

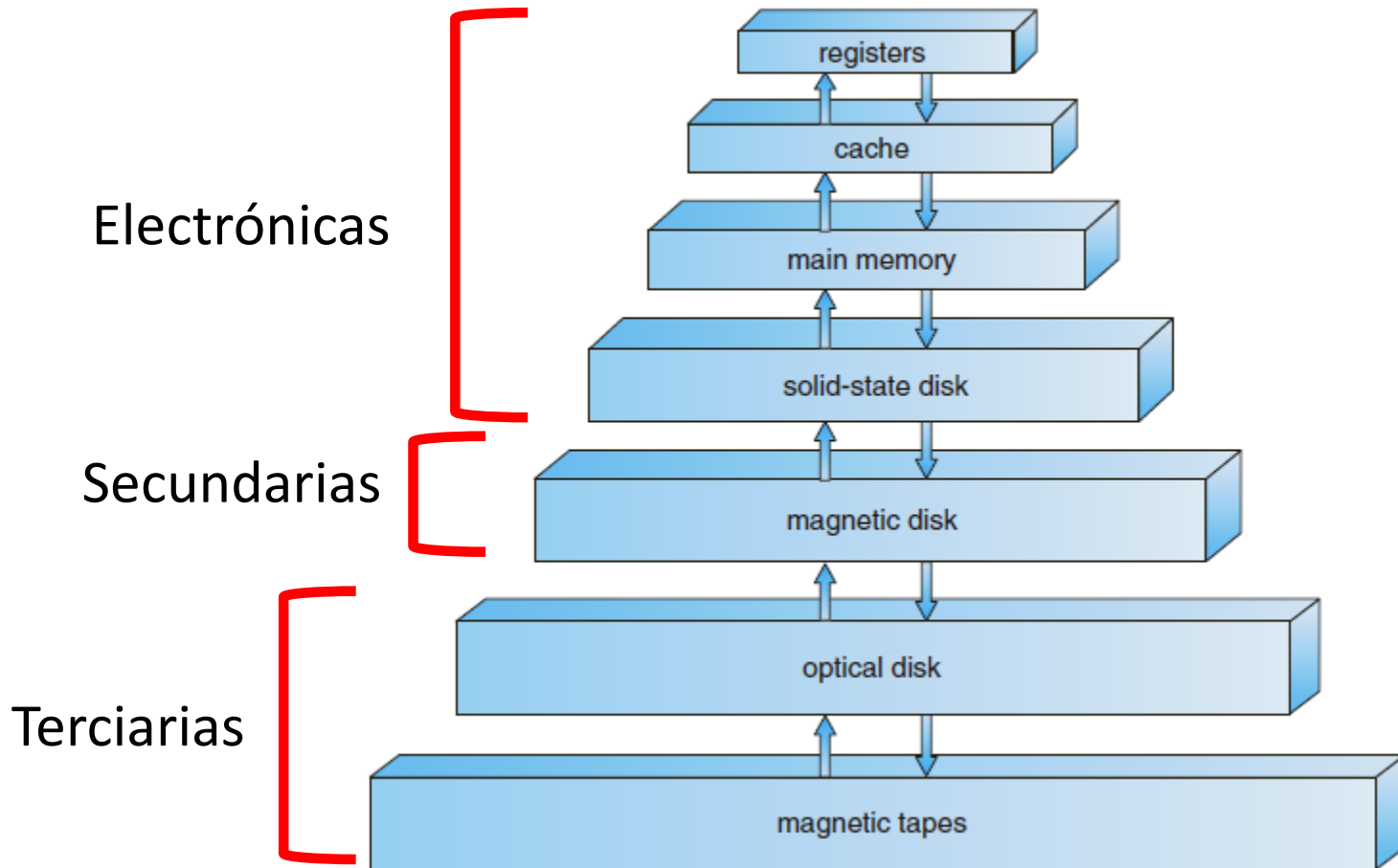
Sistemas Operativos

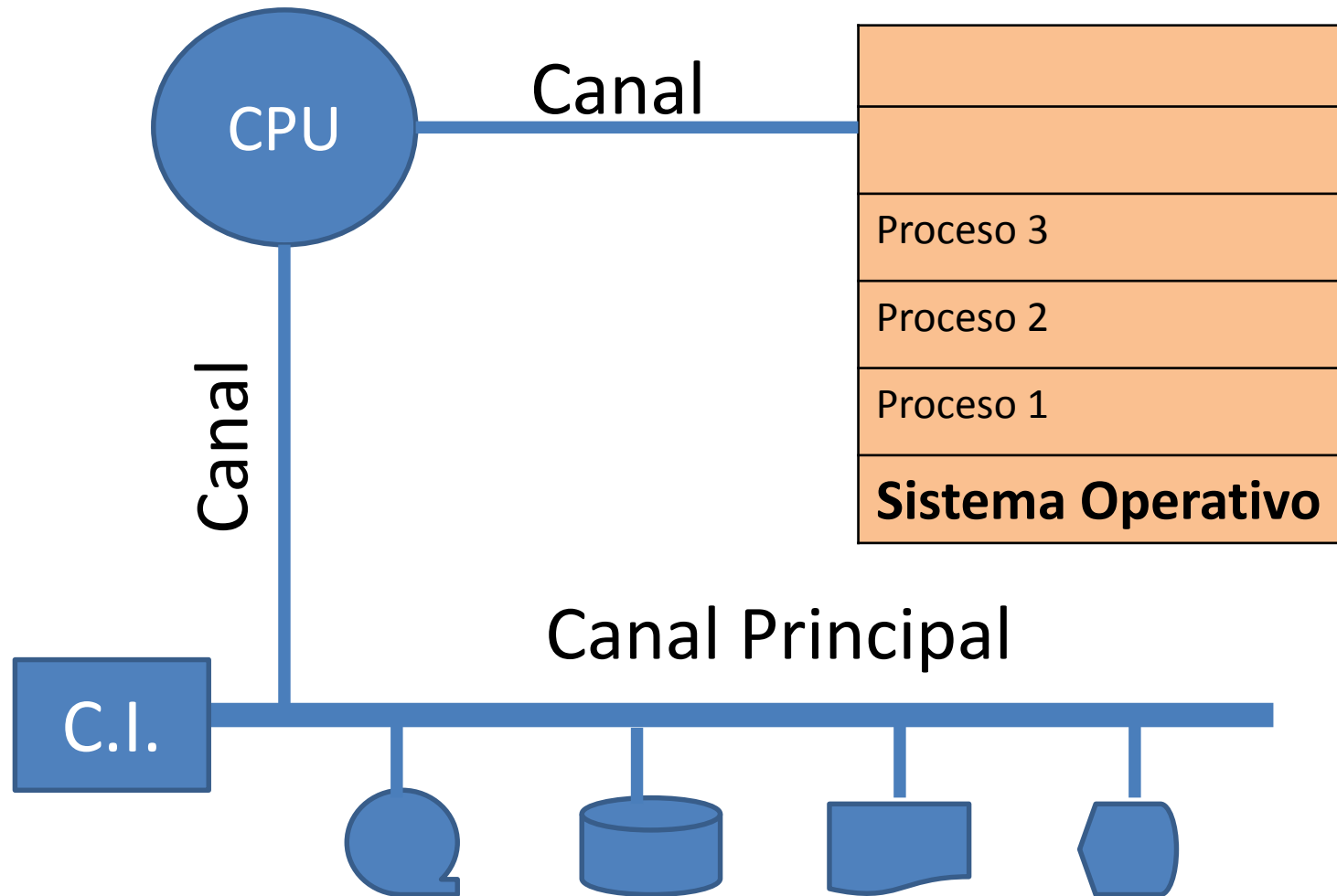
Cursada 2022

Comisión S21 y S22

Administración de Memoria

Jerarquía de Memorias (en general)





Memoria

Vamos a trabajar sobre la administración de la
Memoria Principal

Definición:

Denominamos Memoria Principal al espacio (Memoria) en donde residen tanto el código a ejecutar como los datos del proceso en ejecución

Denominamos Memorias Periféricas, donde guardamos con mayor o menor grado de permanencia, programas, datos, etc.

