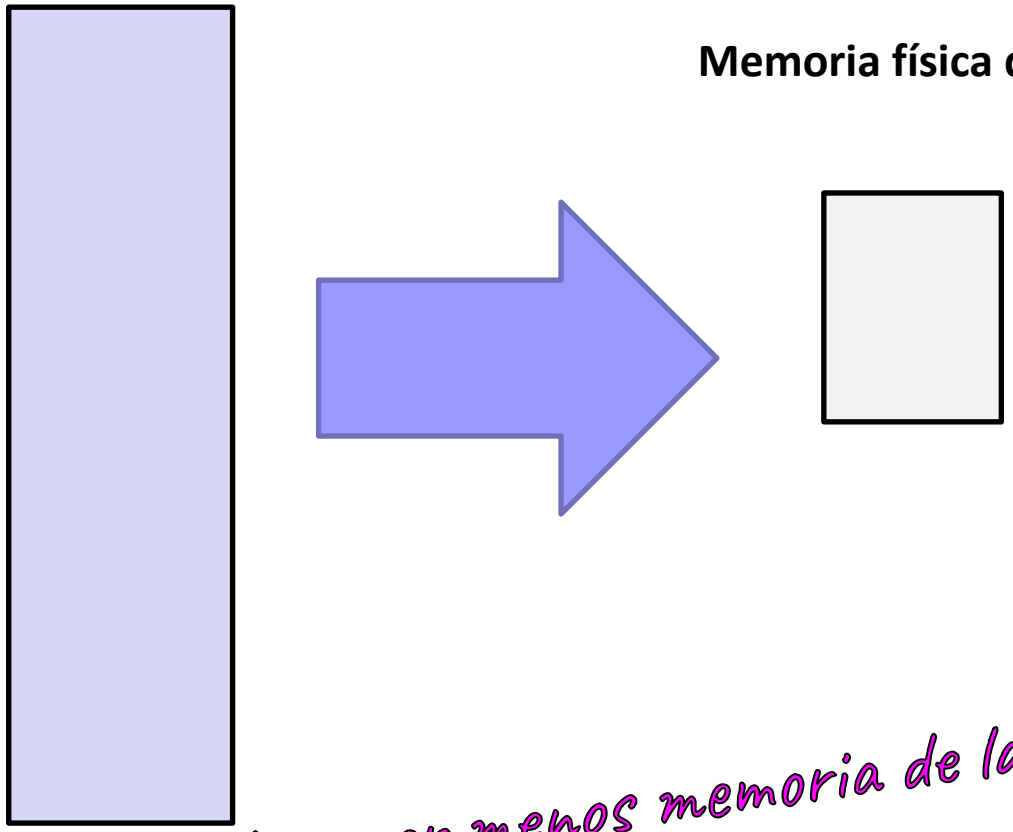
Memoria Virtual

Problemática

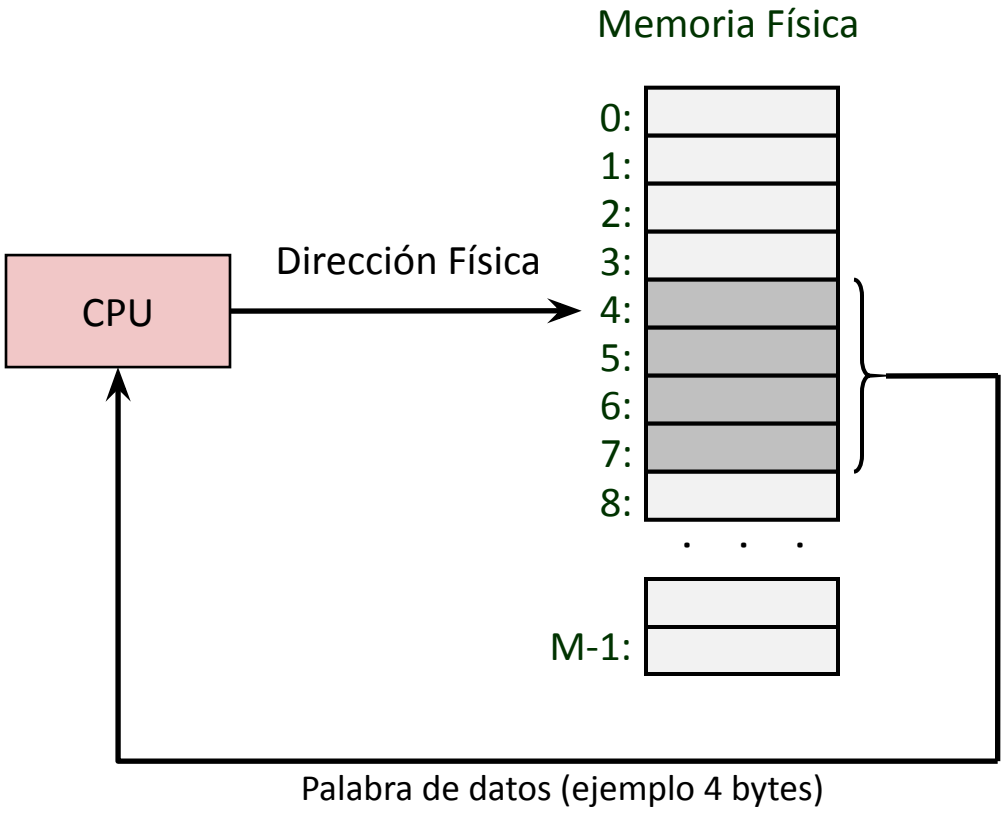
Suponemos un procesador de 32 bits en una Pc con 1GB de Memoria

