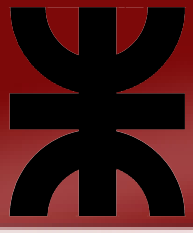
Modo Real
Modo Protegido
Modo Largo



- El Modo Real es el modo inicial de todos los procesadores Intel (y sus clones).
- Su presencia en procesadores modernos es requerida por compatibilidad.
- Todos los sistemas operativos de 32 bits se ejecutan en Modo Protegido.
- Los sistemas operativos de 64 bits se ejecutan en Modo Largo.

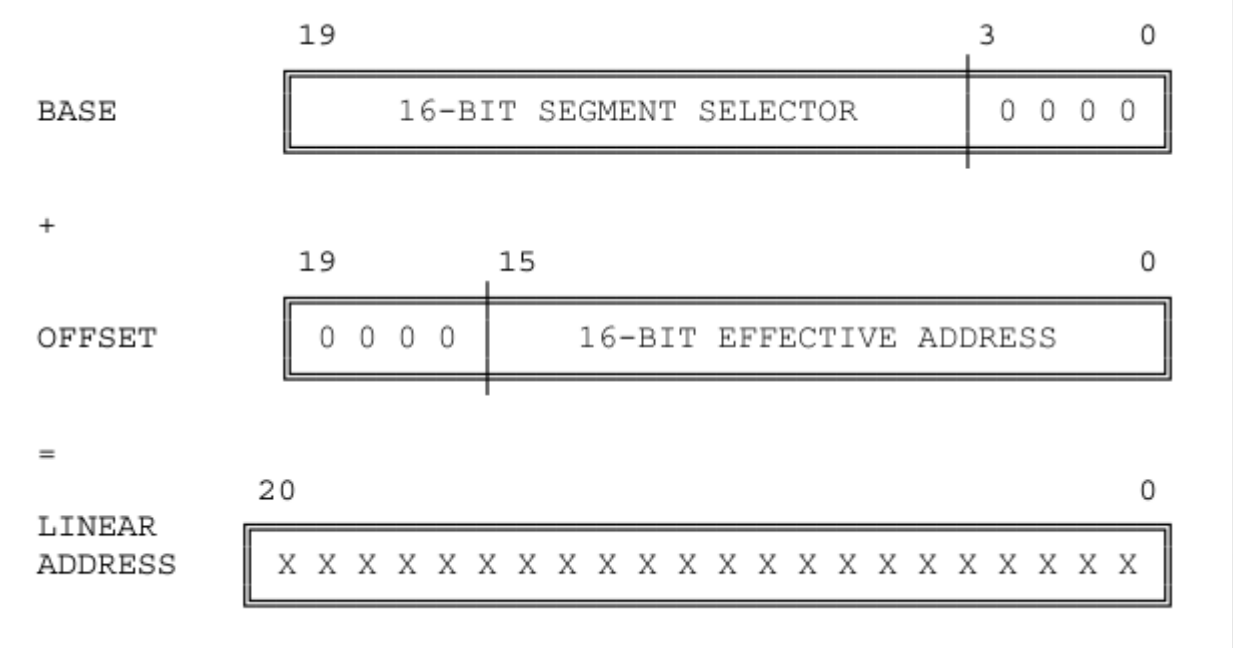
Modo Real	Modo Largo/Protegido
Menos de 1MB de RAM disponible	Toda la RAM disponible
No existe la protección de memoria basada en hardware.	Protección de memoria basada en hardware.
No hay modo de restringir las instrucciones que un programa de usuario puede ejecutar	Existen niveles de privilegio
El tamaño de los operandos es de 16 bits	El tamaño de los operandos es de 32 o 64 bits
Acceso a las funciones del BIOS	No es posible el acceso a las funciones del BIOS

Ingreso a Modo Protegido

```
lgdt [gdtr] ;       ; Carga registro GDT
mov eax,cr0         ; eax = Control Register 0
or eax,1            ; PE = 1 (protection enable)
cli                 ; Se deshabilitan interrupciones
mov cr0,eax         ; Inicia Modo Protegido.
```

