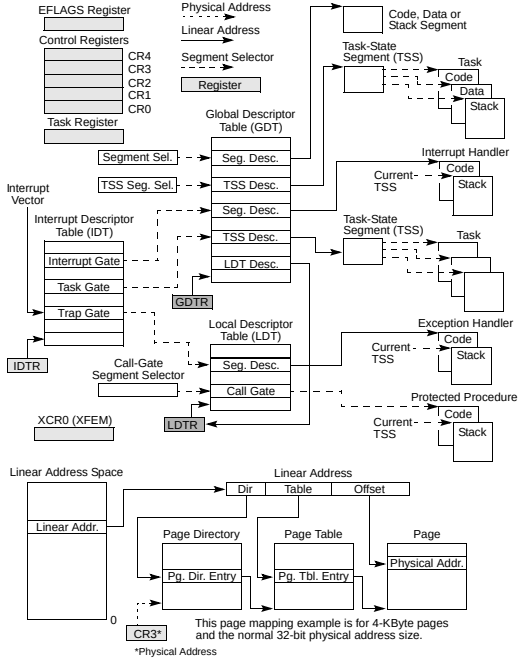
Interrupciones Básicas

Programación de Sistemas Operativos

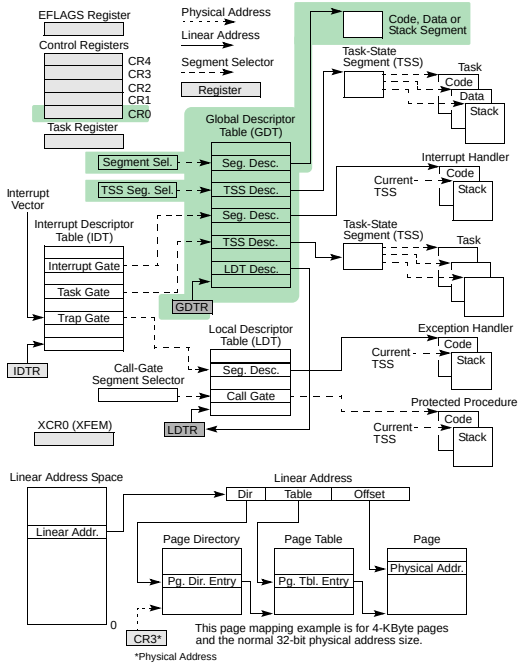
David Alejandro González Márquez

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

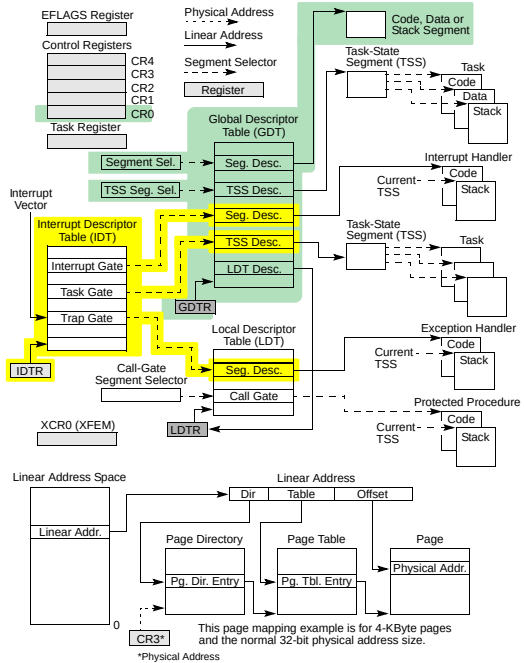
Usted ...


EFLAGS Register
Control Registers
CR4
CR3
CR2
CR1
CR0
Task Register
Physical Address
Linear Address
Segment Selector
Register
Global Descriptor Table (GDT)
Segment Sel.
TSS Seg. Sel.
Seg. Desc.
TSS Desc.
Seg. Desc.
TSS Desc.
LDT Desc.
GDTR
Interrupt Vector
Interrupt Descriptor Table (IDT)
Interrupt Gate
Task Gate
Trap Gate
IDTR
Local Descriptor Table (LDT)
Call-Gate Segment Selector
Seg. Desc.
Call Gate
LDTR
XCR0 (XFEM)
Code, Data or Stack Segment
Task-State Segment (TSS)
Task
Code
Data
Stack
Interrupt Handler
Code
Current-TSS
Stack
Task-State Segment (TSS)
Task
Exception Handler
Code
Current-TSS
Stack
Protected Procedure
Code
Current-TSS
Stack
Linear Address Space
Linear Addr.
0
CR3*
*Physical Address
Linear Address
Dir
Table
Offset
Page Directory
Pg. Dir. Entry
Page Table
Pg. Tbl. Entry
Page
Physical Addr.
This page mapping example is for 4-KByte pages and the normal 32-bit physical address size.