Memoria virtual de 4GB



Suelo tener menos memoria de la que puedo apuntar

Direccionamiento físico



Lo que programo sale por el bus de direcciones

Problema

- **Si asigno a las apps direcciones físicas de memoria luego es complicado:**
  - **Reubicar espacios**
  - **Usar más espacios que lo que tengo de memoria**

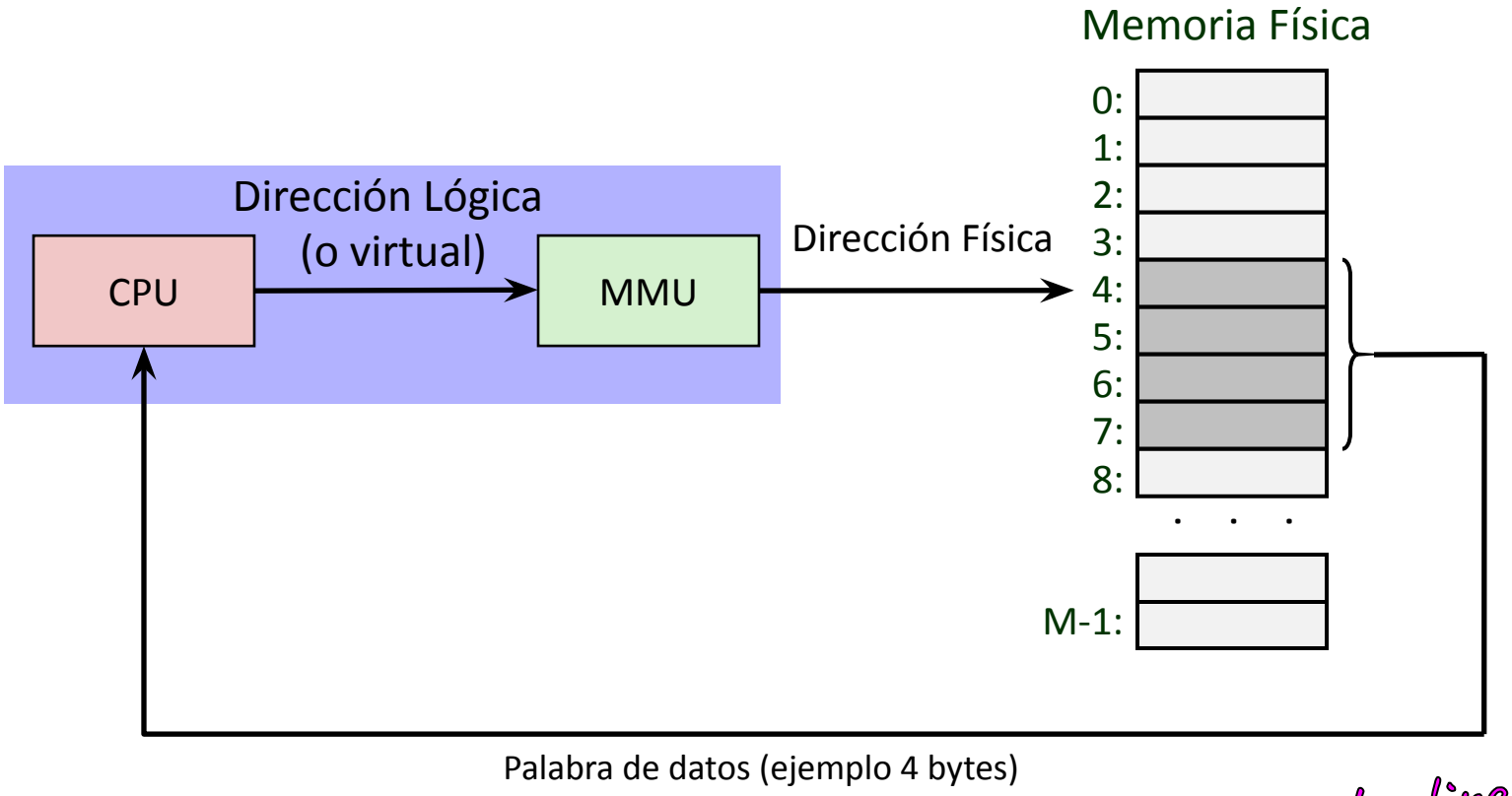
Necesito alguna herramienta para esto……

MMU - Unidad de Maneja de memoria

Permite:

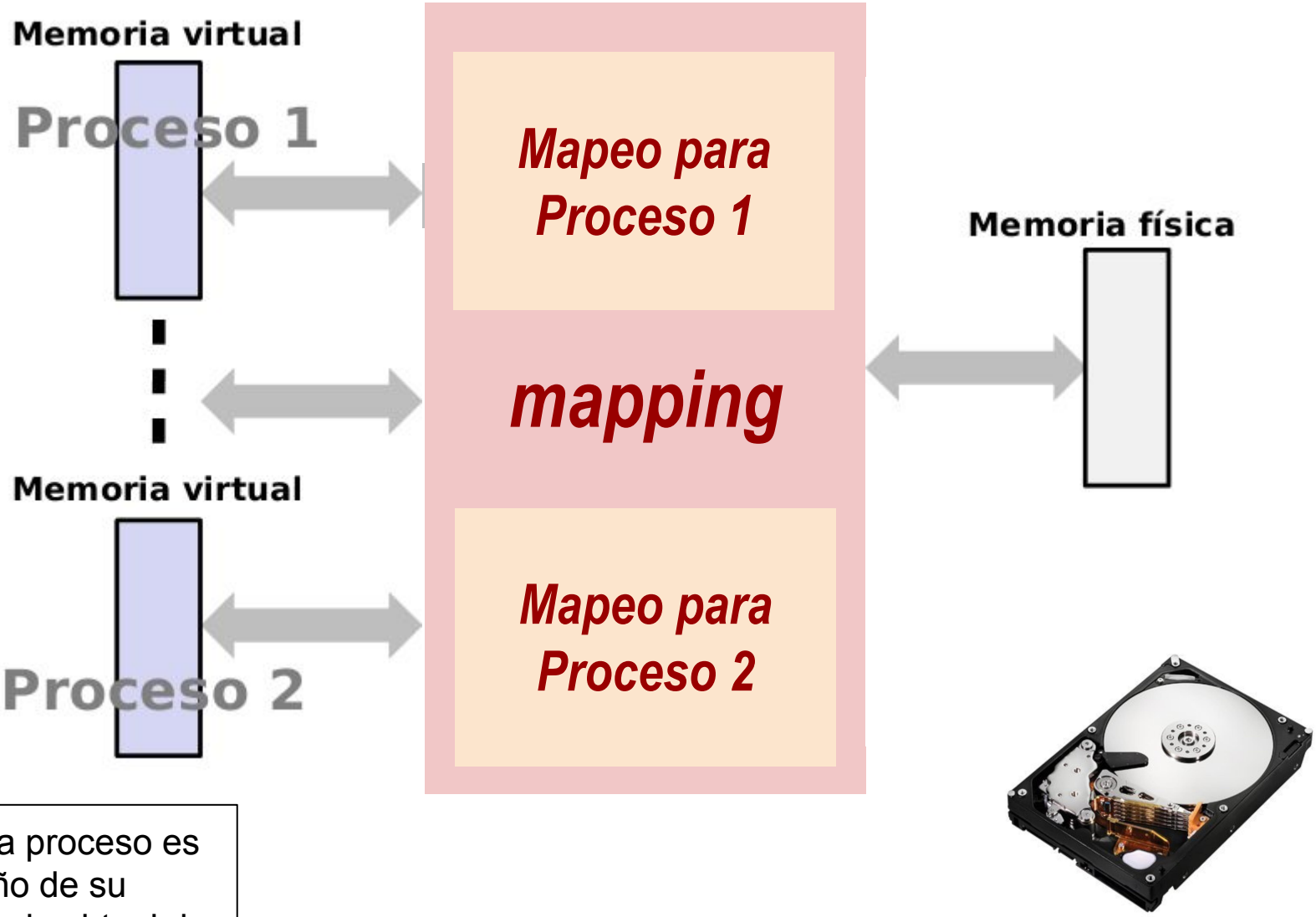
- **Dividir en páginas ó segmentos la memoria**
- **Chequear permisos**
- **Alterar (o no) la dirección “lógica” antes que sea “física”**

Direccionamiento virtual



Lo que programo tiene chequeos y posibles alteraciones de direcciones

Solución: Indirección/Abstracción



Cada proceso es dueño de su espacio virtual de memoria

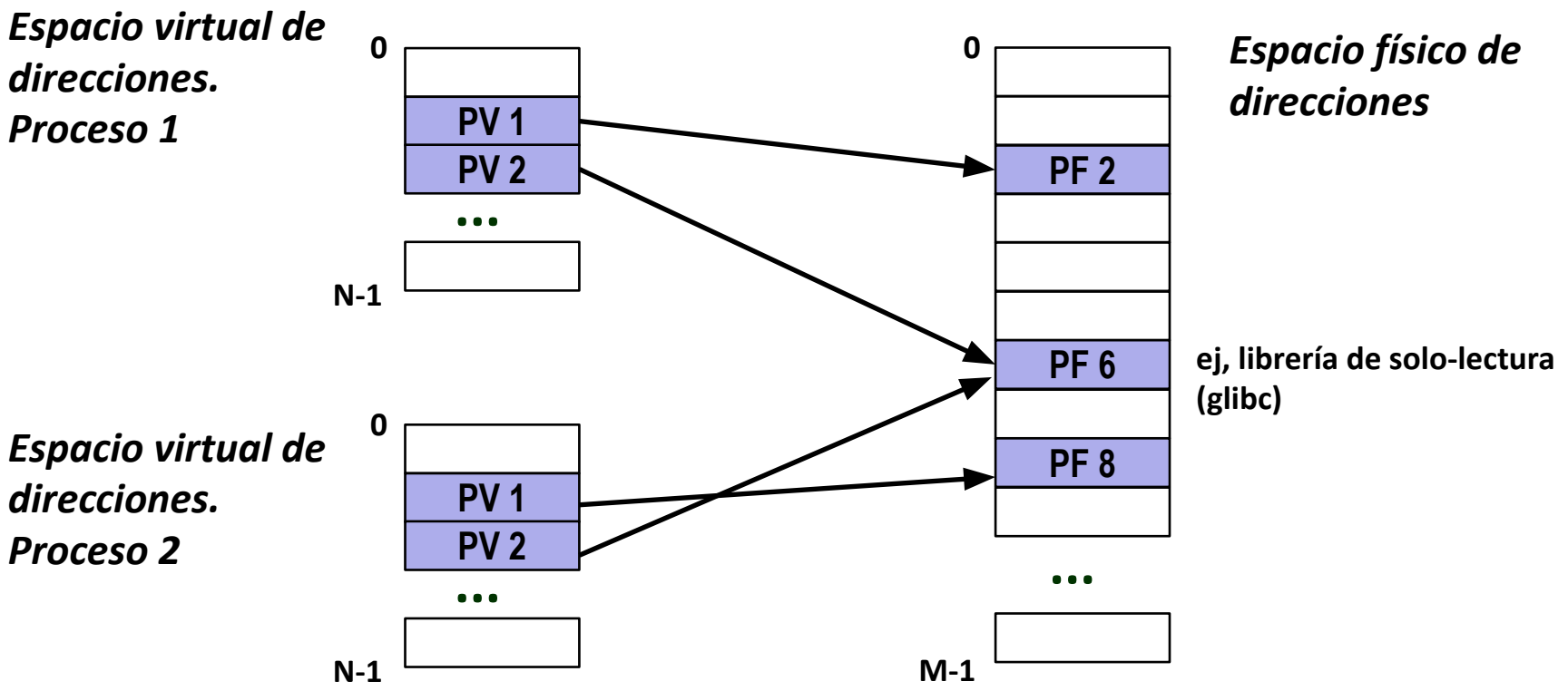
Paginación

Paginación

- **Divide al mapa de memoria y a la memoria física en "páginas"**
- **Cada página tiene un tamaño fijo.**
- **Genera mapeo de Pag-Virtuales (páginas) a Pag-Fisica (marcos)**
- **Tiene esquema de permisos**

