,INT 61/AH=1Ch

-----N-611E-----

INT 61 - PC/TCP kernel v2.05+ - "net_flush" - FLUSH PENDING DATA

AH = 1Eh
BX = network descriptor
Return: CF clear if successful
CF set on error
AX = error code (see #03319 at INT 61"FTP Software")
Range: INT 20 to INT E0, selected by configuration
Notes: buffered data is transmitted immediately, overriding Nagle's algorithm
if necessary

this function is not supported by Beame&Whiteside's BWPCTCP v3.0a shim

SeeAlso: INT 61"FTP Software",INT 61/AH=00h"PC/TCP"

-----b-611E-----

INT 61 - Atari Portfolio - GET/SET CLOCK TICK SPEED

AH = 1Eh
AL = function
00h get speed
01h set speed
BX = clock tick speed
0000h tick every 128 seconds
0001h tick every second

Return: if function 00h
BX = clock tick speed

Note: the faster tick speed also uses more power

SeeAlso: AH=00h"Portfolio",AH=1Ch"Portfolio",AH=1Fh"Portfolio"

-----N-611F-----

INT 61 - PC/TCP kernel v2.05+ - "net_async" - SET UP ASYNCHRONOUS CALL-BACK

AH = 1Fh
BX = network descriptor
CX = event type (see #03466)
DS:SI -> event handler routine (see #03467)
ES:DI = 32-bit hint passed to handler

Return: CF clear if successful
DS:DX -> previous handler

CF set on error
AX = error code (see #03319 at INT 61"FTP Software")

Note: ICMP messages do not trigger events on stream connections

SeeAlso: INT 61"FTP Software",INT 61/AH=00h"PC/TCP"

(Table 03466)

Values for PC/TCP event type:

00h alarm
01h open (successfully opened stream connection)

02h receive (data available)
 03h transmit (ACK received on stream connection)
 04h transmit flush???
 05h foreign close (remote host closed data connection)
 06h close (local host closed connection and protocol is complete)
 07h error (error code passed to handler as arg)

(Table 03467)

Values PC/TCP user event handler is called with:

BX = network descriptor
 CX = event type (see #03466)
 DS:DX -> arg
 ES:DI = 32-bit hint value
 STACK: small stack, possibly the DOS stack

-----b-611F-----

INT 61 - Atari Portfolio - GET KEY/NMI-INVOKED SCREEN REFRESH

AH = 1Fh
 AL = function
 00h get state
 01h set state
 DH = new refresh-on-NMI state
 (00h disabled, 01h enabled, bit 7 set = don't change)
 DL = new refresh-on-key state

Return: DH,DL = current/old states

SeeAlso: AH=00h"Portfolio",AH=1Eh"Portfolio",AH=20h"Portfolio"

-----N-6120-----

INT 61 - PC/TCP kernel v2.05+ - "set_option" - SET AN OPTION ON A DESCRIPTOR

AH = 20h
 BX = network descriptor
 CX = length of buffer (usually 04h)
 DS:DX -> buffer containing option
 SI = (ignored by v2.2-)
 DI = option to set (see #03468)

Return: CF clear if successful

CF set on error
 AX = error code (see #03319 at INT 61"FTP Software")

Notes: Beame&Whiteside's BWPCTCP shim only supports options 01h and 0Bh; all others return CF clear

v2.11 reportedly stores the actual 32-bit value of an option in DS:DX rather than using a buffer in memory

SeeAlso: INT 61"FTP Software",INT 61/AH=00h"PC/TCP",INT 61/AH=21h

(Table 03468)

Values for PC/TCP option to get or set:

01h set non-blocking mode if non-zero
02h timeout of call in milliseconds
03h user-defined 4-byte magic cookie (not used by kernel)
04h TCP window or UDP buffer count (WORD, unsigned)
06h do TCP keep-alives if non-zero
09h (v2.1+) set IP precedence
0Ah (v2.1+) set IP type of service
0Bh use a privileged port if port = 0
0Ch turn off TCP PUSH bit and don't flush buffer every write (see AH=1Ah)

