

Problemas de Estructura de Computadores II

Basilio B. Fraguera

Date: 2007/02/01 15:44:52 V. Revision: 1.2

1. Memorias entrelazadas

1.1 Dada una memoria entrelazada de orden inferior de 4 módulos que debe remitir los resultados de las lecturas en el orden en que han sido solicitados por el procesador y que para ello tiene 16 buffers que permiten guardar los resultados de las lecturas de los módulos hasta el momento en que deben ser enviados al procesador, hágase un estudio del tiempo que tarda en acceder a las posiciones de la secuencia 0, 1, 2, 4, 5, 8, 0, 1, 3, 4, 5, 1 y 2 tanto para el caso de que los módulos tengan latches a la entrada como a la salida.

Datos temporales:

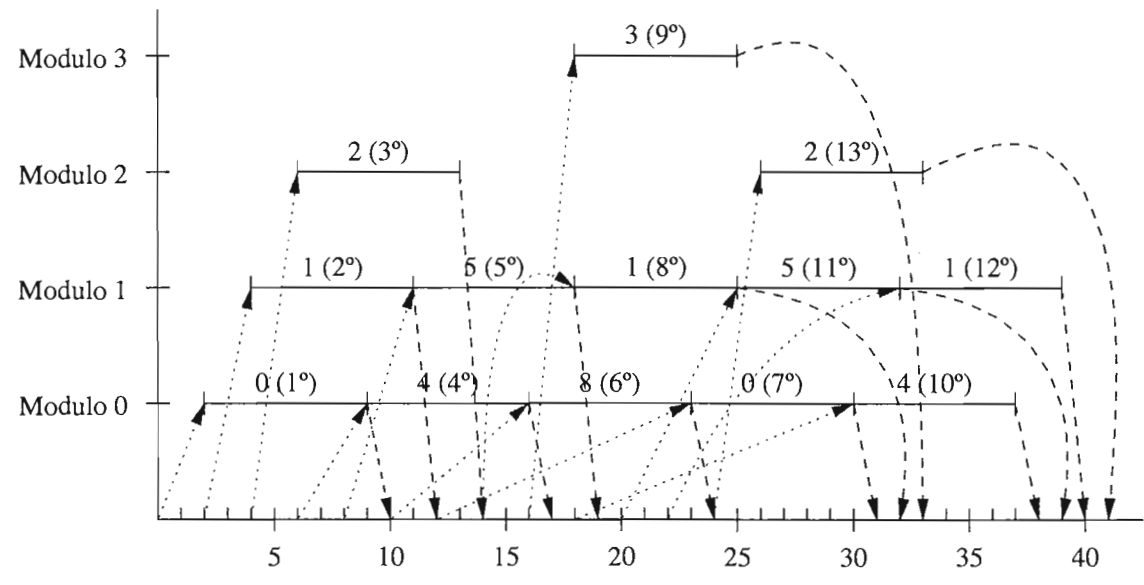
$T_{\text{acceso a un módulo}} = 7$ unidades

$T_{\text{referencia}} = 2$ unidades

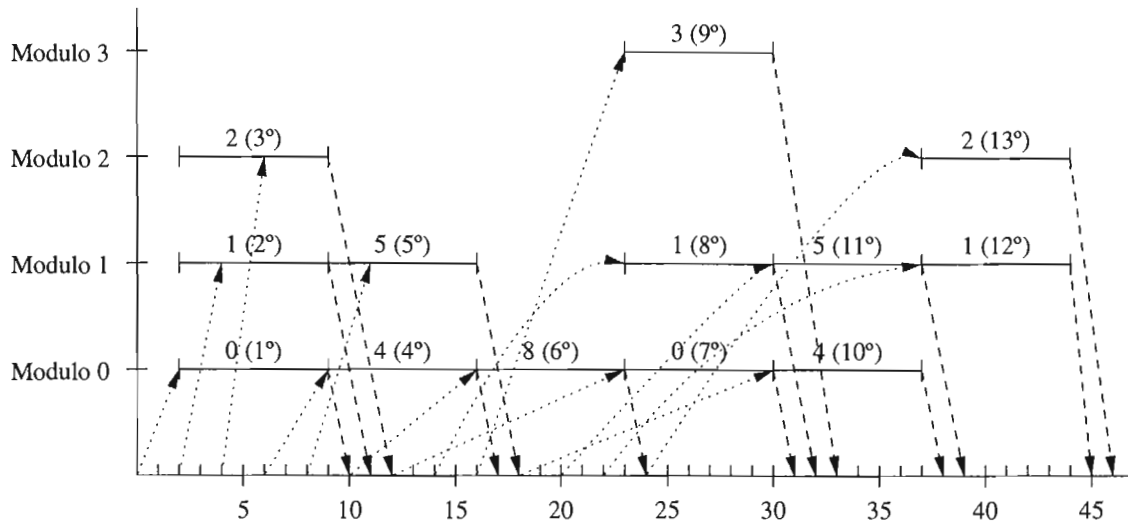
$T_{\text{salida}} = 1$ unidad

Solución:

a) Con latches a la entrada requiere 41 ciclos:



b) Con latches a la salida requiere 46 ciclos:



2. Cachés

2.1 Considerando los datos de la tabla adjunta correspondientes a cachés de correspondencia directa con un tamaño de bloque de 32 bytes, tomados sobre un conjunto de programas de prueba en los que el porcentaje de referencias a instrucciones es del 75 % responde a las siguientes cuestiones:

- ¿Qué sistema presenta una menor razón de fallos, el constituido por una caché de instrucciones de 16 KB y una caché de datos de 16 KB, o el constituido por una caché de 32 KB unificada?
- Supóngase que un acierto caché requiere un ciclo de reloj, un fallo tiene un costo de 50 ciclos, y que un acierto en una instrucción load o store requiere un ciclo extra en el caso de la caché unificada, ya que sólo existe un puerto de acceso a esta caché para satisfacer dos peticiones simultáneas. ¿Cuál es el tiempo medio de acceso a memoria en cada caso?

Tamaño	Caché de instrucciones	Caché de datos	Caché unificada
1 KB	3.06 %	24.61 %	13.34 %
2 KB	2.26 %	20.57 %	9.78 %
4 KB	1.78 %	15.94 %	7.24 %
8 KB	1.10 %	10.19 %	4.57 %
16 KB	0.64 %	6.47 %	2.87 %
32 KB	0.39 %	4.82 %	1.99 %

Solución :

- Dado que el 75 % de los accesos son referencias a instrucciones, el otro 25 % serán accesos a datos, con lo que la razón de fallos conjunta para una la caché separada será

$$75 \% \times 0,64 \% + 25 \% \times 6,47 \% = 2,10 \%$$

tomando los datos de tasas de fallos para una caché de instrucciones y de datos de 16 KB cada una separadas.

La tabla indica que la tasa de fallos de la caché unificada de 32 KB es el 1.99%, lo cual es ligeramente inferior.

- b) El tiempo medio de acceso a memoria puede dividirse en términos, el correspondiente a instrucciones y a datos:

$$\begin{aligned} \text{Tiempo medio de acceso a memoria} = & \\ & \text{Porcentaje de accesos a instrucciones} \times \\ & (\text{Tiempo de acierto} + \text{Tasa de Fallos} \times \text{Penalización de fallos}) + \\ & \text{Porcentaje de accesos a datos} \times \\ & (\text{Tiempo de acierto} + \text{Tasa de Fallos} \times \text{Penalización de fallos}) \end{aligned}$$

El tiempo de acceso medio a memoria para la caché dividida es:

$$75\% \times (1 + 0,64\% \times 50) + 25\% \times (1 + 6,47\% \times 50) = 2,05$$

El tiempo de acceso medio a memoria para la caché unificada es:

$$75\% \times (1 + 1,99\% \times 50) + 25\% \times (1 + 1 + 1,99\% \times 50) = 2,24$$

Así pues, aunque la tasa de fallos del sistema con caché separadas sea superior al de la caché unificada, su tiempo medio de acceso a memoria es menor que el de la caché unificada.

- 2.2 La tabla adjunta muestra diferentes porcentajes de la razón de fallos de una caché en función de su tamaño y el tamaño de línea elegidos (expresados en KBytes y bytes, respectivamente). Considera que el sistema de memoria requiere 40 ciclos de reloj de iniciación y a continuación es capaz de proporcionar 16 bytes cada dos ciclos de reloj. Es decir, proporciona 16 bytes en 42 ciclos de reloj, 32 bytes en 44 ciclos de reloj, y así sucesivamente. Determina qué tamaño de bloque posee el mínimo tiempo medio de acceso a memoria para las cachés de tamaños 4K, 16K y 64K, respectivamente.

