DOS Function Call Summary

00H	**Terminate Program** (obsolete)
Entry	AH = 00h CS = Program segment prefix address
Exit	Does not return
Description:	This is an obsolete method for terminating a DOS program. Use function 4Ch instead.

01H	**Read Keyboard with Echo**
Entry	AH = 01h
Exit	AL = Character read from standard input device
Description:	This function will read a single character from the standard input device. If no character is waiting to be read, it will wait until a character becomes available.

02H	**Write to Standard Output Device**
Entry	AH = 02h DL = Character to write to standard output device
Exit	None
Description:	This function will write the specified character on the standard output device.

03H	**Read Character From Aux Device**
Entry	AH = 03h
Exit	AL = Character read from Standard Auxiliary Device
Description:	This function will read a character from the Standard Auxiliary Device. If no character is waiting to be read, it will wait until a character becomes available.

04H	**Write Character to Standard Aux Device**
Entry	AH = 04h DL = Character to write to standard auxiliary device
Exit	None
Description:	This function will write the specified character to the Standard Auxiliary Device.

05H	**Print Character**
Entry	AH = 05h DL = Character to write to the standard printer device
Exit	None
Description:	This function will send the specified character to the Standard Printer Device.

06H	**Direct Console Input/Output**

Entry	AH = 06h
	DL = I/O Switch value: 0FFh = read character, 0h – 0FEh = write

Exit	AL = Character read if read requested and character ready
Description:	This function will perform input or output on the standard input and standard output devices. If the value in DL is 0FFh, a read is requested. In this case, the function will return with the Zero flag clear and a character code in AL if a character is waiting to be read. If no character is available, the function returns with the Zero flag set, and the value in AL is undefined. If the value in DL is anything other than 0FFh, then that value is written to the standard output device. When used for input, this function does not perform Control-C (break) checking on the input stream. Any characters read using this function are not echoed back to standard output.

07H	**Direct Console Input Without Echo**
Entry	AH = 07h
Exit	AL = Character read from standard input
Description:	This function will read a character from the standard input device. If no character is ready to be read, DOS will wait until a character becomes available. The character read is not echoed to the standard output device. This function does not perform Control-C (break) checking on the input stream.

08H	**Read Standard Input Without Echo**
Entry	AH = 08h
Exit	AL = Character read from standard input
Description:	This function will read a character from the standard input device, without echoing it to the standard output.

09H	**Display A Character String**
Entry	AH = 09h DS:DX = Address of the string to print
Exit	None
Description:	This function will write a character string to the standard output device. On entry, DS:DX contains the address of the string. The string must be terminated by a dollar character ‘$’. All characters up to, but not including, the terminating dollar character will be written to the standard output device.

0AH	**Buffered Keyboard Input**
Entry	AH = 0Ah DS:DX = Address of input buffer
Exit	No values returned in registers. The characters read are returned in the specified buffer.
Description:	This function reads characters from standard input, echoes them to standard output, and places them in the specified buffer until a carriage return character is seen. On entry, DS:DX specifies the address of a keyboard input buffer structure. Location 0 of this structure specifies the maximum number of characters to return. Location 1 will be filled in by DOS with the actual number of characters returned. Locations 2-N of the structure will contain the characters returned. Characters are read from standard input and placed into the buffer starting at the third byte until a carriage return (0Dh) character is read. When the number of characters in the buffer reaches one fewer than the maximum, additional characters are ignored and DOS will beep for each character typed until a carriage return character is seen.

0BH	**Test Status of Standard Input Device**
Entry	AH = 0Bh
Exit	AL = Standard input status
Description:	This function checks to see if a character is waiting to be read from the standard input device. The return value in AL will be 0 if no character is waiting. It will be 0FFh if a character is waiting.

0CH	**Clear Keyboard Buffer and Read Keyboard**
Entry	AH = 0Ch
Exit	AL = keyboard character read
Description:	This function will clear the keyboard buffer of any characters waiting to be read. It will then wait for a character to be entered and return that character.