Note: use "C" true or false values for boolean options

-----b-6120-----

INT 61 - Atari Portfolio - GET/SET INT 09 REVECTORING

AH = 20h

AL = function

00h get INT 09 revectoring state

01h set state

DL = new state (00h disable, 01h enable)

Return: if function 00h

DL = new state (00h disabled, 01h enabled)

Desc: specify whether INT 09 is automatically revectored to the BIOS,
preventing applications from using their own INT 09 handler on the
incompatible keyboard hardware

SeeAlso: AH=1Eh"Portfolio",AH=1Fh"Portfolio",AH=24h"Portfolio"

-----N-6121-----

INT 61 - PC/TCP kernel v2.05+ - "get_option" - GET OPTIONS APPLIED TO NET DESCR

AH = 21h

BX = network descriptor

CX = length of buffer

DS:DX -> buffer for return values

SI = 0004h (ignored by v2.2-)

DI = option (see #03468)

Return: CF clear if successful

DS:DX -> value (usually 32 bits) returned by selected option

CF set on error

AX = error code (see #03319 at INT 61"FTP Software")

Notes: this function is not supported by Beame&Whiteside's BWPCTCP v3.0a shim
v2.11 reportedly stores the actual 32-bit value of an option in DS:DX
rather than using a buffer in memory

SeeAlso: INT 61"FTP Software",INT 61/AH=00h"PC/TCP",INT 61/AH=20h

-----N-6122-----

INT 61 u - PC/TCP kernel v2.05+ - "net_getdesc" - ALLOCATE NETWORK DESCRIPTOR

AH = 22h

Return: CF clear if successful

AX = network descriptor

CF set on error

AX = error code (see #03319 at INT 61"FTP Software")

Note: the descriptor will be an integer in the range 00h-1Fh, and a DOS call is made to allocate this as a file descriptor. Hence a number may not refer to a PC/TCP network descriptor and a DOS file handle simultaneously.

SeeAlso: INT 61"FTP Software",INT 61/AH=00h"PC/TCP",INT 61/AH=13h,INT 61/AH=29h

-----N-6123-----

INT 61 - PC/TCP kernel v2.05+ - "net_listen" - LISTEN FOR INCOMING CONNECTIONS

AH = 23h

BX = network descriptor or FFFFh to allocate descriptor

DX = type of service

DS:SI -> "addr" structure (see #03462)

Return: CF clear if successful

AX = network descriptor

CF set on error

AX = error code (see #03319 at INT 61"FTP Software")

Notes: type of service and "addr" structure are as for AH=13h
any of the address structure can be zero; normally local socket number is filled in prior to call

SeeAlso: INT 61"FTP Software",INT 61/AH=00h"PC/TCP",INT 61/AH=13h

SeeAlso: INT 63/AH=12h"BW-TCP"

-----N-6124-----

INT 61 - PC/TCP kernel v2.05+ - "net_abortall" - RESET ALL NETWORK CONNECTIONS

AH = 24h

Return: always successful

Note: performs "net_abort" (AH=19h) on all open non-global descriptors

SeeAlso: INT 61"FTP Software",INT 61/AH=00h"PC/TCP",INT 61/AH=19h

-----b-6124-----

INT 61 - Atari Portfolio - GET/SET ROM/CCM SPACE STATE

AH = 24h

AL = function

00h get state

01h set state

DL = new ROM state (see #03469)

DH = new CCM state (see #03470)
Return: CF clear if successful
DL = current/old ROM state
DH = current/old CCM state
CF set on error (invalid option or other error)
Desc: control whether memory cards or an extension ROM are visible in the
range 000C0000h to 000DFFFFh
SeeAlso: AH=00h"Portfolio",AH=20h"Portfolio",AH=26h"Portfolio"

