INT 10H

From Wikipedia, the free encyclopedia

INT 10h, **INT 10H** or **INT 16** is shorthand for BIOS interrupt call 10_{hex}, the 17th interrupt vector in an x86-based computer system. The BIOS typically sets up a real mode interrupt handler at this vector that provides video services. Such services include setting the video mode, character and string output, and graphics primitives (reading and writing pixels in graphics mode).

To use this call, load AH with the number of the desired subfunction, load other required parameters in other registers, and make the call. INT 10h is fairly slow, so many programs bypass this BIOS routine and access the display hardware directly. Setting the video mode, which is done infrequently, can be accomplished by using the BIOS, while drawing graphics on the screen in a game needs to be done quickly, so direct access to video RAM is more appropriate than making a BIOS call for every pixel.

Furthermore, on a modern x86 system, BIOS calls can only be performed in Real mode, or Virtual 8086 mode. v8086 is not an option in Long mode. This means that a modern operating system, which operates in Protected mode (32 bit), or Long mode (64 bit), would need to switch into real mode and back to call the BIOS - a hugely expensive operation. Although most modern systems typically use device drivers that directly set the video mode, it is not feasible for hobbyist systems to have a device driver for every video card - a problem that also plagues older, unsupported systems such as Windows 98. Such systems instead can drop into Real mode to switch the video mode, then draw to the framebuffer directly.

In EFI 1.x systems, the INT 10H and the VESA BIOS Extensions (VBE) are replaced by the EFI UGA protocol. In widely used UEFI 2.x systems, the INT 10H and the VBE are replaced by the UEFI GOP.[1][2]

List of supported functions

The list is incomplete; use Ralf Brown's list for comprehensive information. Please only add IBM/PC or other common standard functions. 00h through 0fh are CGA.

Function	Function code	Parameters	Return
Set video mode	AH=00h	AL = video mode (http://www.columbia.edu/~em36/wpdos/videomodes.txt)	AL = video mode flag / CRT controller mode byte
Set text-mode cursor shape	AH=01h	CH = Scan Row Start, CL = Scan Row End Normally a character cell has 8 scan lines, 0–7. So, CX=0607h is a normal underline cursor, CX=0007h is a full-block cursor. If bit 5 of CH is set, that often means "Hide cursor". So CX=2607h is an invisible cursor. Some video cards have 16 scan lines, 00h-0Fh. Some video cards don't use bit 5 of CH. With these, make Start>End (e.g. CX=0706h)	
Set cursor position	AH=02h	BH = Page Number, DH = Row, DL = Column	
Get cursor position and shape	AH=03h	BH = Page Number	AX = 0, CH = Start scan line, CL = End scan line, DH = Row, DL = Column
Read light pen position (Does not work on VGA systems)	AH=04h		AH = Status (0=not triggered, 1=triggered), BX = Pixel X, CH = Pixel Y, CX = Pixel line number for modes 0Fh-10h, DH = Character Y, DL = Character X
Select active display page	AH=05h	AL = Page Number	
Scroll up window	AH=06h	AL = lines to scroll (0 = clear, CH, CL, DH, DL are used), BH = Background Color and Foreground color. BH = 43h, means that background color is red and foreground color is cyan. Refer the BIOS color attributes CH = Upper row number, CL = Left column number, DH = Lower row number, DL = Right column number	
Scroll down window	AH=07h	like above	
Read character and attribute at cursor position	AH=08h	BH = Page Number	AH = Color, AL = Character

Write character and attribute at cursor position	AH=09h	AL = Character, BH = Page Number, BL = Color, CX = Number of times to print character	
Write character only at cursor position	AH=0Ah	AL = Character, BH = Page Number, CX = Number of times to print character	
Set background/border color	AH=0Bh, BH = 00h	BL = Background/Border color (border only in text modes)	
Set palette	AH=0Bh, BH = 01h	BL = Palette ID (was only valid in CGA, but newer cards support it in many or all graphics modes)	
Write graphics pixel	AH=0Ch	AL = Color, BH = Page Number, CX = x, DX = y	
Read graphics pixel	AH=0Dh	BH = Page Number, CX = x, DX = y	AL = Color
Teletype output	AH=0Eh	AL = Character, BH = Page Number, BL = Color (only in graphic mode)	
Get current video mode	AH=0Fh		AL = Video Mode, AH = number of character columns, BH = active page
Change text mode character set[3]	AH=11h	BH = Number of bytes per character, CX = Number of characters to change, DX = Starting character to change, ES:BP = Offset of character data	
Write string (EGA+, meaning PC AT minimum)	AH=13h	AL = Write mode, BH = Page Number, BL = Color, CX = Number of characters in string, DH = Row, DL = Column, ES:BP = Offset of string	
set VESA-Compliant video modes, beginning at 640 by 480 and reaching 1280 by 1024 with 256 colors	AX=4f02h	BX = video mode, if Sign bit (bit 15) set, video memory will not be refreshed	
Other VESA VBE commands	AX=4F00h to 4F15h	See spec	See spec

See also

- BIOS interrupt call
- Mode 13h
- VESA BIOS Extensions
- Ralf Brown's Interrupt List

References

1. "What is efifb? — The Linux Kernel documentation" (https://www.kernel.org/doc/html/latest/fb/efifb.html). *www.kernel.org*. Retrieved 2020-11-24.
2. "What is vesafb? — The Linux Kernel documentation" (https://www.kernel.org/doc/html/latest/fb/vesafb.html). *www.kernel.org*. Retrieved 2020-11-24.
3. "A Font changing routine" (http://www.fysnet.net/font.htm). *Forever Young Software*. Retrieved March 8, 2020.

- INT 10h from Ralf Brown's Interrupt List, online version (http://www.ctyme.com/intr/int-10.htm)
- INT 10h on www.ousob.com (https://web.archive.org/web/20121212035831/http://www.ousob.com/ng/asm/ng6f862.php)