Interrupt 21H

Function Number: **3FH**

Function Name: Read Using Handle

Purpose: Read data from an open file handle

Data Passed:

AH = 3FH
BX = File handle
(Native Mode) ECX = Number of data bytes to read
(Native Mode) DS:EDX = Address of memory area to receive data
(8086 Mode) CX = Number of data bytes to read
(8086 Mode) DS:DX = Address of memory area to receive data

Data Returned:

AX = Error code if carry is set; otherwise
(Native Mode) EAX = Actual length read (0 = end of file)
(8086 Mode) AX = Actual length read (0 = end of file)

Error Codes:

5 - Access denied (file or directory can't be read/written to)
6 - Invalid handle
33 - Area locked by someone else

Extended Error Codes:

Error Code	Type	Action	Location
5	3	4	1
6	7	4	1
33	10	2	1
Critical Error	11	7	2
	11	5	2
	11	1	2
	11	1	1
	9	4	1
	7	4	1
	5	1	1

(See function 59H for Extended Error Code meanings)

Comments: Reading begins with the byte addressed by the I/O pointer, and causes the I/O pointer to be repositioned to the byte following the last byte read. When using this call to read from the "CON" device, all of the features of MOS's command line editing are available.

Interrupt 21H

Function Number: **40H**

Function Name: Write Using Handle

Purpose: Write data to an open file handle

Data Passed:

AH = 40H
BX = File handle
(Native Mode) ECX = Number of data bytes to write
(Native Mode) DS:EDX = Address of data to be written
(8086 Mode) CX = Number of data bytes to write
(8086 Mode) DS:DX = Address of data to be written

Data Returned:

AX = Error code if carry is set; otherwise
(Native Mode) EAX = The actual length written (if less than ECX then disk is full)
(8086 Mode) AX = The actual length written (if less than CX then disk is full)

Error Codes:

5 - Access denied (file or directory can't be read/written to)
6 - Invalid handle
33 - Area locked by someone else

Extended Error Codes:

Error Code	Type	Action	Location
5	3	4	1
6	7	4	1
33	10	2	1
Critical Error	11	7	2
	11	5	2
	11	1	2
	11	1	1
	9	4	1
	7	4	1
	5	1	1

(See function 59H for Extended Error Code meanings)

Comments: Writing begins with the byte addressed by the I/O pointer, and causes the I/O pointer to be repositioned to the byte following the last byte written. Writing a zero-length record will cause the file to be truncated to the current position of its I/O pointer.

Interrupt 21H

Function Number: **41H**

Function Name: Delete File

Purpose: Removes a file from disk

Data Passed:

AH = 41H
(Native Mode) DS:EDX = Address of filename
(8086 Mode) DS:DX = Address of filename

Data Returned: AX = Error code if carry is set

Error Codes:

3 - Invalid path
5 - Access denied (file or directory can't be read/written to)

Extended Error Codes:

Error Code	Type	Action	Location
3	8	3	2
5	3	4	1
Critical Error	11	7	2
	11	5	2
	11	1	2
	11	1	1
	9	4	1
	7	4	1
	5	1	1

(See function 59H for Extended Error Code meanings)

Comments: See function 39H for the format of the name string.

Interrupt 21H

Function Number: **42H**

Function Name: Seek File Position

Purpose: Sets the value of a file's I/O pointer

Data Passed:

AH = 42H
AL = 0 if relative to beginning of file; 1 if relative to current I/O pointer; 2 if relative to end of file
BX = File handle
(Native Mode) ECX = Distance or offset in bytes
(8086 Mode) CX:DX = Distance or offset in bytes

Data Returned:

AX = Error code if carry is set; otherwise
(Native Mode) EAX = The new I/O pointer value
(8086 Mode) DX:AX = The new I/O pointer value

Error Codes:

1 - Invalid function number
6 - Invalid handle

Extended Error Codes:

Error Code	Type	Action	Location
1	7	4	1
6	7	4	1

(See function 59H for Extended Error Code meanings)

Interrupt 21H

Function Number: **43H**

Function Name: Get/Set File Attributes

Purpose: Interrogate or modify attribute flags in a file's directory entry

Data Passed:

AH = 43H
AL = 0 to get current attributes; 1 to modify attributes
CX = Attribute bits (if AL=1)
(Native Mode) DS:EDX = Address of filename
(8086 Mode) DS:DX = Address of filename

Data Returned:

AX = Error code if carry is set
CX = Attribute bits (if AL was 0)

Error Codes:

2 - Invalid filename
3 - Invalid path
5 - Access denied (file or directory can't be read/written to)

Extended Error Codes:

Error Code	Type	Action	Location
2	8	3	2
3	8	3	2
5	3	4	1
Critical Error	11	7	2
	11	5	2
	11	1	2
	11	1	1
	9	4	1
	7	4	1
	5	1	1

(See function 59H for Extended Error Code meanings)

Comments: See function 39H for the format of the name string.

The "attributes" passed or returned in CX are the sum of the following hexadecimal values that apply to the file:

01H - Read-only (cannot be opened for output)
02H - Hidden file (bypassed in normal directory searches)
04H - System File (bypassed in normal directory searches)
20H - Archive flag (set when changed, cleared by EXPORT)

Interrupt 21H

Function Number:	**44H**
Function Name:	Device I/O Control
Purpose:	To provide applications IOCTL to devices.
Data Passed:	There are several sub-functions. Each class will be discussed separately on the next several pages.
Data Returned:	See individual sub-functions

Error Codes:

1 - Invalid function (or sub-function) number

See individual sub-functions for additional error codes.

Extended Error Codes:

Error Code	Type	Action	Location
1	7	4	1

See individual sub-functions for additional extended error codes.

Comments: Function 44H provides sub-functions for device information, reading/writing, status checks, device queries, share/lock retries, and logical unit functions. Logical unit functions are: getting/ setting device parameters, reading/writing logical tracks, formatting/verifying logical device and getting/setting logical device. Function 44H may be used with files, character devices and block devices.

Interrupt 21H

Function Number: **44H**

Sub-Function: **00H, 01H**

Sub-Function Name: Device Information Function

Data Passed:

AH = 44H
AL = 00H - Get device info
AL = 01 - Set device info
BX = device handle
DH = 0 if AL = 01H
DL = Device info

Data Returned:

DX = Device information. See below

Error Codes:

6 - Invalid handle

Extended Error Codes:

Error Code	Type	Action	Location
6	7	4	1

(See function 59H for extended error code meanings)

Comments: Bit 7 is the file/device indicator. If it's 0, it's a file, otherwise it's a device. If bit 7 is 0, bits 0-5 are the unit number, bit 6 is 0 if the file has been written to or changed. The remaining bits are reserved and shouldn't be changed.

If Bit 7 indicates a device, the following table defines the rest of the bits.

Bit	Definition
0	1 = standard input device
1	1 = standard output device
2	1 = NUL device
3	1 = Clock$ device
4	Reserved
5	1 = binary mode. 0 = ASCII mode
6	0 = EOF on read request
14	1 = device processes control strings

Valid calls to process control strings are 2, 3, 4 and 5. If bit 14 is set, calls 2 and 5 can be made via IOCTL.

Bits 8 - 15 are the upper byte of the device driver's attribute word. See the chapter on Device Drivers for more information.

Interrupt 21H

Function Number: **44H**

Sub-Function: **02H, 03H, 04H, 05H**

Sub-Function Name: Read/Write Functions

Data Passed:

AH = 44H
AL = 02H Character device read control string
AL = 03H Character device write control string
AL = 04H Block device read control string
AL = 05H Block device write control string
BX = Handle (Character device)
BL = Drive number (0 = default, 1 = A, 2 = B, etc.)
CX = Read/write byte count
(Native Mode) DS:EDX = data buffer
(8086 Mode) DS:DX = data buffer

Data Returned:

AX = number of characters transferred.