VM para múltiples procesos

Traducción de direcciones



Paginación – Ejercicio 1

Desarrolle un sistema de mapeo para un procesador Intel de 32 bits que puede manejar hasta 4GB de memoria física.

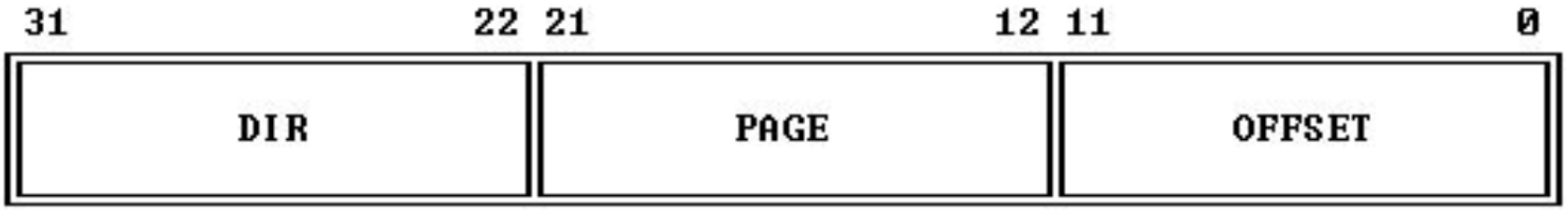
1. **¿Qué tamaño de página elige?**
2. **¿En cuántas páginas quedó divida la memoria física y la memoria virtual ?**
3. **¿Cómo reacciona su sistema para asignar y para liberar memoria?**

Paginación – Ejercicio 2

Repita el Ejercicio 1 para un sistema con 1GB de memoria física.

Elección de Intel para 32 bits

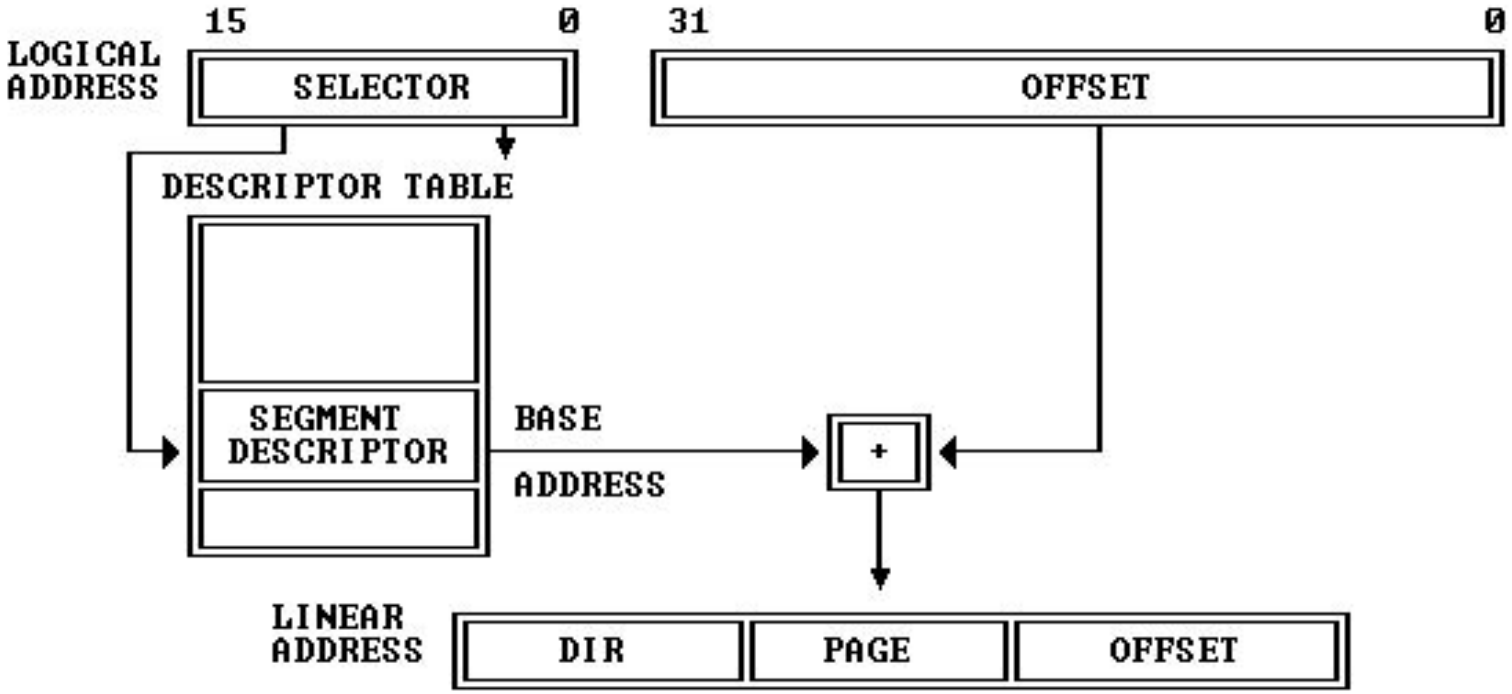
Figure 5-8. Format of a Linear Address



Paginación

Con un selector y el offset, a través de una tabla de descriptores (GDT o LDT), se obtiene una dirección lineal (de 32 bits)

Figure 5-2. Segment Translation



Paginación

La dirección Lineal al pasar el módulo de paginación se convierte en dirección física. El registro CR3 de 32 bits indica la ubicación del Directorio de páginas.