0DH	**Reset Disk Drive**
Entry	AH = 0Dh
Exit	None
Description:	This function will flush all disk buffers for open data files by performing all pending write operations. It does not update directory entries. This function is normally used by Ctrl-C interrupt handlers.

0EH	**Set Default Drive**

Entry	AH = 0Eh
	DL = Drive number (0=A, 1=B, 2=C, etc)

Exit	AL = Number of logical drives on the system
Description:	This function sets the specified drive as the default drive for disk operations.

0FH	**Open File Using FCB**
Entry	
Exit	
Description:	

10H	**Close File Using FCB**
Entry	
Exit	
Description:	

11H	**Search For First Matching File Using FCB**
Entry	
Exit	
Description:	

12H	**Search For Next Matching File Using FCB**
Entry	
Exit	
Description:	

13H	**Delete File Using FCB**
Entry	
Exit	
Description:	

14H	**Sequential Read Using FCB**
Entry	
Exit	
Description:	

15H	**Sequential Write Using FCB**
Entry	
Exit	
Description:	

16H	**Create File Using FCB**
Entry	
Exit	
Description:	

17H	**Rename File Using FCB**
Entry	
Exit	
Description:	

19H	**Return Current Disk Drive**
Entry	
Exit	
Description:	

1AH	**Set Disk Transfer Area**
Entry	AH = 1Ah DS:DX = New disk transfer address
Exit	None
Description:	This function will set the disk transfer address (DTA). This is the address of the buffer that MS-DOS uses file i/o (using FCB’s) and disk directory searches (with or without using FCB’s).

1BH	**Get Default Drive Allocation Table (FAT)**
Entry	
Exit	
Description:	

1CH	**Get Drive File Allocation Table**
Entry	
Exit	
Description:	

21H	**Random File Read Using FCB**
Entry	
Exit	
Description:	

22H	**Random File Write Using FCB**
Entry	
Exit	
Description:	

23H	**Return Number of Records**
Entry	
Exit	
Description:	

24H	**Set Relative Record Size**
Entry	
Exit	
Description:	

25H	**Set Interrupt Vector**

Entry	AH = 25h
	AL = Interrupt number
	DS:DX = Address of new interrupt handler

Exit	None
Description:	This function will store the value specified by DS:DX into the interrupt vector table entry specified by AL.

26H	**Create Program Segment Prefix**
Entry	
Exit	
Description:	

27H	**Random File Block Read**
Entry	
Exit	
Description:	

28H	**Random File Block Write**
Entry	
Exit	
Description:	

29H	**Parse Command Line**
Entry	
Exit	
Description:	

2AH	**Read System Date**
Entry	AH = 2Ah

Exit	AL = Day of week (0=Sunday, 1=Monday, etc)
	CX = Year (1980 – 2099)
	DH = Month (1=Jan, 2=Feb, etc)
	DL = Day of Month (1 – 31)

Description:	This function will return the current MS-DOS system date.

2BH	**Set System Date**

Entry	AH = 2Bh
	CX = Year (1980 – 2099)
	DH = Month (1=Jan, 2=Feb, etc)
	DL = Day of Month (1 – 31)

Exit	AL = Status (0=Success, 0FFh=Error)
Description:	This function will set the current MS-DOS system date to the specified values.

2CH	**Read System Time**
Entry	AH = 2Ch

Exit	CH = Hour in 24 hour format (13=1pm, 14=2pm, etc)
	CL = Minutes (0 – 59)
	DH = Seconds (0 – 59)
	DL = Hundredths of seconds (0 – 99)

Description:	This function will return the current MS-DOS system time.

2DH	**Set System Time**

Entry	AH = 2Dh
	CH = Hour (0 – 23)
	CL = Minutes (0 – 59)
	DH = Seconds (0 – 59)
	DL = Hundredths of seconds (0 – 99)

Exit	AL = Status (0=Success, 0FFh=Error)
Description:	This function will set the current MS-DOS system time to the specified value.

2EH	**Set Disk Write Verify Flag**

Entry	AH = 2Eh
	AL = Flag value (0 = Verify off, 1 = Verify on)

Exit	None
Description:	This function will set or reset the disk verify after write flag. If the verify flag is set, MS-DOS will reread and verify the data after each disk write.