Usted estabá aquí



Usted estará aquí



IDT - Interruption Descriptor Table

- Soporta 256 tipos de interrupciones

IDT - Interruption Descriptor Table

- Soporta 256 tipos de interrupciones
- Se utiliza una tabla denominada `IDT`

IDT - Interruption Descriptor Table

- Soporta 256 tipos de interrupciones
- Se utiliza una tabla denominada `IDT`
- La `IDT` almacena **descriptores de interrupción**

IDT - Interruption Descriptor Table

- Soporta 256 tipos de interrupciones
- Se utiliza una tabla denominada `IDT`
- La `IDT` almacena **descriptores de interrupción**
- El registro `IDTR` almacena la dirección de la `IDT`

IDTR and IDT

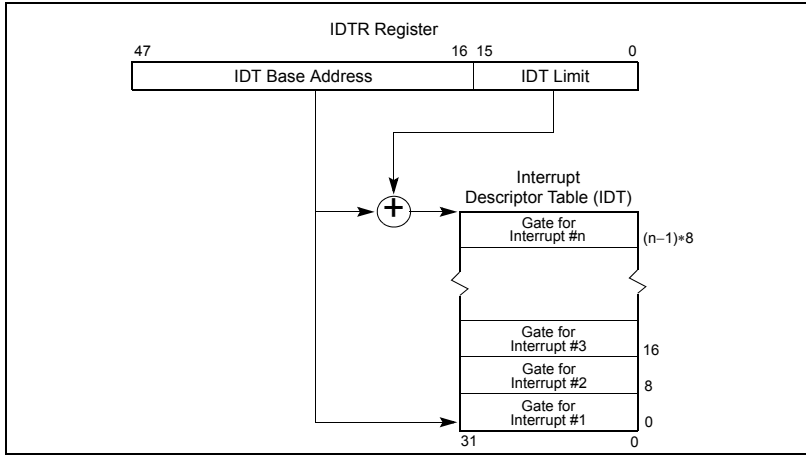
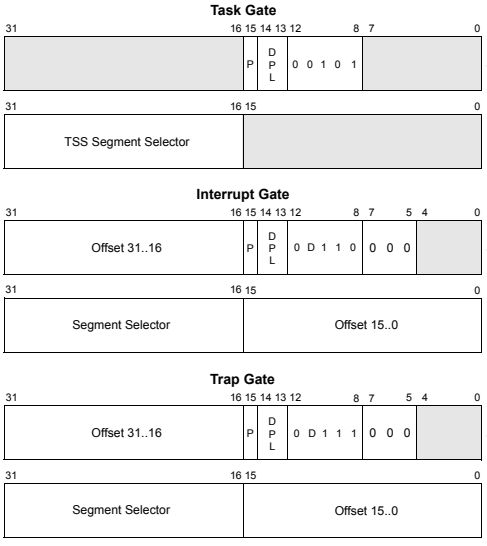


Figure 5-1. Relationship of the IDTR and IDT

Gate Descriptors

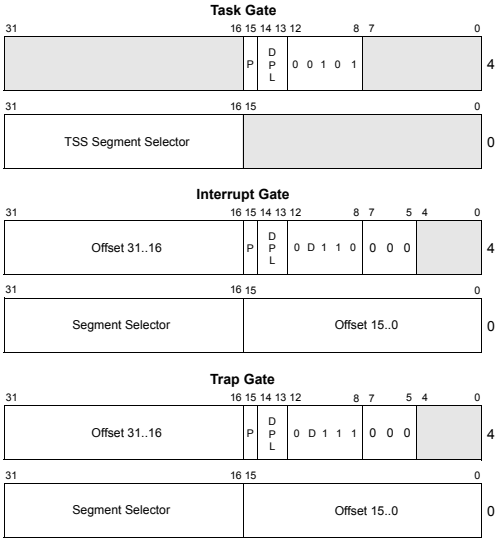


DPL	Descriptor Privilege Level
Offset	Offset to procedure entry point
P	Segment Present flag
Selector	Segment Selector for destination code segment
D	Size of gate: 1 = 32 bits; 0 = 16 bits