Error Codes:

6 - Invalid handle (02H, 03H)
15 - Invalid drive (04H, 05H)

Extended Error Codes:

Error Code	Type	Action	Location
6	7	4	1
15	8	3	2

Interrupt 21H

Function Number: **44H**

Sub-Function: **06H, 07H**

Sub-Function Name: I/O Status Functions

Data Passed:

AH = 44H
AL = 06H for read status
AL = 07H for write status
BX = File/Device Handle

Data Returned:

For Files:

AL = 0FFH while not EOF
AL = 00H upon EOF

For Devices:

AL = 00H - device not ready
AL = 0FFH - device is ready

Error Codes:

6 - Invalid handle

Extended Error Codes:

Error Code	Type	Action	Location
6	7	4	1

(See function 59H for extended error code meanings)

Interrupt 21H

Function Number: **44H**

Sub-Function: **08H, 09H, 0AH**

Sub-Function Name: Query Functions

Data Passed:

AH = 044H
AL = 08H - Removable media
AL = 09H - Device local/remote
AL = 0AH - File local/remote
BL = Drive number (0 = default, 1 = A, 2 = B,
BX = etc.) for functions 08H, 09H
Handle (function 0AH)

Data Returned:

For Function 08H:

Carry Flag set:

AX = 0FH BL value is invalid

No Carry:

AX = 00H if media is removable
AX = 01H if media is fixed

For Function 09H:

DX = device attribute word. If bit 12 is set,
device is remote.

For Function 0AH:

DX = device attribute word. If bit 15 is set, it's a remote handle.

Error Codes:

6 - Invalid handle (0AH)
15 - Invalid drive (08H, 09H)

Extended Error Codes:

Error Code	Type	Action	Location
6	7	4	1
15	8	3	2

(See function 59H for extended error code meanings)

Interrupt 21H

Function Number: **44H**

Sub-Function: **0BH**

Sub-Function Name: Set Share/Lock Retry Function

Data Passed:

AH = 44H
AL = 0BH
CX = Delay count
DX = Retry count

Data Returned:

Carry flag set:

AX = error codes

No Carry:

AX = insignificant

Error Codes: None

Extended Errors: None

Comments: Share/Lock conflicts do not return an error until they have been retried a certain number of times. You may alter both the number of times the system retries accessing a locked file, and the number of system ticks (which are approximately 55 milliseconds apart) to wait between retries through this call.

Interrupt 21H

Function Number: **44H**

Sub-Function: **0DH**

Sub-Function Name: Logical Units Functions

Data Passed:

AH = 44H
AL = 0DH
BL = Unit (drive) number (0 = default, 1 = A, 2 = B, etc.)
CH = 08H
CL = sub-function
= 40H - Set device parameters
= 41H - Write track on logical device
= 42H - Format track on logical device
= 60H - Get device parameters
= 61H - Read track
= 62H - Verify track
(Native Mode) DS:EDX = address of parameter block
(8086 Mode) DS:DX = address of parameter block

Data Returned: See individual Logical Unit Sub-Functions

Error Codes:

15 - Invalid drive

If the carry flag is set issue a function 59H call to get extended error information.

Extended Error Codes:

Error Code	Type	Action	Location
15	8	3	2

Sub-Functions 40H,60H - Get/Set Device Parameters

The fields of the parameter block:

Field	Size	Description
SF	1 byte	Sub-function field. See below
DT	1 byte	Device type. See below
DA	1 word	Device attribute. See below
CC	1 word	Device's cylinder count. See below
MT	1 byte	Media type indicator
BPB	31bytes	Device BPB. See Below
TLD	varies	Track Layout Definition. See below

Sub-function field (SF):

If GET call: (CL=60H)
Bit 0 = 1 - Return current BPB
Bit 0 = 0 - " default "
Bits 1 - 7 = 0
NOTE: track layout definition not returned.

If SET call: (CL=40H)
Bit 0 = 1 - Return contents of BPB field for all following BUILD BPB calls.
Bit 0 = 0 - BPB field is new default BPB for device. Subsequent BUILD BPB call will return actual BPB.

Bit 1 = 1 - Read only Track layout definition field. Ignore all others.
Bit 1 = 0 - Read all fields.

Bit 2 = 1 - All sectors in definition field are the same size.
Bit 2 = 0 - All sectors not necessarily the same size.

All other bits must be 0. Don't set bits 0 and 1 simultaneously.