Direccionamiento

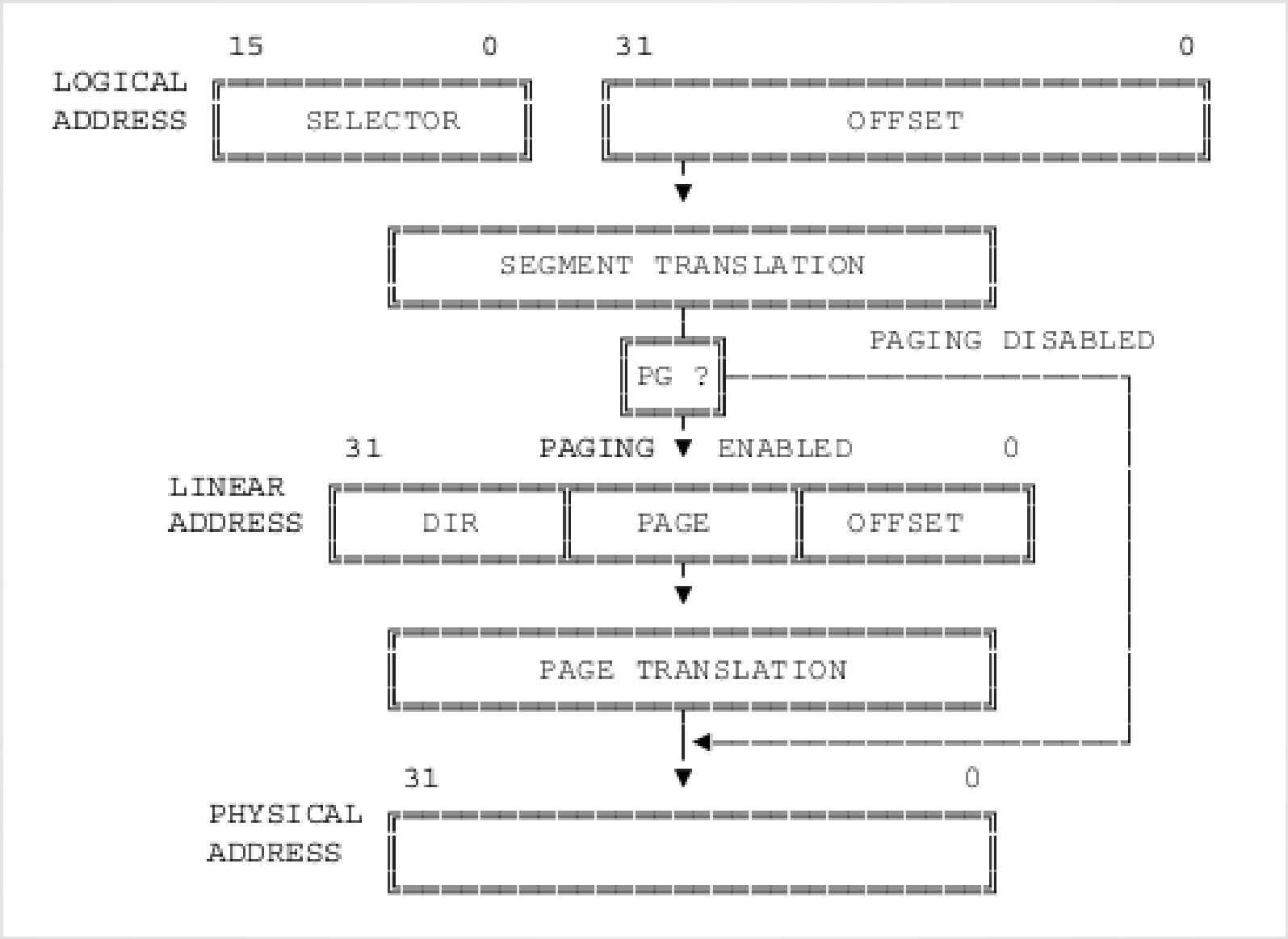
- La dirección física en Modo Real se forma sumando la dirección base con desplazamiento.
- La dirección base se forma rotando cuatro bits a la izquierda el selector de segmento.