Reserved

Figure 5-2. IDT Gate Descriptors

Gate Descriptors



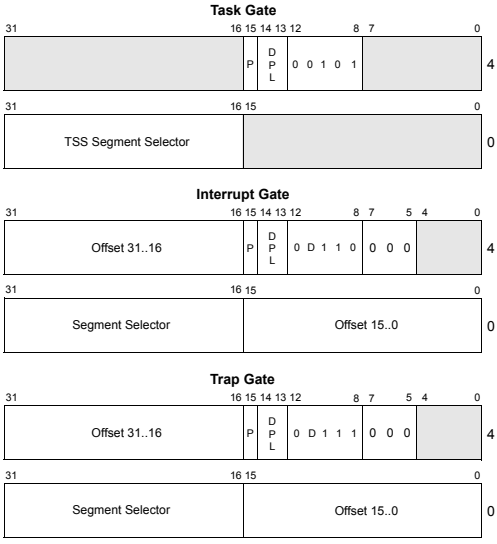
DPL	Descriptor Privilege Level
Offset	Offset to procedure entry point
P	Segment Present flag
Selector	Segment Selector for destination code segment
D	Size of gate: 1 = 32 bits; 0 = 16 bits

Reserved

Figure 5-2. IDT Gate Descriptors

- *Task Gate:* Inicia una tarea para anteder la interrupción.

Gate Descriptors



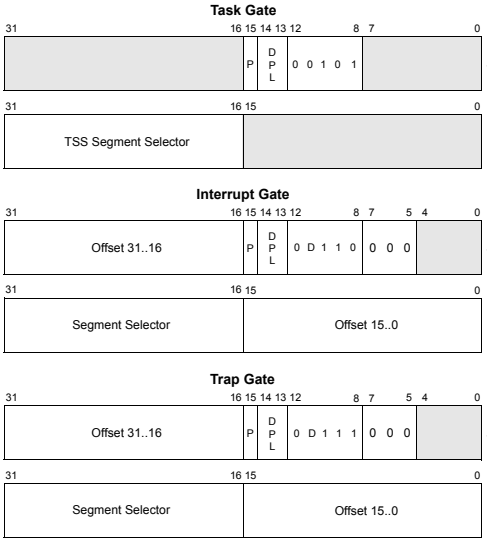
DPL	Descriptor Privilege Level
Offset	Offset to procedure entry point
P	Segment Present flag
Selector	Segment Selector for destination code segment
D	Size of gate: 1 = 32 bits; 0 = 16 bits

Reserved

Figure 5-2. IDT Gate Descriptors

- *Task Gate*: Inicia una tarea para anteder la interrupción.
- *Interrupt Gate*: Detene interrupciones y atiende la interrupción.

Gate Descriptors



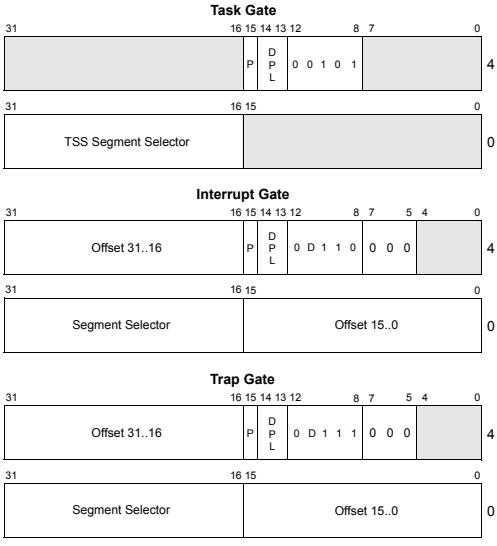
DPL	Descriptor Privilege Level
Offset	Offset to procedure entry point
P	Segment Present flag
Selector	Segment Selector for destination code segment
D	Size of gate: 1 = 32 bits; 0 = 16 bits

Reserved

Figure 5-2. IDT Gate Descriptors

- *Task Gate*: Inicia una tarea para anteder la interrupción.
- *Interrupt Gate*: Detene interrupciones y atiende la interrupción.
- *Trap Gate*: No detene interrupciones y atiende la interrupción.