Device type field (DT):

Value	Drive
0	320K/360K 5.25
1	1.2M 5.25
2	3.5, 720K
3	Reserved
4	Reserved
5	Hard Disk

Device attribute field (DA):

Bit 0 = 0 - The media is removable
Bit 0 = 1 - The media isn't removable

Bit 1 = 1 - drive supports a change line
Bit 1 = 0 - drive doesn't support a change line

Bits 2 - 7 = Reserved

Cylinder count field (CC):

This is the total number of cylinders supported on the logical device. The call cannot set this field.

Media type field (MT):

This byte indicates what type of media is expected to be in the drive.

If the drive supports multiple media types; for example, 48 and 96 TPI media, a value of 0 indicates the default media. A value of 1 would indicate another. Use this if the type of media in the drive can't be determined.

BPB Field: See the following BPB definition table.

Field	Size	Description
BPS	1 word	Sector size in bytes
AUS	1 byte	Allocation unit size (in sectors)
RS	1 word	Reserved sectors
FC	1 byte	FAT Count
REC	1 word	Root directory entry count
TSL	1 word	Low word of total sectors
MD	1 byte	Media descriptor byte
FSC	1 byte	FAT size in sectors
TSH	1 byte	high byte of total sectors
SPT	1 word	Sectors per track
H	1 word	The number of heads
HS	4 bytes	number of hidden sectors
RS1	4 bytes	Reserved
RS2	6 bytes	Reserved

The TSH byte is the high byte of the 24-bit quantity which is the total number of sectors. Total sectors may be calculated by the formula:

$$TS = TSH * 2^{16} + TSL$$

The Track Layout Definition field (TLD):

This field is a variable length table which defines the layout of a track on the media. The sector count word defines the length of the table. There are two words for each sector. A sample table definition is (in C):

```
struct tld {          /* all fields are 16-bits long */
   unsigned SPT; /* sectors per track */
   unsigned S1;   /* Set to actual sector number */
   unsigned SS1; /* Sector size */
   unsigned S2;   /* same as S1 */
   unsigned SS2; /* same as SS1 */
            .
            .
            .
            .
   unsigned Sn;   /* n=SPT */
   unsigned SSn; /* size of nth sector */
   };
```

All sector numbers are 1-based. Set bit 2 of the SF field of the parameter block to 1 if all sectors are the same size.

Sub-Functions 41H,61H - Track Read/Write

The track read/write parameter block is as follows:

Field	Size	Description
SF	1 byte	Sub-function field. See below
HN	1 word	Head number
CN	1 word	Cylinder number
FSN	1 word	First sector number. 0 - based
SC	1 word	Number of sectors to transfer
TA	4 bytes	I/O transfer address

Set all bits in the sub-function field to zero.

Sub-Functions 42H,62H - Track Format/Verify

The parameter block for these functions is defined as:

Field	Size	Description
SF	1 byte	Sub-function field. See below
HN	1 word	Head number
CN	1 word	Cylinder number

Use the sub-function byte when calling this function as follows:

Bit 0 = 1 - Do you support irregular track layouts?
Bit 0 = 0 - Format the track

Upon return from the call, the SF byte will indicate the following:

Bit 0 = 0 - The driver supports irregular track layouts and yours is ok.
Bit 0 = 1 - The driver doesn't support irregular track layouts.

Bit 1= - Your track layout is invalid.

To determine if your track layout is valid, issue this call with bit 0 of the SF byte set (1).

Interrupt 21H

Function Number: **44H**

Sub-Function: **0EH, 0FH**

Sub-Function Name: Logical Unit Select Functions

Data Passed:

AH = 44H
AL = 0EH - Units query
AL = 0FH - Set next logical unit
BL = drive number (0 = default, 1= A, 2 =B, etc.)

Data Returned:

AL = drive number (0 = only 1, 1 = A, 2 = B, etc.)

Error Codes:

15 - Invalid drive

If carry is set upon return, issue function 59H for extended error information.

