

Tema 4.

Administración de memoria

- 1. Gestión de memoria real
- 2. Gestión de memoria virtual

1

1. Gestión de memoria real

- Ordenadores cada vez con más memoria
 - Programas que cada vez ocupan más
- Gestor de memoria:
 - Parte del S.O. que se encarga de gestionar la memoria

2

1.1. Gestión de memoria más simple

- Monoprogramación:
 - Un único proceso en memoria principal que puede acceder a toda la memoria
 - No estaría el S.O
 - Obliga a que cada proceso gestione cada dispositivo de E/S

3

Otra forma

- Dejar sitio para el S.O.
 - Se ejecutan los procesos de 1 en 1.
 - El S.O carga un proceso, cuando termina, carga otro encima.

Programa de Usuario

SO

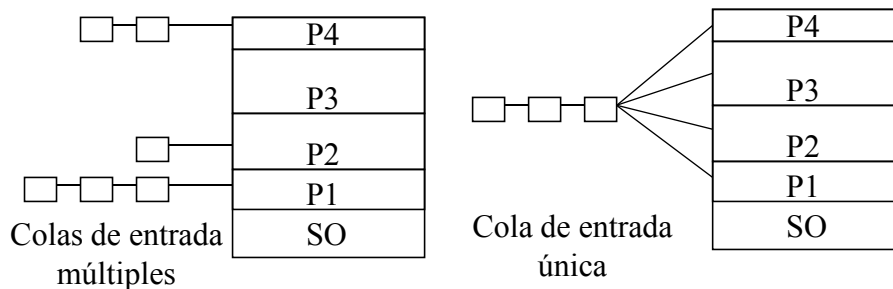
 0

- En ordenadores pequeños, no en grandes sistemas con múltiples usuarios
- Utilización de la multiprogramación para aprovechar tiempos muertos de E/S

4

1.2. Multiprogramación con particiones fijas

- Se divide la memoria en n particiones
- El espacio que no se usa se desperdicia



5

Reubicación

- Se usan direcciones relativas
- Reubicación: convertir direcciones relativas a absolutas una vez conocida la partición en la que se ha cargado
- Se puede hacer en el momento de la carga del programa en memoria
 - El montador de enlace incluye en el fichero ejecutable información referente a las palabras dentro del programa que se han de reubicar

6

Protección

- Hay que evitar que cualquier proceso acceda a una partición que no es la suya
- Posible solución
 - Dividir la memoria en bloques de 2Kbytes
 - Asignar clave de 4 bits a cada bloque
 - Incluir una clave de 4 bits en la palabra de estado del proceso
 - El hardware comprobaba que eran iguales

7

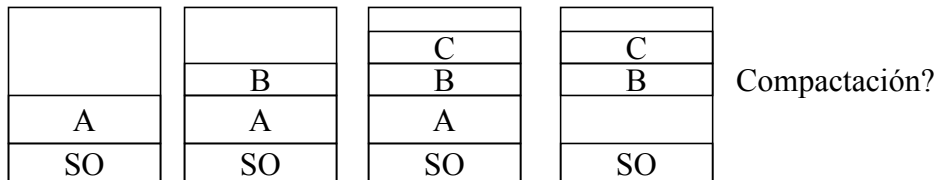
Otra solución al problema de la reubicación y la protección

- 2 registros especiales
 - » Registro de Base
 - » Registro de Límite
- Al asignar el procesador a un proceso
 - Base: dirección de comienzo de la partición
 - Límite: tamaño de la partición
 - Dir. Absoluta = Dir. rel + Base
 - Dir. rel < Límite
- Se puede mover un programa en memoria incluso después de empezar a ejecutarse

8

1.3. Multiprogramación con particiones variables

- Mejora la utilización de memoria
- El nº, la posición y el tamaño de las particiones varía dinámicamente al entrar y salir procesos de la memoria
- Complica la asignación y desocupación de la memoria



9

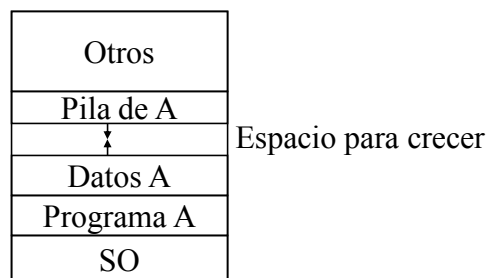
¿Qué cantidad de memoria asignar?