Tamaño de línea	Tamaño de la caché		
	4K	16K	64K
16	8.57 %	3.94 %	2.04 %
32	7.24 %	2.87 %	1.35 %
64	7.00 %	2.64 %	1.06 %

Solución 1

Sabemos que

Tiempo medio de acceso a memoria = Tiempo de acierto + Tasa de Fallos × Penalización de fallos

Supongamos que el tiempo de acierto es de un ciclo de reloj independientemente del tamaño de la línea y calculemos así los tiempos medios de acceso a memoria para cada combinación de la tabla. Por ejemplo, si consideramos la caché de 4 KBytes, según la tabla su tasa de fallos es del 8.57%, y de acuerdo al enunciado el tiempo requerido para traer 16 bytes de la memoria son 42 ciclos. Así pues, su tiempo medio de acceso a memoria será $1 + 8,57\% \times 42 = 4,599$ ciclos. Si considerásemos la misma caché pero con un tamaño de línea de 32 bytes, la tasa de fallos sería del 7.24% y la penalización de fallo de 44 ciclos, puesto que la memoria requiere 44 ciclos para

enviar 32 bytes a la caché. Así pues el tiempo medio de acceso a memoria en este caso sería $1 + 7,24\% \times 44 = 4,186$ ciclos.

Siguiendo esta mecánica podemos calcular el tiempo medio de acceso para todas las combinaciones, obteniendo la siguiente tabla:

Tamaño de línea	Penalización de fallo	Tamaño de la caché		
		4K	16K	64K
16	42	4.599	2.655	1.857
32	44	4.186	2.263	1.594
64	48	4.360	2.267	1.509

Así pues, para las cachés de 4 KBytes y de 16 KBytes el mejor tamaño de bloque es el de 32 bytes, mientras que la caché de 64 KBytes consigue su menor tiempo de acceso medio con un tamaño de línea de 64 bytes.

3. Memoria virtual

3.1 Sea un sistema de memoria virtual paginado en dos niveles con un tamaño de página de 512 bytes. El sistema dispone de un espacio virtual de un 1 MByte y de un espacio físico de 512 KBytes, ambos organizados en bytes y reservados para un único proceso. El sistema dispone de una caché asociativa por conjuntos, que consta de cuatro conjuntos de de dos vías cada uno. Cada línea de la caché consta de 4 bytes.

En la figura adjunta (Fig. 1) se muestra el contenido de la tabla del primer nivel y de la caché en un momento determinado en notación hexadecimal. Cada entrada de la tabla de traducción tiene 16 bits, de forma que los más significativos son de control y los menos significativos contienen un número de página física. El bit más significativo de cada entrada es el de residencia, que vale 1 si la página está residente. El resto de los bits de control no influyen en el problema. Las entradas de las tablas de páginas del segundo nivel tienen el mismo formato. La tabla del primer nivel almacena en registros internos del procesador.

Con estos datos y sabiendo que las tablas del segundo nivel se tratan exactamente igual que el resto de las páginas virtuales, indicar de forma detallada todos los pasos implicados en el proceso de traducción así como los distintos valores que se van obteniendo en el proceso si el procesador emite la dirección virtual B8ED1 para leer un byte. ¿Qué dato es enviado finalmente al procesador?

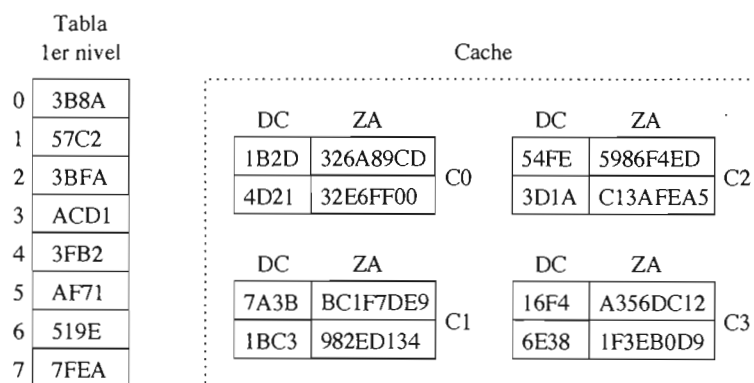


Figura 1: Tabla de páginas del primer nivel y estado de la caché en el problema 3.1

Solución :

Planteamiento:

$|V| = 1 \text{ MByte} = 2^{20} \text{ bytes} \implies 20 \text{ bits de dir. virtual.}$

$|M| = 512 \text{ KBytes} = 2^{19} \text{ bytes} \implies 19 \text{ bits de dir. física.}$

Tamaño de página = 512 bytes = 2^9 bytes $\implies 9$ bits de desplazamiento.

