

# Unidad de Paginación

## Organización del Computador II

David Alejandro González Márquez -> Ezequiel Barrios

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

17-10-2019

# Referencias

→ Teórica: **Gestión de Memoria**

→ **7. Paginación** (págs 74 a 104, págs 117 a 122)

→ Intel® 64 and IA-32 Architectures Software Developer's Manual:

**Volumes 3A, 3B, 3C, and 3D: System Programming Guide**

→ **2.5 CONTROL REGISTERS** (cr0, cr1, cr2, cr3, etc...)

→ **4.3 32-BIT PAGING**

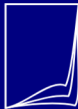
→ **4.7 PAGE-FAULT EXCEPTIONS**

→ **4.8 ACCESSED AND DIRTY FLAGS**

→ **6.15 EXCEPTION AND INTERRUPT REFERENCE**

Interrupt 14—Page-Fault Exception (#PF)

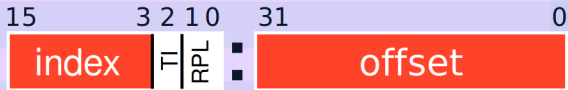
# Unidad de Paginación



## Organización del Computador II

David Alejandro González Márquez

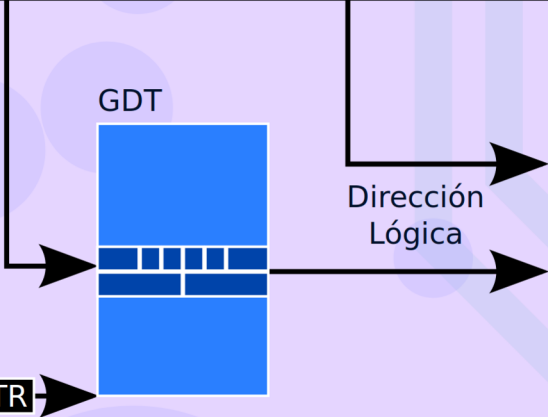
Departamento de Computación  
Facultad de Ciencias Exactas y Naturales  
Universidad de Buenos Aires



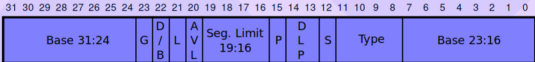
GDT

Dirección  
Lógica

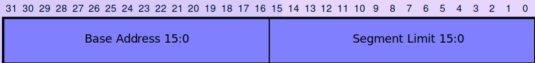
GDTR



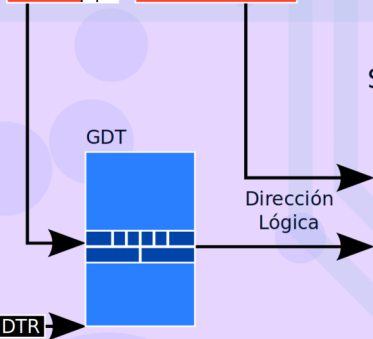
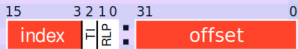
# Segment Descriptor



