

# Estructura de Computadores

## Tema 7. Memoria Virtual

## Problemas con el sistema de memoria

- ▶ ¿Cómo soportar **múltiples programas** de forma **simultánea**?
  - ▶ Compilación/Ensamblado/Enlazado asignan direcciones de memoria absolutas a instrucciones y datos
  - ▶ Hay que reubicar los datos en tiempo de ejecución para evitar conflictos
  - ▶ Hay que evitar que un programa acceda a los datos de otros programas

## Problemas con el sistema de memoria

- ▶ ¿Cómo soportar **múltiples programas** de forma **simultánea**?
  - ▶ Compilación/Ensamblado/Enlazado asignan direcciones de memoria absolutas a instrucciones y datos
  - ▶ Hay que reubicar los datos en tiempo de ejecución para evitar conflictos
  - ▶ Hay que evitar que un programa acceda a los datos de otros programas
- ▶ ¿Qué ocurre si el tamaño de los datos excede la capacidad de la memoria principal?
  - ▶ Añadir otro nivel a la jerarquía de memoria: el disco duro
  - ▶ Hay que mover datos entre memoria principal y el disco duro

## Solución: Memoria virtual

- ▶ Permite que la memoria del computador sea compartida por múltiples programas
  - ▶ Reubicación de datos mediante un sistema de traducción de direcciones
  - ▶ Mecanismos de protección y compartición de datos entre programas

## Solución: Memoria virtual

- ▶ Permite que la memoria del computador sea compartida por múltiples programas
  - ▶ Reubicación de datos mediante un sistema de traducción de direcciones
  - ▶ Mecanismos de protección y compartición de datos entre programas
- ▶ Permite exceder la capacidad de la memoria principal
  - ▶ Utilizando el disco duro como un nivel más de la jerarquía de memoria

## Solución: Memoria virtual

- ▶ Permite que la memoria del computador sea compartida por múltiples programas
  - ▶ Reubicación de datos mediante un sistema de traducción de direcciones
  - ▶ Mecanismos de protección y compartición de datos entre programas
- ▶ Permite exceder la capacidad de la memoria principal
  - ▶ Utilizando el disco duro como un nivel más de la jerarquía de memoria
- ▶ Implementado por el sistema operativo, con soporte hardware para memoria virtual
  - ▶ Transparente para el programador

## Direcciones lógicas vs direcciones físicas

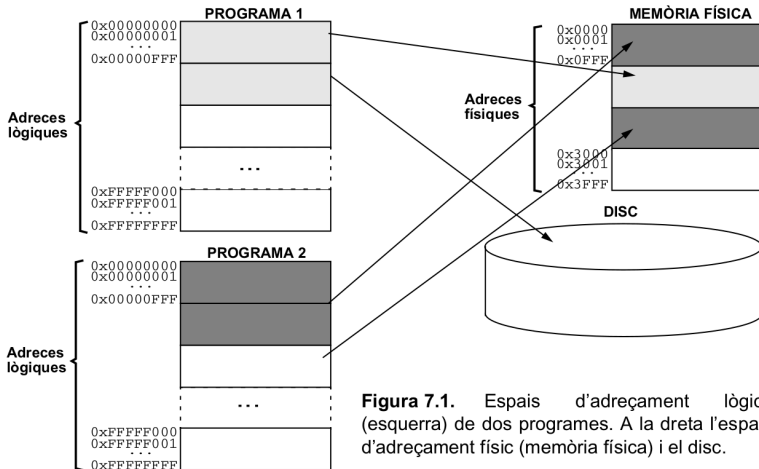
- ▶ Direcciones lógicas (virtuales)
  - ▶ Los programas utilizan un espacio de direcciones lógico o virtual
  - ▶ Exclusivo de cada programa
  - ▶ Tamaño máximo limitado por el número de bits de la dirección
  - ▶ Proporciona aislamiento entre programas



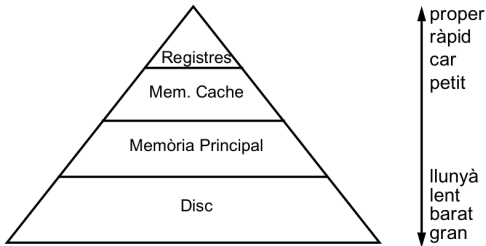
## Direcciones lógicas vs direcciones físicas

- ▶ Direcciones lógicas (virtuales)
  - ▶ Los programas utilizan un espacio de direcciones lógico o virtual
  - ▶ Exclusivo de cada programa
  - ▶ Tamaño máximo limitado por el número de bits de la dirección
  - ▶ Proporciona aislamiento entre programas
- ▶ Direcciones físicas
  - ▶ Direcciones reales de la memoria física
  - ▶ Durante la ejecución de un programa, sus datos e instrucciones se almacenan en la memoria física
  - ▶ Existe un mecanismo para traducir direcciones lógicas a direcciones físicas

# Memoria virtual



**Figura 7.1.** Espais d'adreçament lògic (esquerra) de dos programes. A la dreta l'espai d'adreçament físic (memòria física) i el disc.



