
Problemas de Sistemas Operativos

Dpto. ACYA, Facultad de Informática, UCM

Modulo 4 – Gestión de Memoria

Problemas básicos

1.- El gestor de memoria recibe la siguiente secuencia de peticiones:

1. Asignar el bloque b1 de tamaño 100
2. Asignar el bloque b2 de tamaño 500
3. Asignar el bloque b3 de tamaño 60
4. Asignar el bloque b4 de tamaño 100
5. Liberar el bloque b1
6. Liberar el bloque b3
7. Asignar el bloque b5 de tamaño 50
8. Asignar el bloque b6 de tamaño 90

Suponiendo un tamaño de memoria total de 1024B, indicar las direcciones iniciales y los tamaños de todas las áreas libres para los esquemas de gestión (a) primer ajuste y (b) mejor ajuste, después que se han procesado todas las peticiones.

2.- Considerar los cuatro sistemas siguientes:

	A	B	C	D
Tamaño de página (en palabras)	512	512	1024	1024
Tamaño de palabra (en bits)	16	32	16	32

Para cada sistema determinar el número de entradas de la tabla de páginas (TMP). Suponer que sólo existe una TMP para cada sistema y que cada dirección virtual ocupa una palabra (16 o 32b).

3.- Un sistema de paginación pura tiene un tamaño de página de 512 palabras, una memoria virtual de 512 páginas numeradas de 0 a 511, y una memoria física de 10 marcos de páginas numerados de 0 a 9. El contenido actual de la memoria física es el siguiente:

Memoria	
0	—
1536	Pag. 34
2048	Pag. 9
	—
3072	TMP
3584	Pag. 65
	—
4608	Pag. 10

- a) Mostrar el contenido actual de la tabla de páginas, TMP
- b) Mostrar el contenido de la TMP después de cargar la página 49 en la posición 0 y de sustituir la página 34 por la página 12
- c) ¿Qué direcciones físicas referencian las direcciones virtuales 4608, 5119, 5120 y 33300?
- d) ¿Qué ocurre cuando se referencia la dirección virtual 33000?
- e) Si la página cargada en el marco de página 9 es un procedimiento y otro proceso **q** desea compartirlo, dónde debe aparecer en la TMP de **q**? (Indicar la entrada afectada de la TMP)

4.- Considera los siguientes programas

```
int A[1000], B[1000];
```

```
for (i=0; i<1000; i++)  
    a[i] = a[i] + b[i];
```

```
int A[1000], B[1000];
```

```
for (i=0; i<1000; i++)  
    a[999 - i] = a[i] + b[i];
```

Suponiendo que un entero ocupa una palabra, calcular el número de fallos de página generados por cada programa en cada uno de los casos siguientes:

- Se asignan dos marcos de página de 100 palabras para los arrays A y B
- Se asignan tres marcos de página de 100 palabras para los arrays A y B
- Se asignan diez marcos de página de 100 palabras para los arrays A y B
- Se asignan dos marcos de página de 500 palabras para los arrays A y B

Calcula los resultados para las políticas FIFO y LRU. Suponer que tanto el código como la variable *i* están colocados en otro marco de página y no dan lugar a fallos de página. Suponer también que la memoria está inicialmente vacía.

5.- Considerar la siguiente secuencia de referencias a memoria virtual generadas por un sólo programa en un sistema con paginación pura:

0x10 0x1A 0x1F4 0x17C 0x7C 0x3B9 0x185 0x2FF 0x24C 0x434 0x458 0x36D.

- Deducir la correspondiente cadena de referencias, suponiendo un tamaño de página de 256 Bytes
- Determinar el número de fallos de página para cada una de las siguientes estrategias de sustitución, suponiendo que hay dos marcos de página disponibles para el programa:
 - OPT
 - FIFO
 - Reloj
 - LRU

Problemas adicionales

6.- Un sistema informático tiene sitio suficiente en memoria principal para contener cuatro programas. Los programas están inactivos esperando E/S la mitad del tiempo. ¿Qué fracción del tiempo de CPU se desaprovecha?

