Práctica de Organización del Computador II

System Programming

Segundo Cuatrimestre 2025

Organización del Computador II
DC - UBA

Pasaje a modo protegido

Pasaje a modo protegido

En esta parte vamos a ver:

Pasaje a modo protegido

En esta parte vamos a ver:

- Bootloader

Pasaje a modo protegido

En esta parte vamos a ver:

- Bootloader
- Armado de GDT

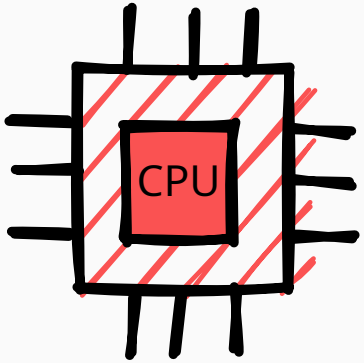
Pasaje a modo protegido

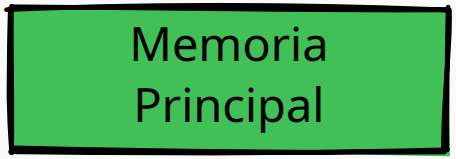
En esta parte vamos a ver:

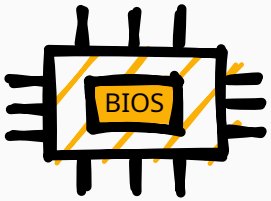
- Bootloader
- Armado de GDT
- Pasaje a modo protegido

