

**2.- (3puntos)**

En un sistema que implementa memoria virtual mediante demanda de páginas, un proceso genera la siguiente secuencia de referencias a páginas de memoria:

2 5 1 3 5 0 4 1 0 8 7 9 5 7 1 0 2 5 9 8 2

Con 5 marcos de página inicialmente vacíos estudiar:

- a) cuantos fallos de página se producen si se utiliza el algoritmo LRU para la sustitución de páginas
- b) cuantos fallos de página se produce si el algoritmo utilizado es el FIFO
- c) razonar si aumentando indefinidamente el número de marcos de página se mejoraría la tasa de fallos de página.

**Solución:**

a) Este algoritmo sustituye la página en memoria que no ha sido referenciada en memoria desde hace mas tiempo.

2	5	1	3	5	0	4	1	0	8	7	9	5	7	1	0	2	5	9	8	2
X	2	5	1	3	5	0	4	1	0	8	7	9	5	7	1	0	2	5	9	8
X	X	2	5	1	3	5	0	4	1	0	8	7	9	5	7	1	0	2	5	9
X	X	X	2	2	1	3	5	5	4	1	0	8	8	9	5	7	1	0	2	5
X	X	X	X	X	2	1	3	3	5	4	1	0	0	8	9	5	7	1	0	0
F	F	F	F		F	F			F	F	F	F		F	F	F		F	F	

Hay 15 fallos de página

b) En este algoritmo se sustituye la página que mas tiempo lleva en memoria

2	5	1	3	3	0	4	4	4	8	7	9	5	5	1	0	2	2	2	8	8
X	2	5	1	1	3	0	0	0	4	8	7	9	9	5	1	0	0	0	2	2
X	X	2	5	5	1	3	3	3	0	4	8	7	7	9	5	1	1	1	0	0
X	X	X	2	2	5	1	1	1	3	0	4	8	8	7	9	5	5	5	1	1
X	X	X	X	X	2	5	5	5	1	3	0	4	4	8	7	9	9	9	5	5
F	F	F	F		F	F			F	F	F	F		F	F	F				F

Hay 14 fallos de página

c) Hay que tener en cuenta que tenemos 9 páginas diferentes (10, si se cuenta la 6, aunque no se ha referenciado en el enunciado), luego como mínimo tenemos que tener 9 fallos de página (10 si no contamos la 6). En cuanto tengamos un número de marcos de página que ya nos permita tener 10 fallos de página, aunque aumentemos el número de marcos, los fallos no los reduciremos.

**10.- La fragmentación externa ocurre cuando:**

- a) Dentro de una partición hay una parte de memoria que no se puede utilizar.
- b) Se desperdicia memoria en variables no utilizadas por el programa.
- c) *No se utiliza una partición libre porque es pequeña para las tareas que esperan.*
- d) Se utilizan todas las particiones libres aunque las tareas que esperan sean mas grandes que ellas.

**2.- (3puntos)**

Una memoria virtual dispone de 9 marcos para programas. En ese momento se está ejecutando el proceso P1, que tiene concedidos 3 marcos, P2 y P3 solicitan memoria para ajustarse (el tamaño de P2 es el doble que el de P3, 1024 y 512 Kb respectivamente). Se realiza un reparto de todos los marcos libres de forma proporcional al tamaño de los procesos y ambos entran en ejecución. **Cuando el proceso P3 demanda por primera vez la página 3 el proceso P1 termina** y sus marcos se reparten entre los procesos P2 y P3 de forma proporcional a sus tamaños.

- a) Determine cuántos marcos de página corresponden a cada proceso en cada instante.
- b) Determine cuántos fallos de página se producen en el proceso P3 si la secuencia de petición de páginas es la siguiente:

1 2 1 5 2 3 7 6 5 4 3 7 5 6 7 3 7 6 4 3 7

Compare los resultados de este apartado para las políticas FIFO y LRU.

**Solución:**

- a) Cuando los procesos P2 y P3 piden memoria el número de marcos libres es 6 pues 3 marcos están asignados al proceso P1. Como el reparto es proporcional al tamaño de los procesos, se tendrá:

Proceso	Tamaño	Factor asignación marcos	Marcos asignados
P2	1024	$1024/(1024+512) = 0,6666$	$0,6666 * 6 = 4$
P3	512	$512/(1024+512) = 0,3333$	$0,3333 * 6 = 2$

Es decir, 2/3 de los marcos se le asignan a P2 y 1/3 a P3.

