

Gestión de la memoria

Equipo "Sistemas Operativos DISCA/DSIC"

Universidad Politécnica de Valencia



Objetivos

- 1.- Introducir los conceptos y los problemas básicos de la gestión de memoria:
 - Memoria lógica vs. memoria física
 - Asignación de memoria a procesos
- 2.- Estudiar las técnicas básicas de asignación dispersa:
 - Paginación.
 - Segmentación.
 - Combinación de técnicas
 - Segmentación Paginada
 - Paginación multinivel.
- 3.- Entender la gestión de memoria virtual: problemas a resolver, implementación y efectos sobre el rendimiento del sistema.

Índice y bibliografía

■ Índice

- 1. Conceptos básicos
- 2. Gestión de memoria
- 3. Asignación contigua
- 4. Asignación dispersa
- 5. Memoria virtual

■ Bibliografía

- **A. Silberschatz, P.B. Galvin**
- Sistemas Operativos. Conceptos Fundamentales. 5ª ed.
 - **Capítulo 8:** Apartados 8.1 (excepto 8.1.4), 8.2, 8.3, 8.4 (excepto 8.4.1), 8.5 (excepto 8.5.4, 8.5.5), 8.6 y 8.7 (excepto 8.7.1 y 8.7.2)
 - **Capítulo 9:** Apartados 9.1, 9.2, 9.3, 9.4, 9.5 (excepto 9.5.4.3), 9.6 (excepto 9.6.1) y 9.7 .

Contenidos



1.- Conceptos básicos

- Memoria física vs. Memoria lógica
- La Unidad de Manejo de Memoria (MMU)

2.- Gestión de memoria

3.- Asignación contigua

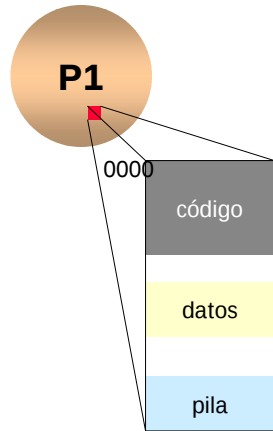
4.- Asignación dispersa

5.- Memoria virtual

Memoria física vs. memoria lógica

■ Memoria lógica de un proceso

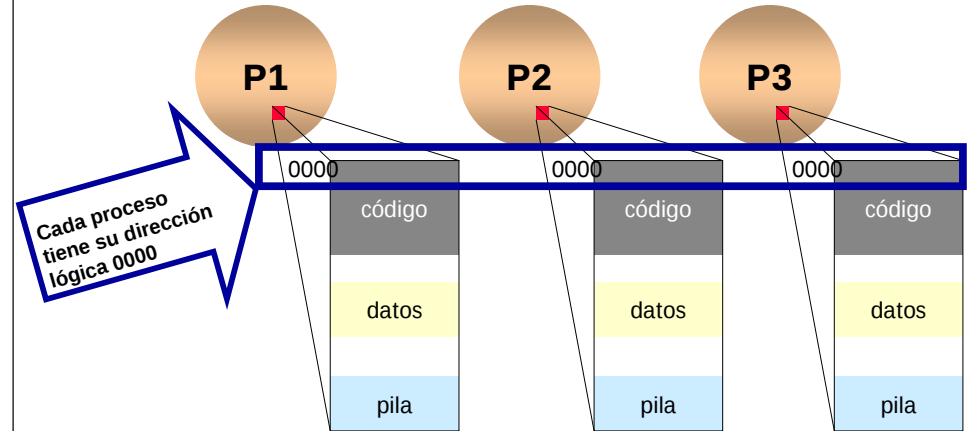
(8.2)



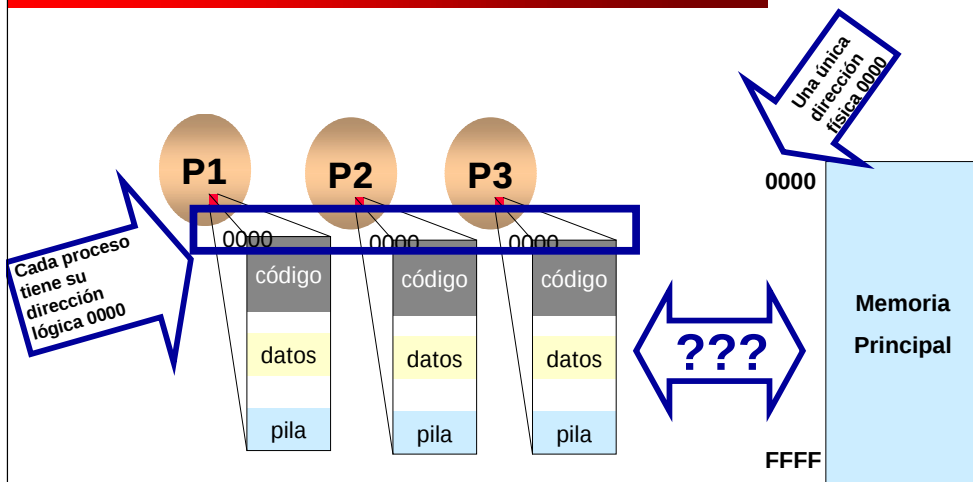
Memoria física vs. memoria lógica

■ Memoria lógica de un proceso

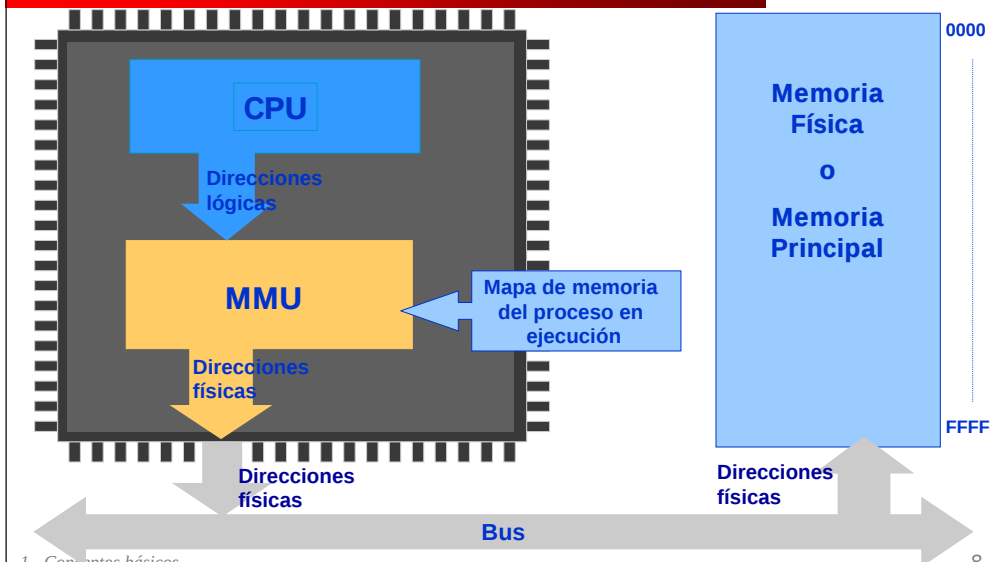
(8.2)



Memoria física vs. memoria lógica



La Unidad de Manejo de Memoria (MMU)



Contenidos

1.- Conceptos básicos



2.- Gestión de memoria

- El problema de gestión de la memoria
- El problema de la reubicación
- El problema de la escasez de memoria
- El problema de la asignación

3.- Asignación contigua

4.- Asignación dispersa

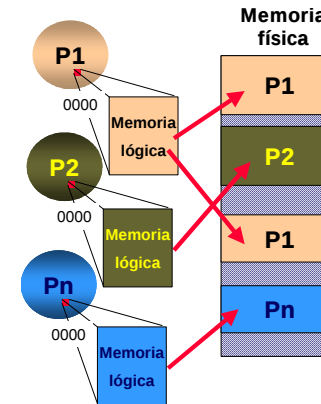
5.- Memoria virtual

9

El problema de gestión de la memoria

- Todo sistema operativo debe ofrecer alguna solución para estos problemas:

- Reubicación
- Escasez
- Protección
- Asignación



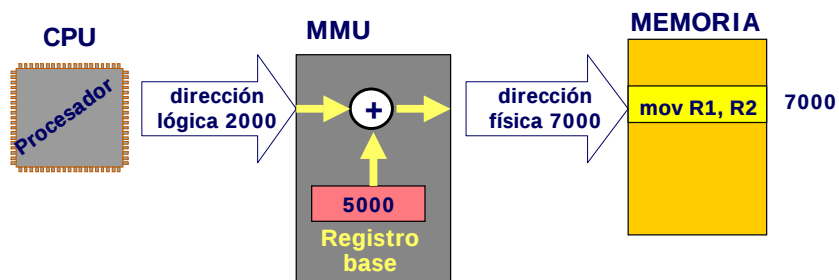
10

El problema de la reubicación

■ Reubicación en tiempo de ejecución: la MMU

(8.1.1)

- Programa reubicable es aquel que puede ejecutarse en cualquier segmento de memoria física.
- Para permitir la reubicación en tiempo de ejecución es necesario un Hardware específico → MMU
- Ejemplo: MMU con registro base.