▪ Descomposición de la dir. virtual

- Al ser la página de 512 bytes, el desplazamiento en la página requiere 9 bits.
- Como la tabla de páginas del primer nivel tiene 8 entradas, se requieren $\log_2(8) = 3$ bits para indexarla.
- Puesto que las DVs son de 20 bits, los restantes $20 - 9 - 3 = 8$ bits indexarán la tabla de páginas del segundo nivel.

Núm PV		Δ
TP1	TP2	
3 bits	8 bits	9 bits

▪ Descomposición de la dir. física

- Al ser la página de 512 bytes, el desplazamiento en la página requiere 9 bits.
- Puesto que las DFs son de 19 bits, los restantes $19 - 9 = 10$ bits serán el número de página física.

Núm PF	Δ
10 bits	9 bits

▪ Descomposición de la dir. física desde el punto de vista de la caché:

- Líneas de la caché de 4 bytes $\implies 2$ bits de desplazamiento
- La caché tiene 4 conjuntos $\implies 2$ bits de campo de índice
- $19 - 2 - 2 = 15$ bits de etiqueta

Etiqueta	Indice	Δ
15 bits	2 bits	2 bits

Resolución:

Descomponemos la dirección virtual B8ED1 generada por el procesador:

B8ED1 en binario es 1011|1000|1110|1101|0001, que descompuesto queda:

TP1	TP2	Δ
101	11000111	011010001

Así pues se lee la entrada 101 (5) de la TP1 que contiene el valor AF71 (véase la Fig. 1). Este valor en binario es 1010|1111|0111|0001. El 1 en el bit más significativo indica que la tabla de páginas del segundo nivel asociada está residente en la memoria física. Como un número de PF consta de 10 bits y el número de PF está en los bits menos significativos de las entradas de las tablas de páginas; el número de página física que contiene la tabla de páginas del segundo nivel asociada es 1101110001. En dicha tabla se necesita leer la entrada 11000111 según indica el segundo campo de la DV B8ED1. Ahora bien, dado que cada entrada de la tabla de páginas, tanto en el primer como en el segundo nivel, consta de 16 bits (2 bytes), la dirección física donde se inicia la entrada de la TP2 a leer se obtiene concatenando al mencionado número de página física el número de entrada multiplicado por dos. Se obtiene así la dirección física

Núm. PF	Núm entrada	$\times 2$
1101110001	11000111	0

Esta dirección corresponde a una posición de memoria física, que será buscada en la caché antes que en memoria principal cuando se la intenta acceder, al igual que cualquier otra posición de memoria física. Si descomponemos la dirección física anterior desde el punto de vista de la caché obtenemos

Etiqueta	Indice	Δ
110111000111000	11	10

es decir, debe buscarse la etiqueta $(110111000111000)_2 = 6E38$ en alguna de las dos líneas del conjunto 3, y de haber acierto, la entrada de la tabla de páginas del segundo nivel que deseamos leer ocupará los bytes 2 y 3 de la línea de la caché. El motivo de este último razonamiento es que cada entrada de las tablas de páginas consta de dos bytes, y como vemos en la descomposición precedente, el desplazamiento del inicio de la entrada dentro de la línea caché es $\Delta = (10)_2 = 2$.

Comprobamos que en efecto dicha etiqueta se encuentra en el conjunto 3 de la caché y los bytes 2 y 3 de su línea contienen el valor hexadecimal B0D9. De esta forma, estos 16 bits son la entrada de la TP2 que estamos buscando, que en binario son 1011|0000|1101|1001. El bit más significativo de la entrada indica que la página física que estamos buscando se encuentra residente en memoria principal, y los 10 últimos bits de la entrada contienen precisamente cuál es el número de esta PF: 0011011001. Para completar el proceso de traducción este número de página física debe concatenarse con el desplazamiento dentro de la página:

Núm PF	Δ
0011011001	011010001

que en hexadecimal es 1B2D1. Al tratarse de una dirección física, el procesador intenta accederla en primer lugar en la caché, y de fallar ahí, la buscará en memoria principal. Si descomponemos la dirección precedente desde el punto de vista de la caché obtenemos

Etiqueta	Indice	Δ
001101100101101	00	01

que indica que en el conjunto 0 debe buscarse la etiqueta $(001101100101101)_2 = 1B2D$, y el byte buscando será aquél que tiene el desplazamiento uno ($\Delta = (01)_2 = 1$) dentro de la línea asociada. Vemos que hay un acierto y que el byte correspondiente contiene un **6A**.