Cuando el proceso P1 termina se reparten sus 3 marcos entre los procesos P2 y P3.

Proceso	Factor asignación marcos	Nuevos Marcos	Total Marcos
P2	0,67	$0,6666 * 3 = 2$	6
P3	0,33	$0,3333 * 3 = 1$	3

- b) La secuencia de peticiones del proceso P3 se puede dividir en dos partes. La primera parte está formada por la secuencia de peticiones anteriores a la primera petición de la página 3, ya que durante este periodo P3 tiene asignados 2 marcos de página. La segunda parte está formada por el resto de la secuencia y durante este tiempo el proceso P3 tiene asignados 3 marcos de página.

Para FIFO

	1	2	1	5	2	3	7	6	5	4	3	7	5	6	7	3	7	6	4	3	7
X	1	2	2	5	5	3	7	6	5	4	3	7	5	6	6	3	7	7	4	4	4
X		1	1	2	2	5	3	7	6	5	4	3	7	5	5	6	3	3	7	7	7
						2	5	3	7	6	5	4	3	7	7	5	6	6	3	3	3
	F	F		F		F	F	F	F	F	F	F	F	F		F	F		F		

En total 15 fallos de páginas.

Para LRU

	1	2	1	5	2	3	7	6	5	4	3	7	5	6	7	3	7	6	4	3	7
X	1	2	1	5	2	3	7	6	5	4	3	7	5	6	7	3	7	6	4	3	7
X		1	2	1	5	2	3	7	6	5	4	3	7	5	6	7	3	7	6	4	3
						5	2	3	7	6	5	4	3	7	5	6	6	3	7	6	4
	F	F		F	F	F	F	F	F	F	F	F	F	F		F		F	F	F	

En total 17 fallos de páginas.

6.- Supongamos un sistema de gestión de memoria virtual con paginación, en el que se utiliza como algoritmo de reemplazo el LRU. Existe un proceso al que se le asignan 4 marcos durante toda su ejecución y que hace referencia a la siguiente lista de páginas: 4 8 9 7 3 8 4 8 4 6 8

**a) Una vez cargadas las cuatro primeras páginas en memoria, tras la referencia al resto de las páginas de la lista se producirán 3 fallos de página.**

b) Si después de la lista anterior se necesita la página 3 se producirá un fallo de página.

c) Si después de la lista anterior se necesita la página 7 se expulsará a la página 4.

d) Ninguna de las afirmaciones anteriores es cierta.

7.- Cuál de las siguientes afirmaciones es correcta:

a) Según el modelo del conjunto de trabajo, las referencias de los programas tienden a agruparse en amplias zonas del espacio de direcciones.

b) Siempre que se produce un fallo de página se generan dos operaciones de E/S, una para guardar la página a expulsar y otra para cargar la página referida.

c) La página que se sustituye con el algoritmo del reloj o segunda oportunidad es la que lleva mas tiempo en memoria.

**d) Ninguna de las respuestas anteriores es correcta.**

2.- (2 puntos). Considere un sistema de gestión de memoria virtual mediante paginación bajo demanda en el que se han medido estos tiempos:

- Tiempo medio de acceso a memoria principal: 50 nanosegundos.
- Tiempo medio de resolución de un fallo de página: 20 milisegundos.
- El resto de los tiempos se consideran despreciables

Calcule cuál es la máxima tasa de fallos de página aceptable si queremos mantener el tiempo medio de acceso a memoria (contando con los fallos de página) por debajo de los 200 nanosegundos.

**Solución:**

El tiempo medio de acceso viene dado por:

$$t_{pa} = (1-p) \times a_m + p \times f_p$$

donde  $a_m$  es el tiempo de acceso a memoria, que será: 50 nanoseg.

$f_p$  es el tiempo que lleva resolver un fallo: 20 miliseg.

$p$  es la probabilidad que ocurra un fallo.