Memoria

Una de las característica esenciales de la memoria principal, es que la podemos direccionar (es direccionable) a nivel de palabra de memoria, hoy en día llamado **byte**, también llamado **octeto**

Podemos hacer una diferencia en lo que seria :

Byte lógico (8bits)

Byte Físico (mas de 8)

Cada lugar donde guardamos un byte tiene una dirección

La velocidad de acceso a cada dirección es la misma

Cada acceso a la RAM es univoca

Memoria

Quien dice, hay que ir a tal dirección de la memoria?

Cada dirección de memoria esta definida en hexa?

Esa dirección hace referencia a un byte?

Que longitud tiene un dirección?

La longitud define la capacidad de direccionamiento?

Las primeras computadoras trabajaban con registros

Los programadores escribían en absoluto las direcc en hexa

Luego cambio al tener memoria y lenguaje ensamblador

Por ejemplo los programadores definen variables (simbolos)

Aparecen los compiladores y empezamos hablar de


Dirección Lógica y Dirección Física

Memoria

Dirección simbólica

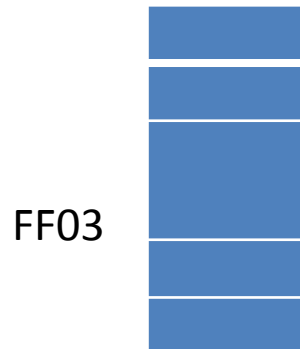
```
#include <stdlib.h>

int main(){
    int i,j,n,cont;
    cont = i+j
    printf("Contador: %d\n",cont);
}
```



Quien pasa de Símbolo a Física ??

Dirección Física



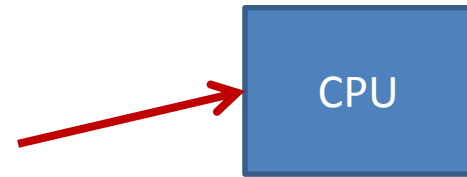
Memoria

Dirección simbólica

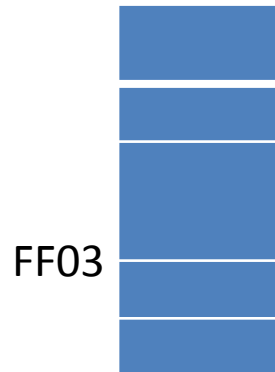
```
#include <stdlib.h>

int main(){
    int i,j,n,cont;
    cont = i+j
    printf("Contador: %d\n",cont);
}
```

Aparece otro tipo llamado **Dirección lógica**



Dirección Física



Memoria

Dirección simbólica

(La escribe el programador en su programa)

Dirección Lógica

Dirección Física



La administración de memoria pasa en su gran parte de como pasar de dirección *lógica a física*

Memoria

Como pasamos de dirección simbólica a lógica:

Dirección simbólica

- Se encargara el Compilador / ensamblador / Interprete

Dirección Lógica

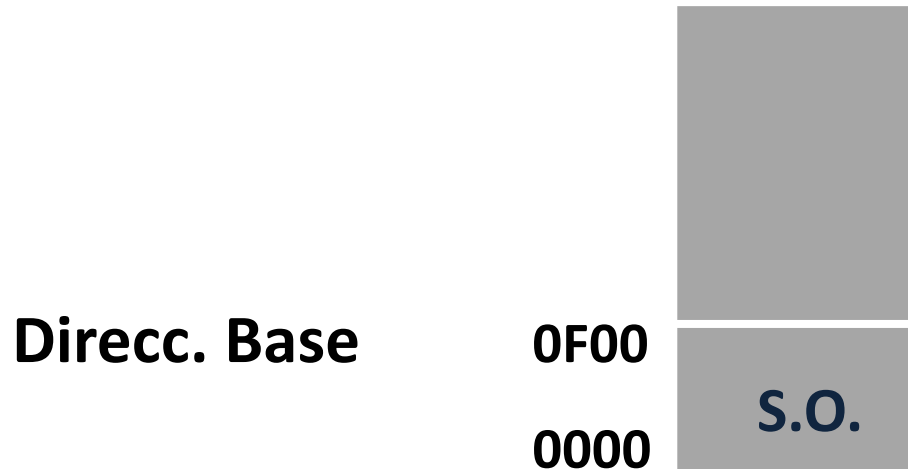
Tengo la Dirección Lógica, como paso a la Física?