Bootloader

Bootloader

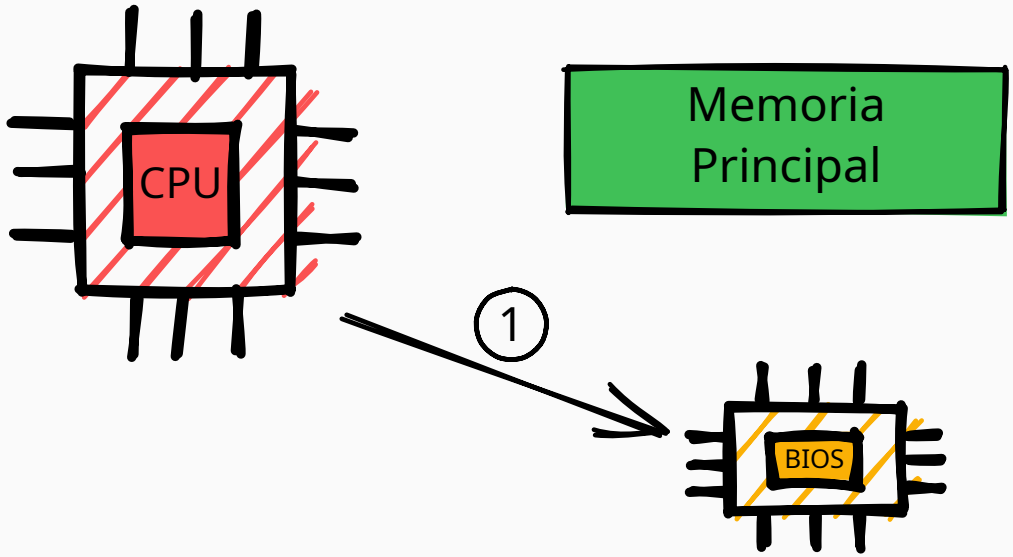






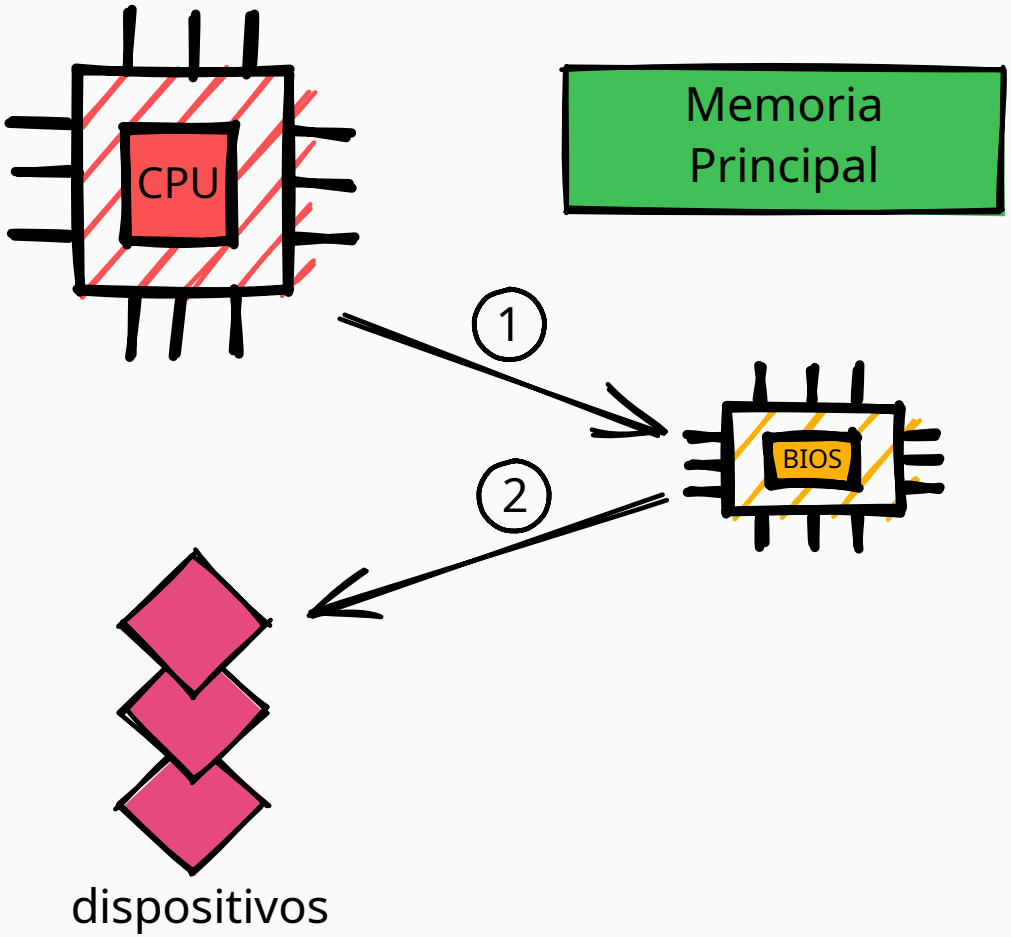
Bootloader

1. CPU ejecuta código residente en memoria flash de BIOS



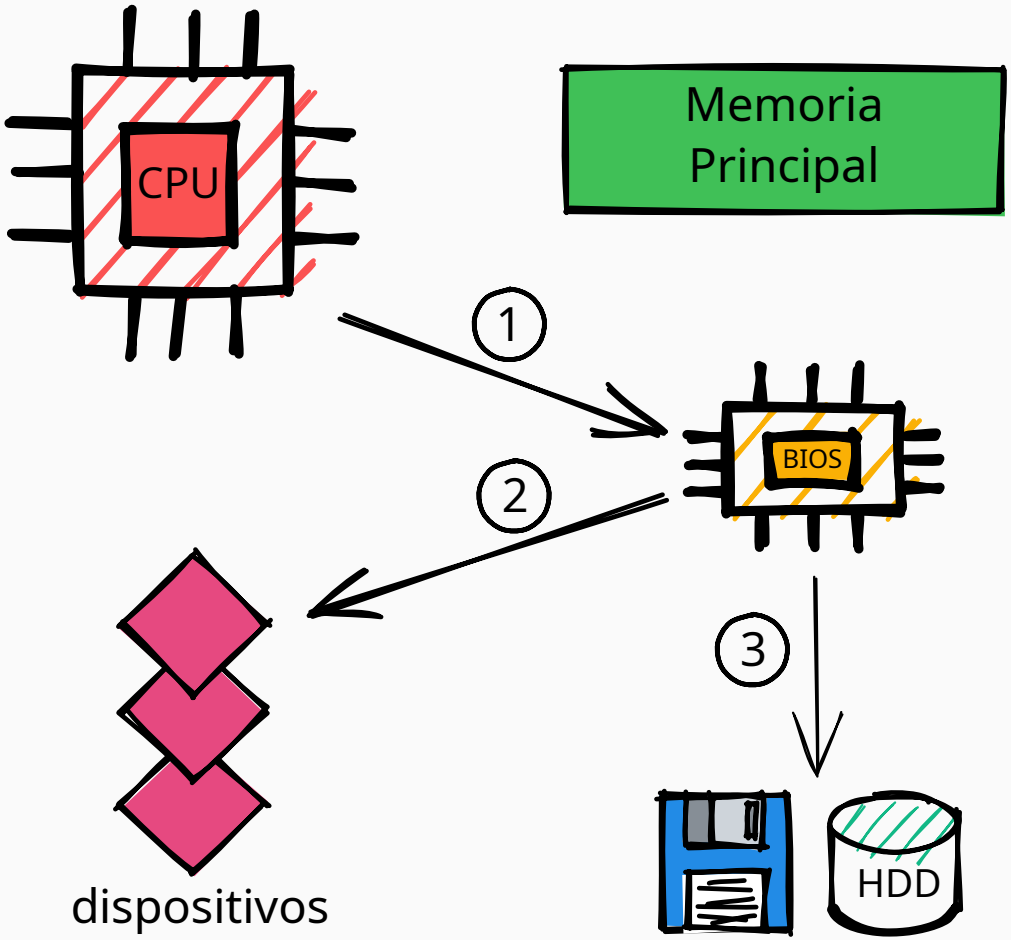
Bootloader

1. CPU ejecuta código residente en memoria flash de BIOS
2. BIOS ejecuta POST (Power On Self Test) en los dispositivos



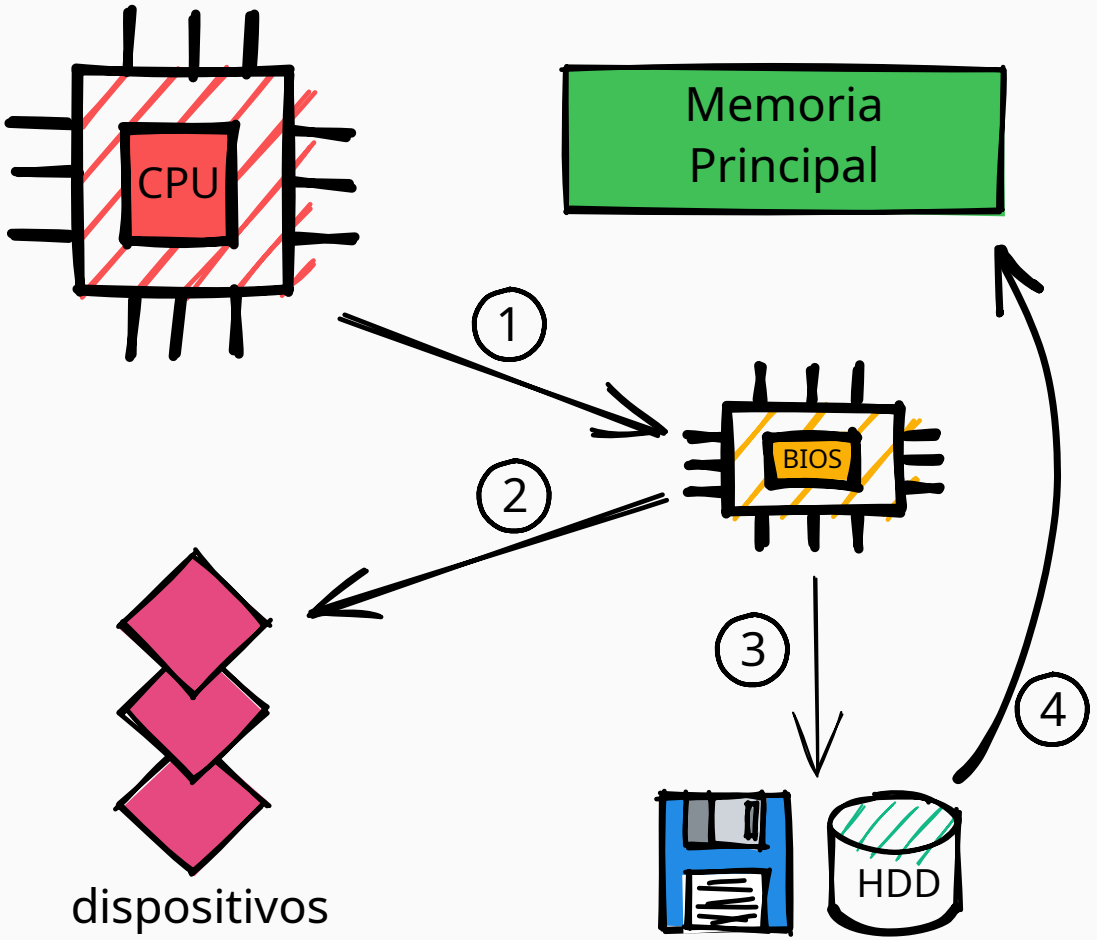
Bootloader

1. CPU ejecuta código residente en memoria flash de BIOS
2. BIOS ejecuta POST (Power On Self Test) en los dispositivos
3. BIOS busca un dispositivo "booteable"