7.- Encontrar una expresión que permita predecir el aumento de productividad de un sistema en función del grado de multiprogramación y de la probabilidad *p* de que los programas se hallen esperando E/S. Suponer que un computador tiene 2MiB de memoria principal, de las cuales el sistema operativo ocupa 512KiB (una cuarta parte) y cada programa de usuario también ocupa 512KiB. Si todos los programas tienen un 60 % de espera por E/S, ¿en qué porcentaje aumentará la productividad si se añade 1MiB más de memoria?

8.- Considerar un sistema de tiempo compartido con *swapping*, un disco para *swap* y tres particiones fijas de memoria. El tiempo de latencia promedio para el disco es de 4ms y el tiempo de transferencia de una partición es de 6ms. La idea es intercambiar una partición mientras se ejecutan los procesos de las otras dos. Un proceso se descarga a disco sólo cuando espera una entrada de usuario y se vuelve a cargar cuando existe una partición libre y el usuario ha introducido una línea.

- Para que la utilización de la UCP y del disco fuera del 100 % ¿cuánto tiempo debería ejecutarse un proceso que esté procesando una línea de entrada antes de esperar a la siguiente línea?
- Si los usuarios enviaran una línea cada segundo para su procesamiento, ¿cuál es el número máximo de usuarios que podría atenderse sin retardo extra?

9.- Considerar un sistema con 4200 palabras de MP que implementa particiones variables. En un cierto instante la memoria contiene las tres particiones siguientes:

	Dirección inicial	Tamaño
P1	1000	1000
P2	2900	500
P3	3400	800

Para cargar un nuevo bloque en memoria se utiliza la siguiente estrategia:

- Probar el algoritmo del mejor ajuste para localizar un hueco del tamaño adecuado
- Si esto falla, crear un hueco mayor desplazando particiones en memoria hacia la posición 0; empezar siempre por la partición de menor dirección y continuar sólo hasta que se haya generado un hueco de tamaño suficiente para albergar la nueva partición.

Suponer que van a cargarse tres nuevos bloques de tamaños 500, 1200 y 200, respectivamente (en el orden indicado). Mostrar el contenido de la MP después de satisfacer las tres peticiones de memoria.

10.- Sea **A** el tamaño medio en bytes de los procesos que se ejecutan en un sistema y sea **b** el número de bytes de cada una de las entradas en la Tabla de Mapa de Páginas (TMP). Deducir, en función de **A** y **b**, el tamaño de página **p** que minimiza la cantidad de memoria desaprovechada. Aplicar al caso en que **A**=1MiB y **b**=4Bytes.

11.- Si se utilizase sustitución de páginas FIFO con cuatro marcos de página y ocho páginas, ¿cuántos fallos de página ocurrirían para la cadena de referencias 0 1 7 2 3 2 7 1 0 3 si los cuatro marcos estuvieran inicialmente vacíos? Repetir el problema para LRU.

12.- Suponiendo una memoria física de cuatro marcos de página, indicar el número de fallos de página que genera la cadena de referencias a b g a d e a b a d e g d e para cada una de las siguientes políticas de sustitución. (Inicialmente todos los marcos están vacíos):

- Optima
- FIFO
- Reloj
- LRU

13.- Considerar la siguiente secuencia de referencias a direcciones de memoria virtual generadas por un sólo programa en un sistema con paginación pura:

0x10, 0x1A, 0x1F4, 0x17C, 0x7C, 0x3B9, 0x185, 0x2FF, 0x24C, 0x434, 0x458, 0x36D

- Deducir la cadena de referencias (secuencia de números de página generadas por el programa), suponiendo un tamaño de página de 256 Bytes.
- Determinar razonadamente el número de fallos de página para cada una de las siguientes estrategias de sustitución, suponiendo que hay dos marcos de página disponibles para el programa: Óptima, FIFO, LRU, Reloj.