Extended Error Codes:

Error Code	Type	Action	Location
15	8	3	2

Comments: Function 0EH queries the driver to determine if it controls more than one logical unit. BL is a drive number upon entry. AL contains the number of the logical unit by which the device was last accessed. Function 0FH sets the driver to the next logical unit. This will become the logical unit for all I/O through the driver until this call is issued again, changing the logical unit or another logical unit is accessed. The function causes MOS to suppress the diskette change prompt.

Interrupt 21H

Function Number: **45H**

Function Name: File Handle Duplicate

Purpose: To create a new file handle for an already open file.

Data Passed:

AH = 45H
BX = existing file handle

Data Returned:

AX = new file handle if carry flag not set.

Error Codes:

4 - Too many files open
6 - Invalid handle

Extended Error Codes:

Error Code	Type	Action	Location
4	1	4	1
6	7	4	1

(See function 59H for extended error code meanings)

Comment: If the read/write pointer is moved for one of the duplicate handles, it is moved for both.

Interrupt 21H

Function Number: **46H**

Function Name: Force Duplicate Handle

Purpose: Forces the file handle in CX to refer to handle in BX.

Data Passed:

AH = 46H
BX = existing file handle
CX = second handle

Data Returned: None

Error Codes:
4 - Too many files open
6 - Invalid handle

Extended Error Codes:

Error Code	Type	Action	Location
4	1	4	1
6	7	4	1
Critical Error	11	7	2
	11	5	2
	11	1	2
	11	1	1
	9	4	1
	7	4	1
	5	1	1

(See function 59H for extended error code meanings)

Comments: If the read/write pointer is moved for one of the duplicate handles, it is moved for both.

Interrupt 21H

Function Number: **47H**

Function Name: Get Current Directory

Purpose: Find the current (default) directory for a particular drive

Data Passed:

AH = 47H
DL = 0 for default drive, 1 for A, ...
(Native Mode) DS:ESI = Address of 64-byte target area
(8086 Mode) DS:SI = Address of 64-byte target area

Data Returned:

AX = Error code if carry is set
(Native Mode) DS:ESI = Full directory name is put into user-supplied area
(8086 Mode) DS:SI = Full directory name is put into user-supplied area

Error Codes:

15 - Invalid drive

Extended Error Codes:

Error Code	Type	Action	Location
15	8	3	2

(See function 59H for Extended Error Code meanings)

Comments: The returned string is in the same form that might be used as input to function 39H; a full specification, null-terminated, but without a drive letter nor a leading backslash.

Interrupt 21H

Function Number: **48H**

Function Name: Memory Allocation (8086 Mode only)

Purpose: To allocate the requested amount of memory.

Data Passed:

AH = 48H
BX = Paragraphs needed

Data Returned:

AX = segment address of allocated memory.
AX = error code if carry flag set
BX = number of paragraphs available if the call fails.

Error Codes:

7 - Memory control blocks destroyed
8 - Insufficient memory
9 - Invalid memory block address

Extended Error Codes:

Error Code	Type	Action	Location
7	7	4	5
8	1	4	5
9	7	4	5

(See function 59H for extended error code meanings)

Interrupt 21H

Function Number: **49H**

Function Name: Free Memory (8086 Mode only)

Purpose: To return allocated memory to system.

Data Passed:

AH = 49H
ES = segment address of block to free

Data Returned: None

Error Codes:

7 - Memory control blocks destroyed
9 - Invalid memory block address

Extended Error Codes:

Error Code	Type	Action	Location
7	7	4	5
9	7	4	5

(See function 59H for extended error code meanings)

Interrupt 21H

Function Number: 4AH

Function Name: Alter Allocated Memory Blocks (8086 Mode Only)

Purpose: To allow for the changing in size of the allocated memory blocks

Data Passed:

AH = 4AH
ES = Segment address of block
BX = New size in paragraphs

Data Returned:

BX = maximum number of paragraphs if "grow" request fails.

Error Codes:

7 - Memory control blocks destroyed
9 - Invalid memory block address

Extended Error Codes:

Error Code	Type	Action	Location
7	7	4	5
9	7	4	5

(See function 59H for extended error code meanings)

Interrupt 21H

Function Number: **4BH**

Function Name: Program Load/Execute (8086 Mode only)

Purpose: To load overlays or programs and execute them.