Luego nos quedará:

$$(1-p) \times 50 \text{ nanoseg} + p \times 20 \text{ miliseg} < 200 \text{ nanoseg}$$

Hay que resolver esta desigualdad

$$20 \times 10^{-3} p + 50 \times 10^{-9} (1-p) < 200 \times 10^{-9}$$

$$20 \times 10^6 p + 50(1-p) < 200$$

$$(20 \times 10^6 - 50)p < 150$$

$$p < \frac{150}{20 \times 10^6 - 50} \approx 7.5 \times 10^{-6}$$

8.- Para la siguiente cadena de referencia: 8,1,2,3,1,4,1,5,3,4,1,4. Suponiendo que se disponen de 3 marcos de página, el algoritmo de sustitución óptimo presenta:

a) 10 fallos de página

b) 0 fallos de página

**c) 7 fallos de página**

d) 5 fallos de página

2.- (3 puntos) En un sistema con gestión de memoria virtual por demanda de páginas, el tamaño de la página es de 1 Kb y el sistema posee 64 Kb de memoria física disponible para programas de usuario. En un determinado momento un programa de usuario que ocupa 9 páginas se carga para su ejecución. Considerando que en ese momento es el único proceso en ejecución, y que inicialmente se cargan las páginas 0, 4, 5 y 8 en los marcos 9, 3, 8 y 5 respectivamente.

a) Dibujar la tabla de páginas para esta situación.

b) Calcular la dirección física para las direcciones virtuales (2,50) y (5,20). Explicar el proceso de traducción de direcciones.

c) Con una política de reemplazo de páginas global, y partiendo de la situación inicial indicada, calcular los fallos de página que se producen con el algoritmo LRU para la siguiente cadena de referencia:

7 5 6 1 0 8 3 4 3 3 1 2 8 6 2 3 5 3 4

d) Calcular los fallos de página para la misma cadena de referencia, pero considerando que sólo se dispone de 6 marcos de página para este proceso (considerar que el orden de carga de páginas inicial fue 0, 4, 5 y 8)

a)

Nº Página	Nº Marco físico	Bit de presente/ausente
0	9	1
1	-	0
2	-	0
3	-	0
4	3	1
5	8	1
6	-	0
7	-	0
8	5	1

b) Dirección virtual (2,50): Al acceder a la tabla indica su bit de presente/ausente indica que no está, luego se da un fallo de página y se debe de iniciar el proceso de carga de página.

Dirección virtual (5,20), su bit de presente/ausente indica que está y según la tabla se encuentra en el marco físico 8, la dirección física inicial del marco será  $8 \times 1024 = 8192$ , y el desplazamiento es 20, luego la dirección será  $8192 + 20 = 8212$ .

c) Inicialmente se tienen ya ocupados 4 marcos, pero el sistema tiene aún libres 60 marcos, (64 Kb de memoria con páginas de 1 Kb dan 64 marcos, de los que inicialmente sólo están 4 ocupados). Luego sólo se producirán 5 fallos de página que corresponden a las referencias a una nueva página.

d) En esta situación, se ha supuesto que todos los marcos están ocupados excepto 5, que son los que puede ocupar este proceso

0	4	5	8	7	5	6	1	0	8	3	4	3	3	1	2	8	6	2	3	5	3	4
	0	4	5	8	7	5	6	1	0	8	3	4	4	3	1	2	8	6	2	3	5	3
		0	4	5	8	7	5	6	1	0	8	8	8	4	3	1	2	8	6	2	2	5
			0	4	4	8	7	5	6	1	0	0	0	8	4	3	1	1	8	6	6	2
				0	0	4	8	7	5	6	1	1	1	0	8	4	3	3	1	8	8	6
					0	4	8	7	5	6	6	6	6	0	0	4	4	4	4	1	1	8
f	f	f	f	F		F	F	F		F	F				F		F			F		F

En las cuatro primeras páginas se habían producido 4 fallos, y a partir de entonces 10 fallos.

2.- (3 puntos) Supongamos que tenemos una máquina con 16 MB de memoria principal y un esquema de gestión de memoria virtual paginado con páginas de 4 KB. Un proceso produce la siguiente secuencia de accesos a direcciones de memoria (mostradas aquí en hexadecimal):

02D4B8, 02D4B9, 02D4EB, 02D4EB, 02D86F, F0B621, F0B815, F0D963, F0B832, F0BA23, D9D6C3, D9B1A7, D9B1A1, F0BA25, 02D4C7, 628A31, F0B328, D9B325, D73425.