3.2 Considera un computador con memoria virtual paginada con un esquema de traducción directa en un nivel, donde el espacio virtual es de 128 MBytes, el espacio físico es de 1 MByte y el tamaño de página es de 4 KBytes. ¿Qué tamaño en bytes tiene la tabla de páginas si éstas contiene la información mínima para hacer la traducción (no hay ningún bit de control)?

Halla el tamaño de las páginas si queremos que la tabla de páginas ocupe sólo 1 KByte.

Solución:

Planteamiento:

$$|V| = 128 \text{ MBytes} = 2^{27} \text{ Bytes} \implies 27 \text{ bits de dir. virtual.}$$

$$|M| = 1 \text{ MByte} = 2^{20} \text{ bytes} \implies 20 \text{ bits de dir. física.}$$

$$\text{Tamaño de página} = 4 \text{ KBytes} = 2^{12} \text{ bytes} \implies 12 \text{ bits de desplazamiento.}$$

Resolución:

Tamaño TP = Núm. Entradas × Tamaño entrada = Núm. Págs. Virtuales × Tamaño entrada =

$$= \frac{2^{27} \text{ bytes de espacio virtual}}{2^{12} \text{ bytes/página}} \times \log_2 \left(\frac{2^{20} \text{ bytes de espacio físico}}{2^{12} \text{ bytes/página}} \right) = 2^{15} \text{ entradas} \times 8 \text{ bits/entrada} =$$

$$= 2^{15} \text{ bytes} = \boxed{32 \text{ KBytes}}$$

Para calcular el tamaño de la página de forma que la TP sólo ocupe un KByte, utilizamos la misma ecuación pero transformando en incógnita el tamaño de página y poniendo como dato conocido el tamaño total de la TP:

$$\left(\frac{2^{27}}{2^p} \right) \text{ entradas} \times \log_2 \left(\frac{2^{20}}{2^p} \right) \text{ bits/entrada} \times \left(\frac{1}{2^3} \right) \text{ bytes/bit} = 2^{10} \text{ bytes}$$

$$\left(\frac{2^{14}}{2^p} \right) \times \log_2 \left(\frac{2^{20}}{2^p} \right) = 1$$

$$2^{(14-p)} \times (20 - p) = 1$$

Por un proceso de prueba y error se obtiene $p = 16$, por lo que el tamaño de la página es $2^p = 2^{16} = 64 \text{ KBytes}$. El número de p 's a probar es reducido puesto que para que el producto de $2^{(14-p)}$ por $(20 - p)$ de uno, es necesario que $(20 - p)$ sea una potencia también de dos, y además de exponente opuesto al de $2^{(14-p)}$.

3.3 Sea el esquema de traducción de direcciones virtuales a direcciones físicas del procesador Intel 386 con los siguientes supuestos:

- Registro Base de la Tabla de Segmentos (RBTS): 00000000
- Registro Base de la Tabla de Páginas del Primer Nivel (RBTP): 000F0000
- Las entradas de la tabla de segmentos y de páginas son de 32 bits.
- En las entradas de las tablas de páginas los bits menos significativos tienen en número de página física asociada.
- La dirección física X de la memoria principal contiene el valor $X \bmod 256$.

Si el procesador emite la dirección virtual 00A200BF8400, de 46 bits, obténgase la dirección física correspondiente en cada uno de los cuatro modos de direccionamiento del procesador suponiendo que no se va a producir ningún fallo.

Solución :

a) Sin memoria virtual:

Las direcciones físicas del Intel 386 son de 32 bits. Cuando no hay memoria virtual los 14 bits altos del registro de segmento del procesador son ignorados y los 32 bits bajos de la misma constituyen la dirección física correspondiente. Así pues, la dirección física asociada a la dirección virtual 00A200BF8400 serían los 32 bits menos significativos de la misma, esto es, $\boxed{00BF8400}$.

b) Segmentación pura:

En la memoria virtual segmentada del Intel 386 los 32 bits bajos de la dirección virtual corresponden al desplazamiento dentro del segmento, en tanto que los 14 bits altos procedentes del registro de segmento constituyen el número de segmento. Así pues, la dirección virtual 00A200BF8400 corresponde al acceso con desplazamiento 00BF8400 dentro del segmento virtual 00A2. La dirección física asociada a esta dirección virtual la obtendremos sumando el desplazamiento a la dirección física en que se inicia el segmento en memoria principal, la cual extraeremos de la tabla de segmentos. Para obtenerla empezamos calculando la dirección de la entrada de la tabla de segmentos asociada al segmento A2:

$$\text{Dir. Física de la entrada para el segmento A2} = \text{RBTS} + 4 \times \text{A2} = 0 + 288 = 288$$

El cálculo precedente se basa en que la tabla de segmentos comienza en la dirección de memoria marcada por el RBTS, y dado que cada entrada ocupa 32 bits, es decir, 4 bytes, la entrada i -ésima se encontrará a $4 \times i$ bytes del inicio de la tabla.