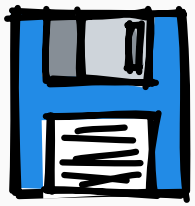


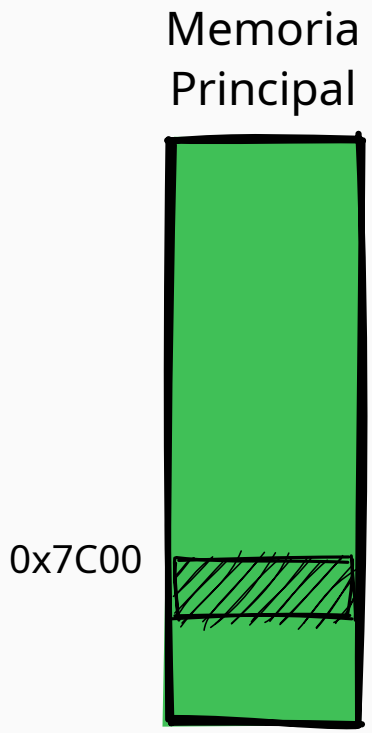
Bootloader

1. CPU ejecuta código residente en memoria flash de BIOS
2. BIOS ejecuta POST (Power On Self Test) en los dispositivos
3. BIOS busca un dispositivo "booteable"
4. Se copia a memoria principal en la posición 0x7C00 el sector de booteo (512 bytes)

