%eax contains info of which system call is being invoked.

Interrupt (int) tells the hardware which trap type this is. Hardware then takes over and does work on behalf of the caller. **One important thing** the hardware does is to raise the *privilege level* of the CPU to kernel mode; on x86 this is usually means moving from a *CPL* *(Current Privilege Level)* of 3 (the level at which user applications run) to CPL 0 (in which the kernel runs). **Second thing** the hardware does is to transfer control to the trap vectors of the system. The hardware knows what code to run when a particular trap occurs because when the OS is booting, it informs the hardware the location of the code to run when a trap takes place.

SETGATE() set the *idt* array to point to the proper code to execute when various traps and interrupts occur. For system calls, it is called independently out of the loop. SETGATE() sets the values of an in-memory data structure. The most important is the **off** parameter, which tells the hardware where the trap handling code is. **Idt[T\_SYSCALL]** is passed in indicates where to run the code when a system call takes place.

The hardware is informed of the code to execute later in the boot sequence.

After the OS has carefully set up the trap handlers