

## BIOS and DOS Interrupts

[INTERRUPT 10h FUNCTIONS \(VIDEO\)](#)  
[INTERRUPT 16h FUNCTIONS \(KEYBOARD\)](#)  
[INTERRUPT 21h FUNCTIONS \(DOS SERVICES\)](#)

Interrupt Number	Description
0	<b>Divide by Zero.</b> Internal: activated when attempting to divide by zero.
1	<b>Single Step.</b> Internal: activated when a memory error is executed.
2	<b>Nonmaskable.</b> Internal: activated when a memory error occurs.
3	<b>Breakpoint.</b> Internal: activated when the 0CCh (INT 3) instruction is executed.
4	<b>Overflow.</b> Activated when the INTO instruction is executed and the Overflow flag is set.
5	<b>Print Screen.</b> Activated either by the INT 5 instruction or tying SHIFT-PRTSCL.
6	<i>(Reserved)</i>
7	<i>(Reserved)</i>
8	<b>Timer Interrupt.</b> Updates the BIOS clock 18.2 times per second. For your own programming, see INT 1Ch.
9	<b>Keyboard Hardware Interrupt.</b> Activated when a key is pressed. Reads the key from the keyboard port and stores it in the keyboard typeahead buffer.
0A-0D	<i>(Reserved)</i>
0E	<b>Diskette Interrupt.</b> Activated when a disk seek is in progress.
0F	<i>(Reserved)</i>
10	<b>Video Services.</b> Routines for manipulating the video display (see the complete list in a subsequent table).
11	<b>Equipment Check.</b> Return a word showing all the peripherals attached to the system.
12	<b>Memory Size.</b> Return the amount of memory (in 1,024-byte blocks) in AX.
13	<b>Disk Services.</b> Reset the disk controller, get the status of the most recent disk access, read and write physical sectors, and format a disk.
14	<b>Asynchronous (Serial) Port Services.</b> Initialize and read or write the asynchronous communications port, and return the port's status.
16	<b>Keyboard Services.</b> Read and inspect keyboard input (see the complete list in a subsequent table).
17	<b>Printer Services.</b> Initialize, print, and return the status of the printer.
18	<b>ROM BASIC.</b> Execute cassette BASIC in ROM.
19	<b>Boot Strap.</b> Reboot DOS.
1A	<b>Time of Day.</b> Get the number of timer ticks since the machine was turned on, or set the counter to a new value. Ticks occur 18.2 times per second.
1B	<b>Keyboard Break.</b> This interrupt handler is executed by INT 9h when CTRL-BREAK is pressed.
1C	<b>User Timer Interrupt.</b> Empty routine, executed 18.2 times per second. May be used by your own program.
1D	<b>Video Parameters.</b> Point to a table containing initialization and information for the Video Controller chip.
1E	<b>Diskette Parameters.</b> Point to a table containing initialization information for the diskette controller.
1F	<b>Graphics Table.</b> Table kept in memory of all extended graphics with ASCII codes greater than 127.
20	<b>Terminal Program.</b> Terminate the current program (INT 21h, function 4Ch should be used instead).
21	<b>DOS Services</b> (see the complete list in the next table).
22	<b>DOS Terminate Address.</b> Point to the address of the parent program or process. When the current program ends, this will be the return address.
23	<b>DOS Break Address.</b> DOS jumps here when CTRL-BREAK is pressed.
24	<b>DOS Error Address.</b> DOS jumps to this address when there is a critical error in the current program, such as a disk media error.
25	<b>DOS Disk Read.</b> Used for reading logical sectors from the disk.
26	<b>DOS Disk Write.</b> Used for writing logical sectors on the disk.
27	<b>Terminate and Stay Resident.</b> Exit to DOS or the calling program, but leave the current program in memory.
28-3F	<i>(Reserved)</i>
40,41	<b>Fixed Disk Services.</b> Fixed disk controller.