Data Passed:

AH = 4BH
AL = 00H - Load & execute a program.
AL = 01H - Load a program (don't execute)
AL = 03H - Load an overlay
DS:DX = Points to ASCIIZ string which is path and filename of file to be loaded.
ES:BX = load parameter block

Data Returned:

AX = error codes if carry set

Error Codes:

1 - Invalid function number
2 - Invalid filename (file not found)
3 - Invalid path
4 - Too many files open
5 - Access denied (file or directory can't be read/written to)
6 - Invalid handle
7 - Memory control blocks destroyed
8 - Insufficient memory
9 - Invalid memory block address
10 - Invalid environment
11 - Invalid format, for EXE file
32 - Attempt to open file locked by someone else
33 - Area locked by someone else

Extended Error Codes:

Error Code	Type	Action	Location
1	7	4	1
2	8	3	2
3	8	3	2
4	1	4	1
5	3	4	1
6	7	4	1
7	7	4	5
8	1	4	5
9	7	4	5
10	9	4	1
11	9	4	1
32	2	2	1
33	10	2	1
Critical Error	11	7	2
	11	5	2
	11	1	2
	11	1	1
	9	4	1
	7	4	1
	5	1	1

(See function 59H for Extended Error Code meanings)

Comments: When AL = 00H, a PSP is created. The terminate address field and the Ctrl-Break field in the new PSP point to the instruction after the 4BH call in the parent program.

All files opened by the parent are inherited by the child unless the parent opened the file with the inheritance bit set (1). The child is passed the environment string by the parent. Before this call can successfully complete, the parent should shrink its memory block (since it gets all of the memory of the task) and reallocate for this function.

When AL = 01H, a PSP is created, and memory for the new program's environment, PSP and the program itself is allocated from free MOS memory. The calling program should make sure there is enough free memory for the new program.

The new program will not be given control of the CPU. Instead, control will be returned to the caller of this function immediately after the new program is loaded.

If the caller wishes the new program to be given control of the CPU, the caller must manually initialize the new program's registers, then jump to the new program's initial CS:IP. This includes: 1) getting the new program's PSP address via interrupt 21H, function 51H; 2) changing the new program's terminate address in its PSP to the location in the calling program to which you wish the new program to return (the default is the instruction immediately after the interrupt 21H instruction); 3) determining if the new program is in .EXE format or .COM format; and 4) for .EXE format programs, placing the PSP address in the ES and DS registers.

When AL = 03H, no PSP is created and control is not transferred.

The parameter block fields are defined as follows:

For sub-function 00H:

word	segment address of parent's environment
dword	Pointer to command line to pass
dword	Pointer to first FCB
dword	Pointer to second FCB

These values are put in the PSP of the child process. The dword pointers are stored offset:segment.

For sub-function 01H:

word	segment address of child's environment
dword	pointer to command line to pass
dword	pointer to first FCB
dword	pointer to second FCB
word	SP for new program (returned)
word	SS for new program (returned)
word	IP for new program (returned)
word	CS for new program (returned)

For sub-function 03H:

word	segment address of where to load file
word	relocation factor

The relocation factor is the value to be added to the segment address in .EXE format programs when loaded by this function.

Interrupt 21H

Function Number:	**4CH**
Function Name:	Terminate Program
Purpose:	Stop the application and return to previous application
Data Passed:	AH = AL =
	4CH Return Code
Data Returned:	Not Applicable
Error Codes:	Not Applicable
Extended Errors:	Not Applicable

Comments: Any files opened by the application are automatically closed. The return code passed in AL is available for IF and ERRORLEVEL testing, or for function 4DH.

Interrupt 21H

Function Number: **4DH**

Function Name: Request Return Code

Purpose: Determine the return code posted by the previous Terminate call (function 4CH or 31H).

Data Passed:

AH = 4DH

Data Returned:

AX = Return Code, as follows:

AH = 0 process ended normally
AH = 1 process ended by ^C
AH = 2 process ended by critical error
AH = 3 process ended by function 31H, terminate and stay resident

AL = contains the return code set by the previous program

Error Codes: None

Extended Errors: None

Comments: See also function 4CH.