- Se puede hacer en 3 momentos 
 - Compilación
 - Carga
 - Ejecucion

Dirección Física

Memoria

➤ **Compilador** (Ya no se hace, salvo en microcontrolador)



Por lo tanto en tiempo de compilación la dirección lógica y física son?

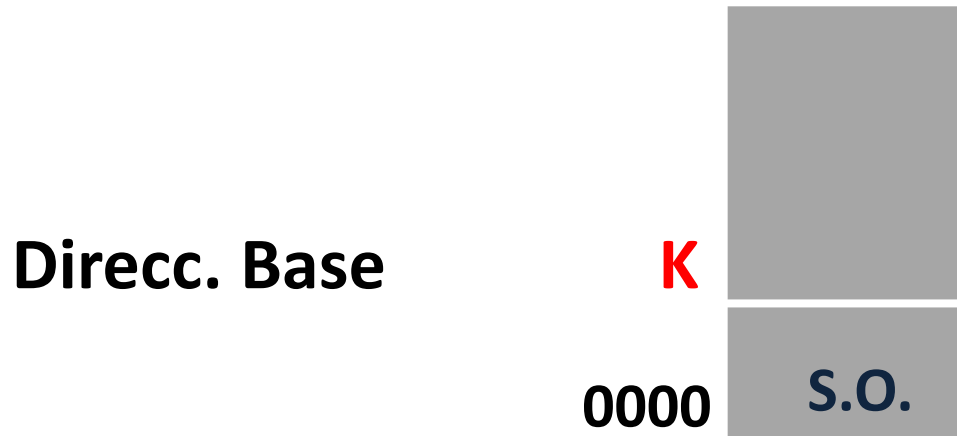
Iguales

Que dificultad puede traer??

Memoria

➤ Carga (Solución al problema anterior)

- Transformación de D.L. a D.F. en tiempo de carga
- Va arrancar el código absoluto de la direcc. 0000
- En el momento de carga lee la D.B. (Constante K)



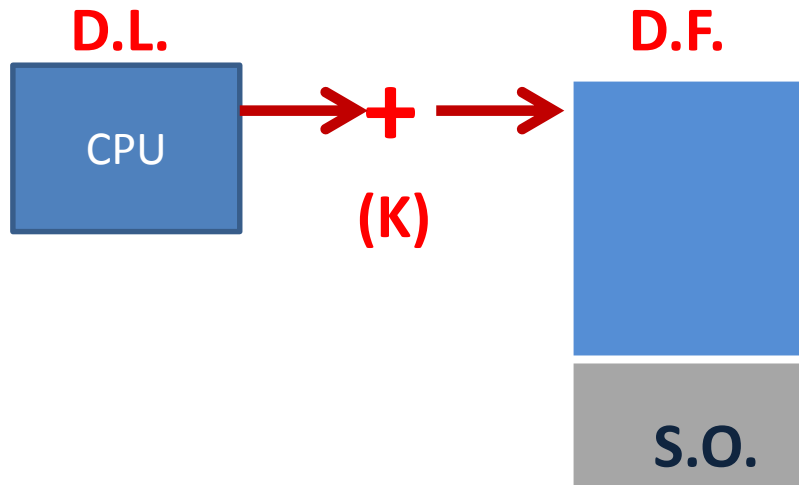
El cargador lee del disco y le va a sumar a la D.L. el valor de **K**
Por lo tanto a cada D.L. el cargador le suma **K**

$$\text{D.F.} = \text{D.L.} + K$$

Memoria

➤ Carga (Solución al problema anterior)

- Una vez que se carga van a estar todas las direcciones efectivas
- Toda la ejecucion sera sobre D.F.. Reales
- En el momento de carga lee la D.B. **(Constante K)**