(Table 03469)

Values for Atari Portfolio ROM space state:

00h normal application ROM
01h CCM drive A:
02h CCM drive B:
03h expansion ROM

SeeAlso: #03470

(Table 03470)

Values for Atari Portfolio CCM space state:

00h CCM drives disabled
01h CCM drive A: permanently enabled
02h CCM drive B: permanently enabled

SeeAlso: #03469

-----N-6125-----

INT 61 - PC/TCP kernel v2.05+ - "ad_res_name" - GET HOST NAME GIVEN ADDRESS

AH = 25h
DX,BX = IP address in network order
CX = length of buffer for name
DS:SI -> buffer for host name
Return: CF clear if successful
buffer filled with ASCIZ host name
CF set on error

AX = error code (see #03319 at INT 61"FTP Software")

Note: this function will use the host table and or DNS to resolve the
address, depending on kernel configuration. Use this call for the
normal gethostbyaddr function.

SeeAlso: INT 61"FTP Software",INT 61/AH=00h"PC/TCP",INT 61/AH=26h,INT 61/AH=27h

-----N-6126-----

INT 61 - PC/TCP kernel v2.05+ - "ad_htable" - GET HOST NAME FROM LOCAL TABLE

AH = 26h
DX,BX = IP address in network order

CX = length of buffer
DS:SI -> buffer for host name
Return: CF clear if successful
 buffer filled with ASCIZ host name
CF set on error
 AX = error code (see #03319 at INT 61"FTP Software")
Notes: normally one would use AH=25h instead of this function
 this function is not supported by Beame&Whiteside's BWPCTCP v3.0a shim
SeeAlso: INT 61"FTP Software",INT 61/AH=00h"PC/TCP",INT 61/AH=25h,INT 61/AH=27h

-----b-6126-----
INT 61 - Atari Portfolio - GET/SET POWER CONTROL

 AH = 26h
 AL = function
 00h get state
 01h set state
 DL = new state
 00h normal power-down on low battery
 01h no power-down, but display warning
 02h no power-down, no warning on low battery

Return: AL = current/old state
SeeAlso: AH=00h"Portfolio",AH=24h"Portfolio",AH=28h"Portfolio"

-----N-6127-----
INT 61 - PC/TCP kernel v2.05+ - "ad_domain" - GET HOST NAME FROM DNS

 AH = 27h
 DX,BX = IP address in network order
 CX = length of buffer
 DS:SI -> buffer for host name
Return: CF clear if successful
 buffer filled with ASCIZ host name
CF set on error
 AX = error code (see #03319 at INT 61"FTP Software")
Note: normally one would use AH=25h instead of this function
SeeAlso: INT 61"FTP Software",INT 61/AH=00h"PC/TCP",INT 61/AH=25h,INT 61/AH=26h

-----N-6128-----
INT 61 - PC/TCP kernel v2.05+ - "net_swap" - EXCHANGE TWO NETWORK DESCRIPTORS

 AH = 28h
 BX = network descriptor 1
 CX = network descriptor 2
Return: CF clear if successful
CF set on error
 AX = error code (see #03319 at INT 61"FTP Software")

Note: the two descriptors will exchange places; both must be local or both

global

SeeAlso: INT 61"FTP Software",INT 61/AH=00h"PC/TCP"

-----b-6128-----

INT 61 - Atari Portfolio - GET/SET TEXT/KEYBOARD LANGUAGE

AH = 28h

AL = function

00h get languages

01h set languages

DH = new text language (set bit 7 to leave unchanged)

DL = new keyboard language (set bit 7 to leave unchanged)

03h get language table pointers

Return: if function 00h or 01h:

DH = current/old text language

DL = current/old keyboard language

if function 03h

ES:CX -> keyboard table (see #03471)

ES:DX -> language table (see #03471)