2FH	**Get Disk Transfer Address**
Entry	AH = 2Fh
Exit	ES:BX = Current disk transfer address
Description:	This function will return the current Disk Transfer Address setting.

30H	**Get DOS Version Number**

Entry	AH = 30h
	AL = Return OEM version or DOS version flags
	(0 = OEM version, 1 = DOS version flags)

Exit	AL = DOS major version number
	AH = DOS minor version number
	BH = OEM version number or DOS version flags
	BL = High 8 bits of 24 bit user serial number
	CX = Low 16 bits of 24 bit user serial number

Description:	This function returns the version number of MS-DOS.

31H	**Terminate and Stay Resident**
Entry	
Exit	
Description:	

33H	Set Control-Break State

Entry	
Exit	
Description:	

34H	**Get Address of INDOS Flag**
Entry	
Exit	
Description:	

35H	**Get Interrupt Vector**

Entry	AH = 35h
	AL = Interrupt number

Exit	ES:BX = Address of specified interrupt handler
Description:	This function will return the contents of the specified interrupt vector table entry.

36H	**Determine Free Disk Space**
Entry	
Exit	
Description:	

38H	**Get Country Code**
Entry	
Exit	
Description:	

39H	**Create Subdirectory**
Entry	AH = 39h DS:DX = Pointer to name of new directory
Exit	CY clear = success, CY set = error AX = Error code if CY set.
Description:	This function creates a new directory. DS:DX points to a zero-terminated ASCII string that specifies the name of the new directory to create.

3Ah	**Delete Subdirectory**
Entry	AH = 3Ah DS:DX = Pointer to the name of the directory to delete
Exit	CY clear = success, CY set = error AX = Error code if CY set.
Description:	This function will delete a subdirectory entry. DS:DX points to a zero-terminated ASCII string specifying the name of the directory to delete. If the specified directory is not empty, error 0005h (ERROR_ACCESS_DENIED) will be returned.

3Bh	**Change Current Subdirectory**
Entry	AH = 3Bh DS:DX = Pointer to new directory name.
Exit	CY clear = success, CY set = error AX = error code if CY set
Description:	This function will change the current directory to the specified directory. DS:DX points to a zero-terminated ASCII string specifying the name of the new current directory.

3Ch	**Create File**

Entry	AH = 3Ch
	CX = Attributes
	DS:DX = Pointer to file name

Exit	CY clear = success, CY set = error
	AX = File handle if CY clear
	AX = Error code if CY set

Description:	This function will create a new file in the current directory and assigns it the next available file handle. If the specified file already exists, MS-DOS will open it and truncate its length to 0. The value in CX specifies attribute bits to be associated with the file. See the File Creation Attributes Table for the meanings of the various bits.

File Creation Attributes

ATTR_NORMAL (0000h)	File can be read from or written to
ATTR_READONLY (0001h)	File is read only (can be read but not written)
ATTR_HIDDEN (0002h)	File is hidden and does not appear in a directory listing.
ATTR_SYSTEM (0004h)	File is a system file.
ATTR_VOLUME (0008)	The File Name is used as a volume name.
ATTR_ARCHIVE (0020h)	This file is marked as having been written to.

3DH	**Open File**

Entry	AH = 3Dh
	AL = Access control
	DS:DX = Pointer to the file name

Exit	CY clear = success, CY set = error
	AX = File handle if CY clear
	AX = Error code if CY set

Description:	This function will open the specified file for access. DS:DX points to a zero-terminated ASCII string specifying the file to open. When the file is opened, the read/write pointer is set to 0. The value in AL specifies access control attributes. See the File Open Access Attributes table for the various settings.