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

- *Task Gate*: Inicia una tarea para anteder la interrupción.
- *Interrupt Gate*: Detene interrupciones y atiende la interrupción.
- *Trap Gate*: No detene interrupciones y atiende la interrupción.

Vamos a utilizar **Interrupt Gate** para atender nuestras interrupciones.

Interrupt Procedure Call

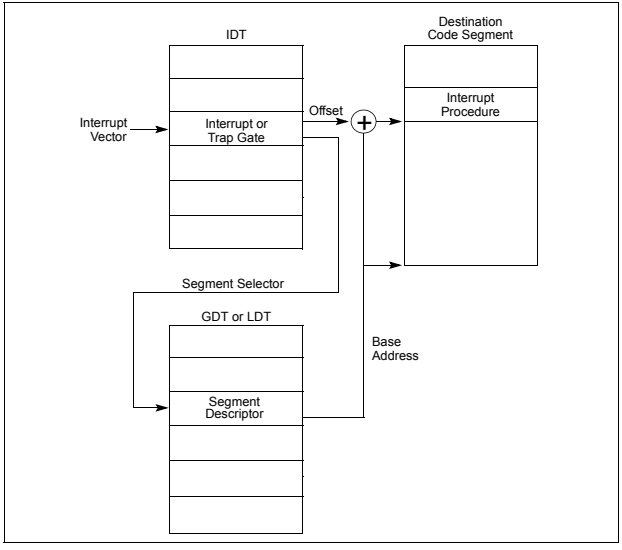


Figure 5-3. Interrupt Procedure Call

Interrupt Procedure Call

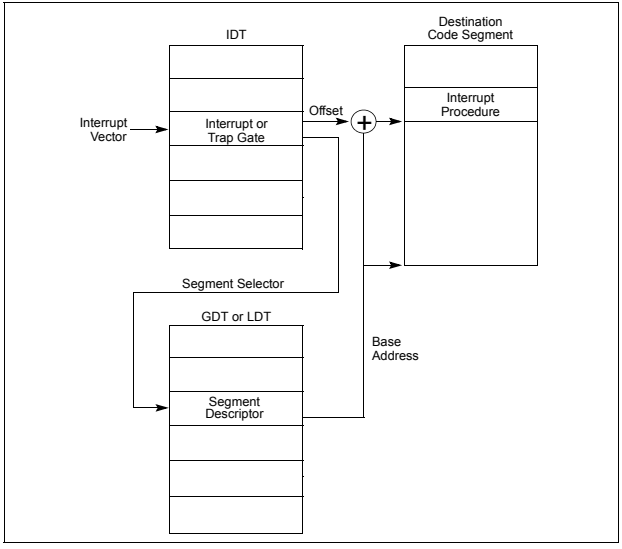


Figure 5-3. Interrupt Procedure Call

- Llega el vector de interrupción (índice) y se atiende según indique su entrada en la IDT.

Interrupt Procedure Call

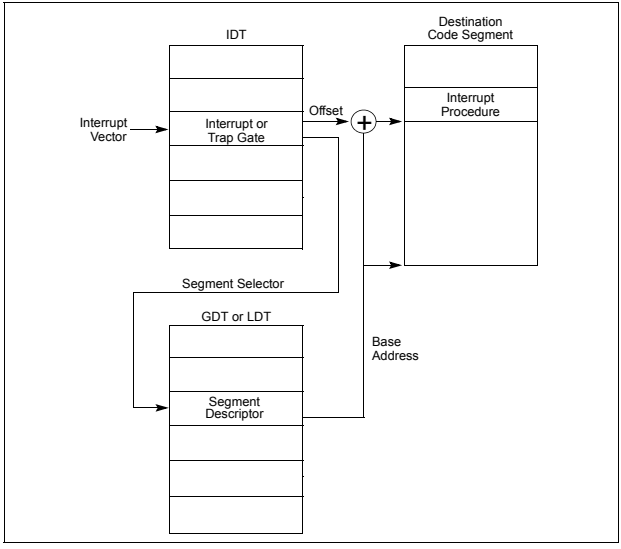


Figure 5-3. Interrupt Procedure Call

- Llega el vector de interrupción (índice) y se atiende según indique su entrada en la IDT.
- Se obtiene el `segment` y se lo busca en la GDT