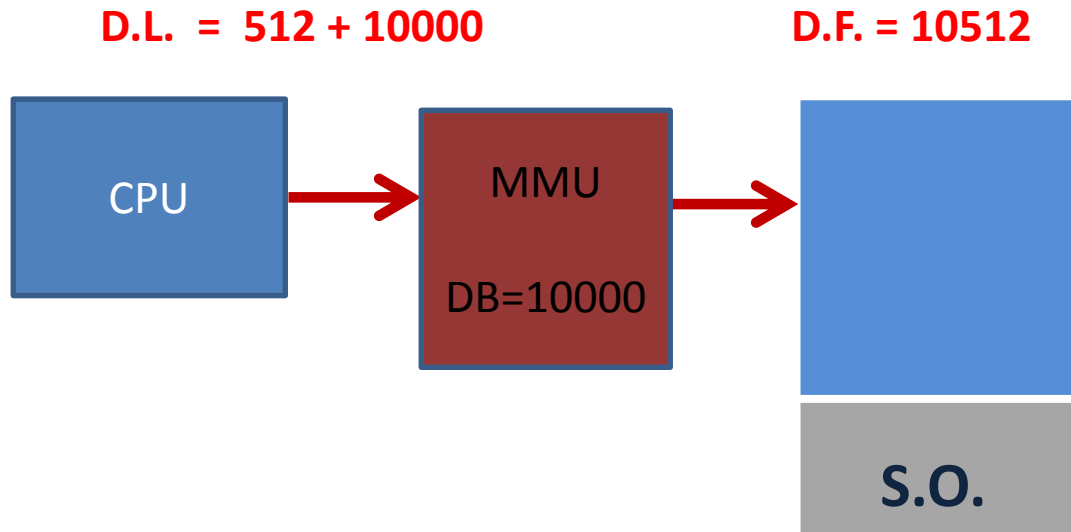
Esto se llama **Código Reubicable** (Ej. DOS)

Memoria

➤ Ejecucion

- Se le sigue sumando K a la D.L. ?
- Ahora en tiempo de Ejecucion
- Quien lo hace?

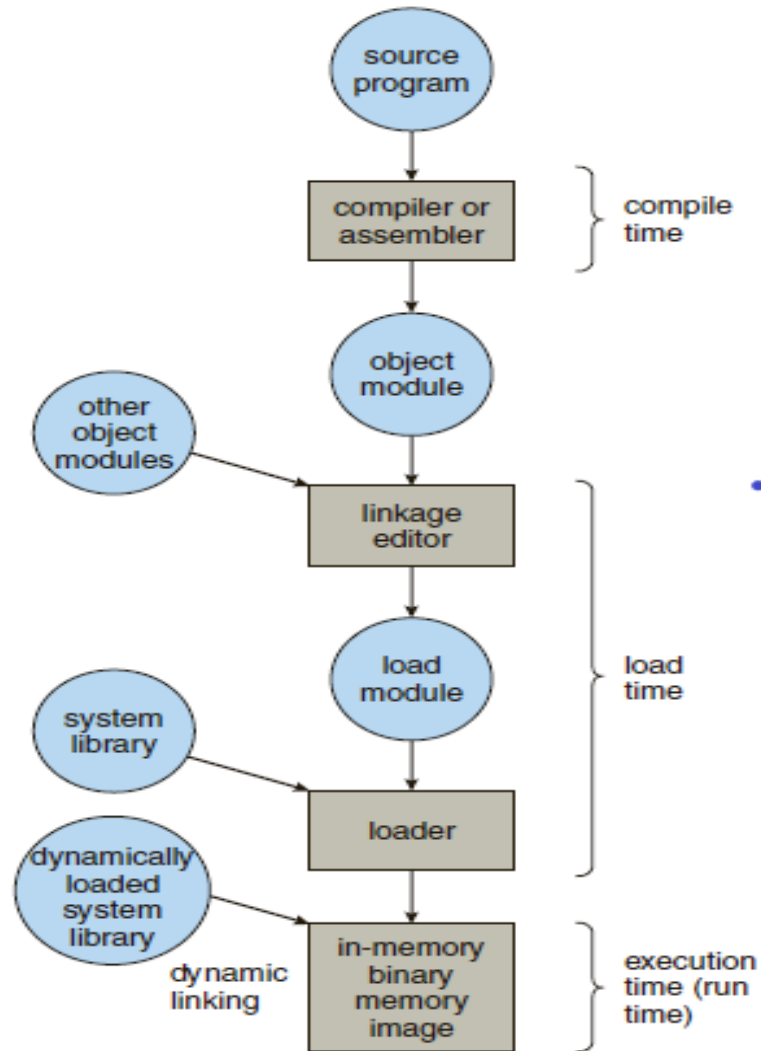
MMU (Management Memory Unit)



Explota la industrialización del Software

Memoria

Vamos hacer un repaso desde que se escribe un programa hasta su ejecucion



Memoria

Espacios de Memoria:

Lo definimos como un conjunto de direcciones de memoria.