SeeAlso: AH=24h"Portfolio",AH=26h"Portfolio",AH=2Ch"Portfolio"

Format of Atari Portfolio language table:

Offset Size Description (Table 03471)

00h BYTE number of languages supported

01h N BYTES supported languages

00h English

01h French

02h German

03h Spanish

04h Italian

05h Swedish

06h Danish

-----N-6129-----

INT 61 - PC/TCP kernel v2.05+ - "net_getglobdesc" - ALLOCATE GLOBAL DESCRIPTOR

AH = 29h

Return: CF clear if successful

AX = network descriptor

CF set on error

AX = error code (see #03319 at INT 61"FTP Software")

Note: use this function rather than AH=22h to avoid a DOS call by the PC/TCP

kernel; the returned descriptor will be >= 40h and cannot be used

with "net_select" (AH=0Eh)

SeeAlso: INT 61"FTP Software",INT 61/AH=00h"PC/TCP",INT 61/AH=07h"PC/TCP"

SeeAlso: INT 61/AH=22h

-----N-612A-----

INT 61 - PC/TCP kernel v2.05+ - GET CONFIGURATION INFORMATION

AH = 2Ah

DS:SI -> 26-byte buffer for configuration information (see #03472)

Return: CF clear

AX = 0000h

buffer filled

Notes: size of buffer may vary with kernel version; 26 bytes is the size for
versions 2.05 through 2.2

this function is not supported by Beame&Whiteside's BWPCTCP v3.0a shim

SeeAlso: INT 61"FTP Software",INT 61/AH=00h"PC/TCP"

Format of PC/TCP configuration information:

Offset Size Description (Table 03472)

00h	BYTE	maximum TCP connections available
01h	BYTE	maximum UDP connections available
02h	BYTE	maximum IP connections available
03h	BYTE	maximum Raw Net connections available
04h	BYTE	number of TCP connections currently in use
05h	BYTE	number of UDP connections currently in use
06h	BYTE	number of IP connections currently in use
07h	BYTE	number of Raw Net connections currently in use
08h	WORD	number of local network descriptors active
0Ah	WORD	number of global network descriptors active
0Ch	BYTE	maximum header size on network
0Dh	BYTE	maximum trailer size on network
0Eh	WORD	size of large packet buffer
10h	WORD	number of network interfaces attached
12h	DWORD	milliseconds since kernel started
16h	DWORD	IP broadcast address

-----N-612B-----

INT 61 - PC/TCP kernel v2.02+ - "net_alarm" - SET TIMED ASYNCHRONOUS EVENT

AH = 2Bh

BX = network descriptor

CX,DX = time before alarm in milliseconds

DS:SI -> handler which will receive call (see #03467 at INT 61/AH=1Fh)

ES:DI = 32-bit cookie passed to handler

Return: CF clear if successful

CF set on error

AX = error code (see #03319 at INT 61"FTP Software")

Notes: this function will cause a NET_AS_ALARM to be generated; it is intended for TSRs, etc. to regain control periodically

this function is not supported by Beame&Whiteside's BWPCTCP v3.0a shim

SeeAlso: INT 61"FTP Software",INT 61/AH=00h"PC/TCP"

-----b-612C-----

INT 61 - Atari Portfolio - GET VERSION

AH = 2Ch

Return: DS:BX -> BIOS version string ('\$'-terminated)

SeeAlso: AH=28h"Portfolio",AH=2Dh"Portfolio"

-----b-612D-----

INT 61 - Atari Portfolio - TURN SYSTEM OFF

AH = 2Dh

Return: nothing

SeeAlso: AH=00h"Portfolio",AH=2Ch"Portfolio",AH=2Eh"Portfolio"

-----b-612E-----

INT 61 - Atari Portfolio - ENABLE/DISABLE SYSTEM STATUS LINE

AH = 2Eh

AL = new state

00h disabled

01h enabled

DH,DL = row,column at which to display

Return: nothing

SeeAlso: AH=00h"Portfolio",AH=2Dh"Portfolio",AH=30h"Portfolio"