Figure 5-8. Format of a Linear Address

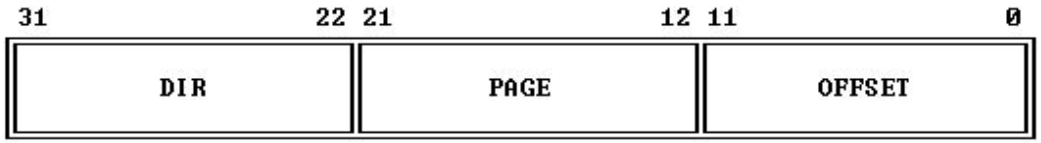
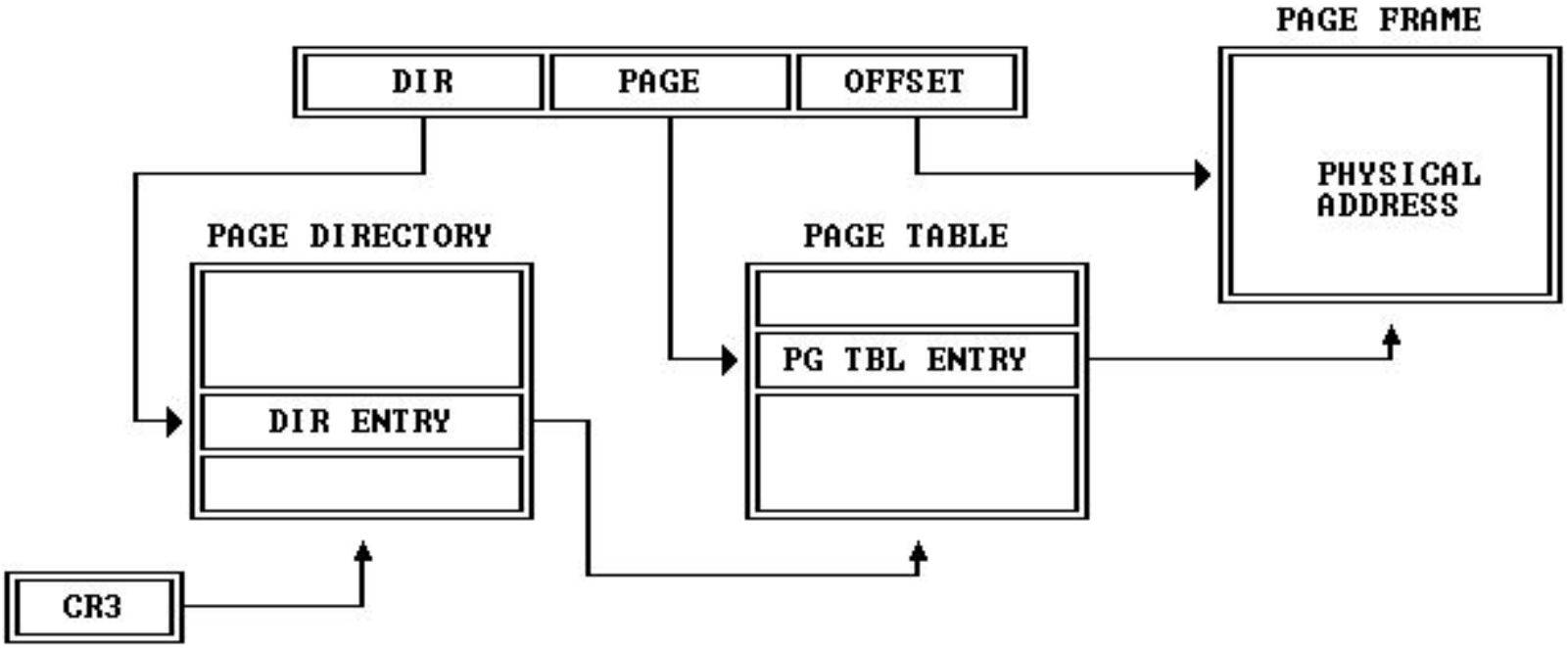


