

# SISTEMAS OPERATIVOS

## PROBLEMAS DE GESTIÓN DE MEMORIA

### PARTE 1

1. 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?
2. 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 2M de memoria principal, de las cuales el sistema operativo ocupa 512K (una cuarta parte) y cada programa de usuario también ocupa 512K. Si todos los programas tienen un 60% de espera por E/S, en qué porcentaje aumentará la productividad si se añade 1M más de memoria.
3. 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 4 ms; el tiempo de transferencia de una partición es de 6 ms. 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; se vuelve a cargar cuando existe una partición libre y el usuario ha introducido una línea.
  - (a) 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?
  - (b) Si los usuarios enviaran una línea cada segundo para su procesamiento, ¿cuál es el número máximo de usuarios que podría atenderse?
4. Considerar un sistema con 4200 palabras de Mp que usa particiones variables. En un cierto instante la memoria contiene las tres particiones siguientes:  
P1 (dirección inicial=1000; tamaño=1000); P2 (2900; 500) y P3 (3400; 800)  
Para cargar un nuevo bloque en memoria, se utiliza la siguiente estrategia:
  - a. Probar el algoritmo del mejor ajuste para localizar un hueco del tamaño adecuado
  - b. 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.
5. 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 90Suponiendo un tamaño de memoria total de 1024, 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.

### PARTE 2

1. 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  $A = 1$  MB y  $b = 4$  bytes

2. Un computador permite que cada proceso ocupe un espacio de direcciones de 65.536 bytes dividido en páginas de 4096 bytes. Un programa particular está formado por tres segmentos: el de texto (código) que ocupa 32768 bytes, el de datos que ocupa 16386 bytes y el de pila que ocupa 15870 bytes. ¿Encajará este programa en el espacio de direcciones del computador? ¿Encajaría si el tamaño de página fuese 512 bytes?.
3. Un computador cuyos procesos tienen 1024 páginas en sus espacios de direcciones mantiene las TMP en memoria. El recargo necesario para leer una palabra de la TMP es de 500 ns. Para reducir este recargo, el computador emplea una memoria asociativa (TLB) que contiene 32 pares (página virtual, marco de página física), y que puede hacer una búsqueda en 100 ns. ¿Qué tasa de aciertos es necesaria para reducir el recargo promedio a 200 ns?
4. 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:

- (a) el tamaño de la tabla de páginas (TMP), es decir su número de entradas, y
- (b) el tamaño en páginas de la memoria virtual .

Suponer que sólo existe una TMP para cada sistema y que cada dirección virtual ( $p, d$ ) ocupa una palabra (16 o 32 bits).

5. Considerar la siguiente tabla de segmentos

Segmento	Base	Límite
0	219	600
1	2300	14
2	90	100
3	1327	580
4	1952	96

¿Cuáles son las direcciones físicas para las siguientes direcciones lógicas:

- (a) 0,430                      (b) 1,10
- (c) 1,11                      (d) 2,500
- (e) 3,400                      (f) 4,112 ?

6. 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 Física
0	---
1536	Página 34
2048	Página 9
	---
3072	TMP
3584	Página 65
	---
4608	Página 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)
7. Considerar dos procesos  $p_1$  y  $p_2$  en un sistema segmentado puro. Sus tablas TDS respectivas contienen actualmente las siguientes entradas (no se muestra el tamaño de los segmentos):

TDS1

TDS2

0	4000
1	6000
2	9000
3	2000
4	7000

0	2000
1	6000
2	9000

- (a) ¿Cuáles son los segmentos compartidos, si hay alguno?
- (b) ¿Cuáles de las siguientes direcciones virtuales sería ilegal cuando fuese utilizada por el segmento ubicado en la posición 6000: (0,0), (1,0), (2,0), (3,0), (4,0), (5,0)? Razonar la respuesta.
- (c) El segmento de la posición 7000 es retirado de memoria y posteriormente vuelto a cargar en la posición 8000. Análogamente el segmento de la posición 2000 es retirado y luego vuelto a cargar en la posición 1000. Mostrar las nuevas tablas TDS.
- (d) Un tercer proceso p3 desea compartir los segmentos de las posiciones 2000, 4000 y 6000. ¿Cuáles de ellos deben ser segmentos de datos (y no de código) para que esto sea posible?

### PARTE 3

- 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.
- 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)
  - OPT
  - FIFO
  - Reloj
  - LRU
  - WS con  $\theta = 3$
- Considerar la siguiente secuencia de referencias a memoria virtual generadas por un sólo programa en un sistema con paginación pura:  
10, 11, 104, 170, 73, 309, 185, 245, 246, 434, 458, 364
  - Deducir la correspondiente cadena de referencias, suponiendo un tamaño de página de 100 palabras
  - 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:
    - FIFO
    - Reloj
    - LRU
- Considerar la cadena de referencias a b c d e b c d c b d d b d d d. Suponiendo que se aplica la estrategia de sustitución Conjunto Operativo (WS), determinar el tamaño mínimo de ventana para que la cadena genere como máximo cinco fallos de página. Mostrar las páginas que hay en memoria en cada referencia. Marca los fallos de página con un asterisco.
- 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[1000 - 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 a los arrays **A** y **B**
- Se asignan tres marcos de página de 100 palabras a los arrays **A** y **B**
- Se asignan diez marcos de página de 100 palabras a los arrays **A** y **B**
- Se asignan dos marcos de página de 500 palabras a 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.

6. Un computador tiene cuatro marcos de página. La lista siguiente muestra los tiempos de carga, y de último acceso, y los valores de los bits R (*Referenciada*) y M(*Modificada*), para cada página (los tiempos están contabilizados en tics de reloj):

Página	Carga	Último acceso	R	M
0	126	279	0	0
1	230	260	1	0
2	120	272	1	1
3	160	280	1	1

¿Qué página sustituirá la estrategia NRU?. ¿Y la FIFO?. ¿Y la LRU?. ¿Y la Reloj?

7. Un pequeño computador tiene cuatro marcos de página. En el primer tic de reloj, los bits R son 0111 (el de la página 0 está a 0, el resto a 1). En los tics de reloj siguientes, los valores son 1011, 1010, 1101, 0010, 1010, 1100 y 0001. Si se aplica el algoritmo de LRU aproximado (*envejecimiento*) con un contador de 8 bits, indicar cuáles son los valores de los cuatro contadores después del último tic.