- Proceso de tamaño fijo: asignar tamaño
- Problema: cuando los procesos pueden crecer pidiendo reserva dinámica
 - Si hay hueco junto al proceso utilizar el hueco
 - En c.c:
 - » mover el proceso a un hueco mayor
 - » llevar algunos procesos a disco

10

Ejemplo de asignación de memoria para poder crecer

- Si se espera que la mayoría crezca:
 - Se les asigna algo más para reducir la probabilidad de que no quepan en la partición asignada



11

Métodos para llevar la cuenta de la ocupación de memoria

- Mapas de bits
- Listas enlazadas

12

Mapas de bits

- Se divide la memoria en unidades de asignación
- 1 bit representa el estado de cada unidad
 - » 1: está asignada
 - » 0: está libre



- Tamaño de la unidad
 - Cuanto menor, mayor el mapa
 - Cuanto mayor, más desperdicio de la memoria

11111000
11111111

13

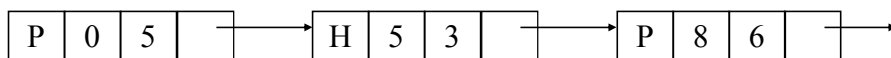
Mapas de bits

- Ventajas:
 - Representación sencilla
 - Tamaño fijo
- Desventaja
 - la localización de una zona de memoria lo suficientemente grande es costosa

14

Listas enlazadas

- Cada nodo representa un proceso o un hueco
 - » Información de si es proceso o hueco
 - » Dirección de comienzo
 - » Tamaño
 - » Puntero al siguiente



- Actualización:
 - PPP -> PHP; HPP -> HP; PPH -> PH; HPH -> H
- Lista doblemente enlazada

15

Métodos de asignación de memoria mediante listas

- Primer ajuste: El gestor de memoria asigna el primer hueco de tamaño suficiente
- Siguiendo ajuste: Se empieza por donde se había quedado en la búsqueda anterior
- Mejor ajuste(más lento y desperdicia más memoria): Asigna el hueco más pequeño de tamaño suficiente
- Peor ajuste: Selecciona el hueco disponible de mayor tamaño

16

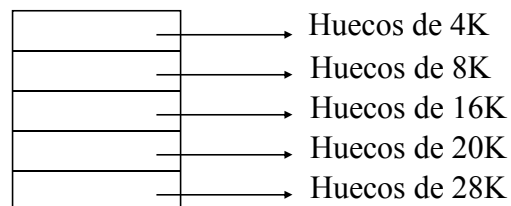
Métodos de asignación de memoria mediante listas

- Para optimizar:
 - Listas separadas de huecos y procesos
 - » La desocupación se vuelve más lenta
 - Ordenar por tamaños de huecos
 - » Mejora el primer ajuste y el mejor ajuste
 - Utilizar los mismos huecos para la lista de huecos:
 - » la primera palabra del hueco contiene el tamaño
 - » la siguiente, el puntero al hueco siguiente

17

Métodos de asignación de memoria mediante listas

- Ajuste rápido
 - Listas independientes para huecos de tamaños más comunes
 - La asignación es muy rápida
 - Es difícil la fusión de huecos



18

Asignación del espacio de intercambio

- En sistemas de tiempo compartido hay más procesos de los que caben en memoria.
 - » Se mantienen en disco.
 - » **Intercambio o swapping**: Operación de mover procesos de memoria a disco y de disco a memoria.
- Al crear el proceso se le asigna espacio en disco. Al acabar el proceso se libera este espacio
- O, se asigna al salir de memoria. Se puede copiar en cualquier posición.
- Algoritmos: los mismos que en memoria