11

El problema de escasez de la memoria

■ ¿Por qué?

(9.1)

- Los procesos tienen una necesidad mayor de memoria.
- Aumentar el grado de multiprogramación

■ Técnicas para solucionarlo:

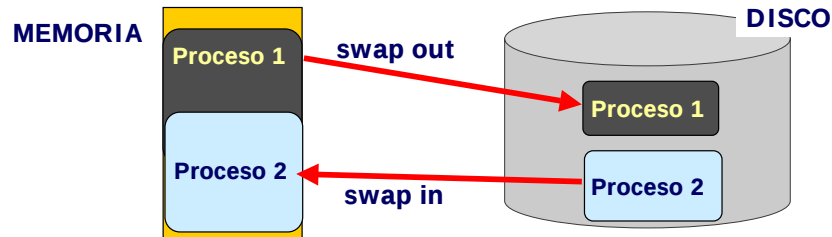
- Intercambios (Swapping)
- Memoria virtual
- Bibliotecas dinámicas

12

El problema de escasez de la memoria

El método de intercambios (*swapping*)

(8.3)

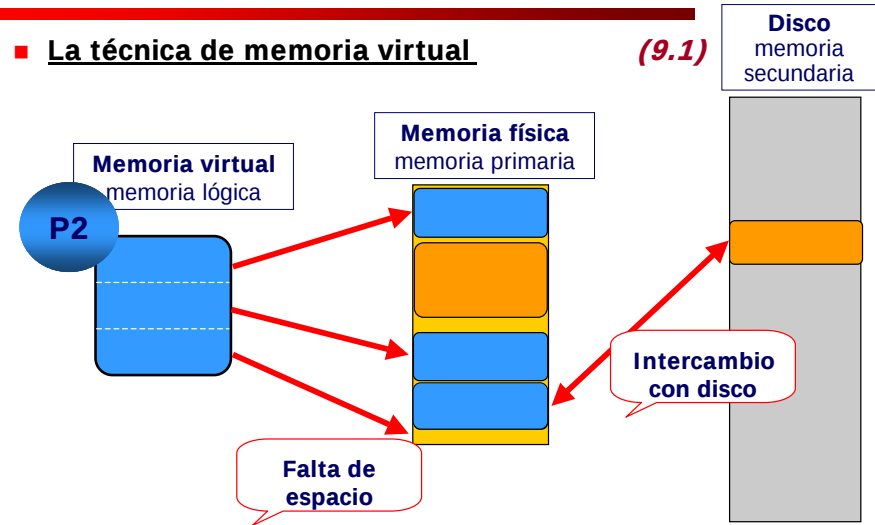


13

El problema de escasez de la memoria

La técnica de memoria virtual

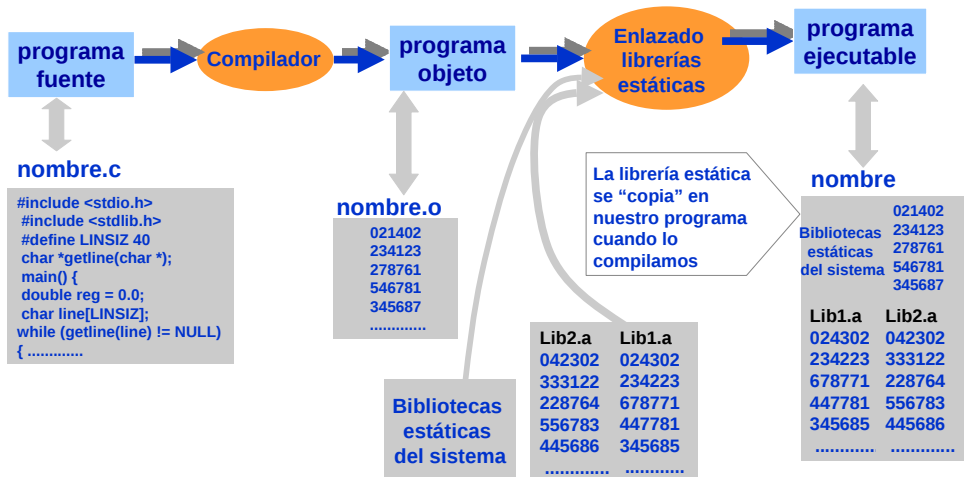
(9.1)



14

El problema de escasez de la memoria

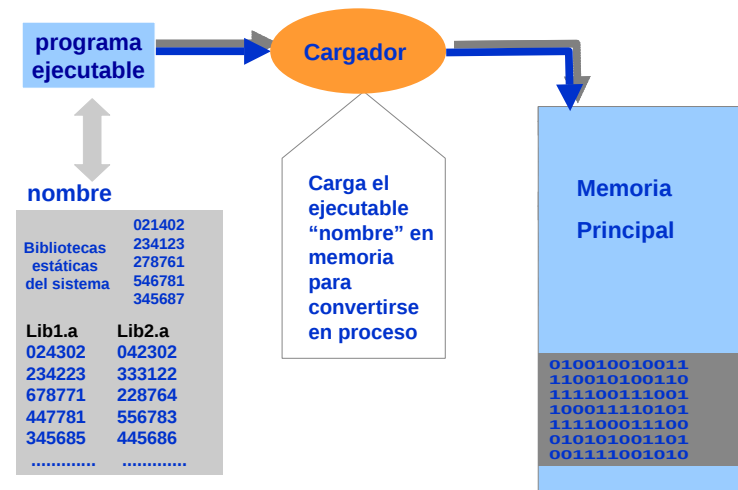
Enlace estático con bibliotecas



15

El problema de escasez de la memoria

Enlace estático con bibliotecas

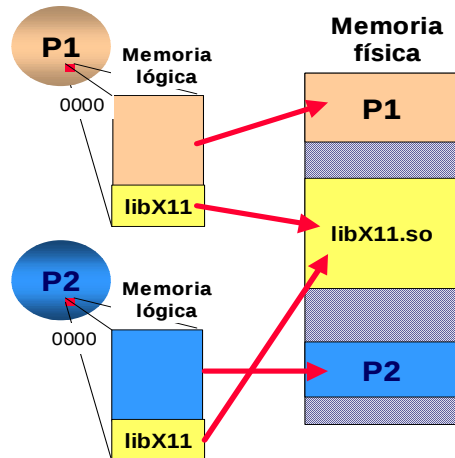


16

El problema de escasez de la memoria

■ Bibliotecas dinámicas

(8.1.3)

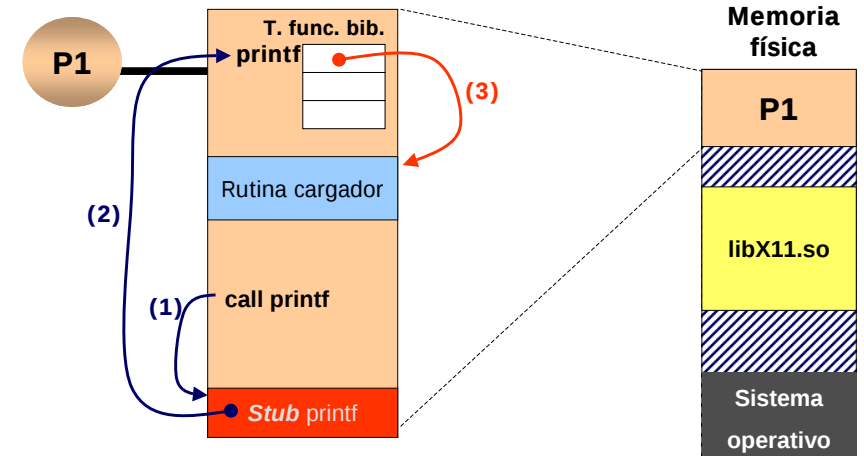


17

El problema de escasez de la memoria

■ Bibliotecas dinámicas

(8.1.3)