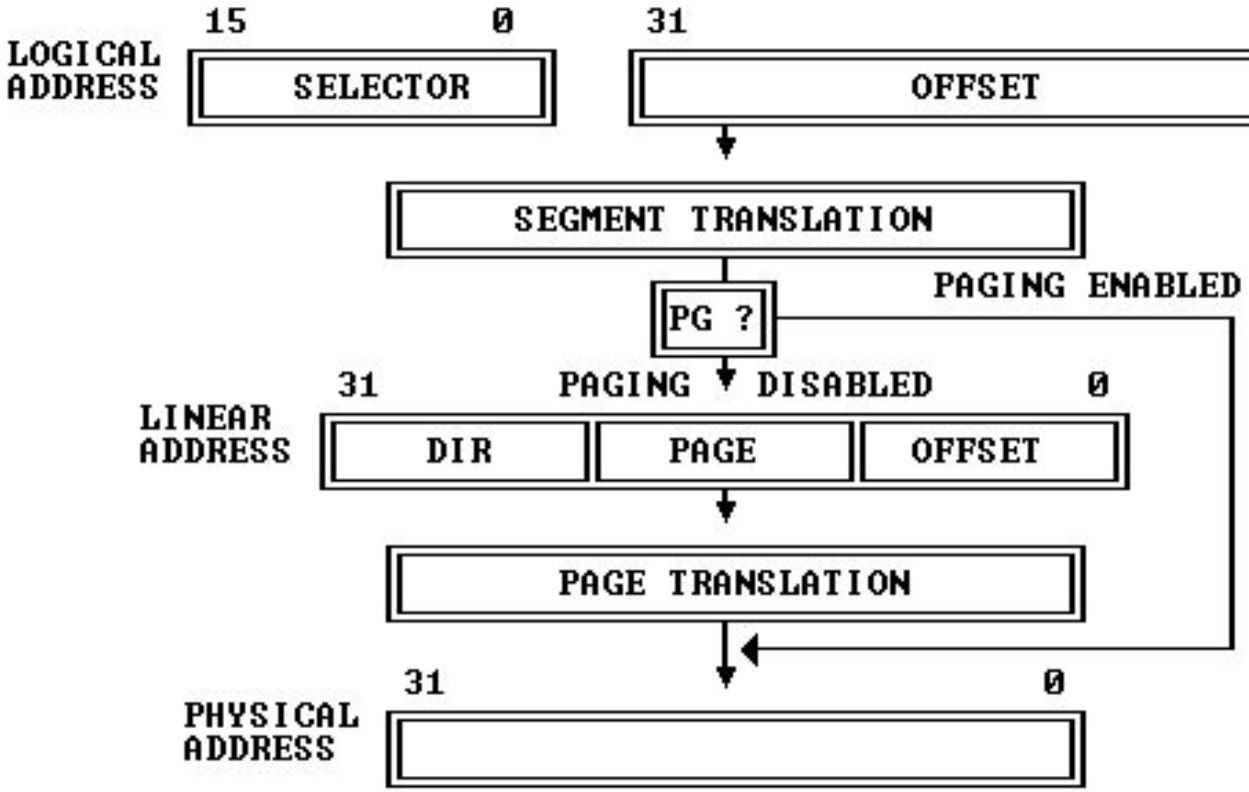
Figure 5-9. Page Translation



Paginación

Recordemos que la unidad de Paginación se puede deshabilitar, con el bit PG que se encuentra en el registro CR0.

Figure 5-1. Address Translation Overview



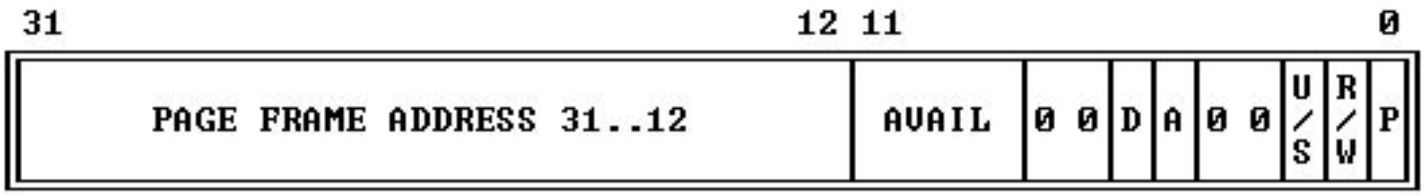
Paginación

El Directorio de paginas , puede referenciar 1024 Tablas de Paginas.

Luego cada una de esas 1024 Tablas de Paginas puede referenciar 1024 Paginas, en la memoria física.

Los elementos de estas tablas tienen el mismo formato:

Figure 5-10. Format of a Page Table Entry



P - PRESENT
R/W - READ/WRITE
U/S - USER/SUPERVISOR
D - DIRTY
AVAIL - AVAILABLE FOR SYSTEMS PROGRAMMER USE

NOTE: 0 INDICATES INTEL RESERVED. DO NOT DEFINE.

Paginación

El bit P, o “Present Bit”, indica cuando tiene valor 1, que esa tabla se puede utilizar para obtener una dirección de memoria. Si tiene valor 0, no se puede utilizar.

