

Ejercicios de Memoria

1) Considere el espacio de direccionamiento lógico de 8 páginas de 1024 bytes cada una, mapeado en una memoria física de 32 frames.

- ¿Cuántos bits hay en la dirección lógica?
- ¿Cuántos bits hay en la dirección física?
- En un esquema de paginación sin memoria virtual, ¿Como tendría que ser la relación entre los tamaños de las direcciones calculadas anteriormente? (dirección lógica *menor, igual o mayor* que dirección física)

2) Dado un sistema con una administración de memoria paginada simple (o sea, sin memoria virtual), con 256 KiB de memoria real, 20 bits de direccionamiento lógico y páginas de 4 KiB, se pide:

- Tamaño máximo posible de un programa a ejecutar en el sistema (ignorando el espacio ocupado por el SO).
- Cantidad máxima en bytes de fragmentación interna y externa que podría existir.

3) Dada la siguiente tabla de segmentos, determine las direcciones físicas:

Segmento	Base	Largo	Dirección Lógica	Dirección Física
0	219	600	0, 430	
1	2300	14	1, 10	
2	90	100	2, 500	
3	1327	580		

4) Considere una máquina con direcciones de 18 bits. Los primeros 2 bits se utilizan para identificar el segmento, y los últimos 16, el offset dentro del segmento. Asumir que las bases y los límites de la tabla de segmentos están configuradas de la siguiente forma:

Segmento	Base	Largo	Protección
0	00000h	0ABCDh	Read-only
1	1B000h	007FFh	Exec-only
2	1B800h	00FFFh	Read-write
3	30000h	01234h	Read-write

Responder *en forma clara y concisa* que sucede cuando el proceso intenta **escribir** en cada una de las siguientes direcciones virtuales: 20000h, 10000h, 0BEEFh, 00ACEh.

5) En un esquema de memoria virtual, con una máquina que tiene 128 KiB de RAM y paginas de 8 KiB. ¿Cuál sería el tamaño mínimo (en bits) de dirección si queremos que un proceso pueda direccionar hasta 1 MiB?.

6) Suponga un esquema de memoria virtual con paginación. Se están ejecutando los programas A, B y C con longitudes totales de 2 KiB, 1.5 KiB y 3 KiB respectivamente. La longitud de la página es de 0.5 KiB.

- ¿Cuál es la cantidad máxima de páginas que podría tener un programa si las instrucciones tienen direcciones de 16 bits y la memoria física es de 32 KiB?
- Proponga un formato para las entradas de la tabla de páginas de un proceso y la tabla de administración de memoria libre/ocupada del sistema, suponiendo ahora que el tamaño de la memoria es 64Kb. ¿Que tamaño ocuparía en memoria la segunda tabla?
- Determinar el contenido de las tablas de páginas para los programas A, B y C, sabiendo que el estado de los primeros 9 frames de memoria es el siguiente:

0	1	2	3	4	5	6	7	8
A0	B0	C5	A1	A2	C3	C1	B2	C4

7) Un esquema de memoria virtual tiene un tamaño de página de 1024 bytes y la memoria física tiene 4 marcos de página. La Tabla de páginas de un proceso es:

Página Virtual	Marco
0	3
1	1
2	--
3	--
4	2
5	--
6	0
7	--

¿Cuáles son las direcciones físicas de las siguientes direcciones virtuales (expresadas en decimal): 0, 3728, 1024, 1025, 4099, 7800?

8) Un proceso de 8 páginas está ejecutando en un sistema de memoria virtual, con asignación fija de 4 marcos por proceso y alcance local. Si se considera que la memoria está inicialmente vacía, determinar el número de fallos de página que ocurren al referenciarse las siguientes páginas: **0, 1, 7, 2, 3, 2, 7, 1, 0, 3, 0, 2, 3, 1**; bajo los siguientes algoritmos:

- Óptimo, FIFO, LRU, Clock

- b) Proponga para el algoritmo Clock un formato para la entrada de la tabla de páginas, sabiendo que los procesos pueden leer/escribir/ejecutar siempre y cuando los permisos de la página lo permitan, con una memoria física de 64 KiB y frames de 2 KiB.

9) Se tiene una PC con procesador Intel de 32 bits de direccionamiento que acepta el uso de páginas de 8KB de tamaño.

Dicha PC cuenta, además, con un sistema operativo que utiliza paginación bajo demanda donde la política de asignación de frames es fija, siendo de 4 frames por proceso. La política de sustitución de páginas es local.

Se está ejecutando un proceso de 159KB de tamaño con la siguiente asignación actual de frames:

Puntero (si aplica)	Marco	Página (Numerada desde 0)	Uso	Modificado	Instante de referencia
	1	14	0	1	28
	5	17	0	0	3
	3	19	1	1	15
-->	8	--	—	--	--

A continuación se detallan las próximas referencias a memoria (con sus respectivos modos de acceso) que el proceso realizará:

100(Lectura) - 122950(Escritura) - 98306(Lectura) - 139264(Escritura) - 122880(Lectura) - 155650(Escritura) - 172100(Lectura) - 100(Lectura)

Se pide:

Para los algoritmos LRU y Clock modificado:

- Indicar el estado de las páginas en memoria luego de cada referencia, así como también los page faults producidos y las páginas que fueron escritas a disco. No es necesario calcular los instantes de referencia.
- Dejando a un lado el costo de ejecución e implementación de un algoritmo de sustitución. Indicar cuál de los dos algoritmos presenta un mejor rendimiento con la secuencia de referencias dada. ¿Qué criterio tiene en cuenta para tomar esa decisión?.

