



# INTERRUPT VECTOR TABLE

## by Mark E. Donaldson

where it resides is the fourth byte which ranges from zero to F. The next four bytes are the interrupts location in that segment. As luck would have it, this interrupts location was the same as the location user by the first IBM PCs using DOS 2.0 long ago. Each BIOS/computer manufacturer spreads them out wherever in memory they wish, but you should find most of them in your computer's segments zero, B and F.

**GET YOUR PC COMPATIBLE COMPUTER'S INTERRUPT VECTORS  
TYPE IN 000 TO 255 TO DISPLAY ITS CODE SEGMENT & ADDRESS  
INPUT IS IN DECIMAL - CODE SEGMENT AND IP IS IN HEX  
THEN PRESS ENTER TO START OVER OR ESC TO EXIT**

**INTERRUPT NUMBER = 02B      VECTOR ADDRESS = <sup>CS</sup> F000:<sup>IP</sup>FF53**

**ACTUAL 20 BIT INTERRUPT ADDRESS = 000FFFF53 HEX  
FIRST 4 NUMBERS IS SEGMENT REST IS LOCATION**

I used decimal 0 through decimal 255 for the input and displayed the CS code segment of its location in memory in hex followed by its IP in memory in hex second which is pretty standard these days. I reverse the CS and IP order in the display above as it is easier to understand. My EDMOD program uses all hex values for all variables including segments, memory locations and values displayed, plus those beneath the mouse or keyboard moved cursor on each page in MODIFY mode.

At the bottom of each page in my program I display the following for the best textbook, all 1200+ pages ever written on interrupts. It is an incredible and thorough work of art!

**The best and most comprehensive textbook on PC Interrupts is:**

**PC INTERRUPTS  
A PROGRAMMER'S REFERENCE TO BIOS, DOS AND THIRD PARTY CALLS  
ISBN 0-201-57797-6**

**This 1200 plus page and nearly 4 pound textbook, list price is  
\$32.95 and available at a discount from:**

**ABYSS BOOKS - phone toll free: (888) WWW-BOOK and ask for Maria.**

In the offered download below I included my Edmod.com program and its accompanying Edmod.doc program which explains how to use it. That way, when you get an interrupt's code segment, you can then use Edmod.com to go have a look at the actual interrupt routines in real memory's segments 0 through F. I also moved the inter.com working program up to segment 7000h where you can look at it with Edmod.com. Translating the interrupt vector tables CS, (code segment) and IP, (instruction pointer) to real memory addresses is a pain in the fanny, but I did it. I had to use Eric Isaacson's brilliant A386.COM assembler with the 32 bit registers to do it. It may be ordered from Eric on my homepage's [BEGINNER] group, number 8 and is the best Intel 386 through Pentium assembler in the world. It is only \$80 and will assemble 70K of source code in a second or so.

Translating the interrupt vector tables code segments and instruction pointers to real memory is done by:

1. Left shifting by 4 bits, (or multiply by 16 decimal), 10 hex, the interrupt vector's code

# INTERRUPT VECTOR TABLE

## by Mark E. Donaldson

segment. Unless you are a wizard, this requires a 32 bit assembler in most cases.

2. Then adding the interrupt vector's instruction pointer to the above and then displaying it.
3. For instance, if the breakpoint interrupt translates to 00002DB0 hex, it is in sector zero at location 2DB0 hex.

IF you wish to download the inter.zip demonstration program select the underlined line below. It includes: inter.com the working program, inter.asm its source code, two tiny batch files for editing and assembling the source code and the edmod.com and edmod.doc programs.

inter.zip = 28,633 bytes zipped - unzips to 64,235 bytes