

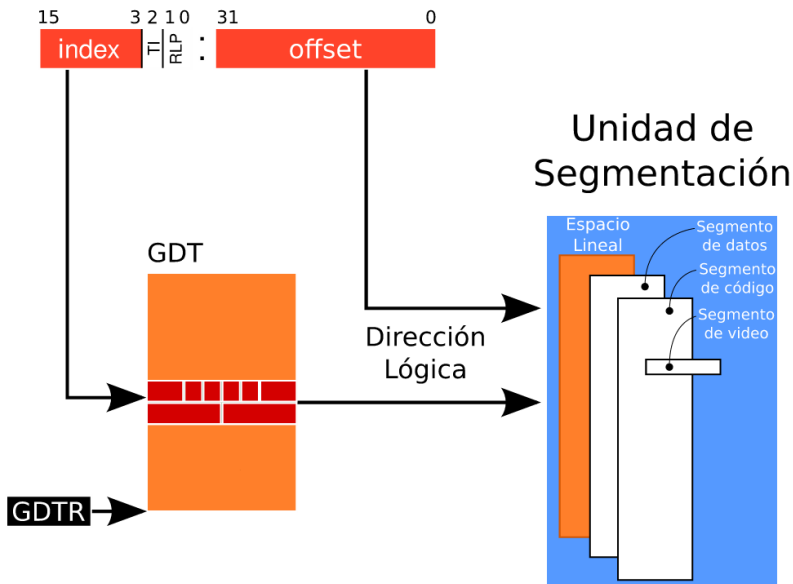
Paginación

Nadia Heredia

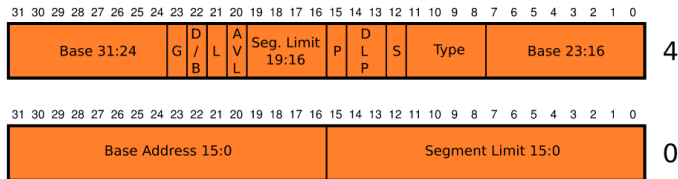
Organización del Computador II, 2do cuat. de 2014



Segmentación



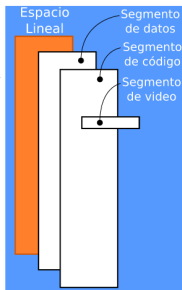
Segment Descriptor



- Crear la GDT.
- Habilitar la A20
- Deshabilitar las interrupciones (cli).
- Cargar el registro GDTR con la dirección base y el límite de la GDT.
- Setear el bit PE del registro CR0.
- Realizar un FAR JUMP a la siguiente instrucción.
- Actualizar la información de los registros DS, ES, FS, GS y SS.
- Mostrar por pantalla la leyenda.