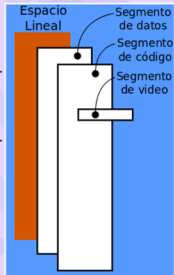
4



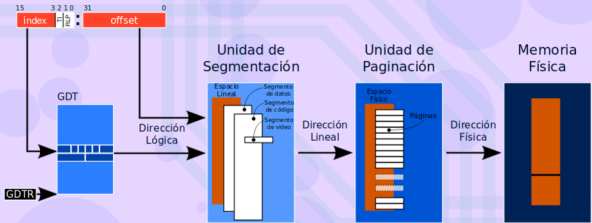
0



## Unidad de Segmentación

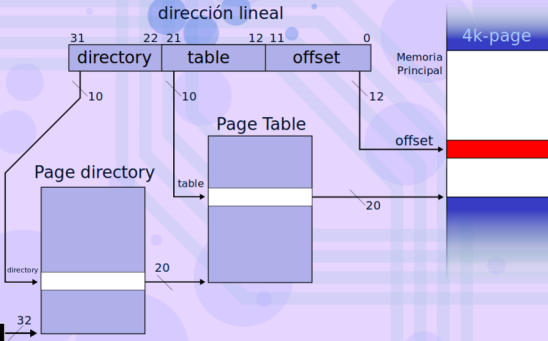


# Unidades de administración de memoria

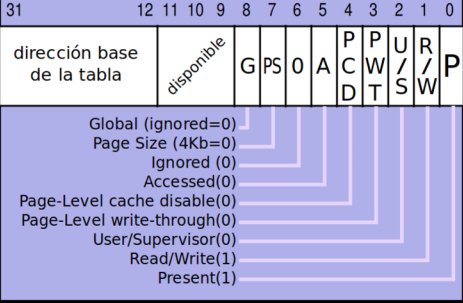




# Mecanismo de Paginación



## PDE (page directory entry)



directory

32

table



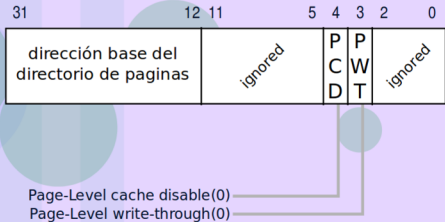
## PTE (page table entry)

31 12 11 10 9 8 7 6 5 4 3 2 1 0

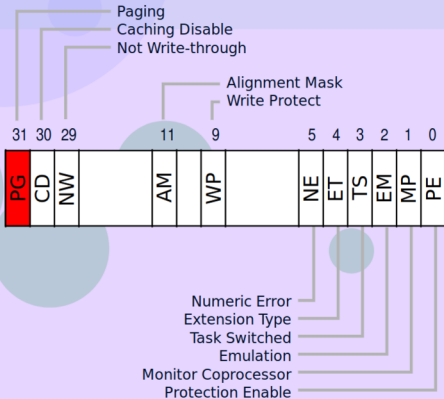
dirección base de la página	disponible	G	P A T	D	A	P C D	P W T	U / S	R / W	P
--------------------------------	------------	---	-------------	---	---	-------------	-------------	-------------	-------------	---

Global (ignored=0)  
Page Table Attribute Index (0)  
Dirty bit (0)  
Accessed(0)  
Page-Level cache disable(0)  
Page-Level write-through(0)  
User/Supervisor(0)  
Read/Write(1)  
Present(1)

CR3

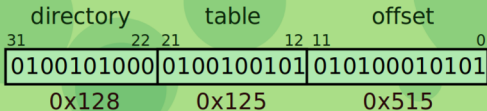


CR0




Ejemplo,  
queremos resolver la siguiente dirección...

Dirección Lineal = 0x4A125515



a ver que hace el CPU...





# CPU

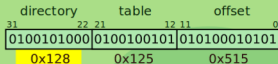
(1) buscar el CR3

CR3

00011	000
-------	-----

# Page Directory

(2) buscamos una entrada dentro del directorio de páginas



(3) decodificamos la PDE

0x11000

00023003

$0x11000 + 0x128 * 4$

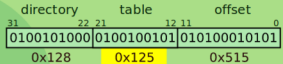


# Page Table

(4) buscamos una entrada dentro del page table

(5) decodificamos la PTE

00002003

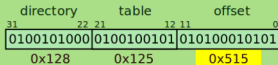


0x23000

$0x23000 + 0x125 * 4$

4k-page

A2C732AB



0x02000

0x02000+0x515

(6) ¡al fin!

ahora buscamos la  
entrada de memoria  
que queremos

## Hasta ahora... (segmentación)

- Creamos y completamos la GDT
- Habilitamos la A20
- Deshabilitamos las interrupciones (cli)
- Cargamos el registro GDTR con la dirección base y el límite de la GDT
- Seteamos el bit PE del registro CR0
- Realizamos un JUMP FAR a la siguiente instrucción
- Actualizamos la información de los registros de segmento DS, ES, GS, FS y SS
- Pintamos el tablero del juego en la pantalla

## Hasta ahora... (interrupciones)

- Creamos y completamos una IDT básica
- Asociamos cada excepción a su rutina de atención
- Asociamos las rutinas de interrupciones externas
- Creamos una entrada para el llamado a sistema (interrupción 47)
- Programamos la rutina de atención de reloj
- Programamos la rutina de atención de teclado
- Programamos una versión básica del llamado a sistema que retorna 0x42 (luego será modificado)
- ...

