



Estructura de Computadores

PROBLEMAS DE MEMORIA VIRTUAL



1. Sea una memoria cache de emplazamiento directo virtualmente accedida físicamente marcada cuyo formato de dirección es el siguiente:

Etiqueta	Marco de bloque	Desplazamiento
63:10	9:4	3:0

- a) ¿Cuál es el tamaño de esta memoria cache?
- b) ¿Cuál es el tamaño de las páginas de la memoria virtual?
- c) ¿Cómo se podría doblar el tamaño de la memoria cache sin modificar el formato de la dirección?
2. Sea un computador con memoria virtual paginada, cada página tiene 1024 palabras, en memoria virtual tiene 8 páginas y la memoria física 4 marcos de página.
- a) Indicar el tamaño en bits de la dirección de memoria virtual y de la dirección de memoria física.
- b) Si la jerarquía tiene una memoria cache de 1024 palabras de acceso directo con 4 bloques de cache de direcciones virtuales, indicar qué tamaño en bits tiene la etiqueta de cache.
- c) ¿Qué valores de etiqueta de cache indicarían que algún bloque de la página virtual cero está en cache?

Página virtual	Validez	Marco de página
0	1	3
1	1	1
2	0	0
3	0	0
4	1	2
5	0	0
6	1	0
7	0	0

3. Sea un sistema de memoria con las siguientes características:

- Memoria virtual paginada de 16MB, política de emplazamiento LRU
- Tamaño de página 4KB
- Memoria principal con 32KB
- Memoria cache de direcciones físicas 8Kbytes, con bloques de 64 bytes, asociativa por conjuntos con 2 bloques/cjtos, política LRU.

Se desea ejecutar el siguiente código:

```
int A[16384];
```

```
for (i=0; i<16384; i++)
```

```
red+= A[i];
```

Suponer que A está almacenado consecutivamente a partir de la dirección virtual 0x100000 y que la variable **red** está almacenada en un registro y únicamente se lleva a memoria al finalizar la ejecución del bucle.

La tabla siguiente indica qué páginas virtuales se encuentran en este momento en memoria principal

Tabla de páginas inversa		
Nº página virtual (hex)	Nº de marco (página física)	LRU*
100	0	0
101	1	1
200	2	2
201	3	3
300	4	4
301	5	5
400	6	6
401	7	7

(*) 0 indica la página más recientemente usada, y 7 la menos recientemente usado.

- Indicar el formato de la dirección virtual y de la dirección física, esta última desde el punto de vista de la memoria virtual y de la memoria cache.
- ¿Cuántas páginas de memoria virtual se visitan? ¿Cuáles son?
- Razonar el número de aciertos y fallos de página que se producen en función de los valores que se muestran en la tabla de páginas anterior.

4. Sea un computador con una memoria virtual paginada y memoria cache con las siguientes características:

- Un procesador que genera direcciones de 20 bits
- Una memoria física de 128 KB y páginas de 32 KB. Reemplazamiento LRU
- Una memoria cache de datos de direcciones físicas de 1 KB. La memoria es de acceso directo con 256 B por bloque.

Sobre esta jerarquía se ejecuta una aplicación que selecciona aleatoriamente una canción dentro de una base de datos de canciones. Cada canción ocupa 8 KB y sus direcciones de comienzo están indicadas en la siguiente tabla:

Canción	1	2	3	4	5	6
Dirección	0x50000	0xF4000	0x66000	0x12000	0x40000	0x36000

- Indicar el formato de la dirección virtual y de la dirección física, esta última desde el punto de vista de la memoria virtual y de la memoria cache.
- Indicar en qué página (o páginas) virtuales está ubicada cada canción.
- Si en un momento dado el contenido de la TLB y de la memoria cache de datos es el indicado en las tablas que se muestran a continuación, ¿cuáles han sido las últimas canciones escuchadas? ¿Cuál es el rango de direcciones físicas de cada una de ellas?