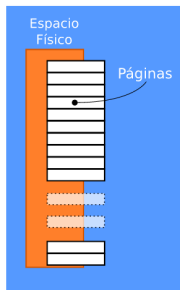
Unidad de Paginación

Unidad de Segmentación



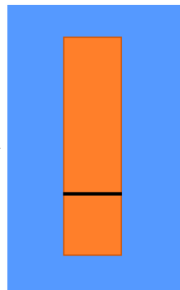
Dirección
Lineal

Unidad de Paginación

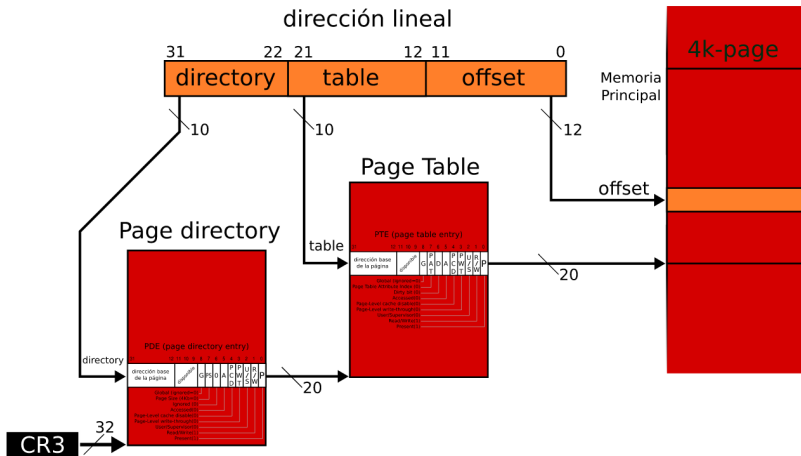


Dirección
Física

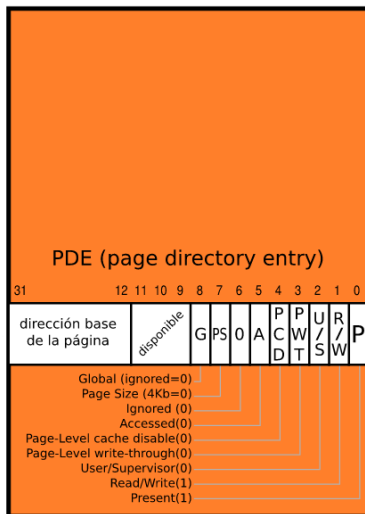
Memoria Física



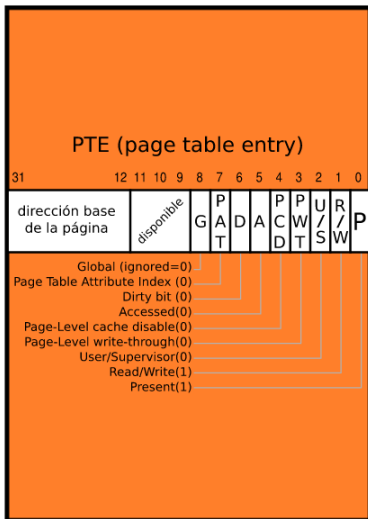
Direcciones Lineales



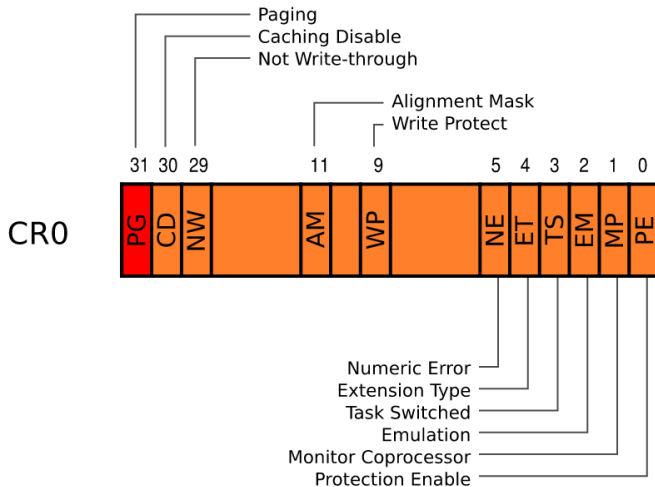
Page directory



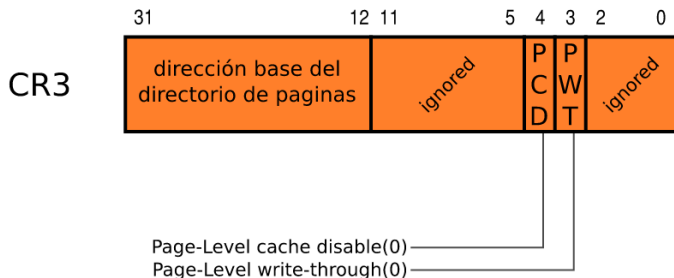
Page Table



Registro de Control CR0



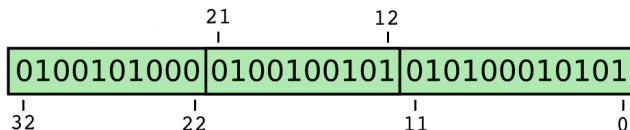
Registro de Control CR3



Ejemplo

Queremos resolver la siguiente dirección lineal 0x4A125515

0x4A125515 = 0100 1010 0001 0010 0101 0101 0001 0101



Separamos

Page Directory Offset	Bits 31 a 22	0001 0010 1000 = 0x128
Page Table Offset	Bits 21 a 12	0001 0010 0101 = 0x125
Page Offset	Bits 11 a 0	0101 0001 0101 = 0x515

Leemos CR3

```
CR3 = 0000 0000 0000 0001 0001 1100 1111 0011
```

Obtenemos la dirección base del directorio de páginas

```
dir_page_directory = 0000 0000 0000 0001 0001 0000 0000  
                    0000
```

```
dir_page_directory = 0x11000
```

Vamos al **Page Directory**, y usando el **Page Directory Offset** buscamos la entrada correspondiente (pde).

```
dir_pde = 0x11000 + 0x128 * 4
```

Leemos la entrada y obtenemos la dirección de la tabla de páginas.

```
pde = 0x23003
```

```
dir_page_table = 0x23000
```

Vamos a la **Page Table**, y usando el **Page Table Offset** buscamos la entrada correspondiente (pte).

```
dir_pte = 0x23000 + 0x125 * 4
```

Leemos la entrada y obtenemos la dirección de la página.

```
pte = 0x2003
```

```
dir_page = 0x2000
```

Vamos a la **Página**, y usando el **Page Offset** buscamos la dirección física correspondiente.

```
dir_fisica = 0x2000 + 0x515
```

¡Terminamos! Ahora tenemos una asociación nueva entre una dirección lineal y una física.

```
0x4A125515 → 0x2515
```


- 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.

```
mov eax, page_directory
mov cr3, eax

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

¡Cuidado!

Antes de habilitar la paginación debemos armar el directorio de páginas y tener al menos una tabla de páginas.

Armar el directorio de páginas y completarlo.

Armar las tablas de páginas necesarias para que

- Haya **Identity Mapping** sobre los primeros 4Mb de memoria.
- La dirección virtual 2Mb esté mapeada a la dirección de video (0xB8000).

```
xor ecx, ecx
mov ebx, ADDR_PAGE_DIR

.pageDirectory:
    cmp ecx, 1024
    je .finPageDirectory

    mov dword [ebx + ecx * 4], 0x0
    inc ecx
    jmp .pageDirectory

.finPageDirectory:
mov dword [ADDR_PAGE_DIR], ADDR_PAGE_TABLE + 0x3
    ;supervisor, read/write, present
```

```
xor ecx, ecx
mov ebx, ADDR_PAGE_TABLE
xor edx, edx
```

```
.pageTable:
    cmp ecx, 1024
    je .finPageTable
```

```
    mov [ebx + ecx * 4], edx
    add dword [ebx + ecx * 4], 0x03
    ;supervisor, read/write, present
    add edx, 0x1000
    inc ecx
    jmp .pageTable
```

```
.finPageTable:
mov dword [ADDR_PAGE_TABLE0 + 512 * 4], 0x000B8003
```

¿Preguntas?