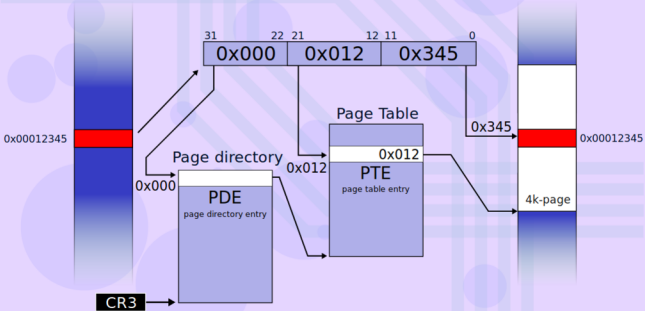
## ¿Y para paginación del Kernel...?

- Armar un directorio de páginas y tablas de páginas
- Poner en CR3 la dirección base del directorio de páginas
- Limpiar bits PCD y PWT de CR3
- Setear el bit PG de CR0

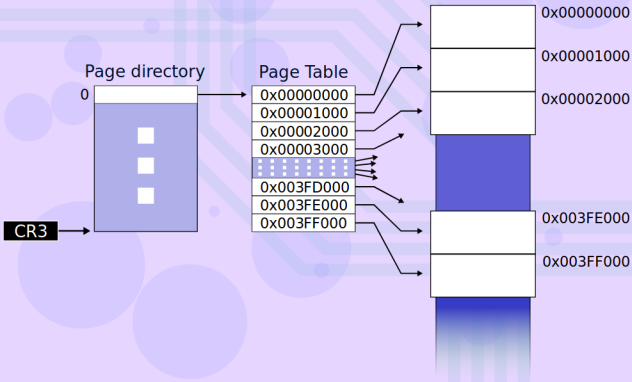
# Identity Mapping

Dirección  
Lineal

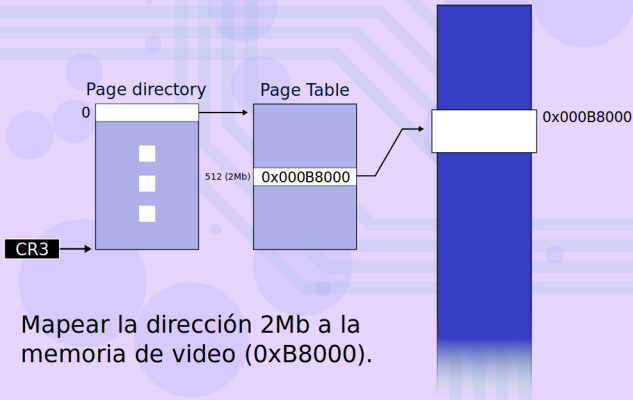
Dirección  
Física



# Ejemplo Identity Mapping



# Ejemplo Pagina para video





# Resolver direcciones - ejemplos



Segmentación	Dirección	
	Lineal	Física
	Paginación	
Ej: dirección directa mov [0xb8000],0x0 lógica = ds : 0xb8000	(segmento flat) lineal = 0xb8000	(identity mapping) física = 0xb8000
Ej: segmento de video mov [es:0x0],0x0 lógica = es : 0x0	(base del segmento de video) lineal = 0xb8000	(identity mapping) física = 0xb8000
Ej: dirección dos megas mov [0x200000],0x0 lógica = ds : 0x200000	(segmento flat) lineal = 0x200000	(mapeo especial 0x200000 -> 0xb8000) física = 0xb8000

... activar paginación ...

```
mov  eax, page_directory  
mov  cr3,  eax
```

```
mov  eax,  cr0  
or   eax,  0x80000000    ;habilito paginacion  
mov  cr0,  eax
```

## Ejercicios ...

- Identity mapping para las direcciones desde la 0x00000000 a la 0x003FFFFFFF
- ~~● La dirección virtual 2Mb mapeada a la memoria de video (0xB8000)~~