-----N-6130-----

INT 61 - PC/TCP kernel v2.05+ - "icmp_ping" - SEND ICMP ECHO REQUEST (PING)

AH = 30h

BX,DX = IP address of host

CX = length of data to send

Return: CF clear if successful (i.e. reply received)

CF set on error

AX = error code (see #03319 at INT 61"FTP Software")

Note: this function is not supported by Beame&Whiteside's BWPCTCP v3.0a shim

SeeAlso: INT 61"FTP Software",INT 61/AH=00h"PC/TCP"

-----b-6130-----

INT 61 - Atari Portfolio - FILE TRANSFER SERVICES

AH = 30h

AL = function

00h transmit block

CX = number of bytes to send

01h receive block

```
CX = buffer size
  02h open ports
  03h close ports
  04h wait 500ms
DS:DX -> start of data buffer
Return: DL = status
  00h successful
  01h buffer too small
  02h transmission timeout
  03h checksum failure
  04h invalid subfunction
  05h peripheral not installed
if function 01h:
  CX = number of bytes received
SeeAlso: AH=00h"Portfolio",AH=2Eh"Portfolio"
-----N-61-----
INT 61 u - PC/TCP kernel v2.05 - NOP for SLIP kernel
  AH = function
    31h "net_add_route"
    32h "net_del_route"
    33h "net_dump_routes"
Notes: these functions are described as "unused" in the v2.2 documentation
  router configuration can be altered using INT 61/AH=01h
  this function is not supported by Beame&Whiteside's BWPCTCP v3.0a shim
-----N-6132-----
INT 61 U - PC/TCP kernel v2.3 - GET ???
  AH = 32h
Return: AX = ???
Note: this function is fully re-entrant, and may be called even when another
  PC/TCP kernel call is already in progress
-----N-6134-----
INT 61 U - PC/TCP kernel v2.1+ - "icmp_destun" - ???
  AH = 34h
  ???
Return: ???
Notes: this function is described as "reserved" in the v2.2 documentation
  this function is not supported by Beame&Whiteside's BWPCTCP v3.0a shim
-----N-6150-----
INT 61 - PC/TCP kernel v2.05+ - "nm_prs_addr" - TRANSLATE NUMERICAL IP ADDRESS
  AH = 50h
  DS:DX -> ASCIIZ IP address as "dotted quad" (max 127 chars)
```

Return: CF clear if successful

DX:AX -> IP address

CF set on error

AX = error code (see #03319 at INT 61"FTP Software")

Range: INT 20 to INT E0, selected by configuration

SeeAlso: INT 61"FTP Software",INT 61/AH=00h"PC/TCP",INT 61/AH=54h

-----N-6151-----

INT 61 - PC/TCP kernel v2.05+ - "nm_hhtable" - RESOLVE NAME USING HOST TABLE

AH = 51h

CX = size of destination buffer

DS:DX -> ASCIZ host name (max 127 chars)

ES:DI -> destination buffer or 0000h:0000h

Return: CF clear if successful

DX:AX -> IP address of host

destination buffer filled with canonical host name

CF set on error

AX = error code (see #03319 at INT 61"FTP Software")

Note: this function calls DOS, and can fail if the DOS call fails

SeeAlso: INT 61"FTP Software",INT 61/AH=00h"PC/TCP",INT 61/AH=54h

-----N-6152-----

INT 61 - PC/TCP kernel v2.05+ - "nm_domain" - RESOLVE NAME USING DNS

AH = 52h

CX = size of destination buffer

DS:DX -> ASCIZ host name (max 127 chars)

ES:DI -> destination buffer or 0000h:0000h

Return: CF clear if successful

DX:AX -> IP address of host

destination buffer filled with canonical host name

CF set on error

AX = error code (see #03319 at INT 61"FTP Software")

