

Interrupciones en Modo Protegido

Organizacion del Computador II

Departamento de Computacion
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Diapositivas de Javier Pimás

Segundo Cuatrimestre, 2015

Refresco de Memoria

- Soporta 256 tipos de interrupciones
- Se utiliza una tabla denominada IDT
- La IDT almacena descriptores de interrupcion
- El registro IDTR almacena la direccion de la IDT
- Ref: Intel Software developer's manual (vol. 3)
capítulo 6, interrupciones (tan sólo 10 pags)

IDTR and IDT

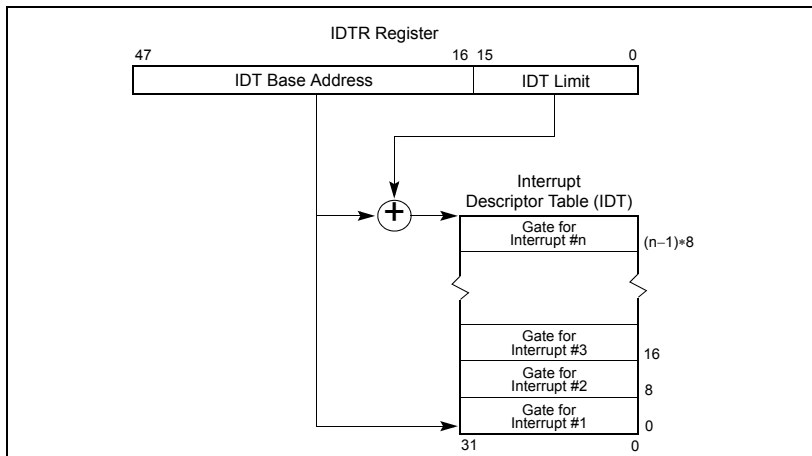
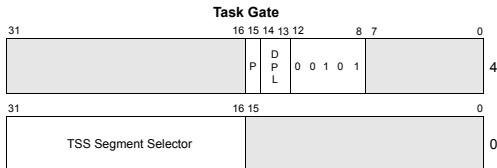


Figure 5-1. Relationship of the IDTR and IDT

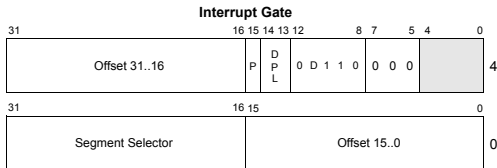
Gate Descriptors



DPL Descriptor Privilege Level
 Offset Offset to procedure entry point
 P Segment Present flag
 Selector Segment Selector for destination code segment
 D Size of gate: 1 = 32 bits; 0 = 16 bits

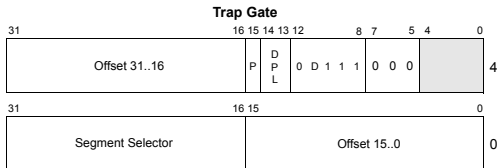
 Reserved

Figure 5-2. IDT Gate Descriptors



Ojo con **los** DPLs (hay 3 en juego)

- Campo DPL (privilegios necesarios para hacer INT n)
- "DPL" Segment Selector (debe ser 0)
- DPL descriptor apuntado por Segment Selector



Interrupt Procedure Call

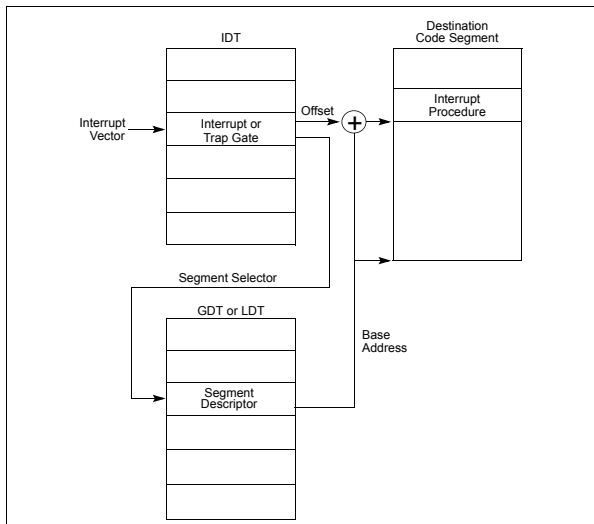


Figure 5-3. Interrupt Procedure Call

Interrupt Task Switch

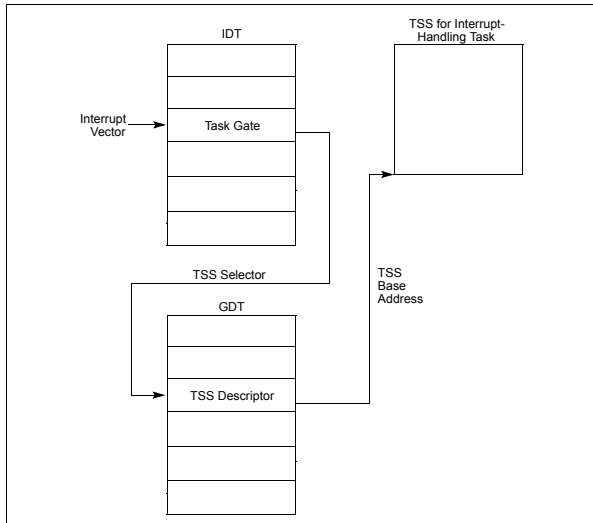


Figure 5-5. Interrupt Task Switch

Tipos de Interrupciones

- **Fault:** Excepción que puede corregirse permitiendo al programa retomar la ejecución de esa instrucción sin perder continuidad. El procesador guarda en la pila la dirección de la instrucción que produjo la falla.
- **Traps:** Excepción producida inmediatamente a continuación de una instrucción de trap. Algunas permiten al procesador retomar la ejecución sin perder continuidad. Otras no. El procesador guarda en la pila la dirección de la instrucción a ejecutarse luego de la instrucción trapeada.
- **Aborts:** Excepción que no siempre puede determinar la instrucción que la causó, ni permite recuperar la ejecución de la tarea que la causó. Reporta errores severos de hardware o inconsistencias en tablas del sistema.