Interrupt Procedure Call

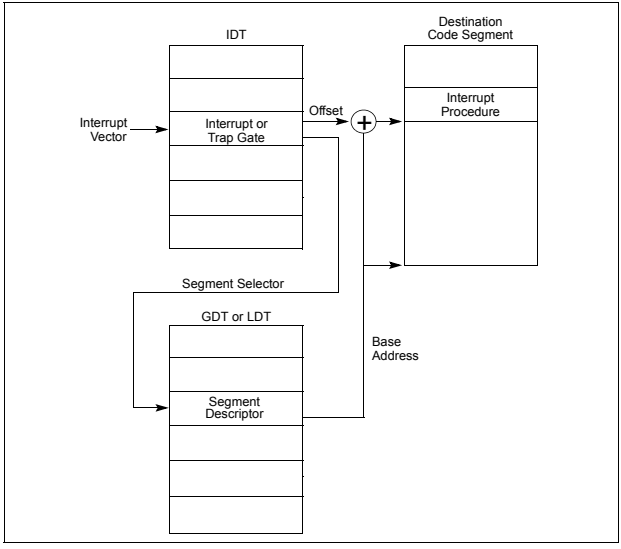


Figure 5-3. Interrupt Procedure Call

- Llega el vector de interrupción (índice) y se atiende según indique su entrada en la IDT.
- Se obtiene el `segment` y se lo busca en la GDT
- Se obtiene la base del segmento, y junto al `offset` se calcula la dirección de la rutina de atención de interrupciones.

Interrupt Procedure Call

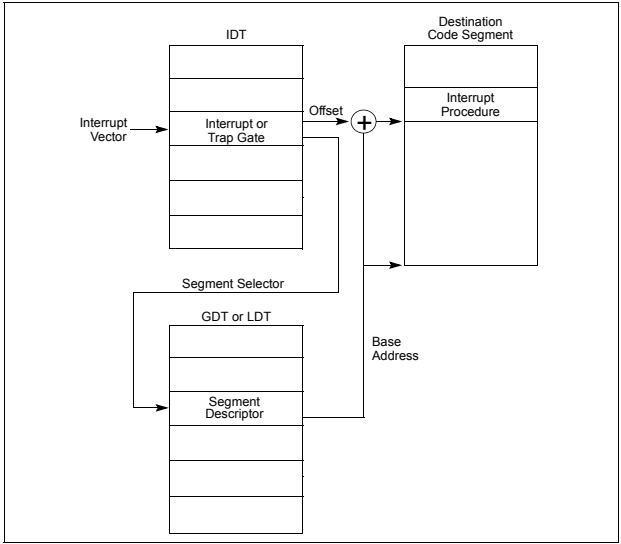


Figure 5-3. Interrupt Procedure Call

- Llega el vector de interrupción (índice) y se atiende según indique su entrada en la IDT.
- Se obtiene el `segment` y se lo busca en la GDT
- Se obtiene la base del segmento, y junto al `offset` se calcula la dirección de la rutina de atención de interrupciones.
- Se comienzá a ejecutar la rutina en la dirección `segment:offset`