Lo podemos dividir en dos conceptos

➤ **Espacio de Direccionamiento:** el conjunto de direcciones posibles

Ej. Si tengo 4bits para manejar direcciones, esto significa que puedo manejar hasta 2^n direcciones de memoria $2^4 = 16$

Entonces define la cantidad de bits que tiene la longitud de la palabra de direccionamientos

Memoria

➤ **Espacio de Direcciones:**

Son todas las direcciones que estoy usando, seria un subconjunto de todo el espacio de direccionamiento.

Por ejemplo

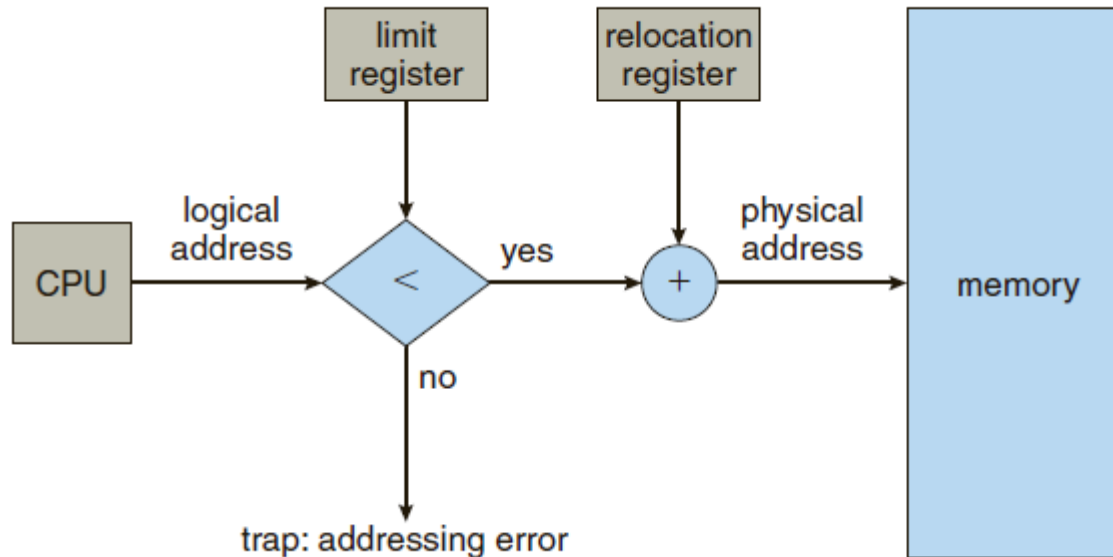
- **El compilador va a generar un espacio de direccionamiento**
- **Que pasa cuando incorporamos librerías?**
- **Esas librerías fueron compiladas**
- **Estas generan otro espacio de direcciones**
- **Por lo tanto se pueden administrar varios espacios cada uno con su identificador**
- **Una librería cargada en memoria se puede compartir por varios procesos**
- **El compilador debe informar el espacio de direcciones que necesita**
- **El cargador lee ese dato, y sabe si tiene espacio disponible para su ejecucion**

Memoria

Asignación Contigua de Memoria

La memoria se divide en lo que es el S.O. y los procesos de usuario.