Interrupt Table

Table 5-1. Protected-Mode Exceptions and Interrupts

Vector No.	Mnemonic	Description	Type	Error Code	Source
0	#DE	Divide Error	Fault	No	DIV and IDIV instructions.
1	#DB	RESERVED	Fault/ Trap	No	For Intel use only.
2	—	NMI Interrupt	Interrupt	No	Nonmaskable external interrupt.
3	#BP	Breakpoint	Trap	No	INT 3 instruction.
4	#OF	Overflow	Trap	No	INTO instruction.
5	#BR	BOUND Range Exceeded	Fault	No	BOUND instruction.
6	#UD	Invalid Opcode (Undefined Opcode)	Fault	No	UD2 instruction or reserved opcode. ¹
7	#NM	Device Not Available (No Math Coprocessor)	Fault	No	Floating-point or WAIT/FWAIT instruction.
8	#DF	Double Fault	Abort	Yes (zero)	Any instruction that can generate an exception, an NMI, or an INTR.
9		Coprocessor Segment Overrun (reserved)	Fault	No	Floating-point instruction. ²
10	#TS	Invalid TSS	Fault	Yes	Task switch or TSS access.
11	#NP	Segment Not Present	Fault	Yes	Loading segment registers or accessing system segments.

Interrupt Table

12	#SS	Stack-Segment Fault	Fault	Yes	Stack operations and SS register loads.
13	#GP	General Protection	Fault	Yes	Any memory reference and other protection checks.
14	#PF	Page Fault	Fault	Yes	Any memory reference.
15	—	(Intel reserved. Do not use.)		No	
16	#MF	x87 FPU Floating-Point Error (Math Fault)	Fault	No	x87 FPU floating-point or WAIT/FWAIT instruction.
17	#AC	Alignment Check	Fault	Yes (Zero)	Any data reference in memory. ³

Table 5-1. Protected-Mode Exceptions and Interrupts (Contd.)

18	#MC	Machine Check	Abort	No	Error codes (if any) and source are model dependent. ⁴
19	#XM	SIMD Floating-Point Exception	Fault	No	SSE/SSE2/SSE3 floating-point instructions ⁵
20-31	—	Intel reserved. Do not use.			
32-255	—	User Defined (Non-reserved) Interrupts	Interrupt		External interrupt or INT <i>n</i> instruction.

NOTES:

1. The UD2 instruction was introduced in the Pentium Pro processor.
2. Processors after the Intel386 processor do not generate this exception.
3. This exception was introduced in the Intel486 processor.
4. This exception was introduced in the Pentium processor and enhanced in the P6 family processors.
5. This exception was introduced in the Pentium III processor.

Stack

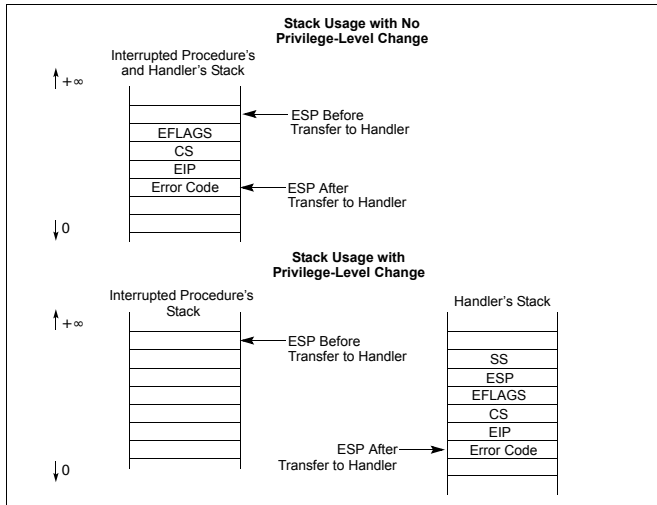


Figure 5-4. Stack Usage on Transfers to Interrupt and Exception-Handling Routines

Ayudas

- ❶ `idt.h`: Descripción de las estructuras.
- ❷ `idt.c`: Estructura de la IDT con cada una de sus entradas
- ❸ `isr.h`: Descripción de las funciones handlers
- ❹ `isr.asm`: Código de los handlers
- `IDT_ENTRY(numero)`: Permite declarar una entrada en la IDT, para la utilizar el handler de nombre `_isrX`
- `void inicializar_idt()`: Función llamada desde el kernel para inicializar las entradas en la idt

IDT

- Struct de descriptor de IDT

```
typedef struct str_idt_descriptor {  
    unsigned short idt_length;  
    unsigned int idt_addr;  
} __attribute__((__packed__)) idt_descriptor;
```

- Struct de una entrada de la IDT

```
typedef struct str_idt_entry_fld {  
    unsigned short offset_0_15;  
    unsigned short segsel;  
    unsigned short attr;  
    unsigned short offset_16_31;  
} __attribute__((__packed__, aligned (8))) idt_entry;
```

Gracias!!!

Recuerden:

Ref: Intel Software developer's manual (vol. 3)
capítulo 6, interrupciones

Ejercicios

- 1 Construir un handler para la interrupción 13 que presente en pantalla la leyenda “General Protection”
- 2 Configurar la entrada correspondiente a la IDT para resolver la interrupción 13
- 3 Armar una rutina que escriba en memoria, utilizando el segmento de video, pero fuera del mismo
- 4 Disfrutar como explota en pedazos