Interrupt Task Switch

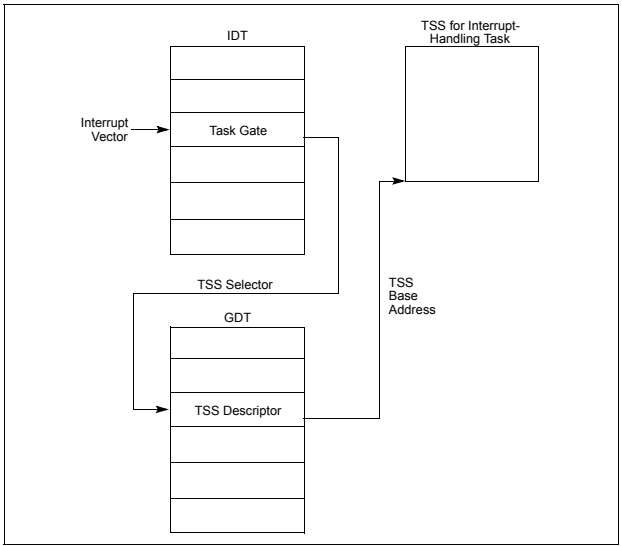


Figure 5-5. Interrupt Task Switch

Interrupt Task Switch

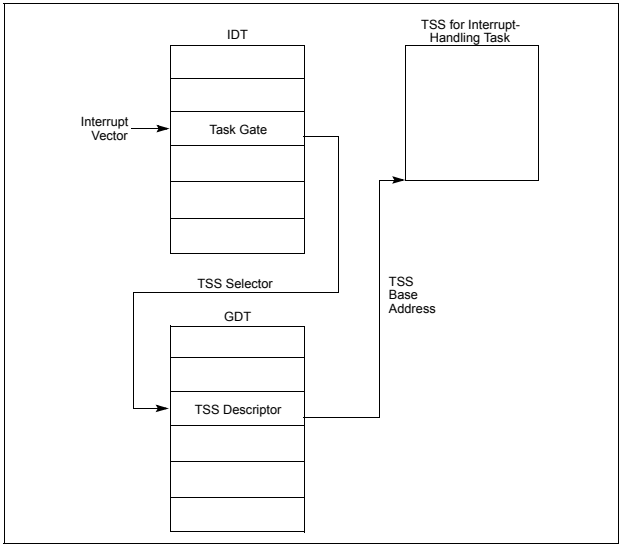


Figure 5-5. Interrupt Task Switch

- Llega el vector de interrupción (índice) y se atiende según indique su entrada en la IDT.

Interrupt Task Switch

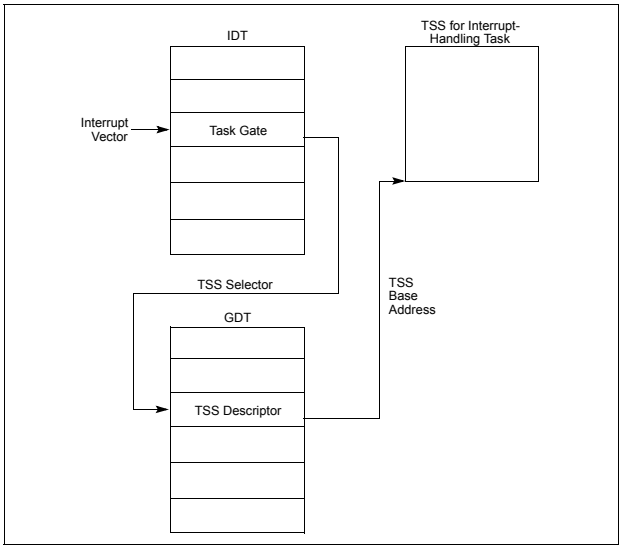


Figure 5-5. Interrupt Task Switch

- Llega el vector de interrupción (índice) y se atiende según indique su entrada en la IDT.
- Se obtiene el `segment` y se lo busca en la GDT

Interrupt Task Switch

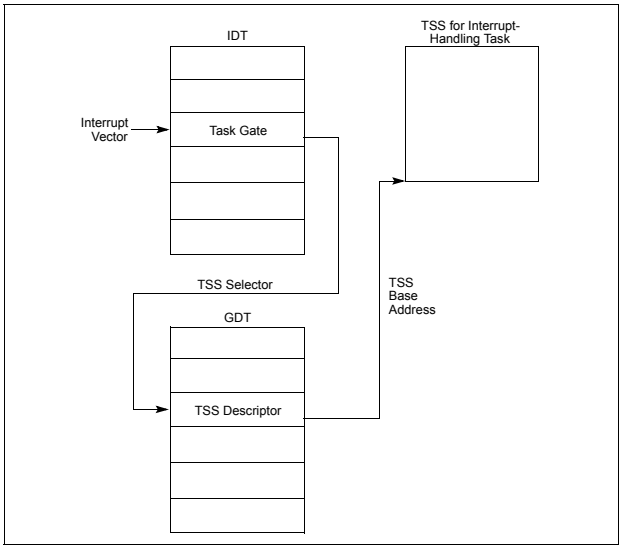


Figure 5-5. Interrupt Task Switch

- Llega el vector de interrupción (índice) y se atiende según indique su entrada en la IDT.
- Se obtiene el `segment` y se lo busca en la GDT
- Se obtiene el contexto de ejecución de la tarea asociada a la entrada en la GDT.