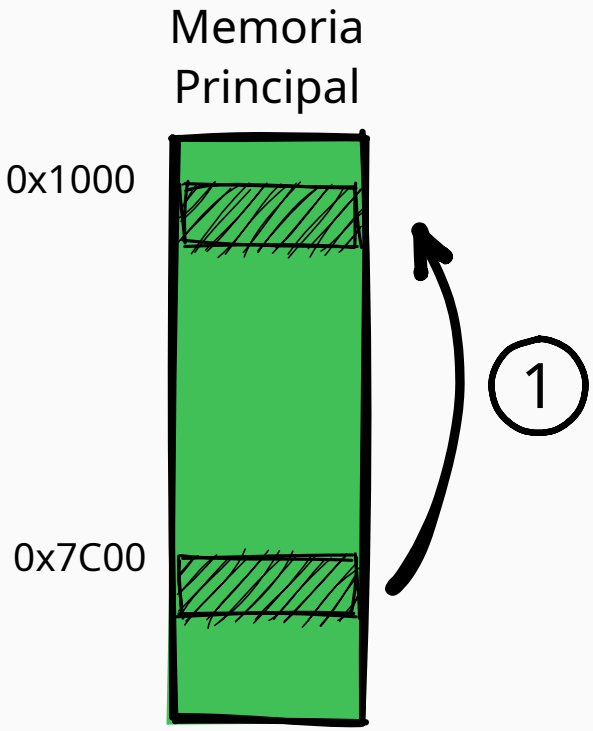


Bootloader de Orga2



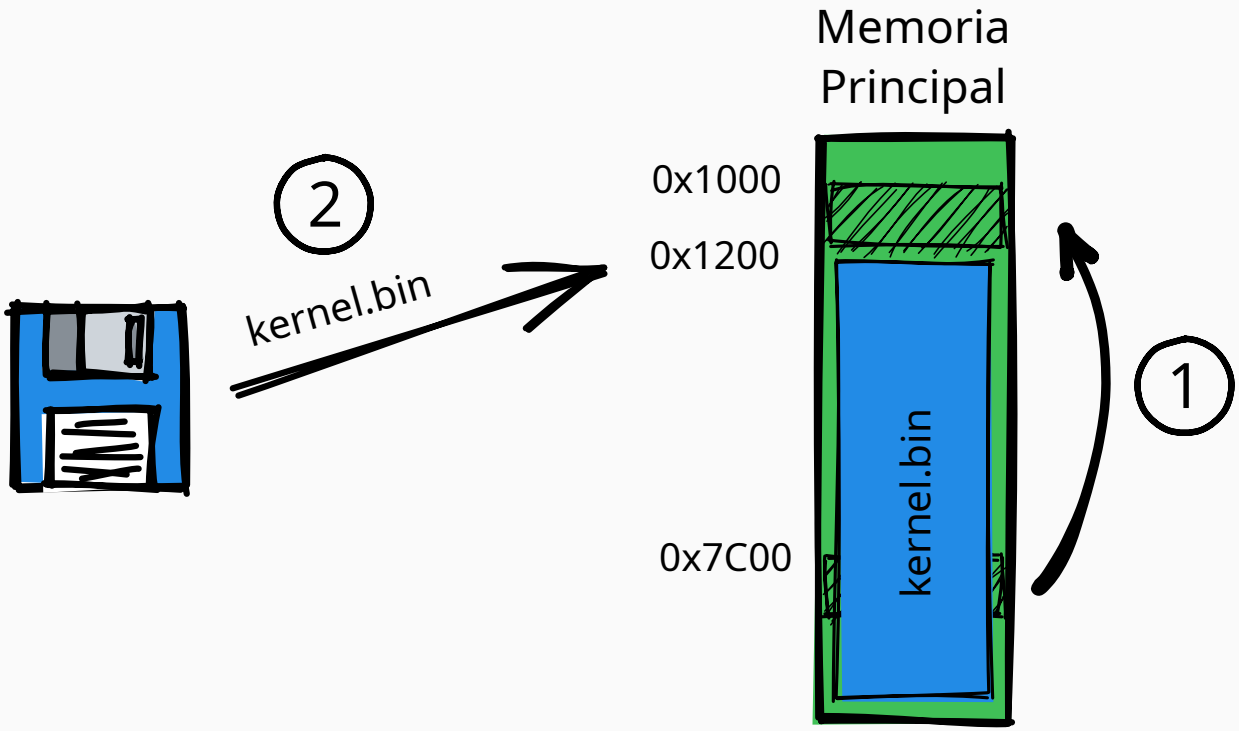


Bootloader de Orga2



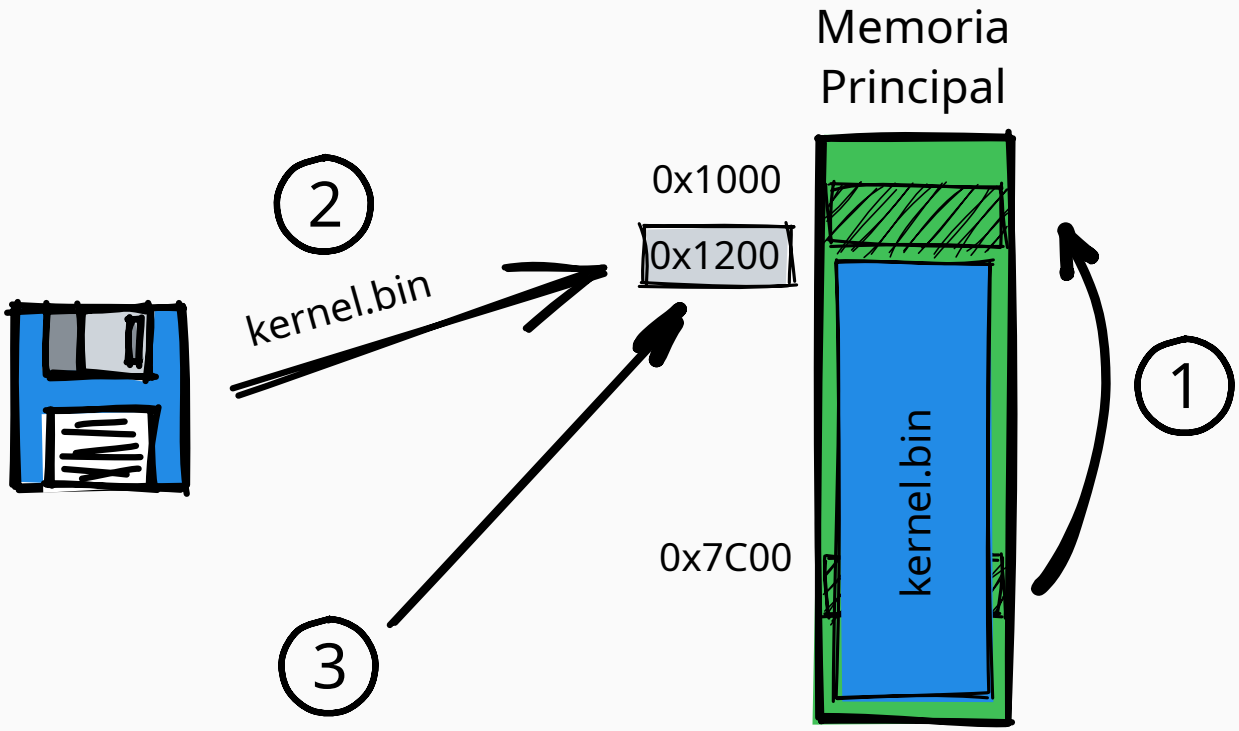
1. Se copia el bootloader a la posición **0x1000**

Bootloader de Orga2



1. Se copia el bootloader a la posición **0x1000**
2. Busca y carga el archivo `kernel.bin` contenido en el diskette y lo copia en la dirección **0x1200**

Bootloader de Orga2

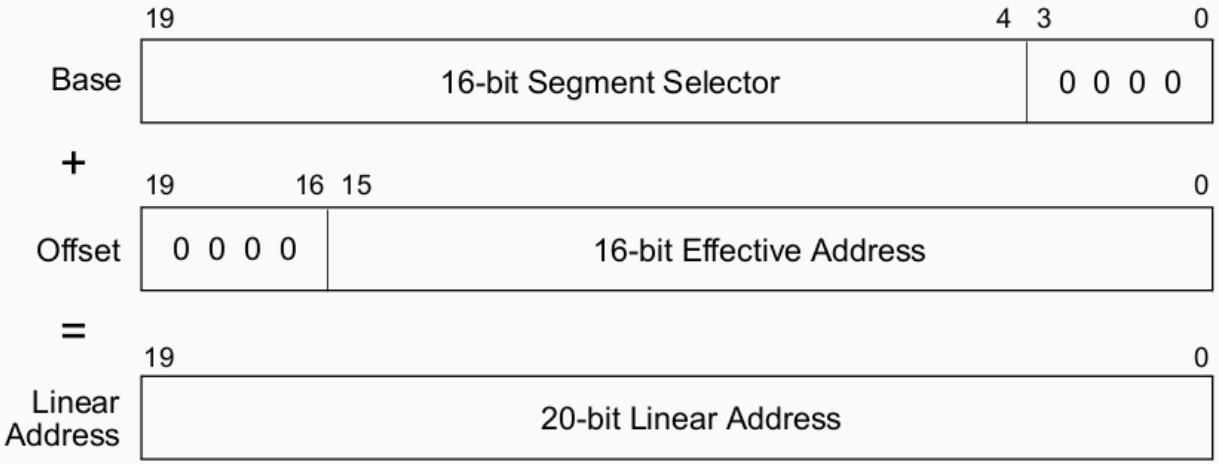


1. Se copia el bootloader a la posición **0x1000**
2. Busca y carga el archivo `kernel.bin` contenido en el diskette y lo copia en la dirección **0x1200**
3. Se salta hacia la dirección **0x1200** y se ejecuta desde ahi