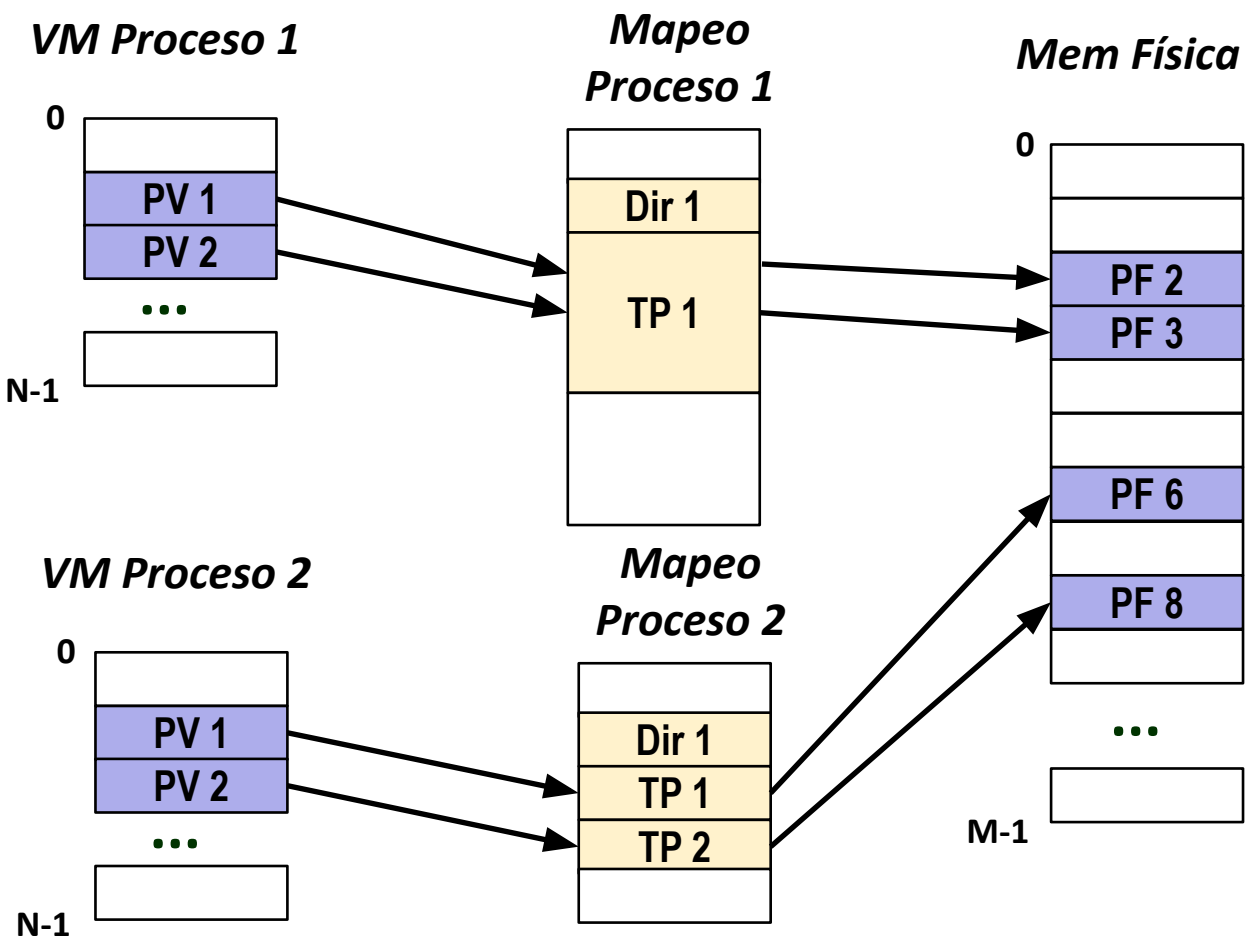
Figure 5-11. Invalid Page Table Entry

Al encontrar un bit P=0, el procesador genera una excepción. Los sistemas operativos que tiene soporte de memoria virtual, utilizan esta excepción para disparar una rutina que lleve a memoria la página faltante. Que seguramente se encontrará en disco.

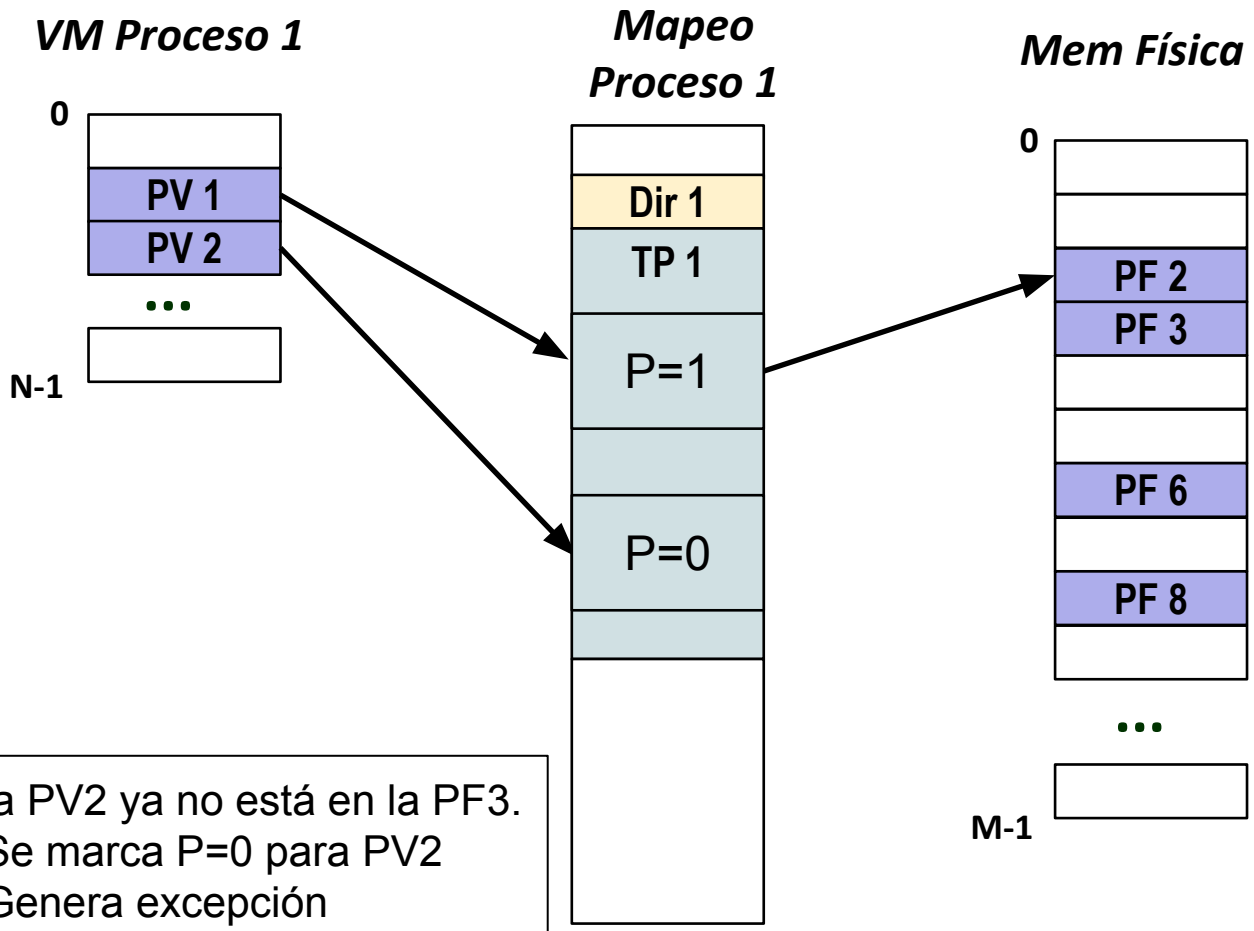
El bit U/S, divide las paginas en 2 niveles de privilegios, (Usuario y Supervisor).