19

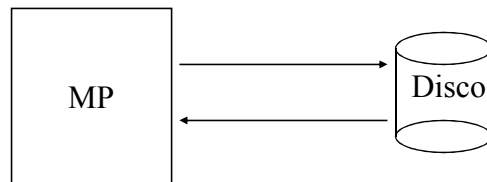
2. Gestión de memoria virtual

- Los programas pueden ser más grandes que la memoria
- Solución adoptada:
 - Dividir el programa en partes llamadas overlays (capas)
 - » La ejecución se empieza en el primer overlay
 - » Cuando termina se llama al siguiente y así
 - » Se pueden tener en memoria varios overlays
 - El SO se encarga del intercambio
 - De la partición del programa el programador

20

Memoria virtual

- Hace que esta operación sea transparente
- Permite que un proceso sea mayor que la cantidad de memoria disponible
- En memoria están las partes del programa que se están utilizando, el resto en disco



21

Memoria virtual

- En multiprogramación:
 - En memoria sólo una parte del programa
 - Se pueden tener más programas
 - Memoria de 2M:
 - 2 programas de 1M completamente en memoria
 - 8 programas con 1/4 de M cada uno en memoria
- Cuando un programa está a la espera de que se cargue en memoria una porción del mismo, está esperando una E/S y se puede seguir con otro programa
- Es usual utilizar **paginación**

22

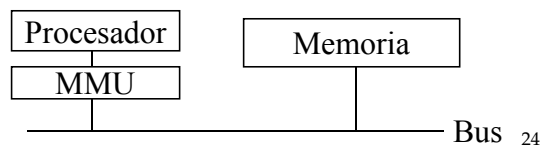
Paginación

- **Dirección virtual:** dirección generada por un programa.
- **Espacio de direcciones virtuales:** direcciones que puede generar un programa.
 - » Se divide en unidades llamadas **páginas**
 - » Las unidades correspondientes en memoria física se llaman **marcos de página** (512bytes, 1K, 2K, 4K)
 - » Las páginas y los marcos tienen idéntico tamaño.
 - » La transferencia entre memoria y disco se realizan siempre en unidades de páginas.

23

Paginación

- Si no hay memoria virtual, las direcciones generadas por un programa son direcciones físicas
- Con memoria virtual, las direcciones se mandan a la unidad de gestión de memoria (MMU)
 - » Dispositivo que se encarga de traducir dinámicamente las direcciones virtuales en físicas



Bus 24

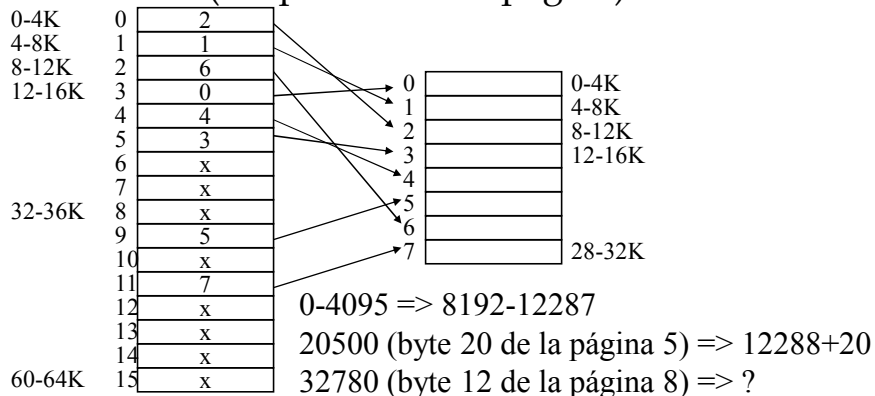
Ejemplo

- Direcciones de 16 bits/ memoria de 32 K:
 - Espacio de direcciones virtuales: 64K (2^{16})
 - Los programas de 64K no caben en memoria (se tiene una copia en disco)
- Páginas (y marcos de página) de 4K:
 - 16 páginas virtuales ($16 \times 4K = 64K$)
 - 8 marcos de página ($8 \times 4K = 32K$)
- Cuando se quiere acceder a una dirección:
 - El programa trabaja con dir. Virtuales
 - Se mandan a la MMU que las convierte

