BIOS interrupt call

From Wikipedia, the free encyclopedia
(Redirected from BIOS call)

BIOS Interrupt Calls are a facility that DOS programs, and some other software such as boot loaders, use to invoke the BIOS's facilities. Some operating systems also use the BIOS to probe and initialise hardware resources during their early stages of booting.

Contents

Interrupt Table

Interrupt	Description
INT 00h	CPU: Division by Zero
INT 01h	CPU: Single Step for debugging
INT 02h	CPU: NMI, used e.g. by POST for memory errors
INT 03h	CPU: Breakpoint for debugging
INT 04h	CPU: Numeric Overflow
INT 05h	Print Screen
INT 08h	IRQ0: Called by system timer 18.2 times per second
INT 09h	IRQ1: Called by keyboard
INT 0Bh	IRQ3: Called by 2nd serial port COM2
INT 0Ch	IRQ4: Called by 1st serial port COM1

INT 0Dh	IRQ5: Called by hard disk controller (PC/XT) or 2nd parallel port LPT2 (AT)
INT 0Eh	IRQ6: Called by floppy disk controller
INT 0Fh	IRQ7: Called by 1st parallel port LPT1 (printer)

INT 10h	Video Services
	AH=00h Set Video Mode
	AH=01h Set Cursor Shape
	AH=02h Set Cursor Position
	AH=03h Get Cursor Position And Shape
	AH=04h Get Light Pen Position
	AH=05h Set Display Page
	AH=06h Clear/Scroll Screen Up
	AH=07h Clear/Scroll Screen Down
	AH=08h Read Character and Attribute at Cursor
	AH=09h Write Character at Cursor
	AH=0Ah Write Character and Attribute at Cursor
	AH=0Bh Set Border Color
	AH=0Eh Write Character in TTY Mode
	AH=0Fh Get Video Mode
	AH=13h Write String

INT 11h	Equipment Installed
INT 12h	Get Conventional Memory Size

INT 13h	Low Level Disk Services
	AH=00h Reset Disk Drives
	AH=01h Check Drive Status
	AH=02h Read Sectors From Drive
	AH=03h Write Sectors To Drive
	AH=04h Verifies Sectors On Drive
	AH=05h Format Track On Drive
	AH=08h Get Drive Parameters
	AH=09h Init Fixed Drive Parameters
	AH=0Ch Seek To Specified Track
	AH=0Dh Reset Fixed Disk Controller
	AH=15h Get Drive Type
	AH=16h Get Floppy Drive Media Change Status

INT 14h	Serial I/O
	AH=00h Serial Port Initialization
	AH=01h Transmit Character
	AH=02h Receive Character
	AH=03h Status

INT 15h	Miscellaneous
	AH=4FH Keyboard Intercept
	AH=83H Event Wait
	AH=84H Read Joystick
	AH=85H Sysreq Key Callout
	AH=86H Wait
	AH=87H Move Block
	AH=88H Get Extended Memory Size
	AH=C0H Get System Parameters
	AH=C1H Get Extended BIOS Data Area Segment
	AH=C2H Pointing Device Functions

INT 16h	Keyboard
	AH=00h Read Character
	AH=01h Read Input Status
	AH=02h Read Keyboard Shift Status
	AH=10h Read Character Extended
	AH=11h Read Input Status Extended
	AH=12h Read Keyboard Shift Status Extended

INT 17h	Print Services
	AH=00h Print Character to Printer
	AH=01h Initialize Printer
	AH=02h Check Printer Status

INT 18h	This interrupt will be called if INT 19h fails, in early IBM BIOS version then ROM Basic was loaded
INT 19h	After POST this interrupt is used by BIOS to load the operating system

INT 1Ah	Real Time Clock Services
	AH=00h Read RTC
	AH=01h Set RTC
	AH=02h Read RTC Time
	AH=03h Set RTC Time
	AH=04h Read RTC Date
	AH=05h Set RTC Date
	AH=06h Set RTC Alarm
	AH=07h Reset RTC Alarm

INT 1Bh	Called by Break key
INT 1Ch	Software Timer Interrupt, called by INT 08h
INT 1Dh	Address pointer: VPT = Video Parameter Table
INT 1Eh	Address pointer: DPT = Diskette Parameter Table
INT 1Fh	Address pointer: VGCT = Video Graphics Character Table
INT 41h	Address pointer: FDPT = Fixed Disk Paramter Table (1st hard drive)

INT 46h	Address pointer: FDPT = Fixed Disk Paramter Table (2nd hard drive)
INT 4Ah	Called by RTC for alarm
INT 70h	IRQ8: Called by RTC
INT 74h	IRQ12: Called by mouse
INT 75h	IRQ13: Called by math coprocessor
INT 76h	IRQ14: Called by primary IDE controller
INT 77h	IRQ15: Called by secondary IDE controller