4B-7F	(Available for application programs)
80-F0	<i>(Reserved: used by BASIC)</i>
F1-FF	(Available for application programs)

## INTERRUPT 21h FUNCTIONS (DOS SERVICES)

Function number is loaded in AH for each function

Function	Description
0	<b>Program Terminate.</b> Terminate the current program (use function 4Ch instead).
1	<b>Keyboard Input.</b> Wait for a character from the standard input device; echo it on the console. <b>Output:</b> AL = character.
2	<b>Display Output.</b> Display the character in DL on the standard output device (console).
3	<b>Auxiliary Input.</b> Wait for a character from the standard auxiliary device (serial). <b>Output:</b> AL = character.
4	<b>Auxiliary Output.</b> Send the character in DL to the standard auxiliary device (serial).
5	<b>Pointer Output.</b> Send the character in DL to the standard printer device.
6	<b>Direct Console Input/Output.</b> If DL = FFh, read a waiting character from the console. <b>Output:</b> AL = character. If DL = 00h-FEh, output a character instead.
7	<b>Direct Console Input Without Echo.</b> Wait for a character from the standard input device. The character is returned in AL but is not echoed.
8	<b>Console Input Without Echo.</b> Wait for a character from the standard input device. The character is returned in AL but is not echoed. May be terminated by using CTRL-BREAK.
9	<b>Print String.</b> Output an ASCII string of characters to the standard output device. <b>Input:</b> DS:DX = address of the string.
0A	<b>Buffered Keyboard Input.</b> Read a string of characters from the standard input device. <b>Input:</b> DS:DX = address of the buffer.
0B	<b>Check Standard Input Status.</b> Check to see if an input character is waiting. <b>Output:</b> AL = 0FFh if the character is ready; otherwise, AL = 0.
0C	<b>Clear Keyboard Buffer, Invoke Input Function.</b> Clear the console input buffer, and then execute an input function. <b>Input:</b> AL = desired function (1, 6, 7, 8, or 0Ah).
0D	<b>Disk Reset.</b> Flush all file buffers.
0E	<b>Select Disk.</b> Set the default drive. <b>Input:</b> DL = drive number (0 = A, 1 = B, etc.)
0F	<b>Open File.</b> Open a file described by an FCB (file control block). <b>Input:</b> DS:DX = address of the unopened FCB.
10	<b>Close File.</b> Close the file whose FCB is pointed to by DS:DX.
11	<b>Search for First Entry.</b> Search for the first matching filename in the current directory. <b>Input:</b> DS:DX = address of an unopened FCB.
12	<b>Search for Next Entry.</b> Search for the next matching filename in the current directory. <b>Input:</b> DS:DX = address of an unopened FCB. <b>Output:</b> AL = 0 if filename found, FFh if not found.
13	<b>Delete File.</b> Delete all matching directory entries that match the filename. <b>Input:</b> DS:DX = address of an unopened FCB with the filename. <b>Output:</b> AL = 0 if the file was deleted. AL = FFh if the file was not found.
14	<b>Sequential Read.</b> Read the next record of the file pointed to by an opened FCB into memory, and increment the record number. <b>Input:</b> DS:DX points to the FCB. <b>Output:</b> AL = 0 if successful.
15	<b>Sequential Write.</b> Write the record identified by the FCB from the current disk transfer address (DTA). <b>Input:</b> DS:DX points to the FCB. <b>Output:</b> AL = 0 if successful.
16	<b>Create File.</b> Create a new file, or open one if it already exist. <b>Input:</b> DS:DX points to an unopened FCB. <b>Output:</b> AL = 0 successful.
17	<b>Rename File.</b> Rename a file. Both the old and new names are in a modified, unopened FCB. <b>Input:</b> DS:DX points to the FCB. <b>Output:</b> AL = 0 if successful; AL = FFh if the file was not found or the new name currently exists.
18	<i>(Reserved)</i>