Armado de GDT

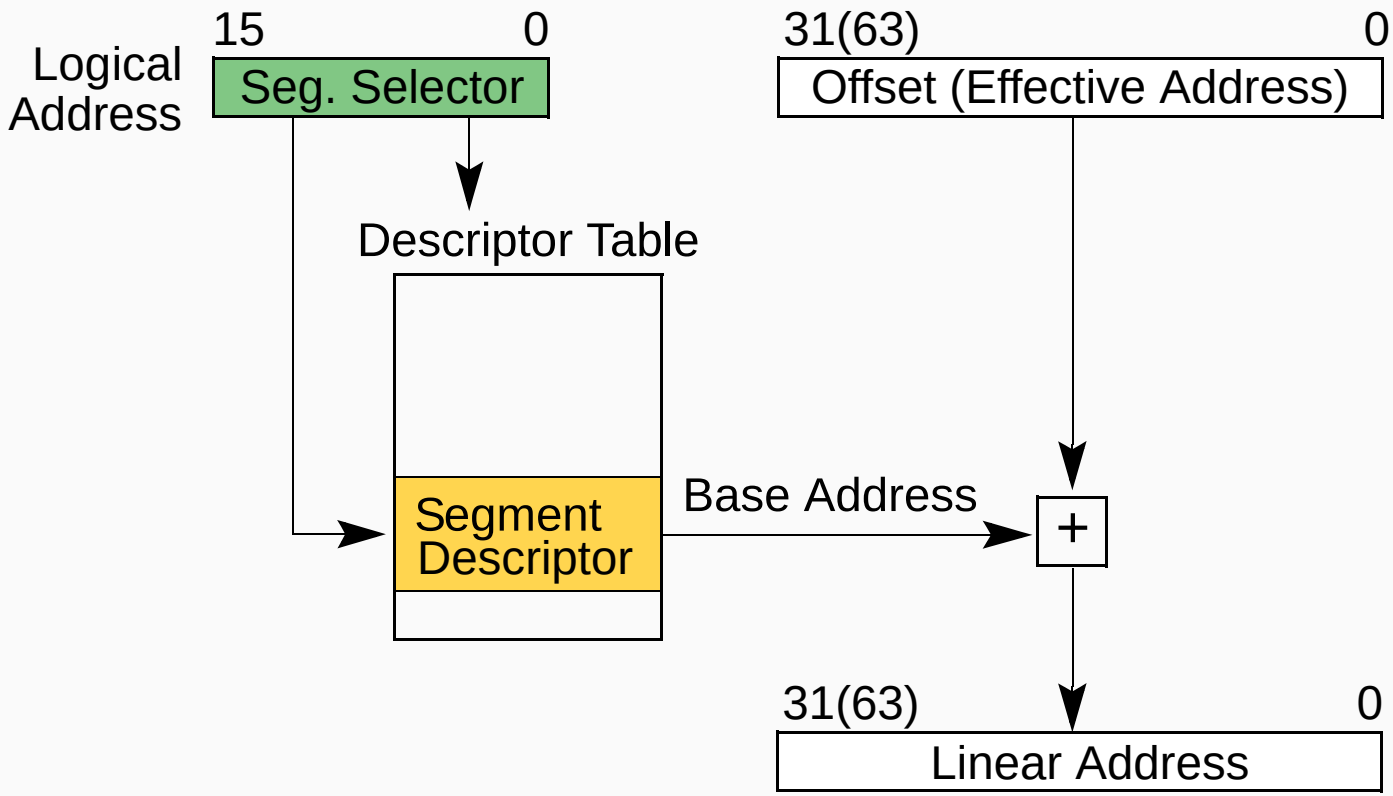
Modo Real

Antes de hablar de la GDT, repasemos segmentación:



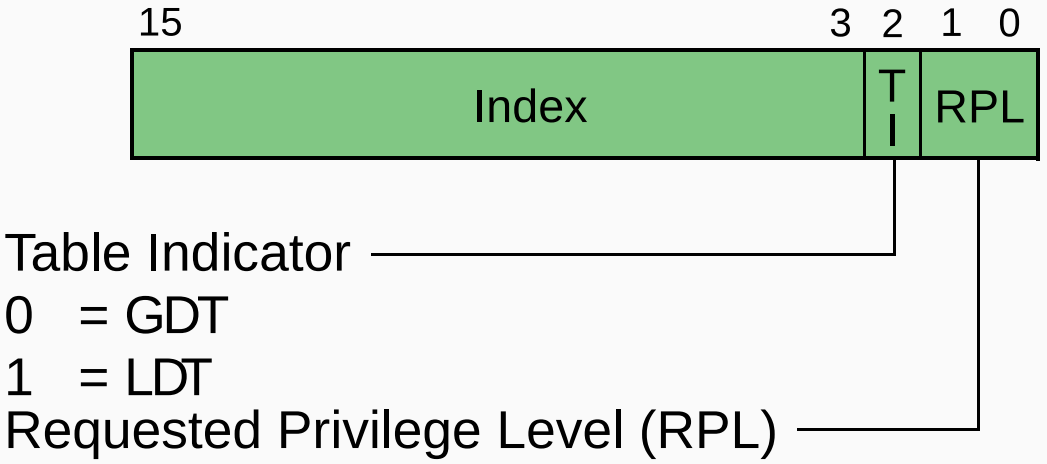
Intel 64 and IA-32 Architectures Software Developer's Manual, Volumen 3, Capítulo 3: Figura 3-5

Modo Protegido



Intel 64 and IA-32 Architectures Software Developer's Manual, Volumen 3, Capítulo 3: Figura 3-5

Selector de Segmento



`CS`: Para acceder a código

`SS`: Para acceder a pila

`DS`: Para acceder a datos (default)