INT 13h: Low Level Disk Services

Drive Table

DH = 00h	1st floppy disk ("drive A:")
DH = 01h	2nd floppy disk ("drive B:")
DH = 80h	1st hard disk
DH = 81h	2nd hard disk

Function Table

AH = 00h		Reset Disk Drives
AH = 01h		Check Drive Status
AH = 02h		Read Sectors From Drive
AH = 03h		Write Sectors To Drive
AH = 04h		Verify Sectors
AH = 05h		Format Track
AH = 08h		Read Drive Parameters
AH = 09h	HD	Initialize Disk Controller
AH = 0Ah	HD	Read Long Sectors From Drive
AH = 0Bh	HD	Write Long Sectors To Drive
AH = 0Ch	HD	Move Drive Head To Cylinder
AH = 0Dh	HD	Reset Disk Drives
AH = 0Eh	PS/2	Controller Read Test
AH = 0Fh	PS/2	Controller Write Test
AH = 10h	HD	Test Whether Drive Is Ready
AH = 11h	HD	Recalibrate Drive
AH = 12h	PS/2	Controller RAM Test
AH = 13h	PS/2	Drive Test

AH = 14h	HD	Controller Diagnostic
AH = 15h		Read Drive Type
AH = 16h	FD	Detect Media Change
AH = 17h	FD	Set Media Type For Format (used by DOS versions <= 3.1)
AH = 18h	FD	Set Media Type For Format (used by DOS versions >= 3.2)
AH = 41h	EXT	Test Whether Extensions Are Available
AH = 42h	EXT	Read Sectors From Drive
AH = 43h	EXT	Write Sectors To Drive
AH = 44h	EXT	Verify Sectors
AH = 45h	EXT	Lock/Unlock Drive
AH = 46h	EXT	Eject Drive
AH = 47h	EXT	Move Drive Head To Sector
AH = 48h	EXT	Read Drive Parameters
AH = 49h	EXT	Detect Media Change

Second column is empty == function may be used both for floppy and hard disk.
"FD" == for floppy disk only.
"HD" == for hard disk only.
"PS/2" == for hard disk on PS/2 system only.
"EXT" == part of the Int 13h Extensions which were written in the 1990s to support hard drives with more than 8 GBytes.

INT 13h AH=00h: Reset Disk Drives

Parameters:

AH	00h
DL	Drive Index

INT 13h AH=01h: Check Drive Status

Parameters:

AH	01h

Results:

	Return Code 00h Success 01h Invalid Command 02h Cannot Find Address Mark 03h Attempted Write On Write Protected Disk

	04h Sector Not Found 05h Reset Failed

INT 13h AH=02h: Read Sectors From Drive

Parameters:

AH	02h
AL	Sectors To Read Count
CX	Track + Sector / See remark
DH	Head
DL	Drive
ES:BX	Buffer Address Pointer

Results:

CF	Set On Error, Clear If No Error
AH	Return Code
AL	Actual Sectors Read Count

Remarks:
Register CX contains both the cylinder number (10 bits, possible values are 0 to 1023) and the sector number (6 bits, possible values are 1 to 63):

```
CX =       ---CH--- ---CL---
cylinder : 76543210 98
sector   :            543210
```

Examples of translation:

```
Turbo Pascal:
CX := ( ( cylinder and 255 ) shl 8 ) or ( ( cylinder and 768 ) shr 2 ) or sector;
cylinder := hi ( CX ) or ( ( lo ( CX ) and 192 ) shl 2 );
sector := CX and 63;
```

Addressing of Buffer should guarantee that the complete buffer is inside the given segment, i.e. (BX + size_of_buffer) <= 10000h. Otherwise the interrupt may fail with some BIOS or hardware versions.
Example: Assume you want to read 16 sectors (= 2000h Bytes) and your buffer starts at memory address 4FF00h. There are different ways to calculate the register values, e.g.:

```
ES = segment         = 4F00h
BX = offset          =  0F00h
sum = memory address = 4FF00h
would be a good choice because 0F00h + 2000h = 2F00h <= 10000h
ES = segment         = 4000h
BX = offset          =  FF00h
```

```
sum = memory address = 4FF00h
would be no good choice because FF00h + 2000h = 11F00h > 10000h
```

Function 02h of interrupt 13h may only read sectors of the first 16,450,560 sectors of your hard drive, to read sectors beyond the 8 GByte limit you should use function 42h of Int 13h Extensions. Another alternate may be DOS interrupt 25h which reads sectors *within* a partition.