# Página de memoria

- ▶ Bloque de memoria contiguo y de tamaño fijo
- ▶ Página virtual
  - ▶ Cada uno de los bloques en los que se divide el espacio de direcciones lógico
  - ▶ Cada página tiene un número de página (**VPN**: Virtual Page Number)
  - ▶ Para páginas de 4 KB ( $2^{12}$  bytes):

	VPN	page offset
@ = 0x10010004 =	0001 0000 0000 0001 0000	0000 0000 0100

# Página de memoria

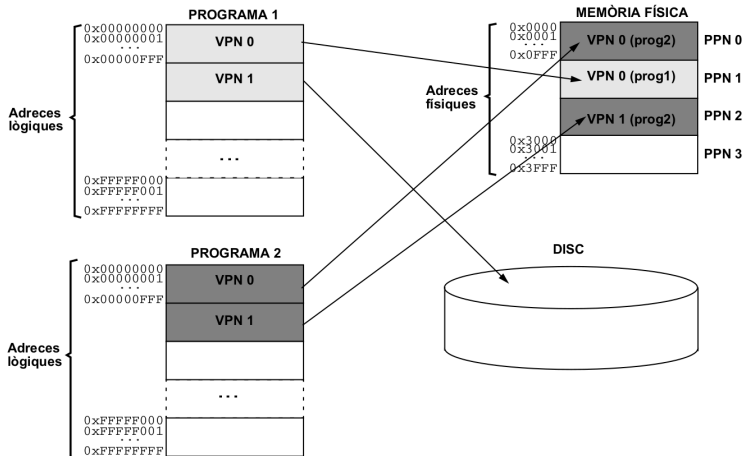
- ▶ La memoria física se divide en marcos de página (Page Frames)
  - ▶ Mismo tamaño que la página virtual
  - ▶ Un marco de página es un contenedor para una página virtual
  - ▶ Cada marco de página tiene un número de página física (**PPN**: Physical Page Number)

## Página de memoria

- ▶ La memoria física se divide en marcos de página (Page Frames)
  - ▶ Mismo tamaño que la página virtual
  - ▶ Un marco de página es un contenedor para una página virtual
  - ▶ Cada marco de página tiene un número de página física (**PPN**: Physical Page Number)
- ▶ Durante la ejecución de un programa:
  - ▶ Sus páginas virtuales (datos, instrucciones) se van almacenando en memoria física en los marcos de página a medida que se necesiten
  - ▶ Si no quedan marcos de página libres, el sistema operativo reemplaza uno ocupado

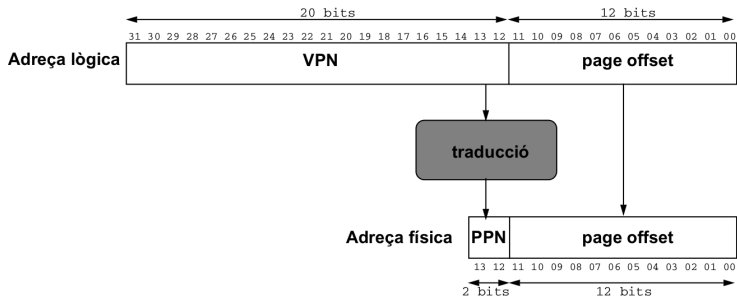
## Ejemplo

- Direcciones de 32 bits, páginas de 4 KB, memoria de 16 KB



## Traducción de direcciones

- ▶ El procesador trabaja con direcciones lógicas
- ▶ Para cada load/store, se debe traducir la dirección lógica a una dirección física
- ▶ MMU (Memory Management Unit):
  - ▶ Traduce el VPN al correspondiente PPN
  - ▶ El offset dentro de la página no cambia