1)

a)

cantidad de páginas: $2^3 \rightarrow 3$ bits

tamaño de página: $2^{10} \text{ b} \rightarrow 10$ bits

total 13 bits

b)

cantidad de frames: $2^5 \rightarrow 5$ bits

tamaño de página: $2^{10} \text{ b} \rightarrow 10$ bits

c) igual o menor

2)

a)

256kb porque no usa Memoria Virtual

b)

interna 4095b

no hay fragmentación externa

3)

0,430 \rightarrow 679

1,10 \rightarrow 2310

2,500 \rightarrow segmentation fault

4)

20000h \rightarrow se escribe correctamente

10000h \rightarrow falla por permisos

0BEEFh \rightarrow Segmentation Fault

00ACEh \rightarrow falla por permisos

5)

20bits ($2^{20}b=1mb$)

6)

tamaño de página: $521b = 2^9b \rightarrow 9$ bits

a) $16-9=7 \rightarrow 2^7 = 128$ páginas

b)

tabla de páginas:

número de página	número de marco

tabla de administración de memoria libre/ocupado = bitvector de 128bits(1 por cada página) $\rightarrow 128$ bits
= 16 bytes

c)

A		
p	m	b
0	0	1
1	3	1
2	4	1
3		0

B		
p	m	b
0	1	1
1		0
2	7	1

C		
p	m	b
0		0
1	6	1
2		0
3	5	1
4	8	1
5	2	1

7)

Dirección lógica	Página	Marco	Offset	Dirección Física
0	0	3	0	3072
3728	3	--	656	Page fault
1024	1	1	0	1024
1025	1	1	1	1025
4099	4	2	3	2051
7800	7	--	632	Page fault

8)

óptimo

```

      2 2 2 2 2 2 2 2 2 2 2 2
    7 7 7 7 7 7 0 0 0 0 0 0
  1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 3 3 3 3 3 3 3 3
F F F F F      F
0 1 7 2 3 2 7 1 0 3 0 2 3 1

```

FIFO

```

      2 3 3 3 3 0 0 0 0 0 1
    7 7 2 2 2 2 3 3 3 3 0
  1 1 1 7 7 7 2 2 2 2 3
0 0 0 0 1 1 1 1 7 7 7 2
F F F F F      F      F
0 1 7 2 3 2 7 1 0 3 0 2 3 1

```

LRU

```

      2 3 2 7 1 0 3 0 2 3 1
    7 7 2 3 2 7 1 0 3 0 2 3
  1 1 1 7 7 3 2 7 1 1 3 0 2
0 0 0 0 1 1 1 3 2 7 7 1 1 0
F F F F F      F F      F
0 1 7 2 3 2 7 1 0 3 0 2 3 1

```

Clock

```

      1 1 2 1 3 0 3 0 3 0 3 0 0 1 3 0 3 0 3 0 3 1 3 1
      1 7 1 7 1 2 0 2 1 2 1 2 1 2 1 0 0 0 1 0 1 0 1 1
    1 1 1 1 1 1 7 0 7 0 7 1 7 1 7 1 2 0 2 0 2 1 2 1
0 1 0 1 0 1 0 1 0 1 0 1 0 1 1 1 1 7 0 7 0 7 0 1 1
F F F F F      F F      F
0 1 7 2 3 2 7 1 0 3 0 2 3 1

```

b)

Marco	Página	Bit de uso	puntero
-------	--------	------------	---------

9)

Dirección lógica	(página,offset)	Operación
100	(0,100)	Lectura
122950	(15,70)	Escritura
98306	(12,2)	Lectura
139264	(17,0)	Escritura
122880	(15,0)	Lectura
155650	(19,2)	Escritura
172100	(21,68)	Lectura
100	(0,100)	Lectura

a)LRU

Marco	Página	Instante de Referencia
1	14	28
5	17	3
3	19	15
8		

1	14	28
5	17	3
3	19	15
8	0	29

F

1	14	28
5	15	30
3	19	15
8	0	29

F

1	14	28
5	15	30
3	12	31
8	0	29

F

1	17	28
5	15	30
3	12	31
8	0	29

F

1	17	28
5	15	30
3	12	31
8	0	29

1	19	33
5	15	30
3	12	31
8	0	29

F

1	19	33
5	15	30
3	12	31
8	21	34

F

b)Clock modificado

Puntero	Marco	Página	Uso	Modificado
	1	14	0	1
	5	17	0	0
	3	19	1	1
-->	8			

	1	14	0	1
	5	17	0	0
	3	19	1	1
-->	8	0	1	0

F

	1	14	0	1
-->	5	15	1	1
	3	19	1	1
	8	0	1	0

F

-->	1	12	1	1
	5	15	1	1
	3	19	1	1
	8	0	1	0

F

	1	12	1	1
	5	15	1	1
	3	19	1	1
-->	8	0	1	1

F

	1	12	1	1
	5	15	1	1
	3	19	1	1
-->	8	0	1	1

	1	12	1	1
	5	15	1	1
	3	19	1	1
-->	8	0	1	1

-->	1	21	0	0
	5	15	0	1
	3	19	0	1
	8	0	0	1

F

b)

El clock modificado es mejor porque le da prioridad a reemplazar las páginas no modificadas para hacer menos pedidos de e/s y aparte con la secuencia dada tuvo menos page faults.