INT 13h AH=08h: Read Drive Parameters

Parameters:

Registers	
AH	08h = function number for read_drive_parameters
DL	drive index (e.g. 1st HDD = 80h)

Results:

CF	Set On Error, Clear If No Error
AH	Return Code
DL	number of hard disk drives
DH	logical last index of heads = number_of - 1 (because index starts with 0)
CX	logical last index of cylinders = number_of - 1 (because index starts with 0) logical last index of sectors per track = number_of (because index starts with 1)

Remarks:
Logical values of function 08h may/should differ from physical CHS values of function 48h.
Result register CX contains both cylinders and sector/track values, see remark of function 02h.

INT 13h AH=0Ah: Read Long Sectors From Drive

The only difference between this function and function 02h (see above) is that function 0Ah reads 516 bytes per sector instead of only 512. The last 4 bytes contains the Error Correction Code ECC, a checksum of sector data.

INT 13h AH=41h: Check Extensions Present

Parameters:

Registers	
AH	41h = function number for extensions check
DL	drive index (e.g. 1st HDD = 80h)
BX	55AAh

Results:

CF	Set On Not Present, Clear If Present
AH	Error Code or Major Version Number
BX	AA55h
CX	Interface support bitmask: 1 - Device Access using the packet structure 2 - Drive Locking and Ejecting 4 - Enhanced Disk Drive Support (EDD)

INT 13h AH=42h: Extended Read Sectors From Drive

Parameters:

Registers	
AH	42h = function number for extended read
DL	drive index (e.g. 1st HDD = 80h)
DS:SI	segment:offset pointer to the DAP, see below

DAP : Disk Address Packet		
offset range	size	description
00h	1 byte	size of DAP = 16 = 10h
01h	1 byte	unused, should be zero
02h	1 byte	number of sectors to be read, 0..127 (= 7Fh)
03h	1 byte	unused, should be zero
04h..07h	4 bytes	segment:offset pointer to the memory buffer to which sectors will be transferred
08h..0Fh	8 bytes	absolute number of the start of the sectors to be read (1st sector of drive has number 0)

Results:

CF	Set On Error, Clear If No Error
AH	Return Code

INT 13h AH=48h: Extended Read Drive Parameters

Parameters:

Registers

AH	48h = function number for extended_read_drive_parameters
DL	drive index (e.g. 1st HDD = 80h)
DS:SI	segment:offset pointer to Result Buffer, see below

Result Buffer		
offset range	size	description
00h..01h	2 bytes	size of Result Buffer = 30 = 1Eh
02h..03h	2 bytes	information flags
04h..07h	4 bytes	physical number of cylinders = last index + 1 (because index starts with 0)
08h..0Bh	4 bytes	physical number of heads = last index + 1 (because index starts with 0)
0Ch..0Fh	4 bytes	physical number of sectors per track = last index (because index starts with 1)
10h..17h	8 bytes	absolute number of sectors = last index + 1 (because index starts with 0)
18h..19h	2 bytes	bytes per sector
1Ah..1Dh	4 bytes	optional pointer to Enhanced Disk Drive (EDD) configuration parameters which may be used for subsequent interrupt 13h Extension calls (if supported)

Results:

CF	Set On Error, Clear If No Error
AH	Return Code

Remark: Physical CHS values of function 48h may/should differ from logical values of function 08h.

INT 18h: Execute BASIC

Description:

This interrupt traditionally jumped to an implementation of BASIC stored in ROM. This call would typically be invoked if the BIOS was unable to identify any bootable volumes on startup. (At the time the original IBM PC was released in 1981, the BASIC in ROM was a key feature.) As time went on and BASIC was no longer shipped on all PCs, this interrupt would simply display an error message indicating that no bootable volume was found (famously, "No ROM BASIC", or more self-explanatory messages in later BIOS versions); in other BIOS versions it would prompt the user to insert a bootable volume and press a key, and then after the user did so it would loop back to the bootstrap loader to try booting again.

See also

- Interrupt
- INT 13
- Input/Output Base Address

External links

- http://www.embeddedarm.com/Manuals/EBIOS-UM.PDF // Embedded BIOS User's Manual
- http://www.23cc.com/free-fdisk/specs-edd11.pdf // Int 13h Extensions
- http://www.missl.cs.umd.edu/winint/index1.html
- http://hdebruijn.soo.dto.tudelft.nl/newpage/interupt/out-0100.htm
- http://home.arcor.de/wzwz.de/wiki/interrupt/i13_en.htm // Turbo Pascal examples for reading sectors
- http://www.ctyme.com/rbrown.htm // HTML version of Ralf Brown Interrupt List
- http://forums.techarena.in/showthread.php?t=389480 // BIOS Beep Codes

Retrieved from "http://en.wikipedia.org/wiki/BIOS_interrupt_call"

Categories: BIOS | Interrupts

- This page was last modified 04:20, 1 September 2006.
-