File Open Access Attributes

OPEN_ACCESS_READONLY (0000h)	Open the file for read only access
OPEN_ACCESS_WRITEONLY (0001h)	Open the file for write only access
OPEN_ACCESS_READWRITE (0002h)	Open the file for read/write access
OPEN_SHARE_COMPATIBILITY (0000h)	Permit other programs to have any kind of access to the file. This is the default sharing value.
OPEN_SHARE_DENYREADWRITE (0010h)	Do not permit any other program to open the file.
OPEN_SHARE_DENYWRITE (0020h)	Do not permit any other program to open the file for write access.
OPEN_SHARE_DENYREAD (0030h)	Do not permit any other program to open the file for read access.
OPEN_SHARE_DENYNONE (0040h)	Permit other programs to have read or write access, but no program may open the file for compatibility access.
OPEN_FLAGS_NOINHERIT (0080h)	A child program created with the Load and Execute Program Function (4B00h) does not inherit the file handle. If this mode is not set, child programs will inherit the file handle.

3EH	**Close File**

Entry	AH = 3Eh
	BX = File handle

Exit	CY clear = success, CY set = error AX = Error code if CY set
Description:	This function is used to close a file opened or created with any file handle function. When this function is processed, any pending writes to the file will be completed, any locked sections of the file will be unlocked and the directory will be updated to reflect any changes to the file size, date or time.

3FH	**Read File**

Entry	AH = 3Fh,
	BX = File handle
	CX = Number of bytes to read
	DS:DX = Address of buffer

Exit	CY clear if success, CY set if error AX = Actual number of bytes transferred if CY clear AX = Error code if CY set
Description:	This function will attempt to read the specified number of bytes from the file associated with the specified file handle. The data will be read beginning at the current file point position. Following the read, the file pointer will be updated to point to the next byte following the last one transferred. The data will be placed in the buffer specified by DS:DX If 0 if returned in AX, the file pointer was at the end of the file at the beginning of the operation. If the value returned in AX is greater than zero, but less than the number requested in CX, this means that the end of the file was reached during the function.

40H	**Write File**

Entry	AH = 40h
	BX = File handle
	CX = Number of bytes to write
	DS:DX = Address of buffer

Exit	CY clear if success, CY set if error AX = Actual number of bytes written if CY clear AX = Error code if CY set
Description:	This function will write up to the specified number of byte to the file or device associated with the specified file handle. Data will be written beginning at the current location of the file pointer. The file pointer will updated to reflect the number of bytes written. Writing 0 bytes will truncate the file at the current file pointer position. If the number of bytes written is less than requested, the destination disk is full. The carry flag is not set in the case.

41H	**Delete File**
Entry	AH = 41h DS:DX = Pointer to file name
Exit	CY clear = success, CY set = error AX = Error code if CY set
Description:	The function will delete the specified file. DS:DX points to a zero-terminated ASCII string specifying the name of the file to delete. This function can not be used to delete a directory, volume label, or a read-only file.

42H	**Set File Pointer**

Entry	AH = 42h
	AL = Mode
	BX = File handle
	CX:DX = File pointer value

Exit	CY clear if success, CY set if error AX = Error code if error DX:AX = New file pointer position if CY clear
Description:	This function is used to move the read/write pointer to a new position within an open file. The value is AL is the move method and has the following values: 0 = Move relative to the beginning of the file 1 = Move relative to the current position 2 = Move relative to the end of the file The value in CX:DX on entry is a 32 bit value used with the mode to determine the new pointer position. CX contains the upper 16 bits and DX the lower 16 bits of the 32 bit value. For mode 0, this value is treated as an unsigned value and the pointer can't be moved before the beginning of the file. For modes 1 and 2, this value is treated as a signed value and the pointer can be moved forward or backward within the file. Using mode 1 or 2, it is possible to move the file pointer before the beginning of the file. This will not generate an error, but a subsequent read or write at the position would generate an error. Using any mode, it is possible to move the pointer beyond the end of the file. A read at this position would generate an error. However, a write with the file pointer beyond the end of the file will cause the file to be extended and the write to occur at that location. On return, if successful, DX:AX contains the new file pointer position value. DX contains the high 16 bits and AX the low 16 bits.