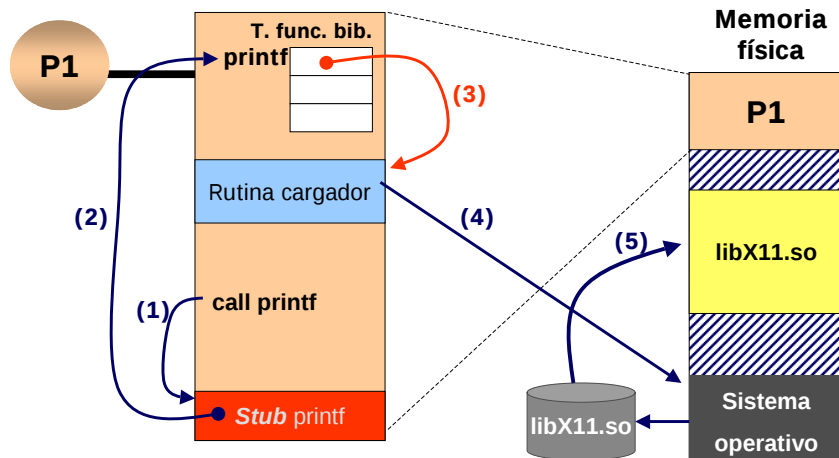


18

El problema de escasez de la memoria

■ Bibliotecas dinámicas

(8.1.3)

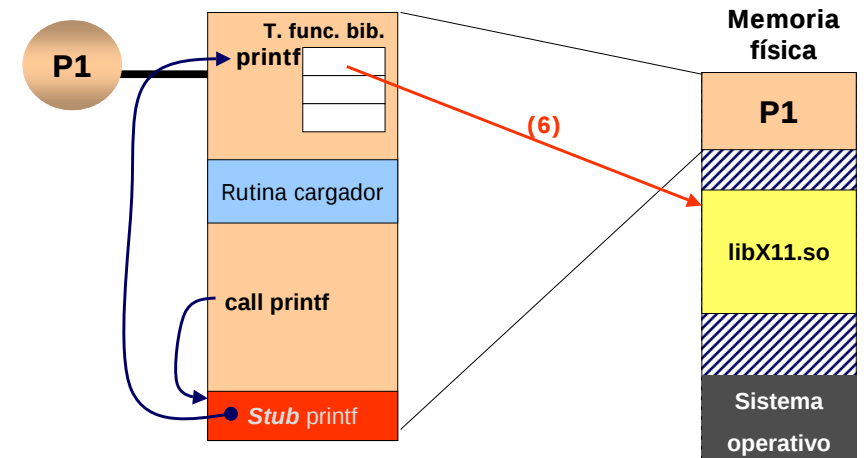


19

El problema de escasez de la memoria

■ Bibliotecas dinámicas

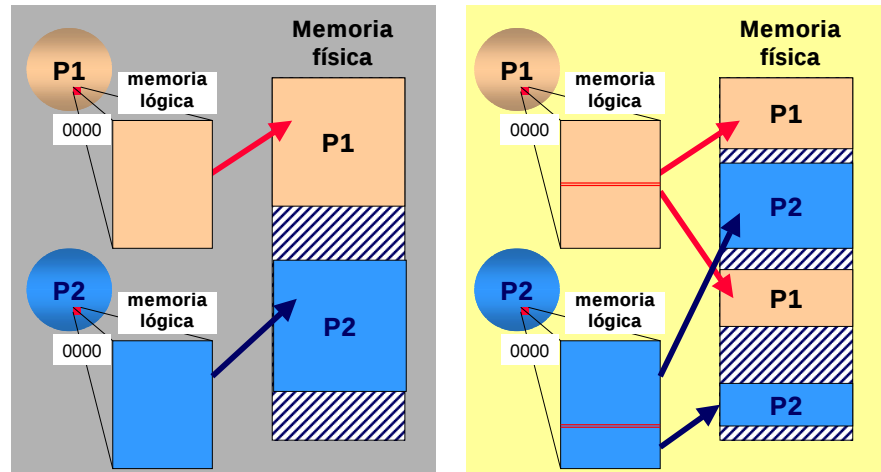
(8.1.3)



20

Asignación de la memoria

■ Asignación contigua y asignación dispersa



21

Asignación de la memoria

■ Métodos de asignación

- Asignación contigua (8.4) :
 - Particiones fijas (8.4.2).
 - Particiones variables (8.4.2)
- Asignación dispersa:
 - Paginación (8.5).
 - Segmentación (8.6).

22

Contenidos

1.- Conceptos básicos

2.- Gestión de memoria

3.- Asignación contigua

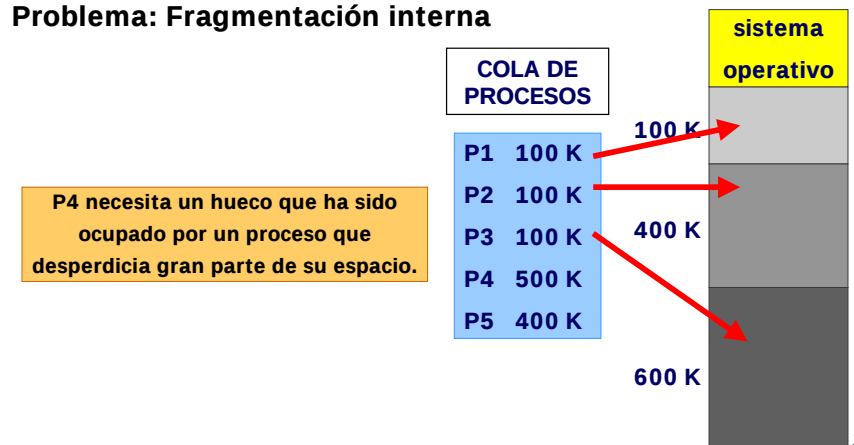
- Método de múltiples particiones fijas
- Método de múltiples particiones variables

4.- Asignación dispersa

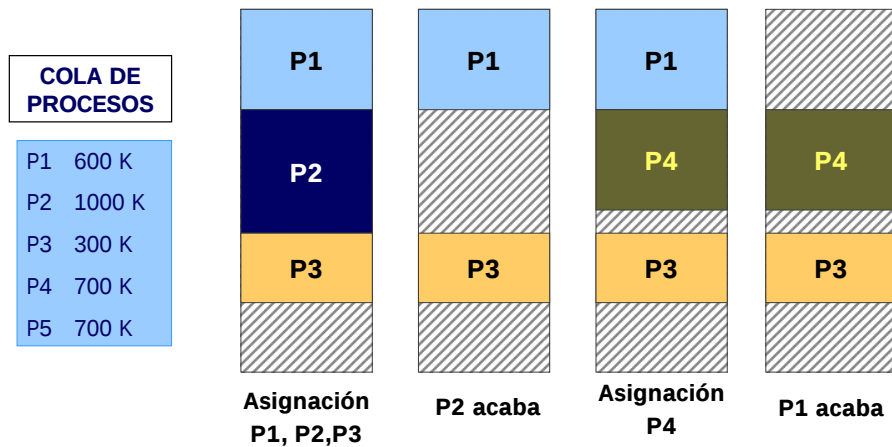
5.- Memoria virtual

Método de múltiples particiones fijas

- Memoria dividida en particiones de diferente tamaño
- Cola única / múltiples colas
- Problema: Fragmentación interna



Método de múltiples particiones variables



25

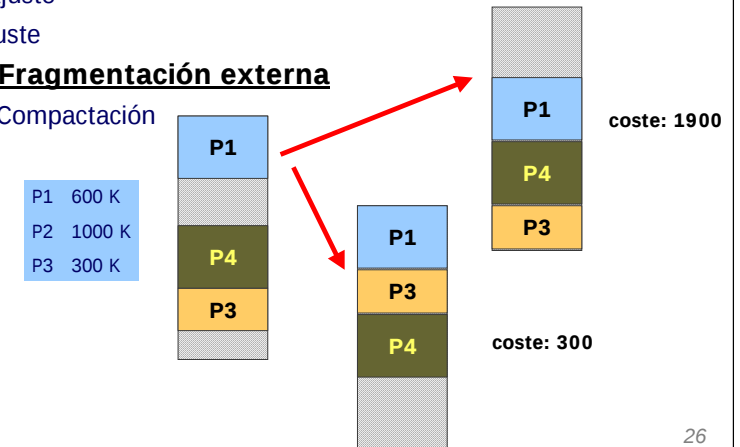
Método de múltiples particiones variables

Algoritmos de asignación de huecos

- El primer ajuste
- El mejor ajuste
- El peor ajuste

Problema: Fragmentación externa

- Solución: Compactación



26

Contenidos

- 1.- Conceptos básicos
- 2.- Gestión de memoria
- 3.- Asignación contigua



4.- Asignación dispersa

- **Paginación** (8.5)
- **Segmentación** (8.6)
- **Técnicas combinadas**
 - Segmentación paginada (8.7)
 - Paginación multinivel (8.5.3)

5.- Memoria virtual

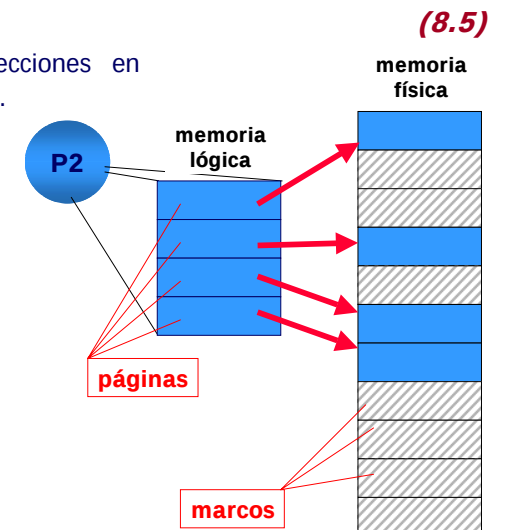
27

La paginación

Concepto básico

Dividir el espacio de direcciones en fragmentos de longitud FIJA.