El enunciado indica que la dirección física X de la memoria principal contiene el valor $X \bmod 256$, así pues, los 4 bytes de la entrada de la tabla de segmentos que se inicia en la posición de memoria principal 288 contendrán los valores

Dirección	288	289	28A	28B
Contenido	88	89	8A	8B

Según el enunciado no puede producirse ningún fallo, y las direcciones de memoria física son de 32 bits, con lo que los 32 bits de la entrada suponen la dirección de memoria principal en que se inicia el segmento. Así pues podemos calcular que la dirección física asociada a la dirección virtual 00A200BF8400 cuando el sistema emplea segmentación pura es

$$88898A8B(\text{base del segmento}) + 00BF8400(\text{Desplazamiento}) = \boxed{89490E8B}$$

c) Paginación pura:

Cuando el Intel 386 está en modo paginado puro los 14 bits de la dirección virtual procedentes del registro de segmento se desprecian y los 32 bits menos significativos constituyen lo que se denomina la dirección lineal. Esta dirección se descompone en tres campos (puesto que la paginación es en dos niveles) de los siguientes tamaños:

TP1	TP2	Desplazamiento
10 bits	10 bits	12 bits

En nuestro caso, la descomposición de los 32 bits menos significativos de la dirección virtual 00A2—00BF8400 es

TP1	TP2	Desplazamiento
$(0000000010)_2 = 2_{16}$	$(1111111000)_2 = 3F8_{16}$	$(010000000000)_2 = 400_{16}$

Con estos valores procedemos como con cualquier otro sistema paginado en dos niveles. Comenzamos por indexar la tabla de páginas del primer nivel, para lo cual debemos calcular la dirección en memoria de su entrada 2 (campo TP1 de la dirección). La tabla de páginas del primer nivel comienza en la dirección marcada por el RBTP, y cada entrada consta de 32 bits, es decir, 4 bytes, con lo que dicha dirección será

$$\text{Dir. Física de la entrada 2} = \text{RBTP} + 4 \times 2 = F0000 + 8 = F0008$$

Según el enunciado, la dirección física X de la memoria principal contiene el valor $X \bmod 256$, con lo que los 4 bytes de la entrada de la tabla de páginas del primer nivel que se inicia en la posición de memoria principal F0008 contendrán los valores

Dirección	F0008	F0009	F000A	F000B
Contenido	08	09	0A	0B

en los que los bits menos significativos contienen el número de página física donde radica la tabla de páginas del segundo nivel. Sabemos que las direcciones físicas son de 32 bits y que el desplazamiento dentro de las páginas, que es el mismo en el espacio virtual y en el espacio físico, es de 12 bits. Así pues, los números de página física deben constar de $32 - 12 = 20$ bits. En nuestra entrada de la tabla de páginas del primer nivel los 20 bits menos significativos son 90A0B (5 dígitos hexadecimales, a razón de 4 bits por dígito hexadecimal). Por tanto, dado que en este ejercicio no hay fallos de traducción, la tabla de páginas del segundo nivel que requerimos está en la página física 90A0B, la cual se inicia en la posición de memoria física 90A0B000. El número de entrada dentro de la tabla de páginas del segundo nivel que deseamos acceder es la 3F8, como indica el campo TP2 de la dirección virtual generada. Como cada entrada ocupa 32 bits, es decir, 4 bytes, la dirección de inicio de la entrada 3F8 será

$$90A0B000 + 4 \times 3F8 = 90A0B000 + FE0 = 90A0BFE0$$

Siguiendo la norma de que la dirección de memoria física X contiene el valor $X \bmod 256$, los 4 bytes de la entrada contendrán los valores

Dirección	90A0BFE0	90A0BFE1	90A0BFE2	90A0BFE3
Contenido	E0	E1	E2	E3

Esta es una entrada de la tabla de páginas del segundo y último nivel, con lo que en su campo de página física encontramos por fin el número de página física en que está mapeada la página virtual que estamos buscando. Para calcular la dirección física deseada sólo tenemos que concatenar este número de página física con el desplazamiento dentro de la página extraído del tercer y último campo de la dirección virtual. Ya hemos razonado que los números de página física son de 20 bits y según el enunciado el número de página física se encuentra en los bits menos significativos de la entrada, con lo que esta entrada señala a la página física 1E2E3. El desplazamiento procedente de la dirección generada por el procesador era 400 (ya en hexadecimal), así pues, la dirección física asociada es la 1E2E3400

d) Segmentación con paginación.