44	<p><b><u>I/O Control for Devices.</u></b>  <b>Input:</b> AL =</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Subfunction</th> </tr> </thead> <tbody> <tr><td>00H</td><td>Get Device Information</td></tr> <tr><td>01H</td><td>Set Device Information</td></tr> <tr><td>02H</td><td>IOCTL: Read Control Data from Character Device Driver</td></tr> <tr><td>03H</td><td>IOCTL: Write Control Data from Character Device Driver</td></tr> <tr><td>04H</td><td>IOCTL: Read Control Data from Block Device Driver</td></tr> <tr><td>05H</td><td>IOCTL: Write Control Data from Block Device Driver</td></tr> <tr><td>06H</td><td>IOCTL: Check Input Status</td></tr> <tr><td>07H</td><td>IOCTL: Check Output Status</td></tr> <tr><td>08H</td><td>IOCTL: Check if Block Device is Removable</td></tr> <tr><td>09H</td><td>IOCTL: Check if Block Device is Remote</td></tr> <tr><td>0AH</td><td>IOCTL: Check if Handle is Remote</td></tr> <tr><td>0BH</td><td>IOCTL: Change Sharing Retry Account</td></tr> <tr><td>0CH</td><td>IOCTL: Generic I/O Control for Character Device</td></tr> <tr><td>0DH</td><td>IOCTL: Generic I/O Control for Block Device</td></tr> <tr><td>0EH</td><td>IOCTL: Get Logical Drive Map</td></tr> <tr><td>0FH</td><td>IOCTL: Set Logical Drive Map</td></tr> <tr><td>10H</td><td>IOCTL: Query Generic IOCTL Capability (Handle)</td></tr> <tr><td>11H</td><td>IOCTL: Query Generic IOCTL Capability (Drive)</td></tr> </tbody> </table>	Value	Subfunction	00H	Get Device Information	01H	Set Device Information	02H	IOCTL: Read Control Data from Character Device Driver	03H	IOCTL: Write Control Data from Character Device Driver	04H	IOCTL: Read Control Data from Block Device Driver	05H	IOCTL: Write Control Data from Block Device Driver	06H	IOCTL: Check Input Status	07H	IOCTL: Check Output Status	08H	IOCTL: Check if Block Device is Removable	09H	IOCTL: Check if Block Device is Remote	0AH	IOCTL: Check if Handle is Remote	0BH	IOCTL: Change Sharing Retry Account	0CH	IOCTL: Generic I/O Control for Character Device	0DH	IOCTL: Generic I/O Control for Block Device	0EH	IOCTL: Get Logical Drive Map	0FH	IOCTL: Set Logical Drive Map	10H	IOCTL: Query Generic IOCTL Capability (Handle)	11H	IOCTL: Query Generic IOCTL Capability (Drive)
Value	Subfunction																																						
00H	Get Device Information																																						
01H	Set Device Information																																						
02H	IOCTL: Read Control Data from Character Device Driver																																						
03H	IOCTL: Write Control Data from Character Device Driver																																						
04H	IOCTL: Read Control Data from Block Device Driver																																						
05H	IOCTL: Write Control Data from Block Device Driver																																						
06H	IOCTL: Check Input Status																																						
07H	IOCTL: Check Output Status																																						
08H	IOCTL: Check if Block Device is Removable																																						
09H	IOCTL: Check if Block Device is Remote																																						
0AH	IOCTL: Check if Handle is Remote																																						
0BH	IOCTL: Change Sharing Retry Account																																						
0CH	IOCTL: Generic I/O Control for Character Device																																						
0DH	IOCTL: Generic I/O Control for Block Device																																						
0EH	IOCTL: Get Logical Drive Map																																						
0FH	IOCTL: Set Logical Drive Map																																						
10H	IOCTL: Query Generic IOCTL Capability (Handle)																																						
11H	IOCTL: Query Generic IOCTL Capability (Drive)																																						
45	<p><b><u>Duplicate a File Handle.</u></b> Return a new file handle for a file that is currently open.  <b>Input:</b> BX = file handle.  <b>Output:</b> AX = error code if the Carry flag is set.</p>																																						
46	<p><b><u>Force a Duplicate of a Handle.</u></b> Force the handle in CX to refer to the same file at the same position as the handle in BX.  <b>Input:</b> BX = existing file handle, and CX = second file handle.  <b>Output:</b> AX = error code if the Carry flag is set.</p>																																						
47	<p><b><u>Get Current Directory.</u></b> Get the full path name of the current directory.  <b>Input:</b> DS:SI points to a 64-byte area to hold the directory path, and DL = drive number.  <b>Output:</b> A buffer at DS:SI is filled with the path, and AX = error code if the Carry flag is set.</p>																																						
48	<p><b><u>Allocate Memory.</u></b> Allocate a requested number of paragraphs of memory, measured in 16-byte blocks.  <b>Input:</b> BX = number of paragraphs requested.  <b>Output:</b> AX = segment of the allocated block, BX = size of the largest block available (in paragraphs), and AX = error code if the Carry flag is set.</p>																																						
49	<p><b><u>Free Allocated Memory.</u></b> Free memory that was previously allocated by function call 48h.  <b>Input:</b> ES = segment of the block to be freed.  <b>Output:</b> AX = error code if the Carry flag is set.</p>																																						
4A	<p><b><u>Modify Memory Blocks.</u></b> Modify allocated memory blocks to contain a new block size. The block will shrink or grow.  <b>Input:</b> EX = segment of the block, and BX = requested number of paragraphs.  <b>Output:</b> AX = error code if the Carry flag is set, and BX = maximum number of available blocks.</p>																																						
4B	<p><b><u>Load or Execute a Program.</u></b> Create a PSP for another program, load it into memory, and execute it.  <b>Input:</b> DS:DX points to an ASCIIZ string with the drive, path, and filename of the program; ES:BX points to a parameter block, and AL = function value. Function values in AL:  0 load and execute the program;  1 load but do not execute (overlay program).  3 load overlay  5 set execution state  <b>Output:</b> AX = error code if the Carry flag is set.</p>																																						
4C	<p><b><u>Terminate a Process.</u></b> Usual way to terminate a program and return to either DOS or a calling program.  <b>Input:</b> AL = 8-bit return code, which can be queried by DOS function 4Dh or by the ERRORLEVEL command in a batch file.</p>																																						
4D	<p><b><u>Get Return Code of a Process.</u></b> Get the return code of a process or program, generated by either function call 31h or function call 4Ch.  <b>Output:</b> AL = 8-bit code returned by the program, AH = type of exit generated: 0 = normal termination, 1 = terminated by CTRL-BREAK, 2 = terminated by a critical device error, and 3 = terminated by a call to function call 31h.</p>																																						
4E	<p><b><u>Find First Matching File.</u></b> Find the first filename that matches a given file specification.  <b>Input:</b> DS:DX points to an ASCIIZ drive, path, and file specification; CX = file attribute to be used when searching.  <b>Output:</b> AX = error code if the Carry flag is set; otherwise, the current DTA is filled with the filename, attribute, time, date, and</p>																																						