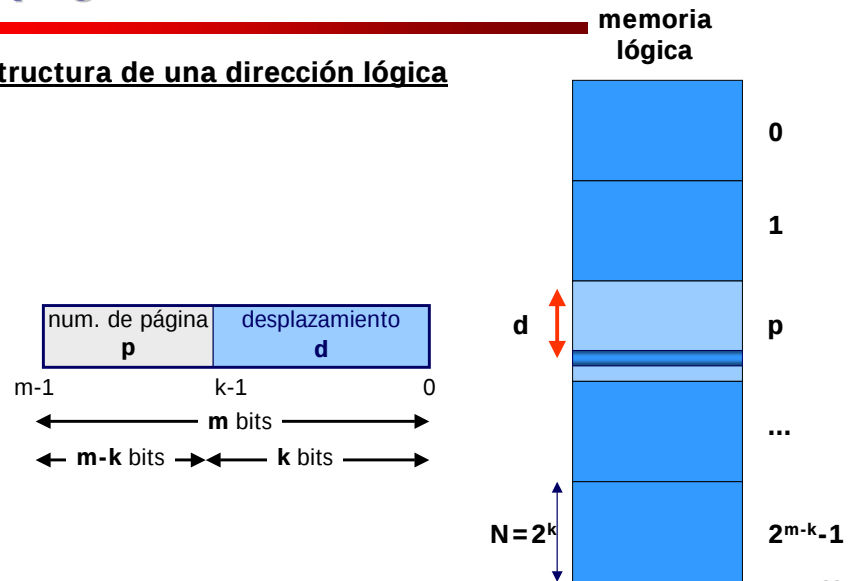
- Páginas.
- Marcos.



28

La paginación

Estructura de una dirección lógica



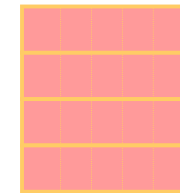
29

La paginación

La tabla de páginas

- Descriptor de página :
 - Número de marco físico
 - Bit de validez
 - Bits de protección (rwx)
 - Bit de modificado

Tabla de páginas



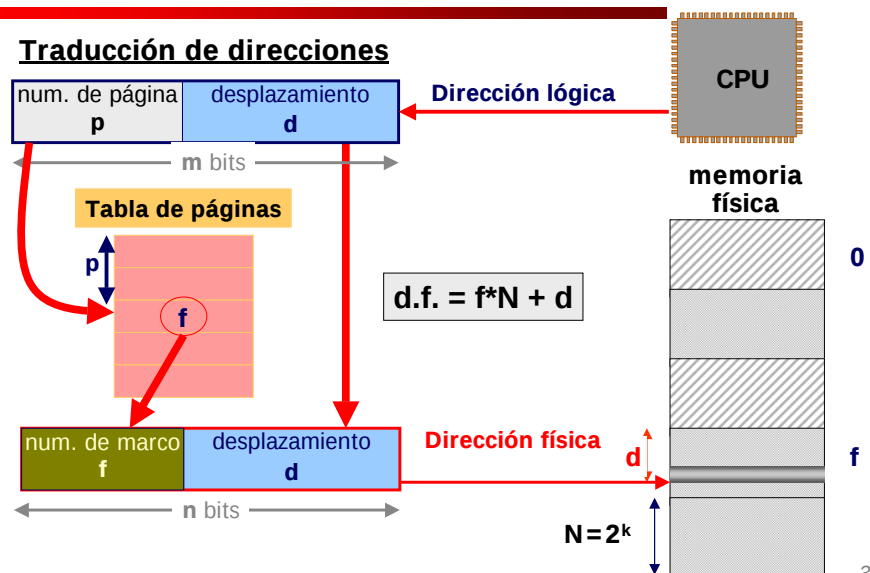
Descriptor de página

número de marco	bit validez	bits protección	bit modificado	otros
				...

30

La paginación

Traducción de direcciones



31

La paginación

Un ejemplo

- Espacio de direcciones físicas: 32 bytes
- Espacio de direcciones lógicas: 16 bytes
- Tamaño de página: 4 bytes

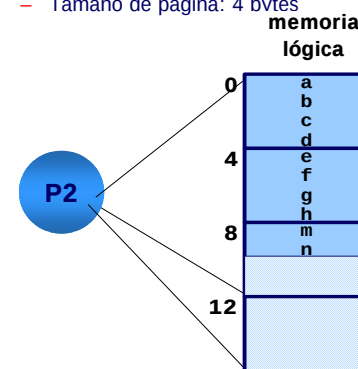
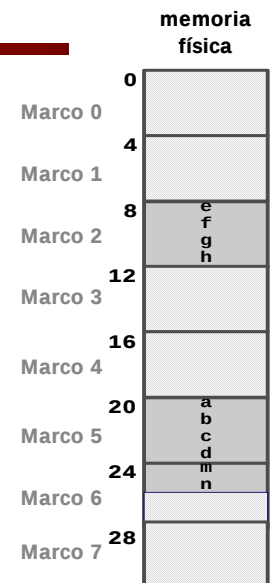


Tabla de páginas

5
2
6
-



32

La paginación

■ Protección

- Un ejemplo

(8.5.2.2)



33

La paginación

■ Análisis

- Ventajas
 - No aparece fragmentación externa.
 - Facilita la reubicación.
 - Proporciona protección.
- Inconvenientes
 - Fragmentación interna.
- Tamaños de página
 - Grandes → mucha fragmentación interna.
 - Pequeños → tablas de página muy grandes.

34

La paginación

■ Implementación de la tabla de páginas

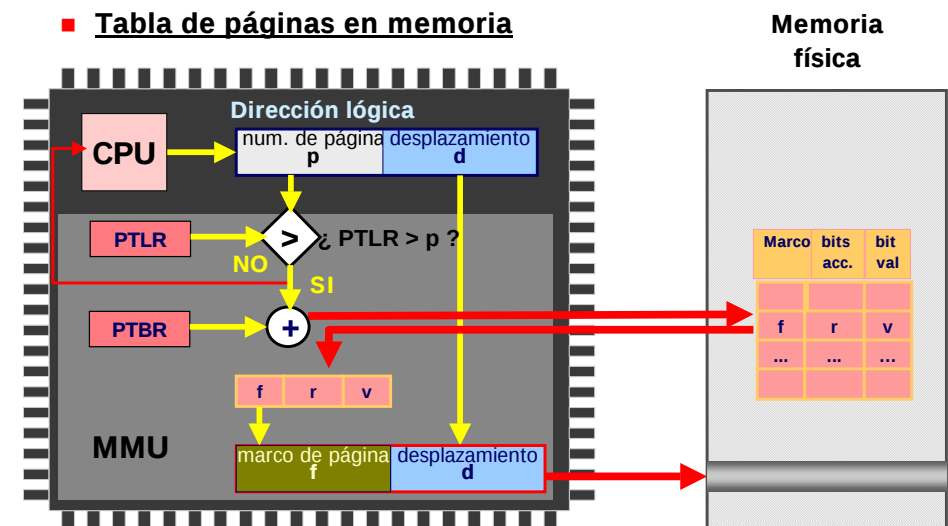
(8.5.2)

- Registros de la MMU
- Memoria
- TLB

35

La paginación

■ Tabla de páginas en memoria



36

La paginación

■ Tabla de páginas en registros asociativos: (8.5.2)