Interrupt 21H

Function Number: **4EH**

Function Name: Find First Directory Entry

Purpose: Find the first directory entry matching a specified filename

Data Passed:

AH = 4EH
CX = Attribute bits
(Native Mode) DS:EDX = Address of filename
(Native Mode) ES:EBX = Address of a 43-byte data area
(8086 Mode) DS:DX = Address of filename
(8086 Mode) DTA = Used as 43-byte data area

Data Returned:

AX = Error code if carry is set

Error Codes:

3 - Invalid path (or filename)
18 - No more files

Extended Error Codes:

Error Code	Type	Action	Location
3	8	3	2
18	1	6	1
Critical Error	11	7	2
	11	5	2
	11	1	2
	11	1	1
	9	4	1
	7	4	1
	5	1	1

(See function 59H for Extended Error Code meanings)

Comments: The specified filename may (and usually will) contain wildcard characters.

The 43-byte data area supplied by the caller is filled in with the following information:

0-20 - Reserved for MOS
21 - Attribute
22-23 - Time of last update
24-25 - Date of last update
26-29 - File size in bytes
30-42 - Found filename, followed by 00H

Interrupt 21H

Function Number: **4FH**

Function Name: Find Next Directory Entry

Purpose: Find the next directory entry after the one found by the previous function 4EH or 4Fh

Data Passed:

AH = 4FH
(Native Mode) DS:EDX = Address of filename
(Native Mode) ES:EBX = Address of the same 43-byte data area used in the previous Find call
(8086 Mode) DTA = The same 43-byte data area used in the previous Find call

Data Returned:

AX = Error code if carry is set

Error Codes:

18 - No more files

Extended Error Codes:

Error Code	Type	Action	Location
18	1	6	1
Critical Error	11	7	2
	11	5	2
	11	1	2
	11	1	1
	9	4	1
	7	4	1
	5	1	1

(See function 59H for Extended Error Code meanings)

Comments: See function 4EH for a description of the data area format.

Interrupt 21H

Function Number: **54H**

Function Name: Get Verify State

Purpose: Tells the application if disk writes are currently being verified by MOS

Data Passed:

AH = 54H

Data Returned:

AL = 1 if verification is on, otherwise 0

Error Codes: None

Extended Errors: None

Comments: See also function 2EH.

Interrupt 21H

Function Number: **56H**

Function Name: Rename File

Purpose: Change the name of a particular file

Data Passed:

AH = 56H
(Native Mode) DS:EDX = Address of current filename
(Native Mode) ES:EDI = Address of new filename
(8086 Mode) DS:DX = Address of current filename
(8086 Mode) ES:DI = Address of new filename

Data Returned:

AX = Error code if carry is set

Error Codes:

2 - Invalid filename
3 - Invalid path
5 - Access denied (file or directory can't be read/written to)
17 - Can't rename files across devices

Extended Error Codes:

Error Code	Type	Action	Location
2	8	3	2
3	8	3	2
5	3	4	1
17	9	3	2
Critical Error	11	7	2
	11	5	2
	11	1	2
	11	1	1
	9	4	1
	7	4	1
	5	1	1

(See function 59H for Extended Error Code meanings)

Comments: Only one file may be renamed per call. The directory associated with the new filename may be different from the file's current directory, in which case the effect is to "move" the file to the new directory. This call cannot, however, be used to move a file to a different disk drive. This function can also be used to "prune" and "graft" portions of the directory tree structure.