El sistema operativo asigna al proceso 4 marcos de memoria principal. Se pide:

- Indicar cuál es el formato de una dirección física de memoria.
- Dar la cadena de referencias de las páginas accedidas por el proceso.
- Si el sistema operativo utiliza 4 marcos de memoria principal, describir el comportamiento del gestor de memoria utilizando cada uno de los siguientes algoritmos de reemplazo de páginas, indicando cuántos fallos de página se producen con los algoritmos FIFO y de la segunda oportunidad.

Solución:

a) A partir del tamaño y organización de la memoria, se calcula el formato de una dirección física:

16 MB =  $2^{24}$  bytes, es decir, se requieren 24 bits para direccionar la memoria física. Como además se nos indica que el tamaño de página es de 4 KB, tendremos que  
 4 KB =  $2^{12}$  bytes, por lo que se necesitan 12 bits para direccionar un byte de una página.

Por tanto, el formato de una dirección física de memoria será:

página	desplazamiento
12 bits	12 bits

b) Dado que la cadena de referencias viene dada en hexadecimal, y para expresar un número hexadecimal se necesitan 4 bits en binario, para direccionar la página se necesitan 3 dígitos hexadecimales y otros 3 para el desplazamiento. Por tanto la dirección física queda expresada según el siguiente formato:

página			desplazamiento		
12 bits			12 bits		
Hex	Hex	Hex	Hex	Hex	Hex
0	2	D	4	B	8

Así, la secuencia de páginas que formará la cadena de referencias es:

02D, F0B, F0D, F0B, D9D, D9B, F0B, 02D, 628, F0B, D9B, D73

c) Algoritmo FIFO

02D	F0B	F0D	F0B	D9D	D9B	F0B	02D	628	F0B	D9B	D73
02D	02D	02D		02D	F0B		F0D	D9D	D9B		02D
	F0B	F0B		F0B	F0D		D9D	D9B	02D		628
		F0D		F0D	D9D		D9B	02D	628		F0B
				D9D	D9B		02D	628	F0B		D73
F	F	F		F	F		F	F	F		F

En total se producen 9 fallos de página.

Algoritmo Segunda Oportunidad

02D	F0B	F0D	F0B	D9D	D9B	F0B	02D	628	F0B	D9B	D73
02D <sub>1</sub>	02D <sub>1</sub>	02D <sub>1</sub>		02D <sub>1</sub>	D9B <sub>1</sub>		D73 <sub>1</sub>				
	F0B <sub>1</sub>	F0B <sub>1</sub>		F0B <sub>1</sub>	F0B <sub>0</sub>	F0B <sub>1</sub>	F0B <sub>0</sub>	F0B <sub>0</sub>	F0B <sub>1</sub>		F0B <sub>0</sub>
		F0D <sub>1</sub>		F0D <sub>1</sub>	F0D <sub>0</sub>	F0D <sub>0</sub>	02D <sub>1</sub>	02D <sub>1</sub>	02D <sub>1</sub>		02D <sub>0</sub>
				D9D <sub>1</sub>	D9D <sub>0</sub>	D9D <sub>0</sub>	D9D <sub>0</sub>	628 <sub>1</sub>	628 <sub>1</sub>		628 <sub>0</sub>
F	F	F		F	F		F	F			F

En total se producen 8 fallos de página.

1.- (3 puntos) Un sistema con memoria virtual mediante demanda de páginas utiliza el algoritmo LRU para la sustitución de páginas. Un proceso genera la siguiente secuencia de referencias a páginas de memoria:

1 3 2 4 1 5 7 4 3 2 8 9 4 5 4 9 1 8 3 2

- (2 puntos) Determinar cuantos fallos de página se producen cuando se dispone de 4 o 5 marcos de página para este proceso.
- (1 punto) Explicar razonadamente si mejoraría la tasa de fallos de página si se aumentase el número de marcos de página a  $N$ , siendo  $N > 5$ .