Debemos proteger la memoria



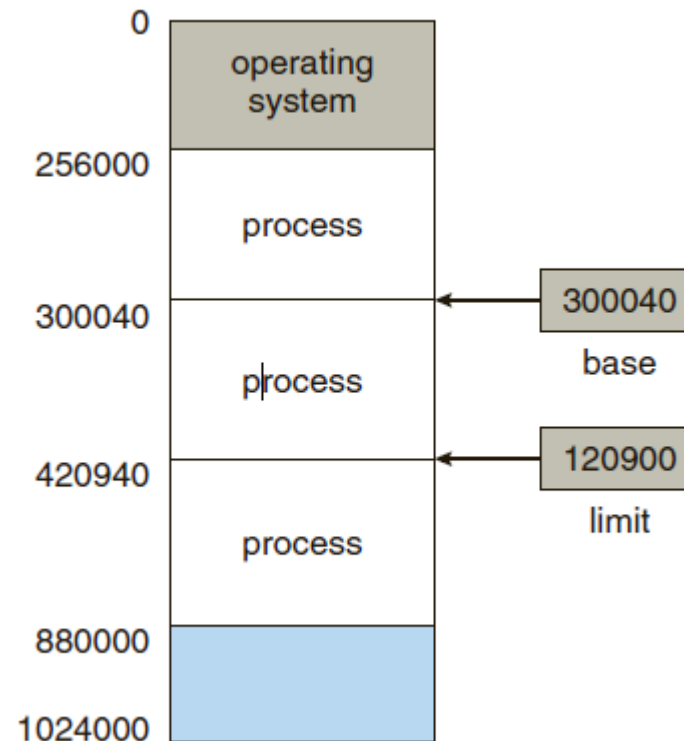
Memoria

Asignación Contigua de Memoria

En este tipo de asignación se producen huecos

Hay algunas técnicas para la asignación del espacio

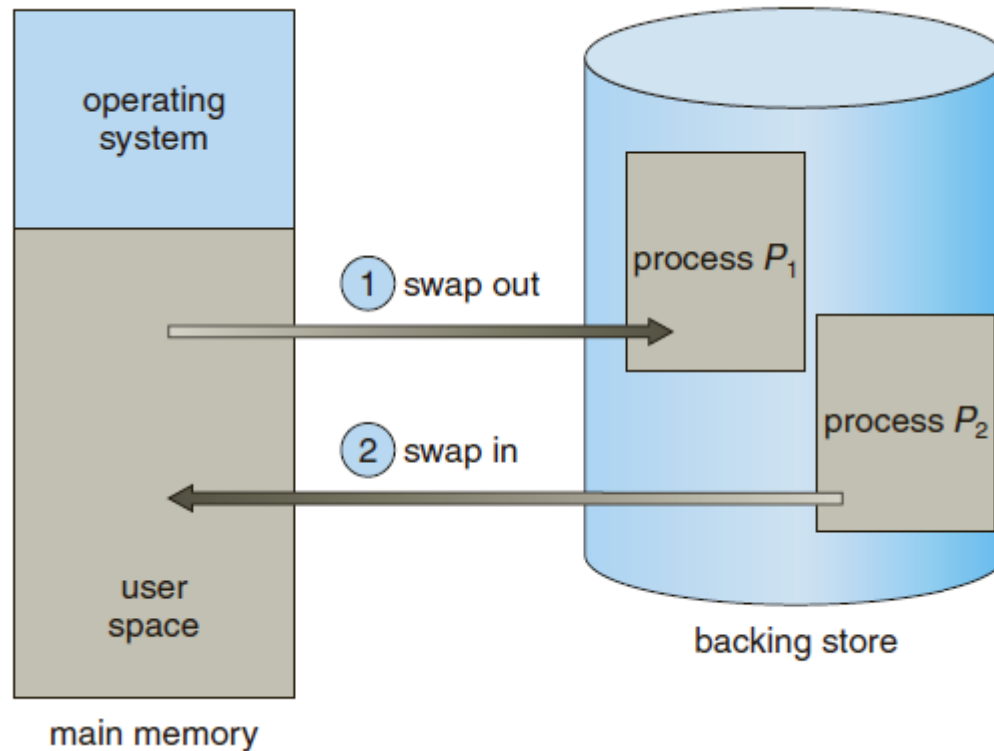
- **Primer Ajuste:** Asigna el 1ero lo suficientemente grande
- **Mejor Ajuste:** El mas pequeño lo suficientemente grande
- **Peor Ajuste:** Asigna el hueco mas grande



Memoria

Intercambio (Swapping)

Los procesos pueden ser intercambiados temporalmente



Fin del Tema