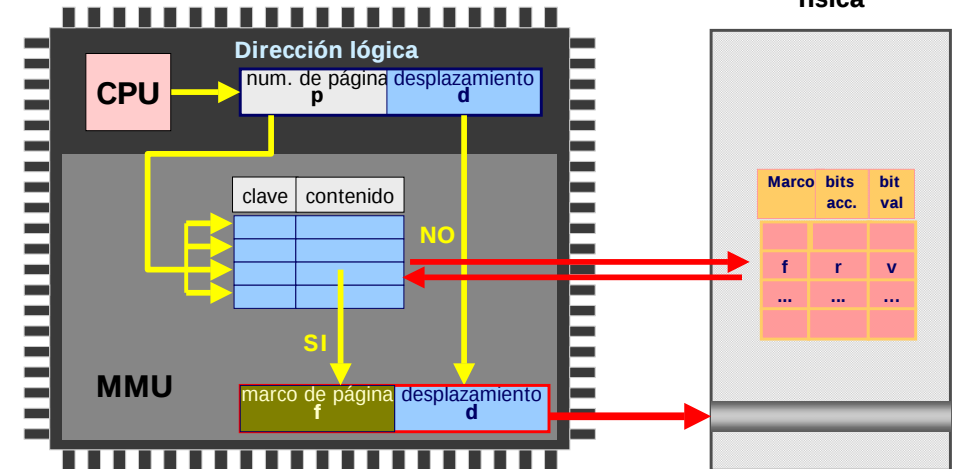
TLB: *Translation Look-aside Buffers*

- Los registros contienen:
 - clave.
 - contenido.
- La búsqueda se realiza por la clave en paralelo para todo el conjunto de registros asociativos

37

La paginación

■ Tabla de páginas con TLB



38

Contenidos

- 1.- Conceptos básicos
- 2.- Gestión de memoria
- 3.- Asignación contigua
- 4.- Asignación dispersa
 - **Paginación (8.5)**
 - **Segmentación (8.6)**
 - **Técnicas combinadas**
 - Segmentación paginada (8.7)
 - Paginación multinivel (8.5.3)
- 5.- Memoria virtual



39

La segmentación

■ Concepto de segmentación (8.6)

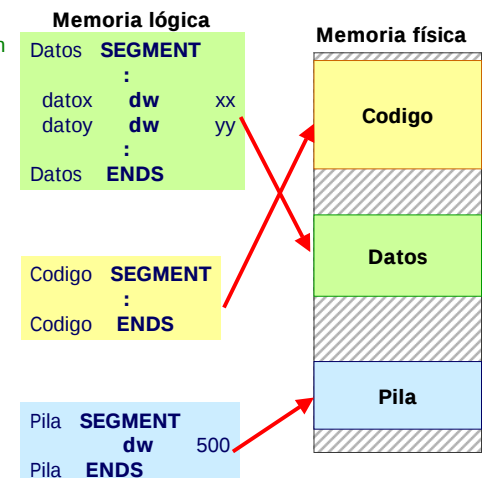
– Definición

Dividir el espacio de direcciones en fragmentos de longitud VARIABLE.

– Tipos de Segmento

- Código
- Datos
- Pila

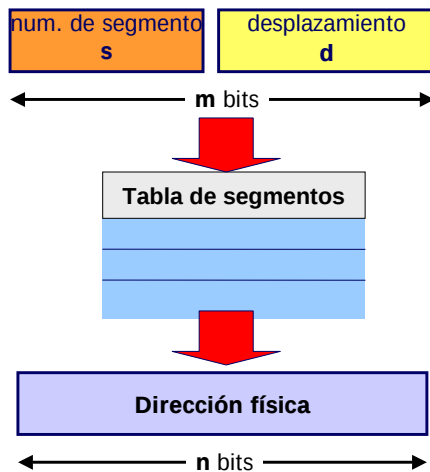
– Propiedades de los segmentos



40

La segmentación

■ Estructura de una dirección lógica



41

La segmentación

■ La tabla de segmentos

- Contiene un descriptor por cada segmento.
- Cada descriptor de segmento contiene, al menos:
 - Base del segmento:
 - Límite:
 - Otros, ...

Tabla de segmentos

base	límite	etc.

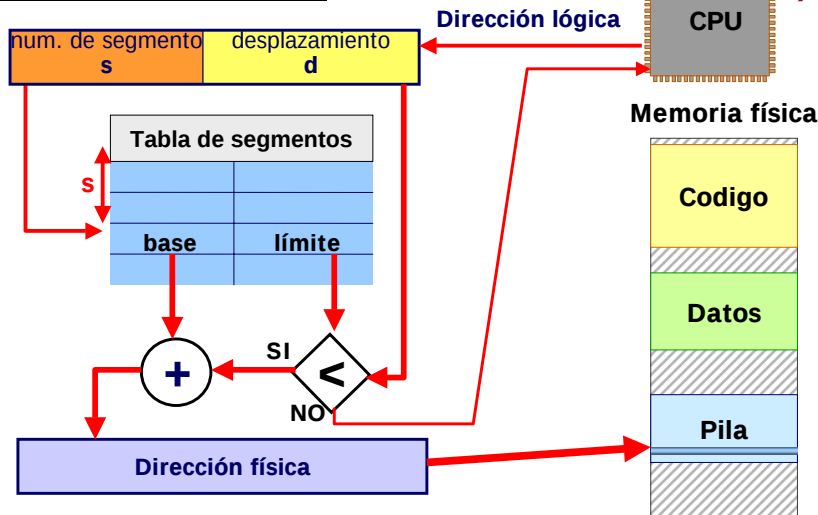
Descriptor de segmento



42

La segmentación

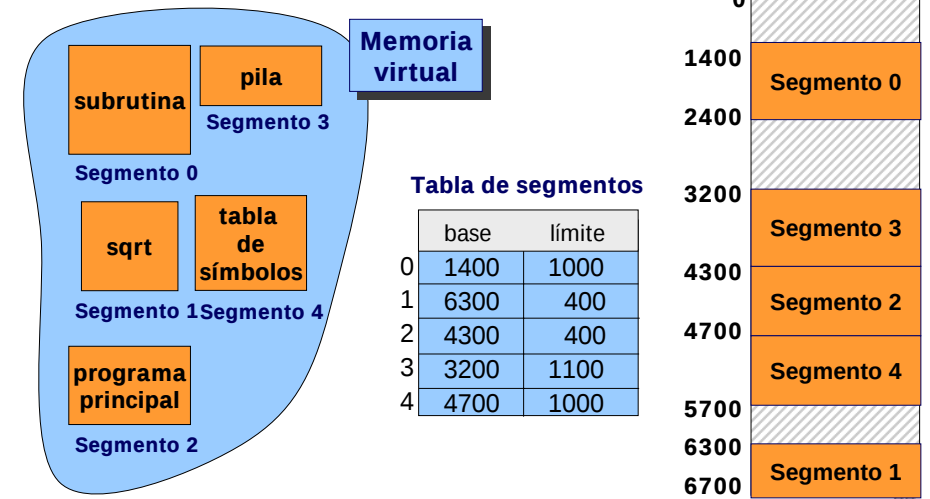
■ Traducción de direcciones



43

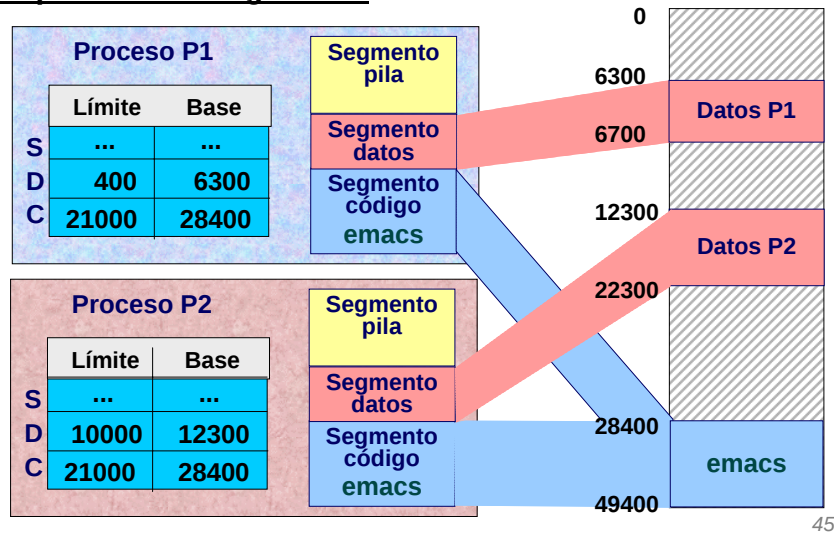
La segmentación

■ Ejemplo de segmentación



La segmentación

■ Compartición de segmentos

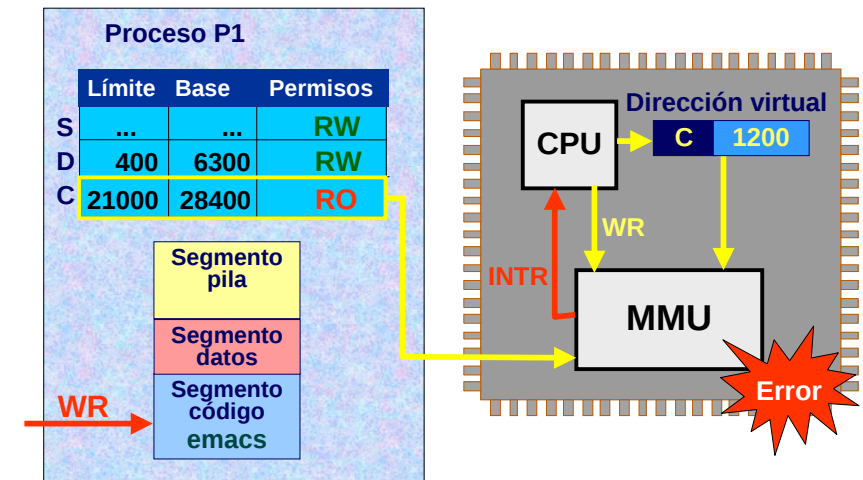


45

La segmentación

■ Facilidad para protección

(8.6.4)