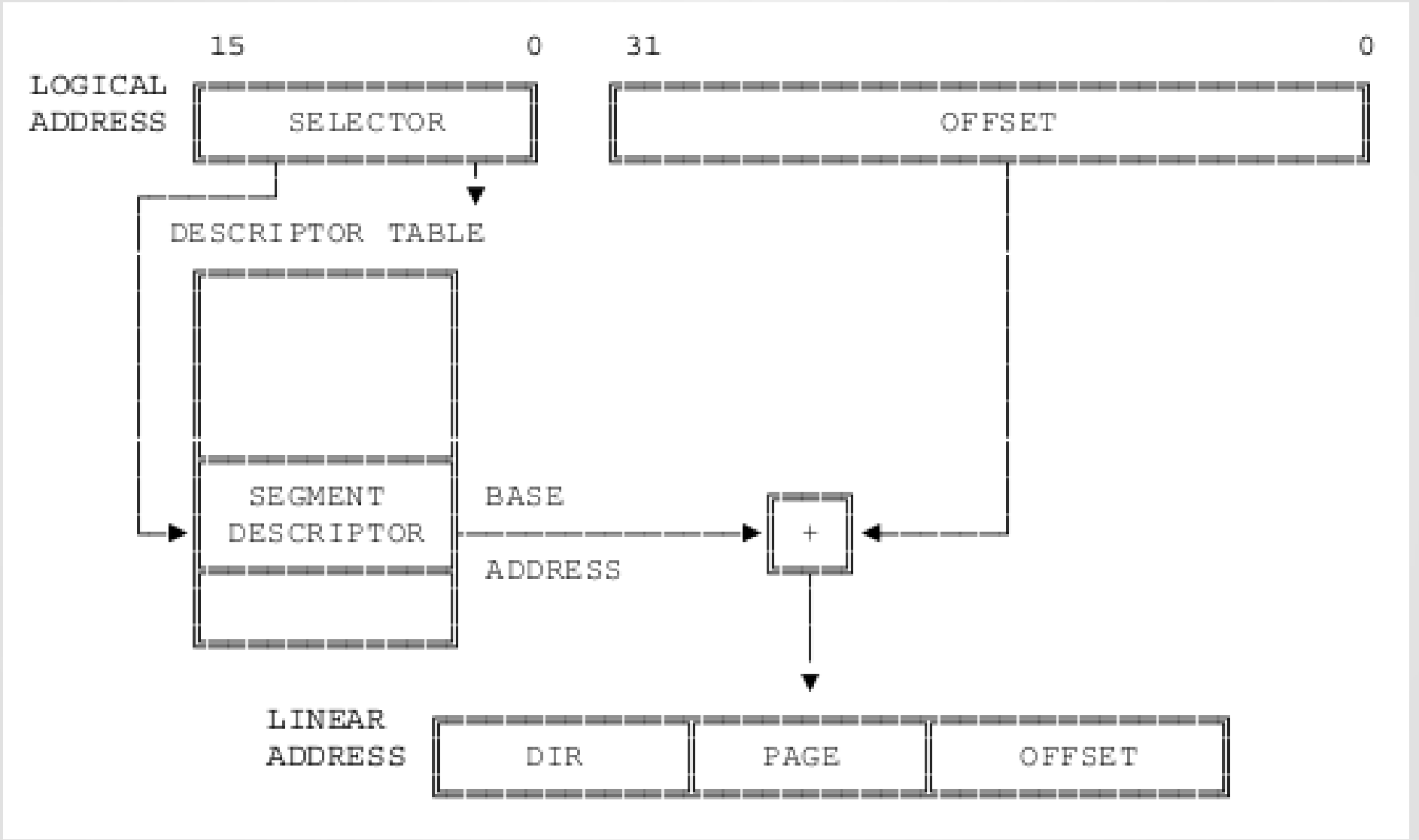
Direccionamiento

- En Modo Protegido, el procesador transforma direcciones lógicas (utilizadas por los programadores) en direcciones físicas (la dirección real en la memoria física) en dos pasos:
  - Traducción de Segmento
  - Traducción de Página
- Estas traducciones son realizadas de tal manera que no son visibles al programador