43H	**Get/Set File Attributes**

Entry	AH = 43h
	AL = Get/Set code (0=Get, 1=Set)
	CX = Attributes to set (for AL=1)
	DS:DX = Pointer to the file name

Exit	CY clear if success, CY set if error
	AX = error code if CY set
	CX = File attributes (if CY clear and AL=0)

Description:	This function is used to Get or Set the attributes associated with a file. DS:DX contains the address of a zero-terminated ASCII string that specifies the file name. AL contains a code to indicate whether to get the current attributes or set new attributes. See the File Attributes Table for the meanings of the file attributes bits.

File Attributes

ATTR_NORMAL (0000h)	File can be read from or written to.
ATTR_READONLY (0001h)	File is read only and cannot be written to or deleted.
ATTR_HIDDEN (0002h)	File is hidden and does not appear in directory listings
ATTR_SYSTEM (0004h)	File is a system file.
ATTR_VOLUME (0008h)	Filename is the current volume label
ATTR_DIRECTORY (0010h)	Filename is a directory and not a file.
ATTR_ARCHIVE (0020h)	File has been written to.

44H	**Device I/O Control**
Entry	
Exit	
Description:	

45H	**Duplicate File Handle**
Entry	AH =45h BX = File handle to duplicate
Exit	CY clear = success, CY set = error AX = New file handle if CY clear AX = Error code if CY set
Description:	This function is used to create a new file handle that can be used to read or write the same file or device as the specified file handle. When the new handle is created the file pointer will point to the same location as in the original handle. Either handle can then be used to read or write the file. A read or write on either handle will update the pointer for both handles.

46H	**Force Duplicate File Handle**
Entry	AH = 46h BX = File handle to duplicate CX = New handle for file or device
Exit	CY clear if success, CY set if error AX = Error code if CY set
Description:	This function is used to create a duplicate file handle having a particular value. Either handle can then be used to read or write the file or device. If the handle value specified in CX is already in use, that file will be closed before the handle in BX is duplicated.

47H	**Get Current Directory**

Entry	AH = 47h
	DL = Drive number
	DS:SI = Pointer to buffer

Exit	CY clear if success, CY set if error AX = Error code if CY set
Description:	This function will return the path and directory name of the current directory for the specified drive. DS:SI points to a buffer to receive the zero-terminated ASCII string for the current directory name. This must point to a buffer sufficiently large to hold the largest possible path for the directory name.

48H	**Allocate Memory Block**
Entry	AH = 48h BX = Number of paragraphs of memory to allocate
Exit	CY clear if success, CY set if error AX = Segment address of allocated memory if CY clear AX = Error code if CY set
Description:	This function will allocate the requested amount of memory and return the segment address of the allocated memory block.

49H	**Free Memory Block**
Entry	AH = 49h ES = Segment address of memory block to free
Exit	CY clear if success, CY set if error AX = Error code if CY set
Description:	This function will free a memory block allocated via function 48h.

4AH	**Modify Allocated Memory Block**
Entry	AH = 4Ah BX = Requested size in paragraphs ES = Segment address of memory block to modify
Exit	CY clear if success, CY set if error AX = Error code if CY set
Description:	This function is used to modify the size of a memory block allocates using function 48h.

4BH	**Load or Execute Program**
Entry	
Exit	
Description:	

4CH	**Terminate Program**

Entry	AH = 4Ch
	AL = Program status code

Exit	Does not return
Description:	This function is used to terminate execution of a program. The value passed in AL is a status code that will be saved by DOS and can be queried by the parent program using function 4Dh.

4DH	**Get Program Termination Status Code**
Entry	AH = 4Dh

Exit	AH = Termination method
	AL = Termination status code

Description:	This function can be used by a parent program to determine the termination status of a child program. The termination status code will be the value returned by the child program when it terminated. The value in AH will indicate what caused the child program to terminate: 00h = Normal termination 01h = Terminated by Ctrl-C 02h = Terminated by critical error 03h = Terminate and stay resident

4EH	**Find First Matching File**

Entry	AH = 4Eh
	CX = Attributes
	DS:DX = Pointer to search string

Exit	CY clear if success, CY set if error AX = Error code if CY set
Description:	This function is used to search a directory for the first file whose name and attributes match the specified name and attributes. DS:DX points to a zero terminated ASCII string the specifies the file or directory to search for. This string may contain wild card characters. If the function is successful, a FILEINFO structure at the current DTA will be filled in with the relevant information. If the DTA hasn't be explicitly set by a call to function 1Ah, the default DAT at offset 80h in the PSP will be used. If the caller specifies any combination of ATTR_SYSTEM, ATTR_HIDDEN, and ATTR_DIRECTORY, this function will return normal files as well as files matching those attributes.