	<p>size. DOS function call 1Ah (Set DTA) is usually called before this function.</p>										
4F	<p><b><u>Find Next Matching File.</u></b> Find the next filename that matches a given file specification. This is always called after DOS function 4Eh.  <b>Output:</b> AX = error code if the Carry flag is set; otherwise, the current DTA is filled with the file's information.</p>										
50	<p><b><u>Set Current Process ID (Set PSP address).</u></b> Sets a new value of PID into "current process" variable  <b>Input:</b> BX = segment address of new PSP</p>										
51	<p><b><u>Get Current Process ID (Get PSP Address).</u></b> Reads the segment address of the current PSP.  <b>Output:</b> BX = segment address of current PSP</p>										
52	(Reserved)										
53	<p><b><u>Translate BIOS Parameter Block to DOS Drive Parameter Block.</u></b> Transforms drive Parameter information from the BIOS Parameter Block which is stored on the disk to the format used by DOS.  <b>Input:</b> DS:SI = segment:offset of BIOS Parameter Block (BPB) ES:DI = segment:offset for Driver Parameter Block (DPB)  <b>Output:</b> ES:BP address of buffer with information</p>										
54	<p><b><u>Get Verify Setting.</u></b> Get the current value of the DOS verify flag (see DOS function 2Eh).  <b>Output:</b> AL = current verify flag (1 = on, 0 = off).</p>										
55	(Reserved)										
56	<p><b><u>Rename a File.</u></b> Rename a file or move a file to another directory.  <b>Input:</b> DS:DX points to a ASCIIZ string that specifies the current drive, path, and filename; ES:DI points to the new path and filename.  <b>Output:</b> AX = error code if the Carry flag is set.</p>										
57	<p><b><u>Get/Set a File's Date and Time.</u></b> Get or set the date and time stamp for a file.  <b>Input:</b> AL = to get the date/time, or AL = 1 to set the date/time; BX = file handle, CX = new file time, and DX = new file date.  <b>Output:</b> AX = error code if the Carry flag is set; otherwise, CS = current file time, and DX = current file date.</p>										
58	<p><b><u>Memory Functions.</u></b> Memory allocation:  <b>Input:</b> AL =</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Subfunction</th> </tr> </thead> <tbody> <tr><td>00H</td><td>Get Allocation Strategy</td></tr> <tr><td>01H</td><td>Set Allocation Strategy</td></tr> <tr><td>02H</td><td>Get Upper-Memory Link</td></tr> <tr><td>03H</td><td>Set Upper-Memory Link</td></tr> </tbody> </table>	Value	Subfunction	00H	Get Allocation Strategy	01H	Set Allocation Strategy	02H	Get Upper-Memory Link	03H	Set Upper-Memory Link
Value	Subfunction										
00H	Get Allocation Strategy										
01H	Set Allocation Strategy										
02H	Get Upper-Memory Link										
03H	Set Upper-Memory Link										
59	<p><b><u>Get extended Error Information.</u></b> Return additional information about a DOS error, including the error class, locus, and recommended action.  <b>Input:</b> BX = DOS version number (0 for version 3.xx).  <b>Output:</b> AX = extended error code, BH = error class, BL = suggested action, and CH = locus.</p>										
5A	<p><b><u>Create Unique File.</u></b> Generate a unique (temporary) filename in a specified directory.  <b>Input:</b> DS:DX points to a ASCIIZ pathname, ending with a backslash (\); CX = desired file attribute.  <b>Output:</b> AX = error code if the Carry flag is set; otherwise, DS:DX points to the path with the new filename appended.</p>										
5B	<p><b><u>Create New File.</u></b> Try to create a new file, but fail if the filename already exists. This prevents you from overwriting an existing file.  <b>Input:</b> DS:DX points to an ASCIIZ string with the path and filename.  <b>Output:</b> AX = error code if the Carry flag is set.</p>										
5C	<p><b><u>Lock/Unlock File Access.</u></b> Lock or unlock a range of bytes in an opened file (networking).</p>										
5D	(Reserved)										
5E	<p><b><u>Functions designed for Networks.</u></b>  <b>Input:</b> AL = Subfunction</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Subfunction</th> </tr> </thead> <tbody> <tr><td>00</td><td>Get Local Machine Name</td></tr> <tr><td>01</td><td>Set Redirection Mode</td></tr> <tr><td>02</td><td>Set Printer Setup String</td></tr> <tr><td>03</td><td>Get Printer Setup String</td></tr> </tbody> </table>	Value	Subfunction	00	Get Local Machine Name	01	Set Redirection Mode	02	Set Printer Setup String	03	Get Printer Setup String
Value	Subfunction										
00	Get Local Machine Name										
01	Set Redirection Mode										
02	Set Printer Setup String										
03	Get Printer Setup String										
5F	<p><b><u>I/O Redirection.</u></b>  <b>Input:</b> AL = Subfunction</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Subfunction</th> </tr> </thead> <tbody> <tr><td>00</td><td>Get Redirection Mode</td></tr> <tr><td>01</td><td>Set Redirection Mode</td></tr> <tr><td>02</td><td>Get Redirection List Entry</td></tr> <tr><td>03</td><td>Redirect Device</td></tr> </tbody> </table>	Value	Subfunction	00	Get Redirection Mode	01	Set Redirection Mode	02	Get Redirection List Entry	03	Redirect Device
Value	Subfunction										
00	Get Redirection Mode										
01	Set Redirection Mode										
02	Get Redirection List Entry										
03	Redirect Device										