Traducción de Direcciones

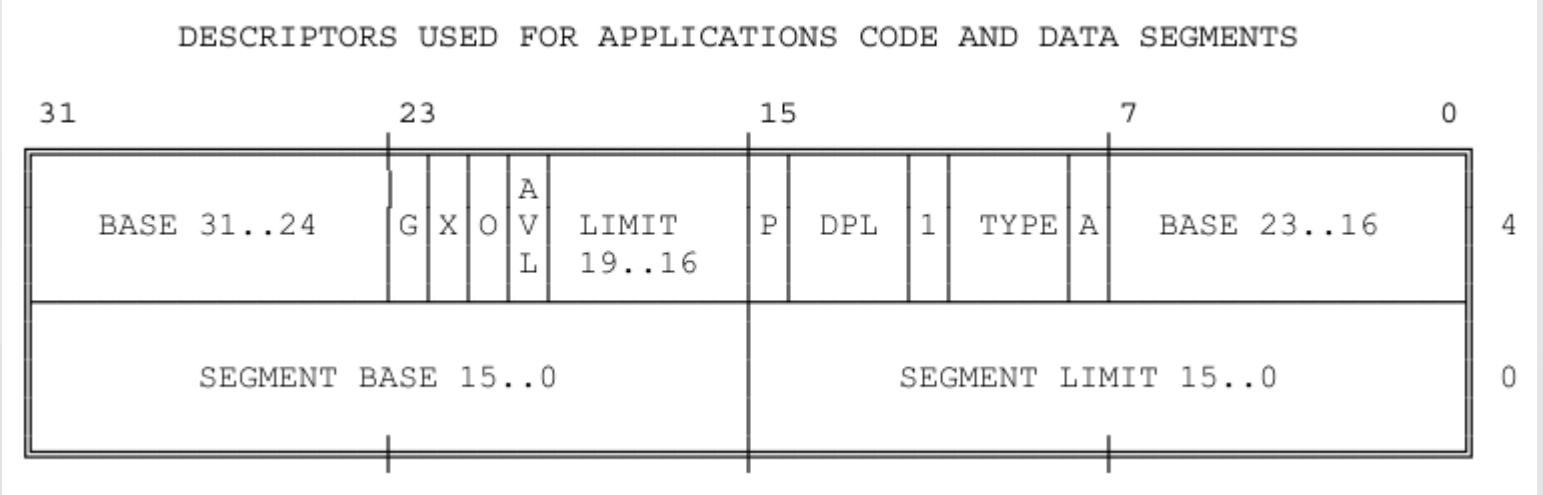


Traducción de Segmento



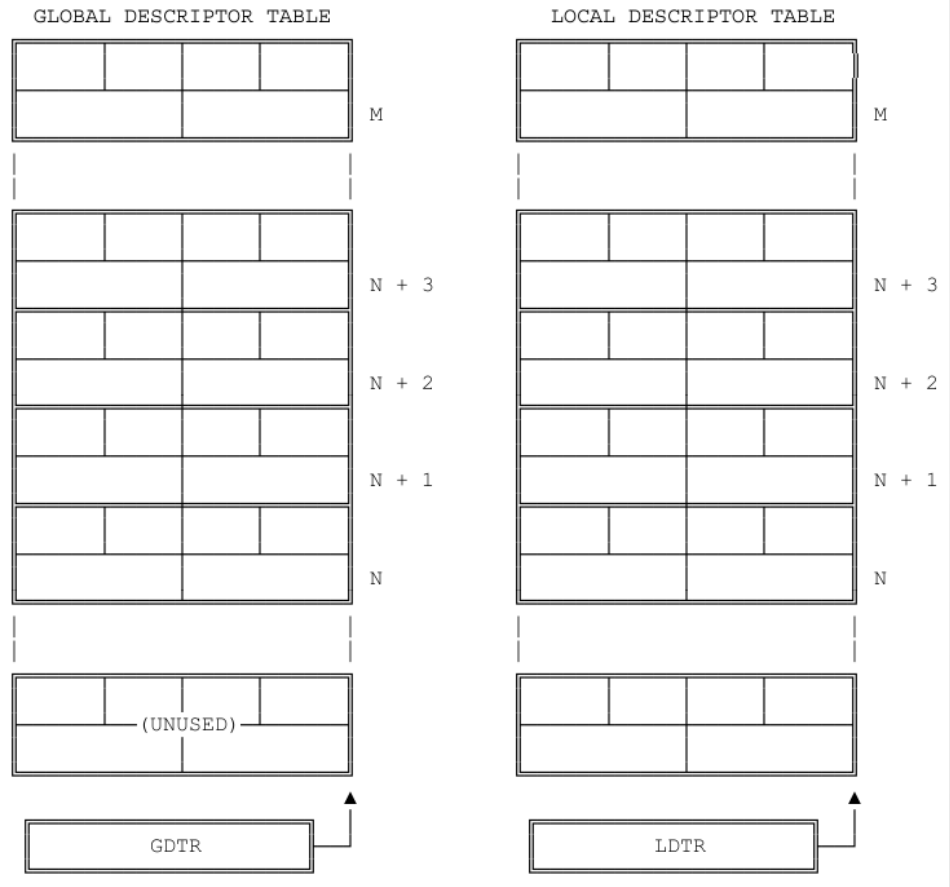
Descriptores

- El descriptor provee al procesador con la información que necesita para mapear una dirección lógica en una dirección lineal.
- Son creados por compiladores, enlazadores, cargadores, o el sistema operativo, no por programadores de aplicaciones.



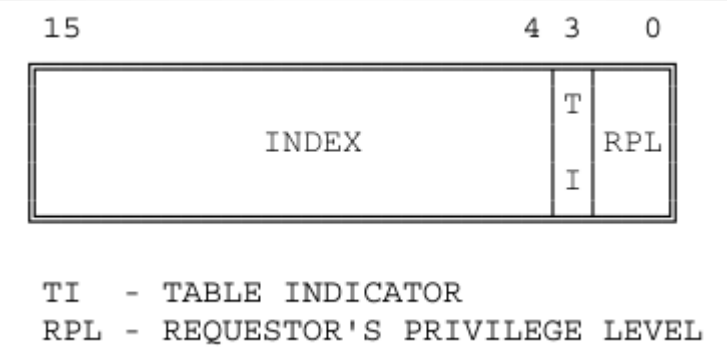
Tablas de Descriptores

- Una tabla de descriptores es simplemente un arreglo de memoria con elementos de 8 bytes.



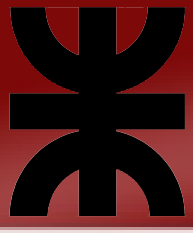
Selectores

- Esta porción de una dirección lógica especifica una tabla e indexa un descriptor en esa tabla.



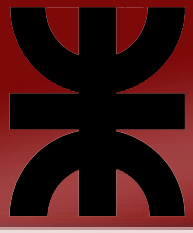
Protección

- El Modo Protegido abarca estos aspectos:
  - Chequeo de tipo
  - Chequeo de límite
  - Restricción de dominio direccionable
  - Restricción de puntos de entrada a procedimientos
  - Restricción del set de instrucciones



Protección

- Chequeo de Tipo:
  - Distingue entre diferentes formatos de descriptores
  - Especifica el uso previsto del segmento
  - Ejemplos:
    - El registro CS solo puede contener un selector de un segmento ejecutable
    - Solo selectores de segmentos con permiso de escritura pueden cargarse en SS.
    - Ninguna instrucción puede escribir en un segmento ejecutable.
    - Ninguna instrucción puede leer un segmento ejecutable si el bit de lectura no está en uno.