46

La segmentación

■ Análisis

- Ventajas
 - No aparece fragmentación interna.
 - Facilita la reubicación.
 - Proporciona protección.
- Inconvenientes
 - Fragmentación externa.
- Tamaños de segmento
 - Muy Grandes → aproximación a particiones variables.
 - Muy Pequeños → eliminaría la fragmentación externa, pero aumentaría el tamaño destinado a registros
 - Tamaño Fijo → Paginación

47

La segmentación

■ Implementación de la tabla de segmentos

(8.6.3)

- Registros de la MMU
- Memoria
- TLB

48

Contenidos

- 1.- Conceptos básicos
- 2.- Gestión de memoria
- 3.- Asignación contigua
- 4.- Asignación dispersa

- **Paginación** (8.5)
- **Segmentación** (8.6)
- **Técnicas combinadas**



- Segmentación paginada (8.7)
- Paginación multinivel (8.5.3)

- 5.- Memoria virtual

La segmentación paginada

■ Motivación

Cuando los segmentos crecen:

- Aumenta la fragmentación externa.
- Aumenta el problema de encontrar un hueco libre en memoria para ubicarlo.
- Problema de asignación dinámica de memoria debido a la diferencia de tamaño de los segmentos.

Límite segmento	Base Tabla páginas	Otras protec, ..

■ Solución:

- Paginar los segmentos → segmentación paginada

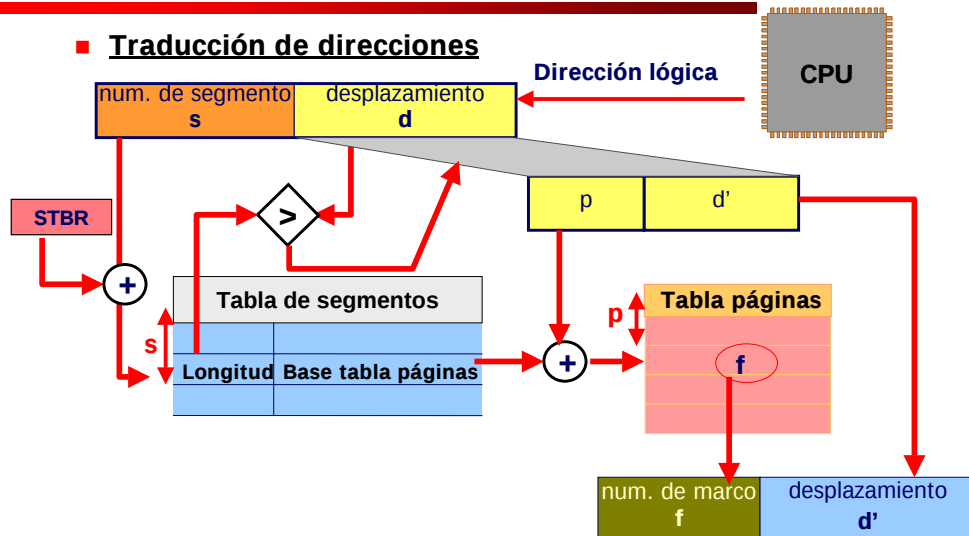
■ La tabla de segmentos

- Base de la tabla de páginas.
- Tamaño del segmento

(8.7)

La segmentación paginada

■ Traducción de direcciones



La segmentación paginada

■ Análisis

- Ventajas

- Ventajas de Segmentación
- Ventajas de Paginación
- Proporciona protección.

- Inconvenientes

- Fragmentación interna.

Contenidos

- 1.- Conceptos básicos
- 2.- Gestión de memoria
- 3.- Asignación contigua
- 4.- Asignación dispersa

- **Paginación** (8.5)
- **Segmentación** (8.6)
- **Técnicas combinadas**
 - Segmentación paginada (8.7)
 - Paginación multinivel (8.5.3)



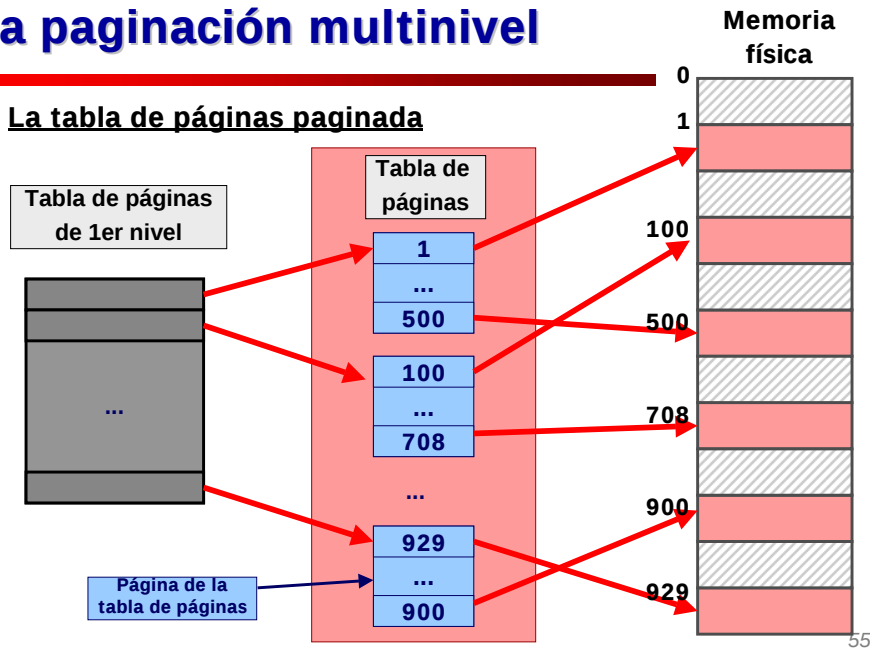
- 5.- Memoria virtual

La paginación multinivel

- **Motivación** (8.5.3)
 - Para espacios de direcciones muy grandes, la tabla de páginas puede ser excesivamente grande.
- **Solución:**
 - Paginar la propia tabla de páginas

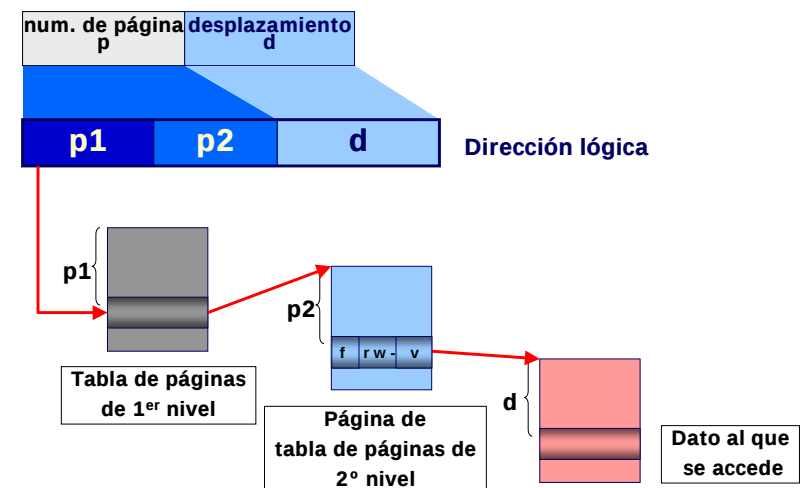
La paginación multinivel

■ La tabla de páginas paginada



La paginación multinivel

■ Traducción de direcciones



La paginación multinivel

■ Ejemplos reales de sistemas con paginación multinivel

- La arquitectura SPARC, con 32 bits de dirección soporta un esquema con 3 niveles de paginación.
- La arquitectura del Motorola 68030, con 32 bits de dirección soporta un esquema con 4 niveles de paginación.
- El Pentium de Intel, con 32 bits de dirección, soporta un esquema con 2 niveles de paginación.

■ Inconveniente:

Cada nivel de paginación requiere incrementar en uno el número de accesos a memoria para acceder a una dirección física.