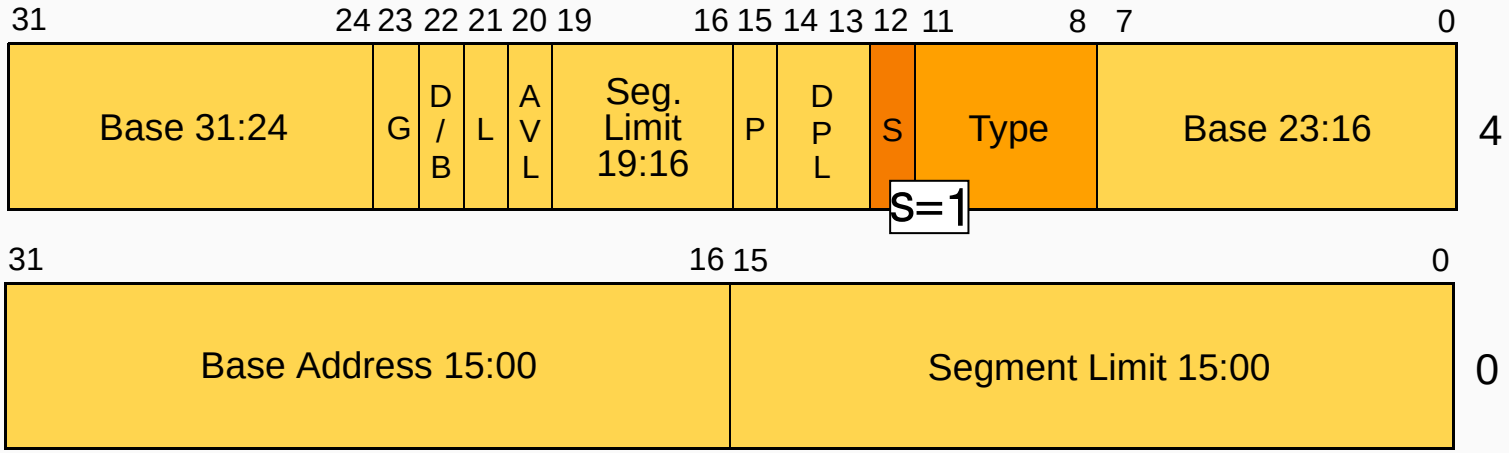
`ES`: Para acceder a datos

`GS`: Para acceder a datos

`FS`: Para acceder a datos

Intel 64 and IA-32 Architectures Software Developer's Manual, Volumen 3, Capítulo 3: Figura 3-6

Descriptor de Segmento



L — 64-bit code segment (IA-32e mode only)
AVL — Available for use by system software
BASE — Segment base address
D/B — Default operation size (0 = 16-bit segment; 1 = 32-bit segment)
DPL — Descriptor privilege level
G — Granularity
LIMIT — Segment Limit
P — Segment present
S — Descriptor type (0 = system; 1 = code or data)
TYPE — Segment type

Intel 64 and IA-32 Architectures Software Developer's Manual, Volumen 3, Capítulo 3: Figura 3-8

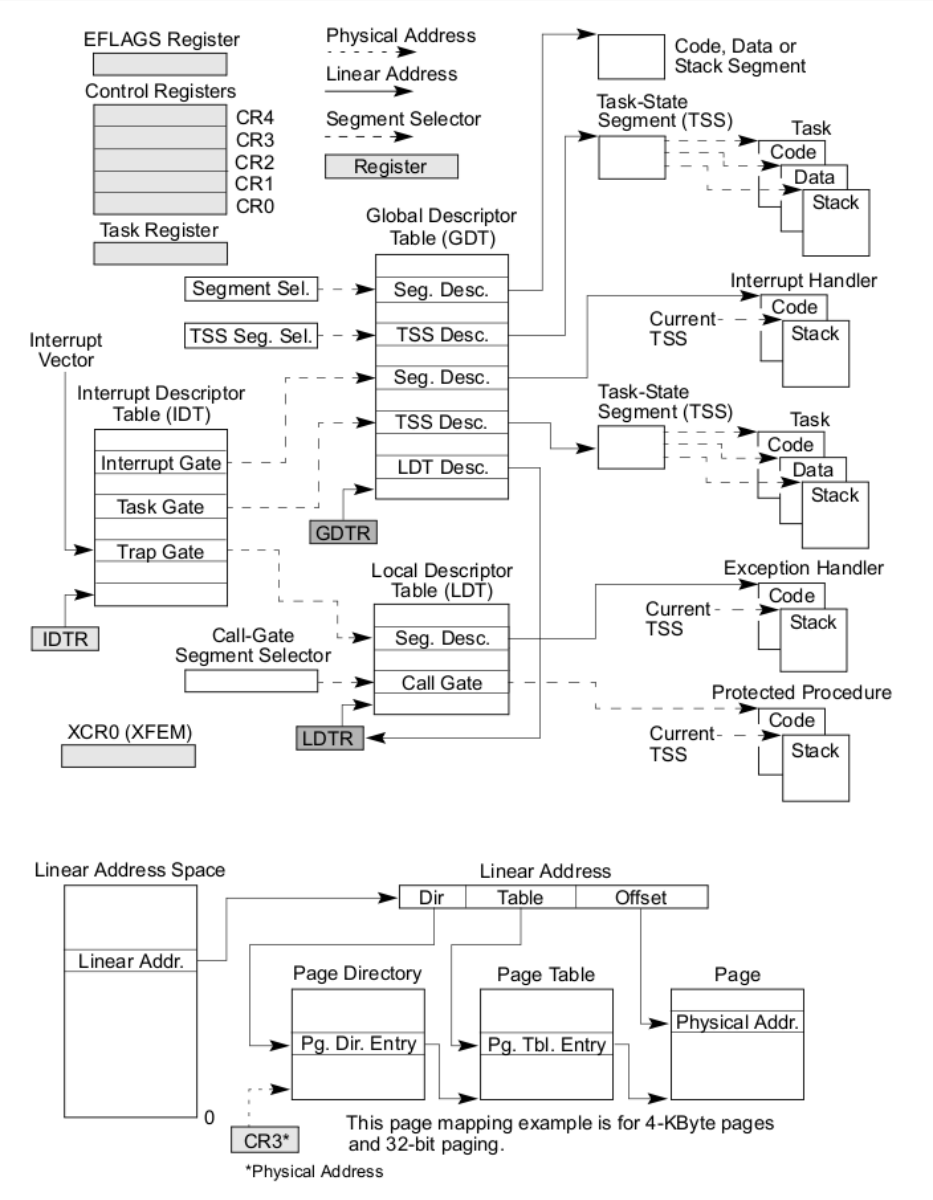
Tipo de Selector de segmento

Type

Type Field					Descriptor Type	Description
Decimal	11	10 E	9 W	8 A		
0	0	0	0	0	Data	Read-Only
1	0	0	0	1	Data	Read-Only, accessed
2	0	0	1	0	Data	Read/Write
3	0	0	1	1	Data	Read/Write, accessed
4	0	1	0	0	Data	Read-Only, expand-down
5	0	1	0	1	Data	Read-Only, expand-down, accessed
6	0	1	1	0	Data	Read/Write, expand-down
7	0	1	1	1	Data	Read/Write, expand-down, accessed
		C	R	A		
8	1	0	0	0	Code	Execute-Only
9	1	0	0	1	Code	Execute-Only, accessed
10	1	0	1	0	Code	Execute/Read
11	1	0	1	1	Code	Execute/Read, accessed
12	1	1	0	0	Code	Execute-Only, conforming
13	1	1	0	1	Code	Execute-Only, conforming, accessed
14	1	1	1	0	Code	Execute/Read, conforming
15	1	1	1	1	Code	Execute/Read, conforming, accessed