	04	Cancel Device Redirection																				
60		(Reserved)																				
61		(Reserved)																				
62		<b>Get Program Segment Prefix (PSP) Address.</b> <b>Output:</b> BX = PSP address of the current program.																				
63		(Reserved)																				
64		(Reserved)																				
65		<b>Subfunctions deal with Extended Country information.</b> <b>Input:</b> AL = <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>value</th> <th>subfunction</th> </tr> </thead> <tbody> <tr> <td>01H</td> <td>Get general international information</td> </tr> <tr> <td>02H</td> <td>Get pointer to uppercase table</td> </tr> <tr> <td>04H</td> <td>Get pointer to filename uppercase table</td> </tr> <tr> <td>05H</td> <td>Get pointer to filename terminator table</td> </tr> <tr> <td>06H</td> <td>Get pointer to collating table</td> </tr> <tr> <td>07H</td> <td>Get pointer to double-byte character set vector</td> </tr> <tr> <td>20H</td> <td>Capitalize character</td> </tr> <tr> <td>21H</td> <td>Capitalize string</td> </tr> <tr> <td>22H</td> <td>Capitalize ASCIIZ string</td> </tr> </tbody> </table> <b>Output:</b>	value	subfunction	01H	Get general international information	02H	Get pointer to uppercase table	04H	Get pointer to filename uppercase table	05H	Get pointer to filename terminator table	06H	Get pointer to collating table	07H	Get pointer to double-byte character set vector	20H	Capitalize character	21H	Capitalize string	22H	Capitalize ASCIIZ string
value	subfunction																					
01H	Get general international information																					
02H	Get pointer to uppercase table																					
04H	Get pointer to filename uppercase table																					
05H	Get pointer to filename terminator table																					
06H	Get pointer to collating table																					
07H	Get pointer to double-byte character set vector																					
20H	Capitalize character																					
21H	Capitalize string																					
22H	Capitalize ASCIIZ string																					
66		<b>Get or Set Code Page.</b> DOS reads or sets the current code page. <b>Input:</b> BX = code page if AL = 02H. <b>Output:</b> If successful, BX = active code page, DX = default code page.																				
67		<b>Set Handle Count.</b> Sets the maximum number of handles available to the calling programs. (Effectively, this only reduces the number available.) <b>Input:</b> BX = number of desired handles.																				
68		<b>Commit File Handle.</b> Forces all data in DOS buffers associated with a specific handle to be written immediately to the specified device. If the handle points to a file which has been updated, the file's directory is also updated.																				
69		<b>Get/Set Disk Serial Number.</b> Reads or writes the volume label and serial number of a specified disk. <b>Input:</b> BL = drive, DS:DX = segment:offset disk serial number info																				
6C		<b>Extended Open/Create function.</b> Combines functions "create", "open", and "commit" into a common function. <b>Input:</b> File attributes and filename <b>Output:</b> CS = 1 if file opened, 2 if file created, 3 if file replaced.																				