57

La paginación multinivel

■ Solución:

- Utilizar un TLB
 - i386 de Intel, tiene un TLB con 32 entradas y consigue una tasa de aciertos del 98%
 - Motorola 68030 tiene un TLB de 22 entradas.
 - Pentium de Intel tiene dos TLBs:
 - el primero gestiona las páginas estándar de 4Kbytes y tiene una capacidad de 64 entradas.
 - el segundo para implementar páginas de 4Mb y tiene una capacidad de 8 entradas.
- Utilizar memoria caché para almacenar tablas de páginas. Con esto el sistema se ralentizará, pero únicamente entre un 20% y un 40%.

58

Contenidos

- 1.- Conceptos básicos
- 2.- Gestión de memoria
- 3.- Asignación contigua
- 4.- Asignación dispersa



5.- Memoria virtual

- Paginación por demanda
- Algoritmos de reemplazo
- Asignación de marcos
- Hiperpaginación

Memoria virtual

■ Concepto de memoria virtual

(9.1)

- Paginación por demanda.
- Segmentación por demanda.

■ Ventajas

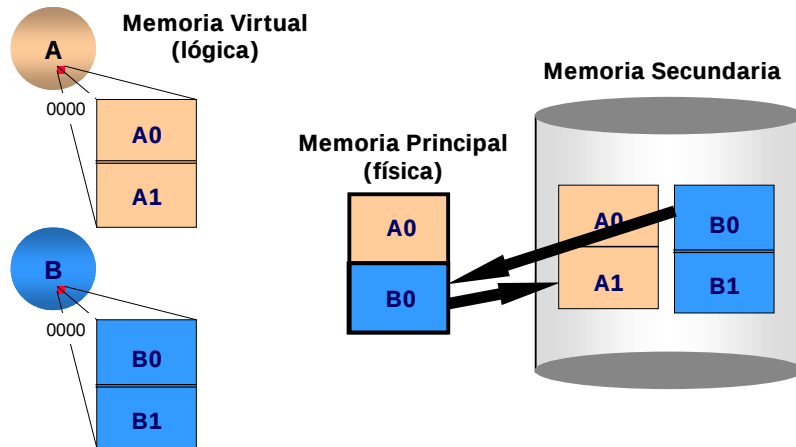
- Aumentar el grado de multiprogramación
- Aumentar el tamaño de los programas
- Ahorro de memoria.

■ Inconvenientes

- Reducir el rendimiento del sistema si no se diseña y sintoniza adecuadamente.
- Mayor complejidad.

Memoria virtual

■ Intercambios



61

Contenidos

- 1.- Conceptos básicos
- 2.- Gestión de memoria
- 3.- Asignación contigua
- 4.- Asignación dispersa

5.- Memoria virtual

- Paginación por demanda
- Algoritmos de reemplazo
- Asignación de marcos
- Hiperpaginación

TEMA 3.- Gestión de la memoria

62

Paginación por demanda

■ Concepto de paginación por demanda

(9.2)

- Paginación + Intercambios entre memoria principal y secundaria (*swapping*).

■ Descriptores de página

- Bit de validez
- Bit de referencia
- Bit de modificación
 - Problema con las páginas compartidas.

63

Paginación por demanda

■ Fallos de página

– Definición:

- Se produce cuando referenciamos una página que tiene su bit de validez igual a 0

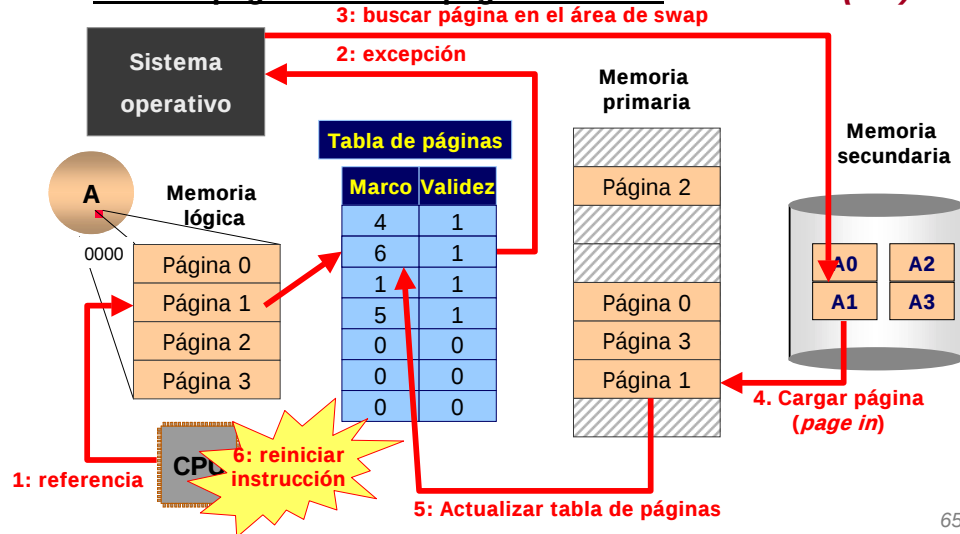
– Manejo de fallos de página

- Página en disco.
 - Se carga la página en un marco
- Error de acceso.
 - Se aborta el proceso
- El proceso “crece” y reclama nuevas páginas.
 - Si el SO lo permite, se le asigna una nueva página al proceso, se valida y se le asigna un marco.

64

Paginación por demanda

■ Fallo de página: caso de página en disco (9.2)



Paginación por demanda

■ Algoritmo de fallo de página: caso de página en disco (9.2)

- Encontrar la página demandada en disco.
- Encontrar un marco libre:
 - Si existe un marco libre, utilizarlo.
 - Si no, utilizar un algoritmo de reemplazo de páginas.
 - Si el bit de modificación es 1, escribir la víctima en disco (*page out*).
 - Actualizar la tabla de páginas, invalidando la víctima, y la tabla de marcos.
- Leer la página demandada del disco (*page in*) y ubicarla en el marco libre, actualizando la tabla de páginas y la tabla de marcos libres.
- Transferir control al proceso de usuario, reejecutando la instrucción que provocó el fallo de página.

66

Paginación por demanda

■ Reemplazo de páginas (9.4)

- Si la memoria principal está completamente ocupada y se produce un fallo de página:
 - Una página ubicada en memoria principal, denominada **víctima**, debe dejar su marco a la página demandada.
 - Si el bit de modificación es 1 hay que salvar la víctima a disco (*page out*)
 - Leer la página demandada sobre el marco de la víctima (*page in*)
- Existen diversos algoritmos para seleccionar la víctima.

67

Contenidos

- 1.- Conceptos básicos
- 2.- Gestión de memoria
- 3.- Asignación contigua
- 4.- Asignación dispersa
- 5.- Memoria virtual



- Paginación por demanda
- Algoritmos de reemplazo
- Asignación de marcos
- Hyperpaginación

TEMA 3.- Gestión de la memoria

68

Algoritmos de reemplazo de páginas

- Serie de referencias (9.5)
- Algoritmos de sustitución de páginas
 - Algoritmo FIFO
 - Algoritmo óptimo
 - Algoritmo LRU
 - Algoritmo de aproximación al LRU
 - Segunda oportunidad

69

Algoritmos de reemplazo de páginas

■ Algoritmo FIFO (9.5.1)

3 marcos de página: 9 fallos de página (6 reempl.)

	1	2	3	4	1	2	5	1	2	3	4	5
	1	1	1	4	4	4	5	5	5	5	5	5
		2	2	2	1	1	1	1	1	3	3	3
			3	3	3	2	2	2	2	2	4	4

4 marcos de página: 10 fallos de página (6 reempl.)

	1	1	1	1	1	1	5	5	5	5	4	4
	1	2	2	2	2	2	2	1	1	1	1	5
			3	3	3	3	3	3	2	2	2	2
			4	4	4	4	4	4	4	3	3	3

70

Algoritmos de reemplazo de páginas

- **Algoritmo FIFO**
 - Filosofía
 - La víctima será aquella página que hace mas tiempo que ha sido CARGADA en memoria.
 - Ventajas
 - Muy simple.
 - Inconvenientes
 - Presenta la anomalía de Belady

71

Algoritmos de reemplazo de páginas

■ Algoritmo óptimo (9.5.2)

3 marcos de página: 7 fallos de página (4 reempl.)

	1	2	3	4	1	2	5	1	2	3	4	5
	1	1	1	1	1	1	1	1	1	3	4	4
		2	2	2	2	2	2	2	2	2	2	2
			3	4	4	4	5	5	5	5	5	5

4 marcos de página: 6 fallos de página (2 reempl.)