**Solución:**

a) El algoritmo LRU asocia a cada página el tiempo de la última vez que se utilizó. Cuando una página debe ser sustituida, se elige a aquella que no ha sido utilizada durante un periodo mayor de tiempo. Una posible forma de implementar este algoritmo es mediante una pila que mantiene los números de las páginas, cada vez que una página se referencia, su número se elimina de la pila y se coloca en la cumbre de la pila. De esta forma, en la parte superior de la pila se tiene siempre el número de la última página usada y en el fondo el de la página que hace más tiempo que se usó.

A continuación, se muestra el contenido de dicha pila para la secuencia de referencias a páginas de memoria dadas en el enunciado. Asimismo, cuando se produce un fallo se marca con una F y cuando se produce un acierto se indica con una A.

■ 4 marcos de página:

1	3	2	4	1	5	7	4	3	2	8	9	4	5	4	9	1	8	3	2
1	3	2	4	1	5	7	4	3	2	8	9	4	5	4	9	1	8	3	2
	1	3	2	4	1	5	7	4	3	2	8	9	4	5	4	9	1	8	3
		1	3	2	4	1	5	7	4	3	2	8	9	9	5	4	9	1	8
			1	3	2	4	1	5	7	4	3	2	8	8	8	5	4	9	1
F	F	F	F	A	F	F	A	F	F	F	F	F	F	A	A	F	F	F	F

En total se producen **16 fallos de página**.

■ 5 marcos de página:

1	3	2	4	1	5	7	4	3	2	8	9	4	5	4	9	1	8	3	2
1	3	2	4	1	5	7	4	3	2	8	9	4	5	4	9	1	8	3	2
	1	3	2	4	1	5	7	4	3	2	8	9	4	5	4	9	1	8	3
		1	3	2	4	1	5	7	4	3	2	8	9	9	5	4	9	1	8
			1	3	2	4	1	5	7	4	3	2	8	8	8	5	4	9	1
					3	2	2	1	5	7	4	3	2	2	2	8	5	4	9
F	F	F	F	A	F	F	A	F	F	F	F	A	F	A	A	F	A	F	F

En total se producen **14 fallos de página**.

b) Analizando la secuencia de páginas referenciadas se observa que existen 8 páginas distintas {1 2 3 4 5 7 8 9}, por lo tanto habrán 8 fallos de página como mínimo, al estar la memoria inicialmente vacía. Como con 5 marcos de página se producen 14 fallos de página todavía existe un margen de mejora en la tasa de fallos de página (pasar de 14 a 8) si se aumenta el número de marcos de página  $N$  por encima de 5.

---

2.- En un sistema de gestión de la memoria con particiones fijas se dispone de 7 particiones de 1 Mb y la cola de tareas contiene tareas con requerimientos de 400 Kb, 1600 Kb, 300 Kb, 900 Kb, 200 Kb, 500 Kb y 800 Kb, Decir si las siguientes afirmaciones son ciertas:

I) La fragmentación externa es de 1600 Kb.

II) La fragmentación interna es de 3040 Kb

a) I: si, II: si.

b) I: si, II: no.

c) I: no, II: si.

d) I: no, II: no

---

**Solución:** pp. 218 del libro base de la asignatura.

Se dispone de siete particiones de 1 Mb, es decir, 1024 Kb. Se va a calcular el espacio desperdiciado en cada partición al alojar cada una de las tareas de la cola:

- Tarea de 400 Kb. Espacio desperdiciado  $1024-400=624$  Kb.
- Tarea de 1600 Kb. Su tamaño es mayor que el tamaño de una partición por lo tanto no podrá ejecutarse nunca.
- Tarea de 300 Kb. Espacio desperdiciado  $1024-300=724$  Kb.
- Tarea de 900 Kb. Espacio desperdiciado  $1024-900=124$  Kb.
- Tarea de 200 Kb. Espacio desperdiciado  $1024-200=824$  Kb.
- Tarea de 500 Kb. Espacio desperdiciado  $1024-500=524$  Kb.
- Tarea de 800 Kb. Espacio desperdiciado  $1024-800=224$  Kb.

La *fragmentación externa* se produce cuando una partición disponible no se emplea porque es muy pequeña para cualquiera de las tareas que se esperan. En este caso, una partición completa de las siete disponible ha quedado vacía luego la fragmentación externa es de 1024 Kb.