Note: this function will poll all configured domain name servers if necessary

SeeAlso: INT 61"FTP Software",INT 61/AH=00h"PC/TCP",INT 61/AH=54h

-----N-6153-----

INT 61 - PC/TCP kernel v2.05- - "nm_ien116" - RESOLVE HOST NAME USING IEN116

AH = 53h

DS:DX -> ASCIZ name to be resolved (max 127 chars)

Return: CF clear if successful

DX:AX -> IP address of host

CF set on error

AX = error code (see #03319 at INT 61"FTP Software")

Range: INT 20 to INT E0, selected by configuration

Note: this function is not supported by v2.10+

SeeAlso: INT 61"FTP Software",INT 61/AH=00h"PC/TCP",INT 61/AH=54h

-----N-6154-----

INT 61 - PC/TCP kernel v2.05+ - "nm_res_name" - RESOLVE HOST NAME

AH = 54h

CX = size of destination buffer

DS:DX -> ASCIZ host name (max 127 chars)

ES:DI -> destination buffer or 0000h:0000h

Return: CF clear if successful

DX:AX -> IP address of host

destination buffer filled with canonical host name

CF set on error

AX = error code (see #03319 at INT 61"FTP Software")

Note: this function uses all configured methods in turn to resolve the

name (numerical, then host table, then DNS, then IEN116)

BUG: the SLIP kernel for v2.05 bounds-checks the wrong register, so values

greater than 54h in AH may crash the system. Other kernels may have

this bug as well; it has been fixed in the v2.2 SLIPDRV kernel.

SeeAlso: INT 61"FTP Software",INT 61/AH=00h"PC/TCP",INT 61/AH=50h,INT 61/AH=51h

SeeAlso: INT 61/AH=52h,INT 61/AH=53h

-----N-6155-----

INT 61 U - PC/TCP kernel v2.3 - ???

AH = 55h

???

Return: ???

SeeAlso: INT 61"FTP Software"

-----N-6156-----

INT 61 U - PC/TCP kernel v2.3 - ???

AH = 56h

???

Return: ???

SeeAlso: INT 61"FTP Software"

-----N-6157-----

INT 61 U - PC/TCP kernel v2.3 - ???

AH = 57h

???

Return: ???

SeeAlso: INT 61"FTP Software"

-----N-6158-----

INT 61 U - PC/TCP kernel v2.3 - ???

AH = 58h


```
    ???  
Return: ???  
SeeAlso: INT 61"FTP Software"  
-----N-6159-----  
INT 61 U - PC/TCP kernel v2.3 - ???  
    AH = 59h  
    ???  
Return: ???  
SeeAlso: INT 61"FTP Software"  
-----N-615A-----  
INT 61 U - PC/TCP kernel v2.3 - ???  
    AH = 5Ah  
    ???  
Return: ???  
SeeAlso: INT 61"FTP Software"  
-----N-615B-----  
INT 61 U - PC/TCP kernel v2.3 - ???  
    AH = 5Bh  
    ???  
Return: ???  
SeeAlso: INT 61"FTP Software"  
-----N-615C-----  
INT 61 U - PC/TCP kernel v2.3 - ???  
    AH = 5Ch  
    ???  
Return: ???  
SeeAlso: INT 61"FTP Software"  
-----N-615D-----  
INT 61 U - PC/TCP kernel v2.3 - ???  
    AH = 5Dh  
    ???  
Return: ???  
SeeAlso: INT 61"FTP Software"  
-----N-615E-----  
INT 61 U - PC/TCP kernel v2.3 - ???  
    AH = 5Eh  
    ???  
Return: ???  
SeeAlso: INT 61"FTP Software"  
-----N-615F-----  
INT 61 U - PC/TCP kernel v2.3 - ???
```

AH = 5Fh

???

Return: ???

SeeAlso: INT 61"FTP Software"

-----!---Section-----