	1	1	1	1	1	1	1	1	1	1	4	4
	1	2	2	2	2	2	2	2	2	2	1	1
			3	3	3	3	3	3	3	3	2	2
			4	4	4	4	5	5	5	5	5	5

72

Algoritmos de reemplazo de páginas

■ Algoritmo óptimo

- Filosofía
 - La víctima es aquella página que se tardará más en utilizar.
- Ventajas
 - Ofrece un número de fallos mínimo.
- Inconvenientes
 - Implementación imposible.

73

Algoritmos de reemplazo de páginas

■ Algoritmos de pila

(9.5.1)

- Definición
 - Garantiza que un conjunto de páginas mantenido con N marcos es un subconjunto del que se mantiene con N+1 marcos
- Propiedad
 - Si un algoritmo es de pila NUNCA presenta la anomalía de Belady

74

Algoritmos de reemplazo de páginas

■ Algoritmos de pila

- FIFO no es un algoritmo de pila

3 marcos de página: 9 fallos de página (6 reempl.)

	1	2	3	4	1	2	5	1	2	3	4	5
1	1	1	1	4	4	4	5	5	5	5	5	5
2		2	2	2	1	1	1	1	1	3	3	3
3			3	3	3	2	2	2	2	2	4	4

4 marcos de página: 10 fallos de página (6 reempl.)

	1	2	3	4	1	2	5	5	5	5	4	4
1	1	1	1	1	1	1	5	5	5	5	4	4
2		2	2	2	2	2	2	1	1	1	1	5
3			3	3	3	3	3	3	2	2	2	2
4				4	4	4	4	4	4	3	3	3

75

Algoritmos de reemplazo de páginas

■ Algoritmo LRU

(9.5.3)

- Filosofía:
 - La víctima será aquella página que hace más tiempo que ha sido REFERENCIADA.
- Propiedades:
 - Es un algoritmo de pila.

76

Algoritmos de reemplazo de páginas

■ Ejemplo algoritmo LRU

3 marcos de página: 10 fallos de página (7 reempl.)

	1	2	3	4	1	2	5	1	2	3	4	5
Marco 1	1	1	1	4	4	4	5	5	5	3	3	3
Marco 2		2	2	2	1	1	1	1	1	1	4	4
Marco 3			3	3	3	2	2	2	2	2	2	5

4 marcos de página: 8 fallos de página (4 reempl.)

	1	1	1	1	1	1	1	1	1	1	1	5
Marco 1	1	1	1	1	1	1	1	1	1	1	1	5
Marco 2		2	2	2	2	2	2	2	2	2	2	2
Marco 3			3	3	3	3	5	5	5	5	4	4
Marco 4				4	4	4	4	4	4	3	3	3

77

Algoritmos de reemplazo de páginas

■ Implementaciones del algoritmo LRU

(9.5.3)

– Contadores

– Cola

■ Ventajas

– Buena aproximación al óptimo

■ Inconvenientes

– Difícil de implementar

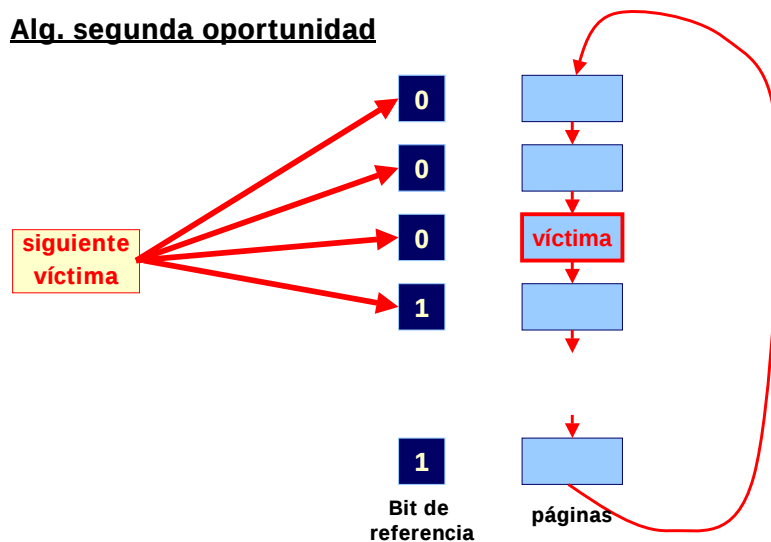
– Solución:

- Utilizar aproximaciones al algoritmo.

78

Algoritmos de reemplazo de páginas

■ Alg. segunda oportunidad



79

Contenidos

1.- Conceptos básicos

2.- Gestión de memoria

3.- Asignación contigua

4.- Asignación dispersa

5.- Memoria virtual

- Paginación por demanda
- Algoritmos de reemplazo
- Asignación de marcos
- Hiperpaginación



Asignación de marcos

- **Problema de asignación de marcos**
 - Lista de marcos libres
 - Reparto de marcos entre los procesos y el SO
- **Algoritmos de asignación de marcos**
 - Asignación equitativa
 - Asignación proporcional
 - Asignación prioritaria
- **Ambito de las políticas de reemplazo**
 - Reemplazo local
 - Reemplazo global

81

Contenidos

- 1.- Conceptos básicos
- 2.- Gestión de memoria
- 3.- Asignación contigua
- 4.- Asignación dispersa
- 5.- Memoria virtual
 - Paginación por demanda
 - Algoritmos de reemplazo
 - Asignación de marcos
 - Hiperpaginación

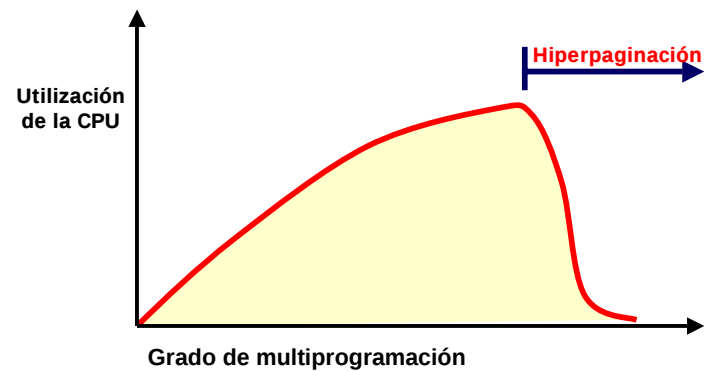


TEMA 3.- Gestión de la memoria

82

Hiperpaginación (*thrashing*)

- **El problema de la hiperpaginación** (9.7)
 - Causa de la hiperpaginación



83

Hiperpaginación (*thrashing*)

- **Principio de localidad de referencia** (9.7.1)
 - Localidad:
Conjunto de páginas que un proceso utiliza conjuntamente.
 - Principio de localidad de referencia.
 - **Hiperpaginación** $\equiv \Sigma(\text{tamaños de localidad}) > (\text{tamaño memoria total})$

84

Hiperpaginación (*thrashing*)

■ Modelo del área activa

(9.7.2)

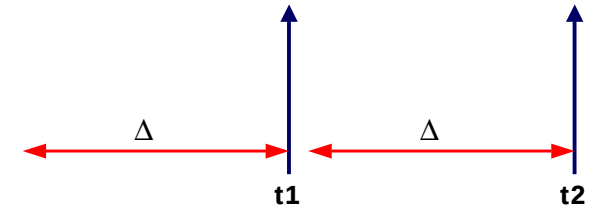
- Asume el principio de localidad de referencia.
- Determinar el número de páginas que un proceso que deben tener en memoria para obtener un buen rendimiento y evitar la hiperpaginación.
- **Área activa**: conjunto de páginas accedidas en las últimas Δ referencias.
- **Ventana de área activa Δ** Es un número fijo de referencias.

85

Hiperpaginación (*thrashing*)

■ Modelo del área activa

... 5 3 5 2 3 5 2 3 5 4 5 4 5 9 8 9 4 6 9 2 3 9 2



$AA(t1) = \{2,3,5\}$

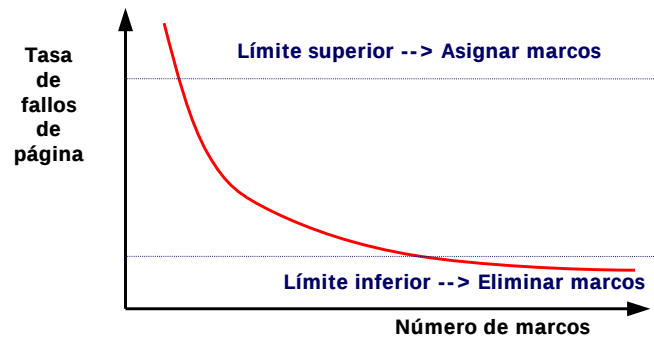
$AA(t2) = \{4,5,6,8,9\}$

86

Hiperpaginación (*thrashing*)

■ Control de la tasa de fallos de página

(9.7.3)



87