25

Tabla de Páginas

- Establece dónde se encuentra cada página virtual (en qué marco de página)



26

Fallo de página

- Trap generado por la MMU que captura el SO
- El SO:
 - Selecciona un marco de página
 - Copia su contenido en disco
 - Carga la página que se necesita en ese marco
 - Cambia la tabla de traducción de la MMU
 - Comienza de nuevo esa instrucción interrumpida

27

Funcionamiento de la MMU

Dirección Virtual => 8196

0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

0	010	1
1	001	1
2	110	1
3	000	1
4	100	1
5	011	1
6	x	0
7	x	0
8	x	0
9	101	1
10	x	0
11	111	1
12	x	0
13	x	0
14	x	0
15	x	0

Dirección Física => 24580

Bit de presente/ausente

28

Aspectos de la traducción

- La tabla de páginas puede ser muy grande (cada proceso cuenta con su propia tabla de páginas)
 - si dir. de 32 bits y páginas de 4K
 - » 1M de páginas ($2^{32} / 2^{12} = 2^{20}$)
- La asociación/traducción debe ser rápida
 - » Se debe hacer la traducción para cada referencia a memoria
 - » En una instrucción puede haber varias referencias

29

Distintos diseños de la tabla de páginas

- Vector de registros rápidos en hardware
- Tabla de páginas en memoria
- Tablas de páginas de varios niveles
- Memoria asociativa

30

Vector de registros rápidos en hardware

- Una entrada por página virtual
- Al iniciar el proceso se carga la tabla de páginas del proceso en estos registros
- Durante la asociación no hay que acceder a memoria
- Costosa en recursos si la tabla de páginas es grande (1 registro por página)
- Costosa en tiempo
 - En el cambio de contexto habría que cargar la tabla de páginas

31

Tabla de páginas en memoria

- En hardware sólo se necesita un registro que apunte al inicio de la tabla de páginas
- El cambio de contexto:
 - modifica un registro
- Desventaja:
 - Hay que hacer una o más referencias a memoria por cada instrucción para acceder a la tabla de páginas

32

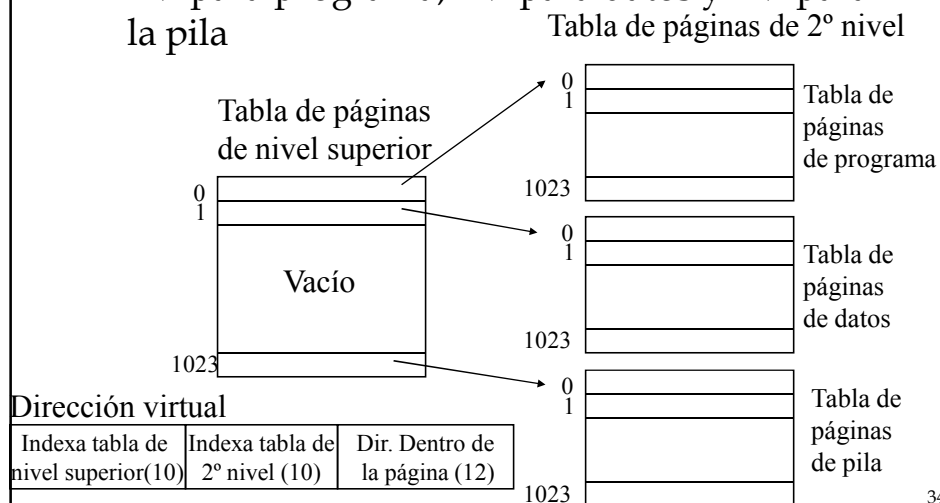
Tablas de páginas de varios niveles

- No todo el espacio de direcciones virtuales es utilizado por todos los procesos
 - Con 32 bits:
 - » se pueden direccionar 4GBytes (2^{32} bytes)
 - » con páginas de 4K: 2^{20} páginas ($2^{32}/2^{12}$)
 - » tabla de páginas con 1 Millón de entradas
 - » sólo se utilizan si el proceso ocupa 4G
 - Para un proceso de 12M:
 - » $(12 \times 2^{20})/2^{12} = 12 \times 2^8$ páginas = 3×2^{10} pág (3K pág)