#### INTERRUPT 10h FUNCTIONS (VIDEO)

Function	Description
0	<b>Set Video Mode.</b> Set the video display to monochrome, text, graphics, or color mode. <b>Input:</b> AL = display mode.
1	<b>Set Cursor Lines.</b> Identify the starting and ending scan lines for the cursor. <b>Input:</b> CH = cursor starting line, and CL = cursor ending line.
2	<b>Set Cursor Position.</b> Position the cursor on the screen. <b>Input:</b> BH = video page. DH = row, and DL = column.
3	<b>Get Cursor Position.</b> Get the cursor's screen position and its size. <b>Input:</b> BH = video page. <b>Output:</b> CH = cursor starting line. CL = cursor ending line, DH = cursor row, DL = cursor column, AH = status.
4	<b>Read Light Pen.</b> Read the position and status of the light pen. <b>Output:</b> CH = pixel row, BX = pixel column, DH = character row, and DL = character column, AH = status.
5	<b>Set Display Page.</b> Select the video page to be displayed. <b>Input:</b> AL = desired page number.
6	<b>Scroll Window Up.</b> Scroll a window on the current video page upward, replacing scrolled lines with blanks. <b>Input:</b> AL = number of lines to scroll, BH = attribute for scrolled lines, CX = upper left corner row and column, and DX = lower right row and column.
7	<b>Scroll Window Down.</b> Scroll a window on the current video page downward, replacing scrolled lines with blanks. <b>Input:</b> AL = number of lines to scroll, BH = attribute for scrolled lines, CX = upper left corner row and column, and DX = lower right row and column.