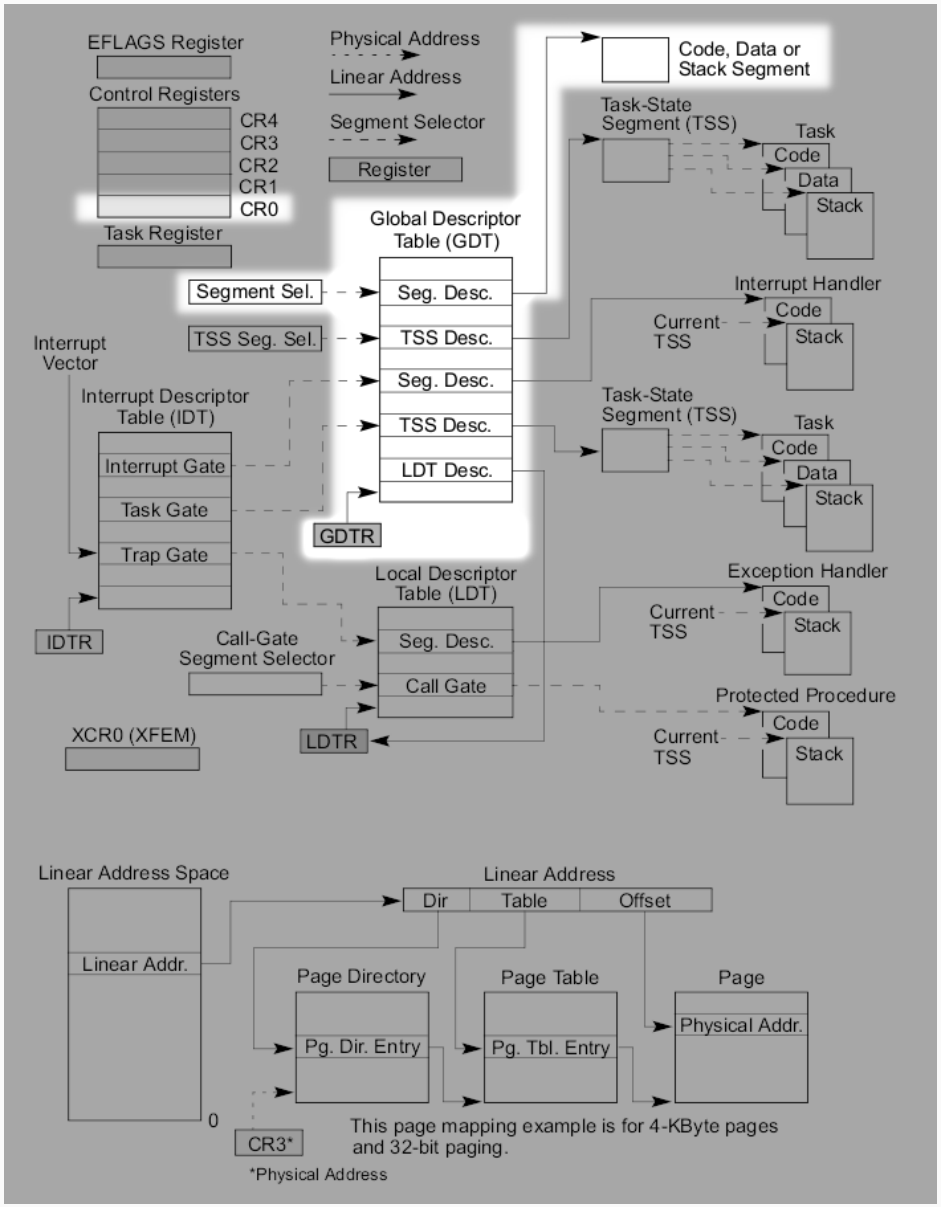
Intel 64 and IA-32 Architectures Software Developer's Manual, Volumen 3, Capítulo 3: Tabla 3-1

Big picture



Intel 64 and IA-32 Architectures Software Developer's Manual, Volumen 3, Capítulo 2: Figura 2-1

Big picture



Intel 64 and IA-32 Architectures Software Developer's Manual, Volumen 3, Capítulo 2: Figura 2-1

Modo real - programación en 16bits

Estamos solos contra el mundo...

Modo real - programación en 16bits

Estamos solos contra el mundo...

No hay librerías no hay `printf`, ¡no hay nada!

Tenemos un binario plano, y chau

¡Ojo con ejecutar los datos!

`section .data section .text` ... ja ja ja

¡Ojo con modificar el código en tiempo de ejecución!

No hay `segmentation fault`

Toda la memoria es casi nuestra

Modo real - programación en 16bits

Estamos solos contra el mundo...

No hay librerías no hay `printf`, ¡no hay nada!

Tenemos un binario plano, y chau

¡Ojo con ejecutar los datos!

`section .data section .text` ... ja ja ja

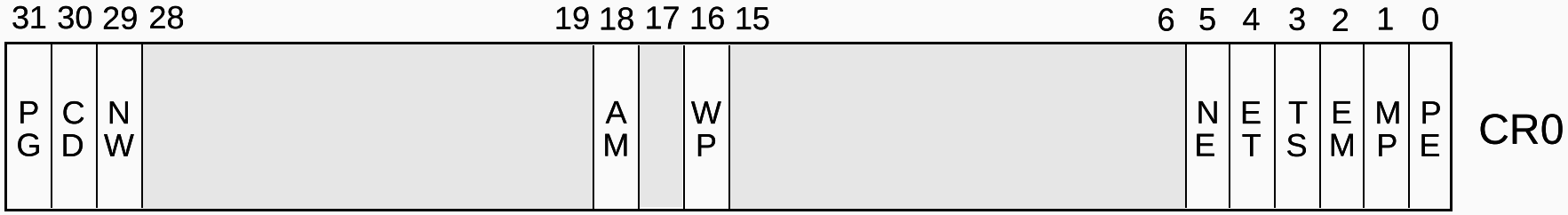
¡Ojo con modificar el código en tiempo de ejecución!

