

Administración de Memoria Principal

Explicación de práctica

Introducción a los Sistemas Operativos
Conceptos de Sistemas Operativos

Facultad de Informática Universidad Nacional de La Plata

2025



- La organización y administración de la *memoria RAM* es uno de los factores más importantes en el diseño de un SO
- Los programas y datos deben residir en ella para:
 - Poder ejecutar
 - Referenciarlos directamente



- La parte del SO que administra esta memoria se llama "*administrador de la memoria*":
 - Lleva un registro de las partes de la memoria que se están utilizando y de aquellas que no
 - Asigna espacio en memoria a los procesos cuando estos la necesitan
 - Libera espacio de memoria asignada a procesos que han terminado
- Se espera que el SO haga uso eficiente de esta memoria con el fin de alojar el mayor número de procesos → repercute en la *multiprogramación*



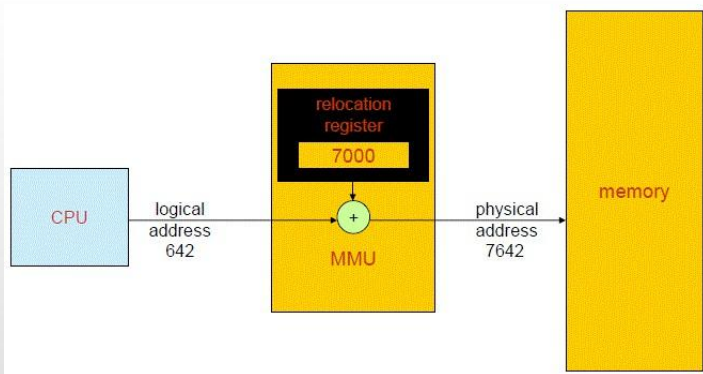
- **Dirección Lógica:**

- Es una dirección que enmascara o abstrae una dirección física
- Referencia a una localidad en memoria
- Se la debe traducir a una dirección física

- **Dirección Física:**

- Es la dirección real. Es con la que se accede efectivamente a memoria
- Representa la dirección absoluta en memoria principal
- La CPU trabaja con direcciones lógicas. Para acceder a la memoria se deben transformar en direcciones físicas
- El mapeo entre direcciones virtuales y físicas se realiza mediante *hardware* → **MMU** (Memory Management Unit)





- **Particiones Fijas:**

- La memoria se divide en particiones o regiones de tamaño fijo → tamaños iguales o diferentes
- Alojan un único proceso

- **Particiones Dinámicas:**

- Las particiones varían en tamaño y número
- Alojan un proceso cada una
- Cada partición se genera en forma dinámica del tamaño justo que necesita el proceso

- **Selección de partición:**

- Cada proceso se coloca en alguna partición de acuerdo a algún criterio:
 - **First Fit**
 - **Best Fit**
 - **Worst Fit**
 - **Next Fit**

¿Qué problemas se generan en cada caso?



- **Particiones Fijas:**

- La memoria se divide en particiones o regiones de tamaño fijo → tamaños iguales o diferentes
- Alojan un único proceso

- **Particiones Dinámicas:**

- Las particiones varían en tamaño y número
- Alojan un proceso cada una
- Cada partición se genera en forma dinámica del tamaño justo que necesita el proceso

- **Selección de partición:**

- Cada proceso se coloca en alguna partición de acuerdo a algún criterio:
 - **First Fit**
 - **Best Fit**
 - **Worst Fit**
 - **Next Fit**

¿Qué problemas se generan en cada caso?



- La fragmentación se produce cuando una localidad de memoria no puede ser utilizada por no encontrarse en forma contigua
- **Fragmentación Interna:**
 - Se produce en el esquema de particiones fijas
 - Es interna a la localidad asignada
 - Es la porción de la localidad que queda sin utilizar
- **Fragmentación Externa:**
 - Se produce en el esquema de particiones dinámicas
 - Son huecos que van quedando en la memoria a medida que los procesos finalizan
 - Al no encontrarse en forma contigua puede darse el caso de que tengamos memoria libre para alojar un proceso, pero que no la podamos utilizar
 - Solución → *compactación* → muy costosa



- La memoria se divide en porciones de igual tamaño llamadas **marcos**
- El espacio de direcciones de los procesos se divide en porciones de igual tamaño denominadas **páginas**
- Tamaño *página* = tamaño *marco* = 512 bytes (generalmente)
- El SO mantiene una tabla de páginas para cada proceso, la cual contiene el *marco* donde se encuentra cada página
- La paginación bajo demanda es una técnica eficiente de manejar esta estrategia



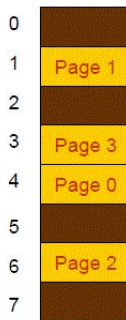
Paginación - Direccionamiento



Programa

0	4
1	1
2	6
3	3

Tabla de Páginas



Memoria



- Un proceso en ejecución hace referencia a una dirección virtual $\rightarrow v = (p, d)$
- El SO busca la página p en la *tabla de páginas* del proceso y determina en qué *marco* se encuentra
- La dirección de almacenamiento real se forma por la concatenación de la resolución de p (dirección inicio del marco que aloca la página) y d , donde p es el número de página y d es el desplazamiento



- Memoria administrada por sistema de paginación
- Tamaño de página \rightarrow 512 Bytes
- Cada dirección de memoria referencia a 1 Byte
- Los *marco* en memoria principal se encuentran desde la dirección física 0
- Tenemos un proceso con un tamaño de 2000 Bytes y con la siguiente tabla de páginas



Página	Marco
0	1
1	2
2	3
3	0



Marco	Inicio-Fin
0	0 - 511
1	512 - 1023
2	1024 - 1535
3	1536 - 2047

F.I. de 48 B.

- Si tenemos una dirección **virtual**, por ejemplo 580:
 - Para averiguar el número de página hacemos $580 \div 512 = 1$. Luego esta dirección corresponde a la página 1 que se encuentra en el *marco* 2
 - Para averiguar el desplazamiento hacemos $580 \bmod 512 = 68$
 - La dirección física es $1024 + 68 = 1092$



Página	Marco
0	1
1	2
2	3
3	0

Marco	Inicio-Fin
0	0 - 511
1	512 - 1023
2	1024 - 1535
3	1536 - 2047

F.l. de 48 B.

- Si tenemos una dirección **física**, por ejemplo 1092:
 - Para averiguar el número de *marco* hacemos $1092 \div 512 = 2$.
En el *marco* número 2 tenemos la página número 1
 - Para averiguar el desplazamiento hacemos $1092 \bmod 512 = 68$
 - La dirección virtual es $512 + 68 = 580$



Segmentación

- La segmentación básicamente la podemos ver como una mejora de la paginación (*no hay F.I., sino externa*)
- Ahora la tabla de segmentos, además de tener la dirección de inicio del mismo, tiene la longitud o límite
- Las direcciones lógicas constan de dos partes → un número de segmento s y un desplazamiento d dentro del segmento
($0 \leq d < \text{límite}$)

