

INT Service	What it Does	Setup for Service	Return Values
INT 21 1	Keyboard Input with Echo	AH = 01	AL = character from standard input device
INT 21 2	Display Output	AH = 02 DL = character to output	Returns nothing
INT 21 7	Direct Console Input Without Echo	AH = 07	on return: AL = character from STDIN - waits for keyboard input until keystroke is ready - character is not echoed to STDOUT - returns 0 for extended keystroke, then function must be called again to return scan code
INT 21 8	Console Input Without Echo	AH = 08	on return: AL = character from STDIN
INT 21 9	Print String	AH = 09 DS:DX = pointer to string ending in "\$"	Returns nothing
INT 21 A	Buffered Keyboard Input	AH = 0A DS:DX = pointer to input buffer	Returns nothing
INT 21 1A	Set Disk Transfer Address (DTA)	AH = 1A DS:DX = pointer to disk transfer address ("DTA")	Returns nothing
INT 21 3C	Create File Using Handle	AH = 3C CX = file attribute DS:DX = pointer to ASCIIZ path name	CF = 0 if successful CF = 1 if error AX = files handle if successful AX = error code if failure
INT 21 41	Delete File	AH = 41h DS:DX = pointer to an ASCIIZ filename	AX = error code if CF set
INT 21 4C	Terminate Process With Return Code	AH = 4C AL = return code (for batch files) AL=00=>No error	returns nothing
INT 21 4E	Find First Matching File	AH = 4E CX = attribute used during search (Use 10h) DS:DX = pointer to ASCIIZ filespec including wildcards	AX = error code if CF set "DTA" = data returned from call in the format: Offset Size Description 00 <u>byte</u> attribute of search 01 <u>byte</u> drive letter used 02 <u>11bytes</u> search name used 0C <u>byte</u> attribute of search 0D <u>word</u> directory entry number 0F <u>word</u> starting cluster number 11 <u>word</u> reserved 13 <u>word</u> starting cluster number of current directory

				15 <u>byte</u> attribute of matching file 16 <u>word</u> file time 18 <u>word</u> file date 1A <u>word</u> file size 1E <u>13bytes</u> ASCIIZ filename and extension in the form NAME.EXT
INT 21	4F	Find Next Matching File	AH = 4F DS:DX = unchanged from previous function 4E	AX = error code if CF set
INT 10	1	Set Cursor Type	AH = 01 CH = cursor starting scan line (cursor top) (low order 5 bits) CL = cursor ending scan line (cursor bottom) (low order 5 bits)^	returns nothing
INT 10	2	Set Cursor Position	AH = 02 BH = page number (0 for graphics modes) DH = row DL = column	returns nothing
INT 10	6	Scroll Window Up	AH = 06 AL = number of lines to scroll, previous lines are blanked, if 0 or AL > screen size, window is blanked BH = attribute to be used on blank line CH = row of upper left corner of scroll window CL = column of upper left corner of scroll window DH = row of lower right corner of scroll window DL = column of lower right corner of scroll window	returns nothing
INT 10	7	Scroll Window Down	AH = 07 AL = number of lines to scroll, previous lines are blanked, if 0 or AL > screen size, window is blanked BH = attribute to be used on blank line CH = row of upper left corner of scroll window CL = column of upper left corner of scroll window DH = row of lower right corner of scroll window DL = column of lower right corner of scroll window	returns nothing
INT 10	8	Read Character and Attribute at Cursor Position	AH = 08 BH = display page	on return: AH = attribute of character (alpha modes only) AL = character at cursor position
INT 10	9	Write Character and Attribute at Cursor Position	AH = 09 AL = ASCII character to write BH = display page (or mode 13h, background pixel value) BL = character attribute (text) foreground color (graphics) CX = count of characters to write (CX >= 1)	returns nothing - does not move the cursor

INT 10	13	Write String (BIOS versions from 1/10/86)	AH = 13h AL = write mode (see bit settings below) = 0 string is chars only, attribute in BL, cursor not moved = 1 string is char only, attribute in BL, cursor moved = 2 string contains chars and attributes, cursor not moved = 3 string contains chars and attributes, cursor moved BH = video page number BL = attribute if mode 0 or 1 (AL bit 1=0) CX = length of string (ignoring attributes) DH = row coordinate DL = column coordinate ES:BP = pointer to string	returns nothing
INT 16	1	Get Keyboard Status	AH = 01	on return: ZF = 0 if a key pressed (even Ctrl-Break) AX = 0 if no scan code is available AH = ~scan code~ AL = ASCII character or zero if special function key
INT 33	1	Show Mouse Cursor	AX = 01	returns nothing
INT 33	2	Hide Mouse Cursor	AX = 02	returns nothing
INT 33	3	Get Mouse Position and Button Status	AX = 03	on return: CX = horizontal (X) position (0..639) DX = vertical (Y) position (0..199) BX = button status: =3 -> Both Buttons Pressed =2 -> Right Button Pressed =1 -> Left Button Pressed