No hay `segmentation fault`

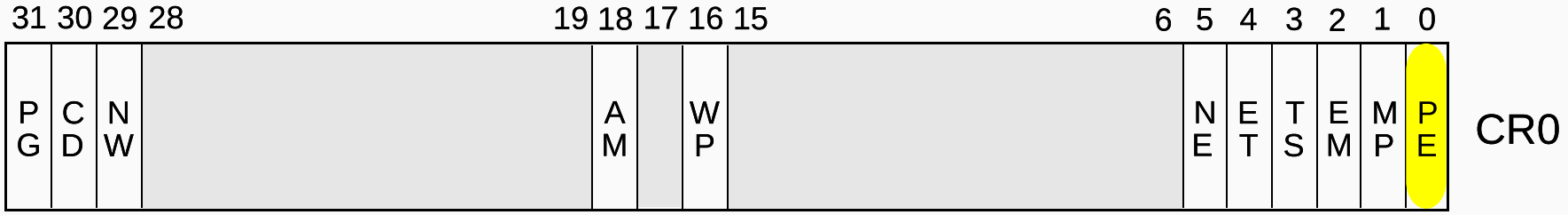
Toda la memoria es casi nuestra

Hasta que no pasemos a modo protegido,
tenemos el mejor 8086 de la historia

Pasar a modo protegido



Pasar a modo protegido



Activar **Modo Protegido** es setear en 1
el bit **PE** del registro de control **CR0**

Protected
Environment

cataplum.

1. onomat. U. para expresar ruido, explosión o golpe.

rae.org

¿Cómo sabemos dónde está la GDT?

En modo real, CS es un offset de memoria (shifteado 4 bits), no un selector de segmento válido. Por lo que hasta no cambiar el CS no estamos efectivamente en modo protegido.

Intel 64 and IA-32 Architectures Software Developer's Manual, Volumen 3, Sección 9.9.1: Switching to Protected Mode

¿Cómo sabemos dónde está la `GDT`?

Cargar el registro GDTR utilizando LGDT

En modo real, CS es un offset de memoria (shifteado 4 bits), no un selector de segmento válido. Por lo que hasta no cambiar el CS no estamos efectivamente en modo protegido.

Intel 64 and IA-32 Architectures Software Developer's Manual, Volumen 3, Sección 9.9.1: Switching to Protected Mode

Pasaje a Modo Protegido

¿Cómo sabemos dónde está la GDT?

Cargar el registro GDTR utilizando LGDT

¿Qué tiene la GDT?

En modo real, CS es un offset de memoria (shifteado 4 bits), no un selector de segmento válido. Por lo que hasta no cambiar el CS no estamos efectivamente en modo protegido.

Intel 64 and IA-32 Architectures Software Developer's Manual, Volumen 3, Sección 9.9.1: Switching to Protected Mode

Pasaje a Modo Protegido

¿Cómo sabemos dónde está la GDT?

Cargar el registro GDTR utilizando LGDT

¿Qué tiene la GDT?

Al menos un descriptor nulo, un descriptor de código y uno de datos

---

En modo real, CS es un offset de memoria (shifteado 4 bits), no un selector de segmento válido. Por lo que hasta no cambiar el CS no estamos efectivamente en modo protegido.

Intel 64 and IA-32 Architectures Software Developer's Manual, Volumen 3, Sección 9.9.1: Switching to Protected Mode

Pasaje a Modo Protegido

¿Cómo sabemos dónde está la `GDT`?

Cargar el registro GDTR utilizando LGDT

¿Qué tiene la `GDT`?

Al menos un descriptor nulo, un descriptor de código y uno de datos

¿Cuál es la próxima instrucción a ejecutar?

En modo real, CS es un offset de memoria (shifteado 4 bits), no un selector de segmento válido. Por lo que hasta no cambiar el CS no estamos efectivamente en modo protegido.

Intel 64 and IA-32 Architectures Software Developer's Manual, Volumen 3, Sección 9.9.1: Switching to Protected Mode

Pasaje a Modo Protegido

¿Cómo sabemos dónde está la GDT?

Cargar el registro GDTR utilizando LGDT

¿Qué tiene la GDT?

Al menos un descriptor nulo, un descriptor de código y uno de datos

¿Cuál es la próxima instrucción a ejecutar?

La instrucción en la dirección CS:EIP

En modo real, CS es un offset de memoria (shifteado 4 bits), no un selector de segmento válido. Por lo que hasta no cambiar el CS no estamos efectivamente en modo protegido.

Intel 64 and IA-32 Architectures Software Developer's Manual, Volumen 3, Sección 9.9.1: Switching to Protected Mode

Pasaje a Modo Protegido

¿Cómo sabemos dónde está la `GDT`?

Cargar el registro GDTR utilizando LGDT

¿Qué tiene la `GDT`?

Al menos un descriptor nulo, un descriptor de código y uno de datos

¿Cuál es la próxima instrucción a ejecutar?

La instrucción en la dirección CS:EIP

¿Qué valor tiene CS y cómo lo cambiamos?

En modo real, CS es un offset de memoria (shifteado 4 bits), no un selector de segmento válido. Por lo que hasta no cambiar el CS no estamos efectivamente en modo protegido.

Intel 64 and IA-32 Architectures Software Developer's Manual, Volumen 3, Sección 9.9.1: Switching to Protected Mode

Pasaje a Modo Protegido

¿Cómo sabemos dónde está la `GDT`?

Cargar el registro GDTR utilizando LGDT

¿Qué tiene la `GDT`?

Al menos un descriptor nulo, un descriptor de código y uno de datos

¿Cuál es la próxima instrucción a ejecutar?

La instrucción en la dirección CS:EIP

¿Qué valor tiene CS y cómo lo cambiamos?

FAR JUMP a la siguiente instrucción

```
JMP <selector>:<offset>
```

En modo real, CS es un offset de memoria (shifteado 4 bits), no un selector de segmento válido. Por lo que hasta no cambiar el CS no estamos efectivamente en modo protegido.

Intel 64 and IA-32 Architectures Software Developer's Manual, Volumen 3, Sección 9.9.1: Switching to Protected Mode

Pasaje a Modo Protegido

- Completar la GDT
- Deshabilitar interrupciones
- Cargar el registro GDTR con la dirección base de la GDT
- Setear el bit PE del registro CR0

```
MOV eax, cr0
OR eax, 1
MOV cr0, eax
```

- FAR JUMP a la siguiente instrucción

```
JMP <selector>:<offset>
```

- Cargar los registros de segmento (`DS`, `ES`, `GS`, `FS` y `SS`)

Intel 64 and IA-32 Architectures Software Developer's Manual, Volumen 3, Sección 9.9.1: Switching to Protected Mode

Pasaje a Modo Protegido