La *fragmentación interna* consiste en aquella parte de la memoria que no se está usando pero que es interna a una partición asignada a una tarea. En este caso la fragmentación interna es de  $624+724+124+824+524+224=3044$  Kb.

Respuesta: **D) I: no, II: no.**

2.- (3 puntos) En un computador con una capacidad de memoria principal de 64 Kpalabras se utiliza gestión de memoria mediante segmentación. La tabla de segmentos (todos los datos numéricos están en decimal) es la siguiente:

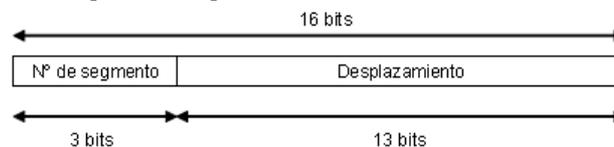
Nº de segmento	Base	Longitud
0	0	7230
1	16384	8191
2	32768	1024
3	8192	356
4	24576	4200

Se pide:

- a) (1 punto) Supuesto que una dirección lógica tiene el mismo tamaño en bits que una dirección física y que consta de los campos [nº de segmento, desplazamiento], determinar el tamaño en bits de cada uno de estos campos.
- b) (2 puntos) Determinar a que direcciones físicas expresadas en decimal corresponden las siguientes direcciones lógicas expresadas en hexadecimal: i)  $11AE_{16}$ , ii)  $6190_{16}$

**Solución:**

a) El tamaño de una dirección de memoria física se puede obtener del dato de la capacidad de la memoria principal que es 64 Kpalabras, o equivalentemente,  $2^{16}$  palabras. Luego se requiere 16 bits para codificar todas las palabras de la memoria, es decir, el tamaño de una dirección de memoria es de 16 bits. Por otra parte en la tabla de segmentos se observa que la memoria principal está dividida en 5 segmentos, luego se requerirán 3 bits para codificarlos, es decir, el tamaño del campo [nº de segmento] es de 3 bits. Finalmente, el tamaño del campo [desplazamiento] se determina como la diferencia entre el tamaño de una dirección y el tamaño del campo [nº de segmento], es decir,  $16-3=13$  bits. Luego el formato de una dirección lógica es el siguiente:



b) i) Hay que pasar la dirección  $11AE$  a binario: 0001 0001 1010 1110. Se observa que de acuerdo con el formato de una dirección lógica, los tres bits más significativos 000 hacen referencia al nº de segmento mientras que los trece bits restantes 1000110101110 hacen referencia al desplazamiento. Pasando estos campos a decimal se obtiene:

$$\text{Nº de segmento} = 000_2 = 0_{10}$$

$$\text{Desplazamiento} = 1000110101110_2 = 2^{12} + 2^8 + 2^7 + 2^6 + 2^3 + 2^2 + 2 = 4096 + 256 + 128 + 32 + 8 + 4 + 2 = 4526_{10}$$

A continuación, hay que comprobar que la dirección lógica es válida, para ello se compara el desplazamiento de esta dirección con la longitud del segmento nº 0 dada en la tabla de segmentos. Puesto que  $4526 \leq 7230$  la dirección lógica es válida.

La dirección física se obtiene sumando la base del segmento nº 0 con el desplazamiento de la dirección lógica, es decir,  $0 + 4526 = 4526$ .

Luego la dirección lógica  $11AE_{16} = (0, 4526)_{10}$  equivale a la dirección física 4526.

b) ii) Hay que pasar la dirección  $6190$  a binario: 0110 0001 1001 0000. Se observa que de acuerdo con el formato de una dirección lógica, los tres bits más significativos 011 hacen referencia al nº de segmento mientras que los trece bits restantes 0000110010000 hacen referencia al desplazamiento. Pasando estos campos a decimal se obtiene:

$$\text{Nº de segmento} = 011_2 = 3_{10}$$

$$\text{Desplazamiento} = 0000110010000_2 = 2^8 + 2^7 + 2^4 = 256 + 128 + 16 = 400_{10}$$

A continuación, hay que comprobar que la dirección lógica es válida, para ello se compara el desplazamiento de esta dirección con la longitud del segmento nº 3 dada en la tabla de segmentos. Puesto que  $400 > 356$  se tiene un error de direccionamiento ya que se está violando el tamaño del segmento.