Protección

- Chequeo de límite
  - El procesador no permite el direccionamiento fuera del segmento.
- Niveles de privilegio
  - Las siguientes entidades poseen niveles de privilegio:
    - Descriptores (DPL)
    - Selectores (RPL)
    - Registro interno del procesador (CPL)
  - El procesador automáticamente evalúa el derecho de un procedimiento de acceder a otro segmento comparando el CPL con uno o más niveles de privilegio.

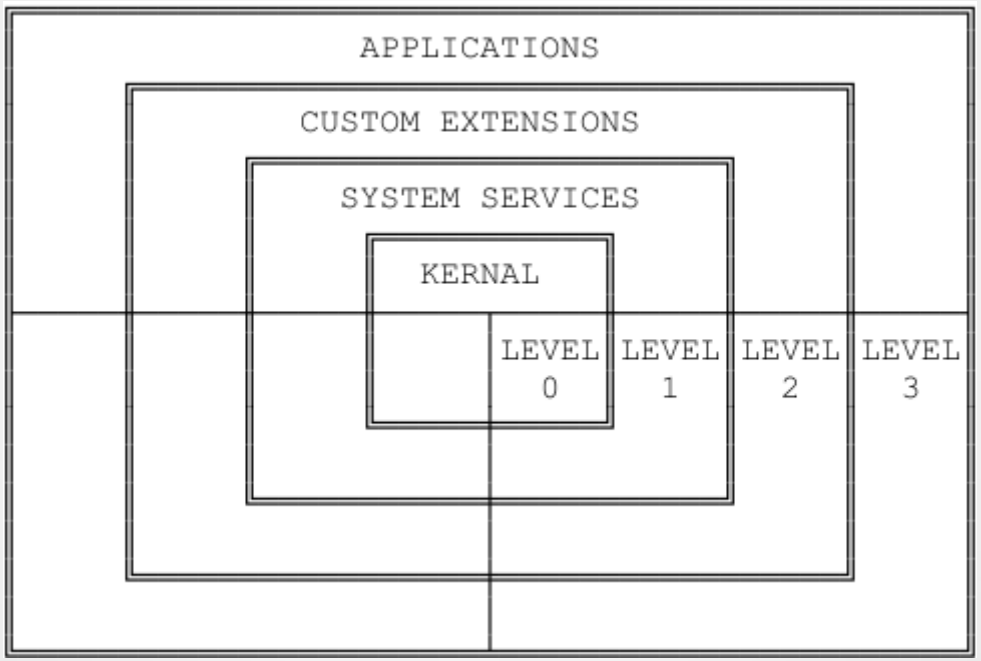
Protección

- Las instrucciones que afectan estructuras de datos del sistema solo pueden ser ejecutas en el anillo cero.

```
CLTS                 — Clear Task-Switched Flag
HLT                  — Halt Processor
LGDT                 — Load GDL Register
LIDT                 — Load IDT Register
LLDT                 — Load LDT Register
LMSW                 — Load Machine Status Word
LTR                  — Load Task Register
MOV to/from CRn      — Move to Control Register n
MOV to /from DRn     — Move to Debug Register n
MOV to/from TRn      — Move to Test Register n
```

Protección

- No es necesario utilizar los cuatro niveles de privilegio.



Interrupciones y Excepciones

- Las Interrupciones y Excepciones son tipos especiales de transferencia de control, funcionan como llamadas (CALLs) no programadas. Alteran el flujo normal del programa para manejar eventos externos, para reportar errores o condiciones excepcionales.
- La diferencia entre Interrupciones y Excepciones es que las interrupciones son usadas para manejar eventos externos asíncronos, mientras que las excepciones manejan condiciones detectadas por el propio procesador mientras ejecuta instrucciones.