Cuando el esquema de segmentación y de paginación se combinan en el Intel 386 la traducción consta de dos fases. En la primera fase los 46 bits de la dirección virtual generada por el procesador se tratan y descomponen como una dirección segmentada pura, siguiendo el mismo esquema visto en el apartado b) de este ejercicio. La dirección obtenida tras esta fase (89490E8B en este problema) es entonces tratada como una dirección lineal de 32 bits y se realiza con ella un proceso análogo al realizado en el apartado c) del ejercicio, es decir, se la procesa como una dirección de un espacio paginado en dos niveles. Si se aplica este procedimiento a la dirección lineal 89490E8B y nos ajustamos a las condiciones impuestas en el enunciado del problema, podemos comprobar que el resultado final del proceso de traducción es la dirección física 14243E8B.

3.4 Supongamos un computador con un espacio virtual de 64 bytes, una memoria principal de 32 bytes y un tamaño de página de 8 bytes. El computador utiliza un esquema de memoria virtual paginada con traducción en un nivel, con el auxilio de una TLB de tres entradas. El algoritmo de reemplazo de las páginas físicas y de las entradas de la TLB es de tipo FIFO.

El sistema también posee una memoria caché de 8 bytes, organizada por conjuntos. En concreto dispone de dos conjuntos con dos líneas por conjunto. Dentro de cada conjunto se utiliza un algoritmo de reemplazo de tipo LRU.

Si el procesador emite la siguiente lista de direcciones virtuales (en hexadecimal), indicar y explicar cómo evoluciona la tabla de páginas, la TLB y el directorio caché partiendo de la situación inicial indicada en la figura 2. Como puede apreciarse, la figura muestra que en la memoria física se hayan cargadas tres páginas virtuales, siendo el orden en que han sido cargadas: primero la 111, luego la 001, y por último la 011.

Direcciones virtuales: 1D, 21, 1C, 0C, 15 y 2A.

En la figura R es el bit de residencia, V es el bit de validez, DC es el directorio caché, ZA la zona de almacenamiento, y LRU es el bit del algoritmo LRU, que indica el número de línea dentro del conjunto que habrá que reemplazar cuando se desee introducir una nueva línea en el conjunto.

Tabla de páginas			TLB		Cache			
R	Pag.	Fis.	Pag. Virt.	Pag. Fis.	LRU	V	DC	ZA
0	0	---	111	10	1	1	000	
1	1	00	001	00	1	1	001	
2	0	---	011	01				
3	1	01						
4	0	---			0	1	010	
5	0	---			1	1	101	
6	0	---						
7	1	10						

Figura 2: Tabla de páginas, TLB estado de la caché en el problema 3.4.

Solución:

Planteamiento:

$|V| = 64 \text{ bytes} \implies 6 \text{ bits de dir. virtual.}$

$|M| = 32 \text{ bytes} \implies 5 \text{ bits de dir. física.}$

Tamaño de página = 8 bytes = $\implies 3 \text{ bits de desplazamiento.}$

Caché de 8 bytes organizada en dos conjuntos de dos líneas cada uno \implies cada línea puede almacenar $\frac{8}{2 \times 2} = 2 \text{ bytes.}$

- Descomposición de la dir. virtual
 - Al ser la página de 8 bytes, el desplazamiento en la página requiere 3 bits.
 - Puesto que las DVs son de 6 bits, y sólo hay un nivel de paginación, los restantes $6 - 3 = 3$ bits indexarán la tabla de páginas.

PV	Δ
3 bits	3 bits

- Descomposición de la dir. física
 - Al ser la página de 8 bytes, el desplazamiento en la página requiere 3 bits.
 - Puesto que las DFs son de 5 bits, los restantes $5 - 3 = 2$ bits serán el número de página física.