4FH	**Find Next Matching File**
Entry	AH = 4Fh
Exit	CY clear if success, CY set if error
Description:	This function is used following a call to function 4Eh (Find First Matching File) to find the next file in a directory matching the specified search criteria. As with function 4Eh, if successful, a FILEINFO structure at the current DTA will be filled in with the relevant information from the file found.

The FILEINFO structure has the following form:

```
        FILEINFO    struc
                FiReserved      db      21 dup (?)      ;reserved
                FiAttribute     db      ?               ;attribute of file found
                FiFileTime      dw      ?               ;time of last write
                FiFileDate      dw      ?               ;date of last write
                FiSize          dd      ?               ;file length
                FiFileName      db      13 dup (?)      ;filename and extension
        FILEINFO    ends
```

50H	**Set Current Program Segment Prefix**
Entry	AH = 50h BX = segment address of new PSP
Exit	none
Description:	This function is used to set a new PSP as the current PSP.

51H	**Get Current Program Segment Prefix**
Entry	AH = 51h
Exit	BX = segment address of current PSP
Description:	This function is used to return the segment address of the Program Segment Prefix (PSP) for the current program.

54H	**Get Disk Verify Status**
Entry	AH = 54h

Exit	AL = Current state of disk verify flag
Description:	This function is used to obtain the current setting of the DOS “disk verify after write flag”.

Entry	
Exit	
Description:	

56H	**Rename File**
Entry	
Exit	
Description:	

57H	**Read File Date and Time**
Entry	
Exit	
Description:	

59H	**Get Extended Error Code**
Entry	
Exit	
Description:	

5AH	**Create Unique File Name**
Entry	
Exit	
Description:	

5BH	**Create New File**
Entry	
Exit	
Description:	

5CH	**Lock/Unlock File**
Entry	
Exit	
Description:	

5DH	**Set Extended Error Information**
Entry	
Exit	
Description:	

67H	**Set Handle Count**
Entry	
Exit	
Description:	

68H	**Commit File**
Entry	
Exit	
Description:	

6CH	**Extended File Open**
Entry	

Exit	
Description:	

File Creation Attributes

ATTR_NORMAL (0000h)	File can be read from or written to
ATTR_READONLY (0001h)	File is read only (can be read but not written)
ATTR_HIDDEN (0002h)	File is hidden and does not appear in a directory listing.
ATTR_SYSTEM (0004h)	File is a system file.
ATTR_VOLUME (0008)	The File Name is used as a volume name.
ATTR_ARCHIVE (0020h)	This file is marked as having been written to.

File Open Access Attributes

OPEN_ACCESS_READONLY (0000h)	Open the file for read only access
OPEN_ACCESS_WRITEONLY (0001h)	Open the file for write only access
OPEN_ACCESS_READWRITE (0002h)	Open the file for read/write access
OPEN_SHARE_COMPATIBILITY (0000h)	Permit other programs to have any kind of access to the file. This is the default sharing value.
OPEN_SHARE_DENYREADWRITE (0010h)	Do not permit any other program to open the file.
OPEN_SHARE_DENYWRITE (0020h)	Do not permit any other program to open the file for write access.
OPEN_SHARE_DENYREAD (0030h)	Do not permit any other program to open the file for read access.
OPEN_SHARE_DENYNONE (0040h)	Permit other programs to have read or write access, but no program may open the file for compatibility access.
OPEN_FLAGS_NOINHERIT (0080h)	A child program created with the Load and Execute Program Function (4B00h) does not inherit the file handle. If this mode is not set, child programs will inherit the file handle.