33

Tabla de páginas de 2 niveles

- 4M para programa, 4M para datos y 4M para la pila



34

Memoria asociativa

- Permite acelerar la traducción
- Se basa en la observación de que la mayoría de los programas hacen un gran número de referencias a un pequeño número de páginas
- Se equipa a la máquina con un dispositivo hardware que permite asociar direcciones virtuales con físicas: la memoria asociativa
 - parte de la MMU
 - número pequeño de entradas
 - Búsqueda en todas las entradas en paralelo

35

Memoria asociativa

- Para la traducción
 - » Se comprueba si está la información en la memoria asociativa
 - » Si está: ya se puede hacer la traducción
 - » En c.c.: búsqueda normal
- Proporción de encuentros: proporción de referencias a memoria que pueden ser satisfechas a partir de la memoria asociativa
 - Cuanto mayor, mayor el rendimiento

Página virtual	Marco de página	
		0
		7

36

Algoritmos de sustitución de páginas

- Cuando se produce un fallo de página: el SO debe decidir qué página que está en memoria debe pasar a disco para traer a memoria la página requerida.
- Si la página que sale ha sido modificada: se reescribe en disco.
- Se puede elegir aleatoriamente, pero es más eficiente sacar una que no se vaya a utilizar pronto.

37

Sustitución óptima de páginas

- Buscar de todas las páginas de memoria aquella a la que se tardará más en hacer una referencia a ella.
- Retarda todo lo posible el fallo de página.
- Irrealizable: no se dispone de esta información

38

Sustitución de la página no usada recientemente (NRU)

- 2 bits asociados a cada página
 - bit R (bit de referencia): se pone a 1 cada vez que se lee o se escribe
 - bit M (bit de modificación): se pone a 1 cuando se escribe
- Cuando se carga en memoria: ambos a 0
- Con cada interrupción: se pone el bit R a 0
- Con el fallo de página:

R	M
0	0
0	1
1	0
1	1

39

NRU

- Se toma aleatoriamente una página del grupo 0, si no hay del grupo 1,
- Es mejor sustituir una página que no se haya usado recientemente aunque haya sido modificada, que una que se haya usado en el último pulso de reloj, ya que tiene mucha probabilidad de usarse de nuevo.
- Ventaja:
 - » fácil de entender
 - » Implementación eficiente (rendimiento adecuado)

40

Primera en entrar primera en salir (FIFO)

- Lista con todas las páginas en memoria
 - » Una página que entra se añade al final
 - » En el fallo de página se saca la página que está a la cabeza de la lista
- Puede que se sustituya una página muy utilizada
 - Se mira el bit R y M
 - Se sustituye la más antigua a la que no se haya hecho referencia
 - Segunda oportunidad: Si R a 1 se pone a 0 y se pone al final de la lista

41

Sustitución de la página usada hace más tiempo (LRU)

- Se sustituye la que lleve más tiempo sin usarse
 - » Probablemente no se use en las próximas instrucciones
- Implementación:
 - Lista de todas las páginas en memoria
 - » Al principio: la usada más recientemente
 - » Al final: la que lleve más tiempo sin usarse
 - » Se actualiza con cada referencia a memoria. Se busca la página y se lleva al principio
 - Es costosa

42

LRU: otra forma (I)

- Contador de 64 bits
 - » Se incrementa después de cada instrucción
- Asociado a cada página:
 - » un campo donde poder guardar el contador
 - » cada vez que se accede a la página se actualiza al contador
- Con el fallo de página
 - » Se examinan los contadores
 - » Se sustituye la página con un contador menor