8	<b>Read Character and Attribute.</b> Read the character and its attribute at the current cursor position. <b>Input:</b> BH = display page, AH = attribute byte, and AL = ASCII character code.
9	<b>Write Character and Attribute.</b> Write the character and its attribute at the current cursor position. <b>Input:</b> AL = ASCII character, BH = video page, BL = attribute or color, and CX = repetition factor.
0A	<b>Write Character.</b> Write a character only (no attribute) at the current cursor position. <b>Input:</b> AL = ASCII character, BH = video page, BL = attribute, and CX = replication factor.
0B	<b>Set Color Palette.</b> Select a group of available colors for the EGA or VGA adapter. <b>Input:</b> BL = color valve, and BH = color palette ID.
0C	<b>Write Graphics Pixel.</b> Write a graphics pixel when in color graphics mode. <b>Input:</b> AL = pixel value, CX = x coordinate, and DX = y coordinate.
0D	<b>Read Graphics Pixel.</b> Read the color of a single graphics pixel at a given location. <b>Input:</b> CX = x coordinate, and DX = y coordinate. <b>Output:</b> AL = pixel color.
0E	<b>Write Character.</b> Write a character to the screen and advance the cursor. <b>Input:</b> AL = ASCII character code, BH = video page, BL = attribute or color
0F	<b>Get current Video Mode.</b> Get the current video mode. <b>Output:</b> AL = video mode, and BH = active video page.
10	<b>Set Video Palette.</b> (EGA and VGA) Set the video palette register, border color, or blink/intensity bit. <b>Input:</b> AL = function code (00 = set palette register, 01 = set border color, 02 = set palette and border color, 03 = set/reset blink/intensity bit), BH = color, BL = palette register to sets. If AL = 02, ES:DX points to a color list.
11	<b>Character Generator.</b> Select the character size for the EGA or VGA display. For example, an 8 by 8 font is used for the 43-line display, and an 8 by 14 font is used for the 25-line display.
12	<b>Alternate Select Function.</b> Return technical information about the EGA or VGA display.
13	<b>Write String.</b> (AT BIOS and later) Write a string of text to the video display. <b>Input:</b> AL = mode, BH = page, BL = attribute, CX = length of string, DH = row, DL = column, and ES:BP points to the string.

#### INTERRUPT 16h FUNCTIONS (KEYBOARD)

Function	Description																
00	<b>Read Character.</b> Wait for an input character and keyboard scan code. <b>Output:</b> AH = scan code, and AL = ASCII character.																
01	<b>Get Keyboard Status.</b> Find out if a character is waiting in the keyboard typeahead buffer. If one is, return the scan code in AH and the ASCII code in AL, and clear the Zero flag. If no key is waiting, set the Zero flag. (Note: the same character will remain in the keyboard buffer.)																
02	<b>Get Keyboard Flags.</b> Return the keyboard flag byte stored in low RAM in AL. The meaning of each bit is: <table border="1" style="margin-left: 20px;"> <tr> <td>Insert on</td> <td>Caps Lock on</td> <td>Num Lock on</td> <td>Scroll Lock on</td> <td>Alt key down</td> <td>Ctrl key down</td> <td>Left shift down</td> <td>Right shift down</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> </tr> </table>	Insert on	Caps Lock on	Num Lock on	Scroll Lock on	Alt key down	Ctrl key down	Left shift down	Right shift down	1	1	1	1	1	1	1	1
Insert on	Caps Lock on	Num Lock on	Scroll Lock on	Alt key down	Ctrl key down	Left shift down	Right shift down										
1	1	1	1	1	1	1	1										
10	Extended keyboard input (F11 + F12)																