Interrupt 21H

Function Number: **57H**

Function Name: Get or change file date/time

Purpose: Interrogate or modify the "date/time of last update" associated with an open file

Data Passed:

AH = 57H
AL = 0 to Get file date/time
1 to Set file date/time
BX = File handle
CX = Time (if AL=1)
DX = Date (if AL=1)

Data Returned:

AX = Error code if carry is set
CX = Time (if AL was 0)
DX = Date (if AL was 0)

Error Codes:

1 - Invalid function number, AL not 0 or 1
5 - Access denied (file or directory can't be read/written to)
6 - Invalid handle

Extended Error Codes:

Error Code	Type	Action	Location
1	7	4	1
5	3	4	1
6	7	4	1
Critical Error	11	7	2
	11	5	2
	11	1	2
	11	1	1
	9	4	1
	7	4	1
	5	1	1

(See function 59H for Extended Error Code meanings)

Comments: The format of information in CX and DX is:

```
-------DL------ -------DH------
7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0
Y Y Y Y Y Y Y M M M M D D D D D

-------CL------ -------CH------
7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0
H H H H H N N N N N N S S S S S
```

where:

YYYYYYY = Year minus 1980 (e.g., 0000001 is 1981)
MMMM = Month 1-12
DDDDD = Day of month 1-31
HHHHH = Hours 0-23
NNNNNN = Minutes 0-59
SSSSS = Seconds times 2 (0-29)

Interrupt 21H

Function Number: **59H**

Function Name: Get Extended Error Code

Purpose: Returns error information associated with the previous function call

Data Passed:

AH = 59H
BX = 0

Data Returned:

AX = Error code
BH = Type of error
BL = Suggested action
CH = Location of error

Error Codes: See following table

Extended Errors: See following table

Comments: Possible error codes are listed with the description of each associated function call in this chapter. The extended error codes that may be returned with each standard error are also shown.

For convenience, both standard and extended error codes are listed on the following tables.

Extended Error Codes:

Possible error TYPES are:

1 - Resource pool is depleted
2 - The cause of the error is temporary
3 - Security violation
4 - Internal system failure
5 - Hardware failure
6 - System configuration problem
7 - Illegal request(s) from the application
8 - Resource not found
9 - Improper format of supplied data
10 - Resource locked
11 - I/O error
12 - Resource name already exists
13 - Other

Possible suggested-ACTION codes are:

1 - Retry the call a few times before aborting
2 - Retry the call a few times, inserting a time delay before each
3 - If the data came from a user workstation, ask the user to enter it again
4 - Abort the application after performing appropriate termination routines
5 - Abort the application immediately
6 - Ignore the error
7 - Retry the call after prompting the user to correct the condition causing the error

Possible LOCATION codes are:

1 - Location cannot be specified
2 - Error associated with a block device driver
3 - Error associated with a network
4 - Error associated with serial device driver
5 - Memory (RAM) error

Standard Error Codes:

1 - Invalid function number
2 - Invalid filename
3 - Invalid path
4 - Too many files open
5 - Access denied
6 - Invalid handle
7 - Memory control blocks destroyed
8 - Insufficient memory
9 - Invalid memory block address
10 - Invalid environment
11 - Invalid format
12 - Invalid access code
13 - Invalid data
15 - Invalid drive
16 - Can't delete current directory
17 - Can't rename files across devices
18 - No more files
19 - Disk is write-protected
20 - Bad disk unit
21 - Drive not ready
22 - Invalid disk command
23 - CRC (Cyclic Redundancy Check) error
24 - Invalid length (disk operation)
25 - Seek error
26 - Not a MOS disk
27 - Sector not found
28 - Out of paper
29 - Write fault
30 - Read fault
31 - General failure
32 - Attempt to open file locked by someone else
33 - Area locked by someone else
34 - Wrong disk
35 - FCB (File Control Block) unavailable
50 - Network request not supported
51 - Remote computer not listening
52 - Duplicate name on network

53 - Network name not found
54 - Network busy
55 - Network device no longer exists
56 - NetBIOS command limit exceeded
57 - Network adapter hardware error
58 - Incorrect response from network
59 - Unexpected network error
60 - Incompatible remote adaptor
61 - Print queue full
62 - Queue not full
63 - Not enough space for print file
64 - Network name was deleted
65 - Access denied
66 - Network device type incorrect
67 - Network name not found
68 - Network name limit exceeded
69 - NetBIOS session limit exceeded
70 - Temporarily paused
71 - Network request not accepted
72 - Print or disk redirection is paused
80 - File already exists
82 - Cannot make
83 - Interrupt 24 failure
84 - Out of structures
85 - Already assigned
86 - Invalid password
87 - Invalid parameter
88 - Network write fault