43

LRU: otra forma (II)

- N marcos de página
- Matriz de NxN a 0
 - Cada vez que se accede al marco K
 - » fila K a 1; columna k a 0
- Lleva más tiempo sin utilizarse la página correspondiente a la fila de menor valor binario. Para: 0 1 2 3 2 1 0 3 2 3

0	<table><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	<table><tr><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	0	0	1	1	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	2	<table><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	0	0	0	1	1	1	0	0	1	1	1	1	0	1	1	0	0	0	0	0	3	<table><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td></tr></table>	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	1	1	1	0	0	2	<table><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td></tr></table>	0	0	0	0	0	1	0	0	0	0	1	1	0	1	1	1	1	0	0	0
0	1	1	1	1																																																																																																									
0	0	0	0	0																																																																																																									
0	0	0	0	0																																																																																																									
0	0	0	0	0																																																																																																									
0	0	1	1	1																																																																																																									
1	0	1	1	1																																																																																																									
0	0	0	0	0																																																																																																									
0	0	0	0	0																																																																																																									
0	0	0	1	1																																																																																																									
1	0	0	1	1																																																																																																									
1	1	0	1	1																																																																																																									
0	0	0	0	0																																																																																																									
0	0	0	0	0																																																																																																									
1	0	0	0	0																																																																																																									
1	1	0	0	0																																																																																																									
1	1	1	0	0																																																																																																									
0	0	0	0	0																																																																																																									
1	0	0	0	0																																																																																																									
1	1	0	1	1																																																																																																									
1	1	0	0	0																																																																																																									

44

Sustitución de página no usada frecuentemente (NFU)

- Contador software asociado a cada página
 - » con valor inicial 0
 - » Con cada interrupción de reloj se suma al contador el bit R
 - » lleva la cuenta del nº de pulsos en los que se ha hecho referencia
- Se sustituye la que tenga el contador más bajo
- Problema: nunca olvida
 - » Puede ser que al principio de un programa se utilicen algunas páginas muy frecuentemente y luego ya no se utilicen.

45

Algoritmo de envejecimiento

- Modificación para simular LRU
 - » Antes de sumar R, los contadores se desplazan a la derecha
 - » El bit R se suma al bit más a la izquierda
 - » Se sustituye la página con contador más pequeño

Bit R	Contador
1	1 0 0 0 0 0 0 0
0	0 1 0 0 0 0 0 0
0	0 0 1 0 0 0 0 0
1	1 0 0 1 0 0 0 0

46

Diferencias

- Entre envejecimiento y LRU
 - si 00100000 y 001010000
 - » en LRU cualquiera de ellas, la que se referenciara antes en el pulso 3
 - Como son contadores finitos, no es capaz de distinguir entre una página que se referenció hace 9 pulsos o una que hace 300
- Ya no es NFU
 - si 00100111 y 00110000
 - » Se sustituye la primera, pero la 2ª es accedida menos frecuentemente

47

Aspectos a tener en cuenta en un sistema de paginación

- Conjunto operativo
- Políticas de asignación global frente a local
- Tamaño de página
- Aspectos de implementación
 - Instrucciones reiniciables
 - Inmovilización de páginas en memoria
 - Páginas compartidas
 - Demonios de paginación

48

Conjunto operativo

- Estrategia de paginación por petición
 - Se arrancan los procesos sin ninguna página en memoria
 - Cuando se inicia, se produce un fallo de página
 - Transcurrido cierto tiempo, el proceso tiene en memoria todas las páginas que le hace falta

49

Conjunto operativo

- Método del conjunto operativo
 - Vecindad de referencias:
 - » en una fase de la ejecución de un programa, sólo se hace referencia a un conjunto pequeño de sus páginas
 - Conjunto operativo
 - Hiperpaginación:
 - » Si se produce un fallo de página cada pocas instrucciones
 - Prepaginación: carga de páginas antes de empezar
 - El S.O. se encarga de que el conjunto operativo quepa en memoria reduciendo el índice de multiprogramación si hace falta
 - El S.O. lleva la cuenta de los conjuntos operativos