Interrupt Task Switch

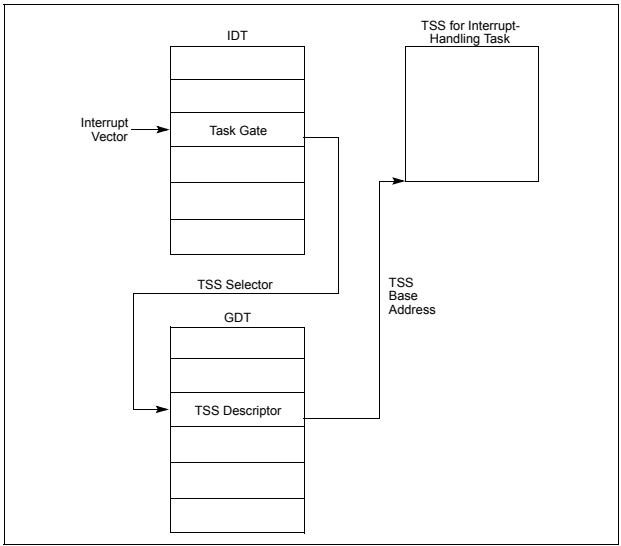


Figure 5-5. Interrupt Task Switch

- Llega el vector de interrupción (índice) y se atiende según indique su entrada en la IDT.
- Se obtiene el `segment` y se lo busca en la GDT
- Se obtiene el contexto de ejecución de la tarea asociada a la entrada en la GDT.
- Se comienzá a ejecutar la tarea, intercambiandola con la tarea actual.

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.

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.

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 causá, ni permite recuperar la ejecución de la tarea que la causá. 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 *n* 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

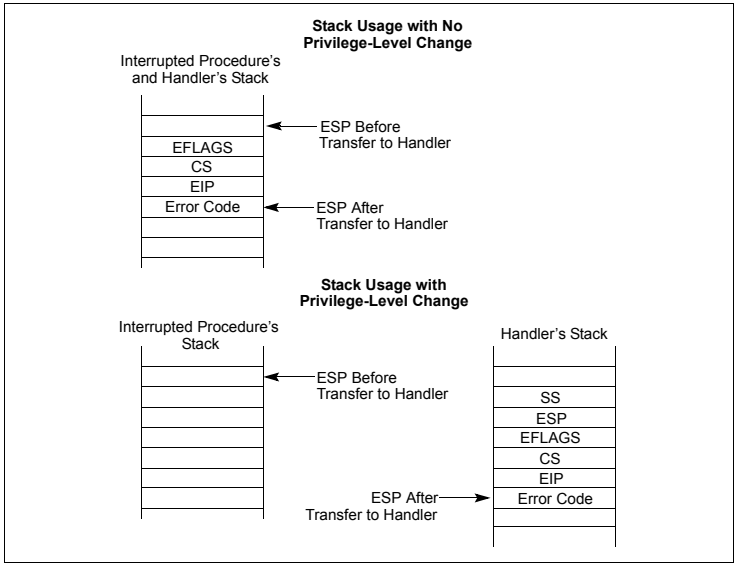


Figure 5-4. Stack Usage on Transfers to Interrupt and Exception-Handling Routines

Error Code

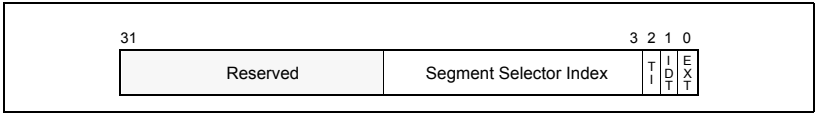


Figure 5-6. Error Code

Error Code

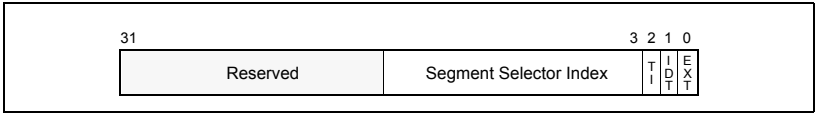


Figure 5-6. Error Code

- **EXT**: (External Event)
  Se setea para indicar que la excepción ha sido causada por un evento externo al procesador.

Error Code

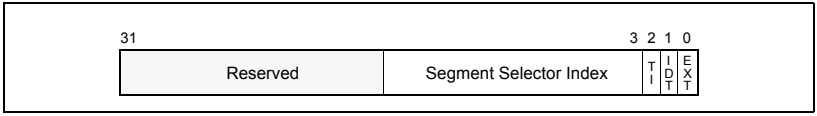


Figure 5-6. Error Code

- **EXT**: (External Event)
  Se setea para indicar que la excepción ha sido causada por un evento externo al procesador.
- **IDT**: (Descriptor Location)
  Cuando está seteado indica que el campo Segment Selector Index se refiere a un descriptor de puerta en la IDT. Cuando est en cero indica que dicho campo se refiere a un descriptor en la GDT o en la LDT de la tarea actual.