Si no se respetan los accesos se produce una excepción. Si el procesador esta ejecutando en niveles 0,1 y 2 es nivel Supervisor, sino ejecuta en nivel 3 es Usuario.

EJ Paginación dos procesos



EJ Page fault



- la PV2 ya no está en la PF3.
- Se marca P=0 para PV2
- Genera excepción
- La busco en disco

Paging y swapping

Cuando se necesita una página en memoria y la memoria está completa, Linux elige un pagina que no ha sido accedida últimamente y realiza un "page out", que consiste en guardar esa página en disco.

Generalmente en una zona especial destinada para ello, puede ser una partición o un archivo en el filesystem

Luego setea en la página "vieja" el bit de Presencia en 0, y guarda en su índice la dirección donde la debe encontrar en el disco rígido.

A este concepto se lo denomina "pagging".

El concepto de swapping se refiere a guardar en disco TODAS las páginas de un proceso.

En Linux existe un thread llamado "kswapd" que se encarga de hacer este trabajo.

Es interesante estudiar los algoritmos que se aplican para decidir qué página debe ser "bajada", ya que si se decide mal, dicha página puede llegar a tener la próxima instrucción a ejecutar, o un dato que debe ser accedido.

Protección combinada

Cuando la paginación está habilitada, el procesador, primero chequea los permisos en la segmentación y luego en la paginación.

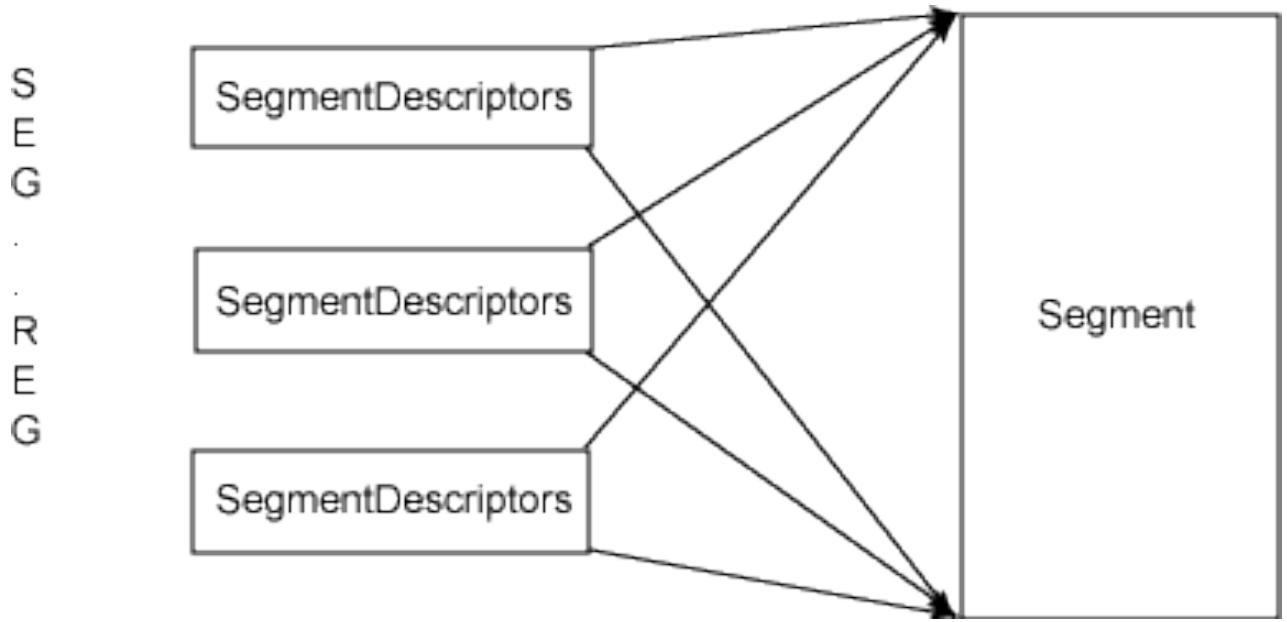
Por ejemplo se puede definir un segmento de datos, que está compuesto por páginas, donde algunas son de escritura y otras se sólo lectura.

Debido a este manejo de memoria, algunos sistemas operativos como Unix, eligen utilizar el modelo flat de memoria.

Modelo de memoria Flat

Unix declara dos segmentos de código y de datos que referencian a toda la memoria.

Esto le permite un manejo mas simple de la mismas y a su vez mas portabilidad ya que muchos procesadores no tiene unidad de segmentación pero si de paginación.



Linux

- **Hasta la versión 2.2 de Kernel, Linux utilizaba los TSS para realizar conmutación de tareas.**
- **A partir de la versión 2.4, realiza la conmutación de tareas mediante funciones.**
- **Los archivos que se utilizan para la arquitectura que estamos viendo se encuentran en: /usr/src/linux/arch**
- **Ahí podremos encontrar los archivos en ASM que setean los registros del microprocesador.**
- **Linux utiliza la paginación de hardware**
- **En microprocesadores de 64 bits, agrega un nivel extra de páginas (en el medio de las de hardware) llamado “middle directory”.**