50

Políticas de asignación global frente a local

- Local
 - Se busca la página a ser sustituida entre las del proceso
 - Se asigna una cantidad fija de memoria
 - Si el conjunto operativo disminuye, se desperdicia memoria
- Global
 - Se busca la página a ser sustituida entre todas
 - Se debe decidir cuanto asignar a cada proceso
 - » Conjunto operativo
 - » Asignación por frecuencia de fallos (PFF)
 - Se intenta mantener entre unos límites mínimo y máximo

51

Tamaño de página

- Si grandes
 - » Se desperdicia en promedio la mitad de una página por proceso, por fragmentación interna
 - » Más espacio se desperdicia
 - » Más porción de programa en memoria sin utilizar
- Si pequeñas
 - » Tablas de páginas más grandes
 - » Se tarda prácticamente lo mismo en transferir

52

Instrucciones reiniciables

- Cuando se accede a una página que no está en memoria, se genera un trap
- El S.O. carga esta página y la instrucción debe comenzar de nuevo
- Se debe saber dónde empieza la instrucción
- Puede haber problemas con el autoincremento y autodecremento

53

Inmovilización de páginas en memoria

- La E/S puede plantear problemas
- Ej: un proceso pide E/S del terminal
 - Se queda bloqueado esperando
 - Se produce fallo de página
 - Se puede sustituir la página que contiene el buffer de E/S
 - Los datos se leen o escriben en la página recién cargada
- Solución
 - Inmovilizar las páginas en memoria
 - Realizar E/S en buffers del núcleo

54

Páginas compartidas

- Si hay más de un usuario ejecutando el mismo programa es más eficiente compartir páginas
- No todas las páginas se pueden compartir
- Si A se saca de memoria, no es eficiente expulsar todas las páginas si las está compartiendo con B, ya que se producirían muchos fallos de páginas
- Igual si A termina, sería fatal para B eliminar todas las páginas de A del disco

55

Demonios de paginación

- Proceso secundario
- Se despierta de forma periódica
 - Mira el estado de la memoria
 - Asegura que haya un n^o mínimo de marcos de página libres
 - Si hay menos, selecciona algunas páginas y las lleva a disco si han sido modificadas
- Mejora el tiempo en cargar una nueva página
- El contenido de las páginas no se elimina

56

Segmentación

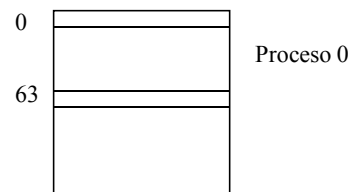
- El espacio de direcciones se divide en segmentos
 - Cada segmento tiene un nombre y una longitud
 - Una dirección se especifica como (s,d)
 - El compilador construye los distintos segmentos
- Para realizar la correspondencia:
 - Tabla de segmentos
 - » base: comienzo del segmento
 - » límite: tamaño del segmento
 - (s,d) => base+d (d< límite)
- Se utiliza segmentación paginada

57

Funcionamiento de la MMU

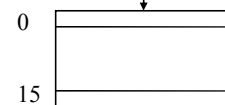
- 16 procesos
- Páginas de 4K
- 1024 páginas por proceso
- espacio de direcciones virtuales del proceso de 4M
- Espacio de direcciones virtuales total de 64 M
- 64 segmentos por proceso
- 16 páginas por segmento
- 64 K por segmento

Tabla de segmentos



Entrada en la tabla de segmentos

Longitud	Protección	P. A tabla de Páginas
----------	------------	-----------------------



58

Traducción de direcciones virtuales

- Dirección virtual de 22 bits (4 M)
- Registro especial de 4 bits indicando el proceso

4	6	4	12
Proceso	Nº de segmento	Nº de página	Posición dentro de la página

- Para cambiar de proceso se cambia el registro
- Se puede compartir un segmento haciendo apuntar a la misma tabla de páginas

59