Error Code

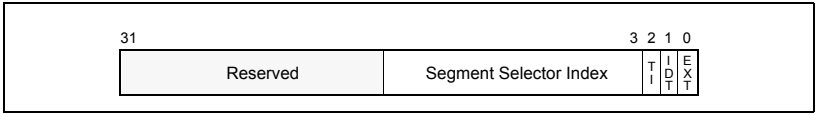


Figure 5-6. Error Code

- **EXT**: (External Event)
  Se setea para indicar que la excepción ha sido causada por un evento externo al procesador.
- **IDT**: (Descriptor Location)
  Cuando está seteado indica que el campo Segment Selector Index se refiere a un descriptor de puerta en la IDT. Cuando est en cero indica que dicho campo se refiere a un descriptor en la GDT o en la LDT de la tarea actual.
- **TI**: (GDT/LDT)
  Tiene significado cuando el bit anterior est en cero. Indica a que tabla de descriptores corresponde el selector del campo Indice. (GDT=0 , LDT=1)

Archivos

1. `idt.h`
   Descripción de las estructuras

Archivos

1. `idt.h`
   Descripción de las estructuras

2. `idt.c`
   Estructura de la `IDT` con cada una de sus entradas

Archivos

1. `idt.h`
   Descripción de las estructuras

2. `idt.c`
   Estructura de la `IDT` con cada una de sus entradas

   - `idt_entry idt[255] = { ... };`

Archivos

1. `idt.h`
   Descripción de las estructuras

2. `idt.c`
   Estructura de la `IDT` con cada una de sus entradas

   - `idt_entry idt[255] = { ... };`
   - `IDT_ENTRY(numero)`
     Permite declarar una entrada en la `IDT`, para la utilizar el handler de nombre `_isrnumero`

Archivos

1. `idt.h`
   Descripción de las estructuras

2. `idt.c`
   Estructura de la `IDT` con cada una de sus entradas

- `idt_entry idt[255] = { ... };`
- `IDT_ENTRY(numero)`
  Permite declarar una entrada en la `IDT`, para la utilizar el handler de nombre `_isrnumero`
- `void init_idt()`
  Función llamada desde el kernel para inicializar las entradas en la `IDT`

Estructuras

- 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;
```

Rutinas de Atención de Interrupción

Cómo manejar correctamente una interrupción

1. Preservar los registros que vayamos a romper

Rutinas de Atención de Interrupción

Cómo manejar correctamente una interrupción

1. Preservar los registros que vayamos a romper → *¡la interrupción debe ser transparente!*

Rutinas de Atención de Interrupción

Cómo manejar correctamente una interrupción

1. Preservar los registros que vayamos a romper → *¡la interrupción debe ser transparente!*
2. Realizar la tarea correspondiente a la interrupción

Rutinas de Atención de Interrupción

Cómo manejar correctamente una interrupción

1. Preservar los registros que vayamos a romper → *¡la interrupción debe ser transparente!*
2. Realizar la tarea correspondiente a la interrupción
3. Restaurar los registros

Rutinas de Atención de Interrupción

Cómo manejar correctamente una interrupción

1. Preservar los registros que vayamos a romper → *¡la interrupción debe ser transparente!*
2. Realizar la tarea correspondiente a la interrupción
3. Restaurar los registros
4. Retornar de la interrupción

Bibliografía: Fuentes y material adicional

- Convenciones de llamados a función en x86:
  `https://en.wikipedia.org/wiki/X86_calling_conventions`
- Notas sobre System V ABI:
  `https://wiki.osdev.org/System_V_ABI`
- Documentación de NASM:
  `https://nasm.us/doc/`
- Artículo sobre el flag `-pie`:
  `https://eklitzke.org/position-independent-executables`
- Documentación de System V ABI:
  `https://uclibc.org/docs/psABI-x86_64.pdf`
- Manuales de Intel:
  `https://software.intel.com/en-us/articles/intel-sdm`

¡Gracias!

Recuerden leer los comentarios al final de este video por aclaraciones o fe de erratas.