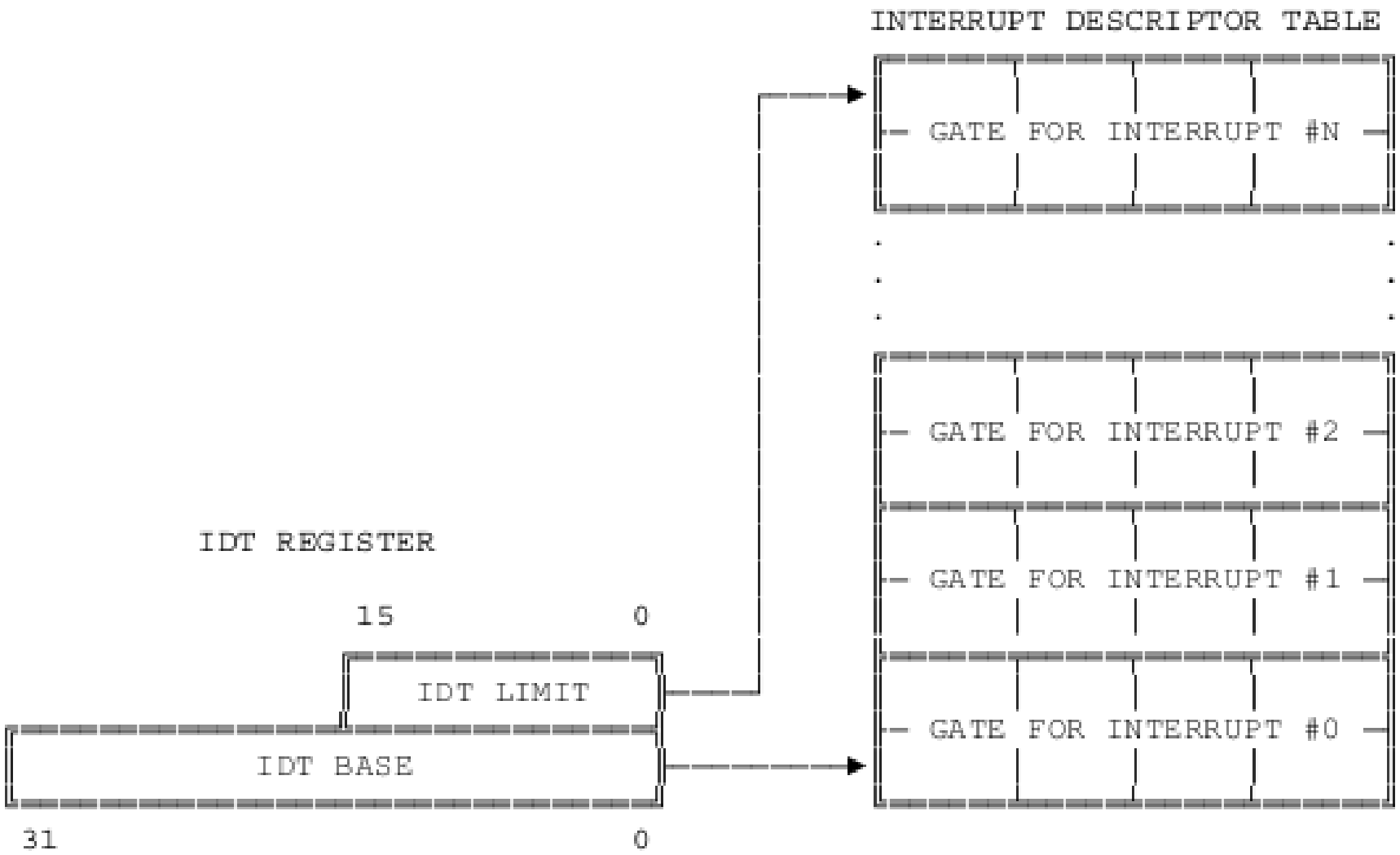
Interrupciones y Excepciones

- Hay dos fuentes de interrupciones y dos fuentes de excepciones:
- Interrupciones:
  - Enmascarables: pin INTR
  - No enmascarables: pin NMI
- Excepciones:
  - Detectadas por el procesador
  - Programadas (instrucción INT), habitualmente llamadas “interrupciones de software”