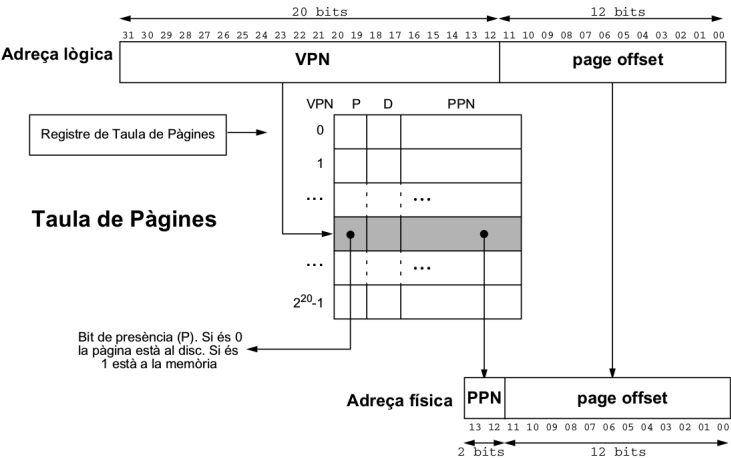




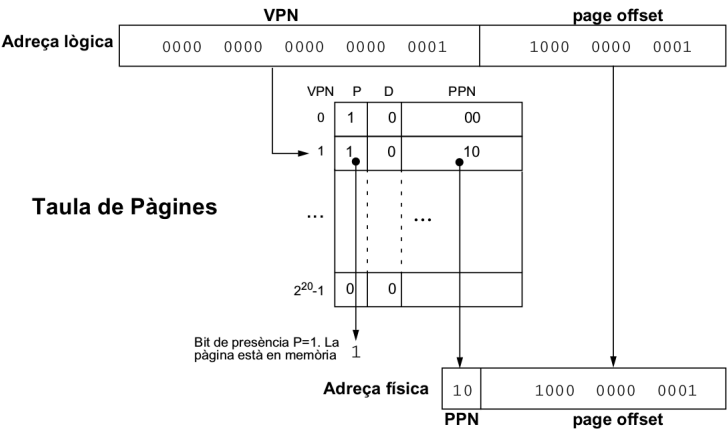
## Tabla de páginas

- ▶ Para traducir una dirección lógica a física, es necesario saber en qué marco de página (PPN) se almacena una página lógica (VPN)
- ▶ El sistema operativo utiliza emplazamiento completamente asociativo
- ▶ Cada programa tiene una tabla de páginas que almacena la correspondencia entre VPN y PPN para cada página lógica
  - ▶ Entrada de la tabla de páginas (**PTE**: Page Table Entry)
  - ▶ Tantas entradas como páginas virtuales
  - ▶ Bit de presencia (P) y bit de modificado (D)
- ▶ Registro de tabla de páginas

# Traducció amb taula de pàgines



# Traducció amb taula de pàgines



## Fallo de página

- ▶ Se produce cuando la CPU solicita a la MMU un acceso a una página que no se encuentra en memoria principal
  - ▶ Bit de presencia (P) vale 0 en la tabla de páginas

## Fallo de página

- ▶ Se produce cuando la CPU solicita a la MMU un acceso a una página que no se encuentra en memoria principal
  - ▶ Bit de presencia (P) vale 0 en la tabla de páginas
- ▶ En caso de fallo de página, el sistema operativo:
  - ▶ Lee la página del disco duro
  - ▶ La escribe en memoria principal (emplazamiento completamente asociativo)
  - ▶ Actualiza la información de la tabla de páginas
  - ▶ Reejecuta la instrucción que causó el fallo de página

## Fallo de página

- ▶ Se produce cuando la CPU solicita a la MMU un acceso a una página que no se encuentra en memoria principal
  - ▶ Bit de presencia (P) vale 0 en la tabla de páginas
- ▶ En caso de fallo de página, el sistema operativo:
  - ▶ Lee la página del disco duro
  - ▶ La escribe en memoria principal (emplazamiento completamente asociativo)
  - ▶ Actualiza la información de la tabla de páginas
  - ▶ Reejecuta la instrucción que causó el fallo de página
- ▶ Política de escritura retardada con asignación
  - ▶ El acceso al disco es extremadamente lento

## Reemplazos de página

- ▶ Se producen cuando el sistema operativo necesita cargar una página en memoria y no queda ningún marco de página libre
- ▶ Se utiliza un algoritmo de reemplazo para elegir la página a reemplazar (LRU)
- ▶ Antes de reemplazar una página modificada (bit  $D=1$ ) hay que escribirla en el disco
- ▶ En Linux la zona del disco dedicada a almacenar las páginas se denomina espacio de intercambio (swap)