Nº pag. virtual	Nº pag. física	Etiqueta	Bloque
0x0A	1	0x70	0
0x0C	2	0x27	1
0x02	0	0x27	2
0x1E	3	0x27	3

TLB CACHE

- Supongamos que a continuación se genera la siguiente cadena de referencias virtuales: 0xF40F5, 0x66000, 0x40040 y 0x51D04. Indicar si se producen fallos o aciertos de cache y fallos de página, mostrando cómo evolucionan los contenidos de la memoria principal y de la memoria cache

5. Sea un computador con memoria virtual paginada y memoria cache con las características siguientes:

- Memoria virtual de 32 páginas de 8 KB cada una, con traducción asociativa y reemplazamiento LRU
- Memoria física de 32 KB
- Memoria cache de direcciones físicas de 512 bytes, asociativa por conjuntos, con bloques de 128 bytes, 2 bloques por conjunto y reemplazamiento LRU.
- La política de actualización de la memoria principal es escritura directa sin asignación en escritura
 - Indicar el formato de la dirección virtual y de la dirección física, esta última desde el punto de vista de la memoria virtual y de la memoria cache.
 - Si en un momento dado los contenidos de la tabla de páginas y de la cache son los siguientes:

Tabla de páginas inversa		Cache		
Nº página virtual	Nº de marco (página física)	Etiqueta (hex)	Conjunto	Vía
06	3	23	0	0
01	0	47	0	1
04	2	73	1	0
0C	1	08	1	1

Expresar en hexadecimal el rango de direcciones virtuales y físicas ubicadas en cada marco de página de la memoria principal, así como el rango de direcciones virtuales y físicas ubicadas en cada bloque de la memoria cache.

- c) Supongamos que un programa realiza la siguiente cadena de referencias virtuales (en hexadecimal): 0x08770, 0x02080-0x0209F, 0x02880-0x0289F, 0x0D3F0-0x0D410, 0x27000-0x2701F. Suponiendo que inicialmente los contenidos de la cache son los que se muestran en la tabla anterior, indicar claramente cómo evolucionan los contenidos de la memoria principal y de la memoria cache (indicando las direcciones virtuales que se ubican en cada bloque y los cambios en las etiquetas), cuando se realiza la cadena de referencias anterior.

6. Sea un sistema de memoria con las siguientes características:

- Memoria virtual paginada de 4GB, política de emplazamiento LRU
- Tamaño de página 2KB
- Memoria principal con 13 bits para la dirección
- Memoria cache de direcciones físicas 2048 bytes, con bloques de 256 bytes, asociativa por conjuntos con 4 bloques/cjto, política LRU

- a) Indicar el formato de la dirección virtual y de la dirección física, esta última desde el punto de vista de la memoria virtual y de la memoria cache.
- b) Se desea acceder a las siguientes direcciones: 0x55118FF8; 0x0A000000 y 0x11281276. Indicar si se produce algún fallo en la gestión de la memoria y los valores de las tablas al finalizar el acceso a cada una de las direcciones. Además indicar en los casos de fallo de página los bloques de memoria cache que se invalidarían.
- c) Si el computador accede a la dirección 0xFFA34500 indicar el rango de direcciones virtuales y físicas correspondientes a su página y a su marco de bloque en memoria cache. Las tablas se mantienen con los valores obtenidos a la finalización del apartado b.

Nº de página	Marco	Edad
0x000001	0	0
0x1B0000	1	1
0x0AA231	2	3
0x100000	3	2

Conjunto	M. Bloque	Etiqueta	Edad
0	0	F	0
	1	0	2
	2	B	3
	3	1	1
1	0	A	1
	1	3	3
	2	1	2
	3	B	0

7. En una empresa dedicada al análisis de datos tienen que utilizar un algoritmo que requiere sumar todos los elementos de cada columna de una matriz de 512x512 elementos enteros. El código C de dicho algoritmo es el siguiente:

```
for (j = 0; j < 512; j++)
    for (i = 0; i < 512; i++)
        red[j] = red[j] + A[i][j];
```

Sabiendo que:

- En C los elementos de una matriz se emplazan en memoria consecutivos por filas
- El procesador es de 32 bits (direcciones virtuales de 32 bits).
- El array de enteros A[512][512] se ubica en el mapa virtual de memoria del proceso a partir de la dirección 0x00458000, y el array de enteros red[512] se ubica justo a continuación de A.
- El sistema operativo (SO) maneja páginas de 4KB con reemplazo LRU. se pide:

- a) Indicar qué páginas de memoria virtual ocupan los arrays *A* y *red*, indicando qué elementos de cada array están en cada página.
- b) El SO le asigna al proceso 8 marcos de página (lo que sería equivalente a decir que, desde el punto de vista de este proceso, la memoria física tiene 8 páginas). Sabiendo que 2 de esos 8 marcos están ocupados por la página de código y la página de pila, accedidos constantemente y que por tanto nunca son elegidos para el reemplazo, y que los 6 restantes no contienen inicialmente ninguna página de *A* ni de *red*; indicar razonadamente cuántos fallos de página se producirán al ejecutar el código y el número de accesos a *A* y *red* que no suponen fallo de página.
- c) Razone si es posible o no reducir significativamente los fallos de página del proceso con alguna modificación en su código, indicando en su caso qué transformación se haría y cuantificando la mejora esperada.

8. En una máquina con una memoria DRAM de 256MB y un procesador de 32 bits (direcciones virtuales de 32 bits) el sistema

operativo gestiona la memoria virtual por paginación, utilizando páginas de 4KB y con algoritmo de reemplazamiento LRU. En dicha máquina se ejecuta una aplicación que accede a distintas estructuras de datos, cada una de las cuales ocupa 1KB. El proceso dispone de los marcos de página 6, 7, 8 y 9 para almacenar el código y los datos (se asignan en ese orden, esto es, la primera página que necesite el proceso se alojará en el marco 6). Se dispone de una TLB de dos entradas totalmente asociativa.

Las direcciones virtuales iniciales de cada una de las estructuras de datos son:

Nombre E. D.	A	B	C	D	E	F
D. V. inicial	0x00128000	0x00941600	0x00002000	0x005110A0	0x00300100	0x00041000

El código empieza en la dirección virtual 0x000C0000 y ocupa 3500B. Tener en cuenta que cada vez que se ejecuta una instrucción se accede a la página de código que almacena esta instrucción.

Se pide:

a) Mostrar el estado final de la tabla de páginas y la TLB cuando la aplicación se ejecuta y accede a las estructuras de datos en el siguiente orden: A, B, A, C, D, A, E, B, A.

b) Si a continuación el proceso intenta acceder a F, indique las estructuras que se consultarían, el orden en que se haría y si habría fallo o acierto en cada caso. En caso de que se produjeran fallos indique cómo quedaría cada una de las estructuras tras resolver el fallo.

c) Si la memoria cache de datos fuera virtualmente accedida físicamente marcada, ¿cuál sería su tamaño máximo para cache asociativa por conjuntos con dos vías?

9. En una máquina con una memoria principal de 4 KB el sistema operativo gestiona la memoria virtual por paginación, utilizando páginas de 1KB y un algoritmo LRU de reemplazo de páginas. Siendo el tamaño de la memoria virtual 8KB, se pide:

a) Completar toda la información posible de la tabla de páginas conociendo el contenido de la TLB, y sabiendo que las últimas referencias a memoria han sido: 0x144C, 0x1F04, 0x0750, 0x008C.

Tabla de páginas			
Nº Pág. Virtual	Validez	LRU	Nº Pág. Física
0			
1			
2			0
3			
4			1
5			2
6			2
7			

TLB	
Nº Pág. Virtual	Nº Pág. Física
0	3
1	1

- b) Si se accede a las referencias de memoria 0x0F6C, 0x120C, 0x0478. Indica cómo queda la tabla de páginas y la TLB.
- c) Se desea añadir al sistema una memoria cache de 8 bloques de 256 Bytes. ¿Qué tipo de emplazamiento podría usarse para que fuera virtualmente accedida físicamente marcada?
- d) En un determinado instante, tras referenciar las direcciones comprendidas en el rango 0x1C00 – 0x1FFF la TLB contiene la siguiente información:

Nº Pág. Virtual	Nº Pág. Física
7	3
4	1

A continuación, se referencia una nueva posición de memoria, cuya dirección se quiere averiguar. Se dispone de una memoria cache de emplazamiento directo de 1 KB y bloques de 256 KB, cuyo contenido tras este acceso es el siguiente:

Marco	Etiqueta
0	11
1	01
2	11
3	11

Indica en qué rango de direcciones físicas y virtuales debe encontrarse la última dirección de memoria accedida