Interrupciones y Excepciones

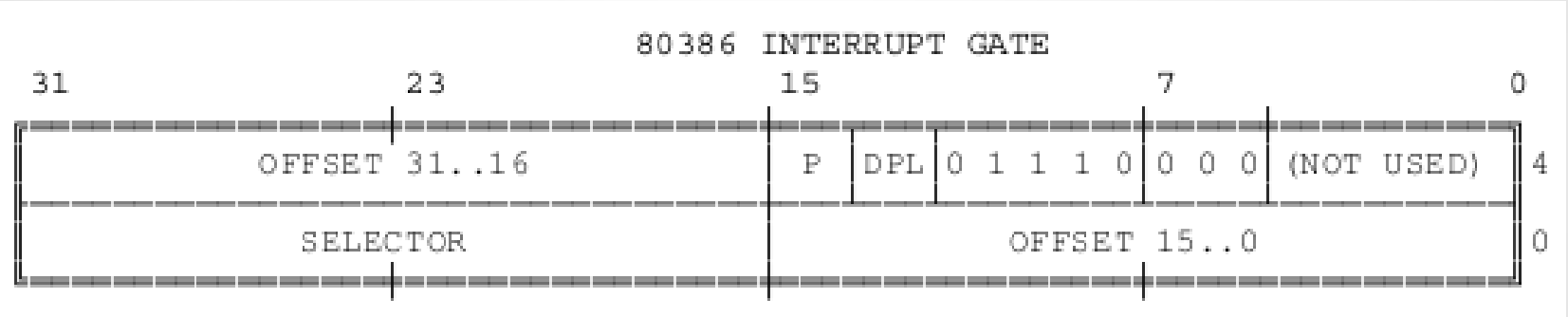
- La Tabla de Descriptores de Interrupción (IDT) asocia cada interrupción o excepción con un descriptor de las instrucciones que sirven el evento asociado.
- La IDT puede residir en cualquier parte de la memoria. El procesador localiza la IDT a través del registro IDTR.
- Las instrucciones LIDT y SIDT operan en el registro IDTR.

Interrupciones y Excepciones



Interrupciones y Excepciones

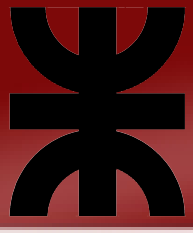
- La IDT puede contener tres tipos de descriptores:
  - Task gates
  - **Interrupt gates**



  - Trap gates

Interrupciones y Excepciones

- Cuando responde a una interrupción o una excepción, el procesador utiliza el identificador del evento para indexar un descriptor en la IDT.
- El selector de la puerta apunta a un descriptor de segmento ejecutable.
- El desplazamiento apunta al inicio del procedimiento de manejo de la interrupción o excepción.



Práctica

Ensamblador

- NASM: Netwide Assembler
- Comando

```
$ nasm boot.asm
```

- Salida en formato binario plano

archivo

- Instalación en el sector de arranque de floppy

```
$ dd if=boot of=boot.img conv=notrunc
```

Emulador x86

- Bochs: IA-32 Emulator Project
- Comando

```
$ bochs -f bochsrc
```

Memoria de Video

- La memoria de video de texto se encuentra localizada a partir de la dirección 0xB8000.
- Ocupa un byte para el carácter ASCII y otro para atributos

0xB8000.	0xB8002	0xB8004	0xB8006	...	0xB8096	0xB8098	0xB809A	0xB809C	0xB809E
0xB80A0									...
...									...
...									...
...									...
...									...
...									...
...									..
...									..
...	...	...	...	...	...	...	...	..	0xB8F9E

Llamadas a BIOS

Función	Código función	Parámetros	Retorno
Asigna posición del cursor	AH = 02h	BH = Página DH = Fila DL = Columna	El cursor se ubica en la posición seleccionada
Escribe carácter y atributo en la posición del cursor	AH = 09h	AL = Caracter BH = Número de Página BL = Color CX = Número de rep.	Caracter/es en pantalla

Hay muchas más!