Núm PF	Δ
2 bits	3 bits

- Descomposición de la dir. física desde el punto de vista de la caché:

- Líneas de la caché de 2 bytes \implies 1 bit de desplazamiento
- La caché tiene 2 conjuntos \implies 1 bit de campo de índice
- $5 - 1 - 1 = 3$ bits de etiqueta

Etiqueta	Indice	Δ
3 bits	1 bit	1 bit

Resolución:

a) Acceso 1D:

En binario usando 6 bits (tamaño de la dirección virtual), $1D_{16} = 011101_2$. Dado que las direcciones virtuales tienen 3 bits de número de página virtual y otros tres de desplazamiento dentro de la misma, este acceso afecta la página virtual 011 con desplazamiento 101.

La traducción se inicia consultando la TLB, en la que se aprecia que la página virtual 011 está mapeada a la página física 01. Así pues, es la dirección física 01101 la que deseamos acceder. Comprobamos, pues, si dicha dirección física se encuentra en la caché. Para ello descomponemos esta dirección física desde el punto de vista de la caché:

Etiqueta	Indice	Δ
011	0	1

lo cual nos indica que debemos buscar la etiqueta 011 en el conjunto 0. Podemos apreciar en la figura 2 que dicha etiqueta no se haya en el conjunto, con lo que nos encontramos ante un fallo de caché. Ya que el bit LRU señala a la línea 1, es ésta la que debe ser expulsada de la caché para poder introducir en el conjunto 0 la línea con etiqueta 011 en que acabamos de fallar. Tras este proceso la línea 0 pasa a ser la menos recientemente accedida del conjunto 0, con lo que el bit LRU pasa a apuntar a ella. De esta forma, tras este acceso las estructuras de traducción de direcciones (tabla de páginas y TLB) no han sido afectadas, en tanto que en la caché se han producido los cambios que hemos comentado en el conjunto 0, como muestra la figura 3.

R	Pag. Fis.
0	---
1	00
2	---
3	01
4	---
5	---
6	---
7	10

Pag. Virt.	Pag. Fis.
111	10
001	00
011	01

LRU	V	DC	ZA
Conjunto 0			
0	1	000	
1	011		
Conjunto 1			
0	1	010	
1	101		

Figura 3: Tabla de páginas, TLB y estado de la caché en el problema 3.4 tras el primer acceso (1D).

b) Acceso 21:

En binario usando 6 bits (tamaño de la dirección virtual), $21_{16} = 100001_2$. Dado que las direcciones virtuales tienen 3 bits de número de página virtual y otros tres de desplazamiento dentro de la misma, este acceso afecta la página virtual 100 con desplazamiento 001.

En la TLB no hay ninguna traducción almacenada para la página virtual 100, con lo que debe consultarse la tabla de páginas, en la que puede comprobarse que nos encontramos

ante un fallo de página. La memoria física consta de 32 bytes y cada página ocupa 8 bytes; así pues hay un total de 4 páginas físicas. Dado que hay 3 ocupadas debido a los accesos precedentes (la 00, la 01 y la 10 según la tabla de páginas), hay una aún disponible para ubicar nuevas páginas virtuales, en concreto, la página física 11. Así pues, en la entrada 4 (100) escribimos que la página virtual correspondiente se ubica en la página física 11 y reflejamos esta misma información en la TLB. Como la TLB está ya llena, debe expulsarse una entrada para introducir la nueva, y dado que el reemplazo es FIFO, la entrada más antigua es la que debe expulsarse. El enunciado del problema dice que la primera página accedida fue la 111, así pues su entrada es la que debe ser reemplazada por la nueva. La segunda página accedida fue la 001, así que dejamos una marca indicando que ahora esa entrada es la más antigua de la TLB.

Una vez la página virtual 100 ha sido ubicada en la página física 11, podemos calcular la dirección física asociada al acceso. Al inicio del apartado habíamos visto que el acceso a la dirección virtual 21_{16} constituía un acceso a la página virtual 100 con desplazamiento 001. Estando esta página virtual en la página física 11, la dirección física asociada será la 11001, resultante de concatenar el número de página física con el desplazamiento, que es el mismo en el espacio virtual y en el espacio físico. Debemos ahora completar el acceso con el procesamiento en la caché. Esta dirección física queda descompuesta desde el punto de vista de la caché en los siguientes campos:

Etiqueta	Índice	Δ
110	0	1

Como es natural, el conjunto 0 de la caché no contiene ninguna línea con etiqueta 110, puesto que la página acaba de ser traída de la memoria secundaria, con lo que era imposible que hubiera copias de la misma en la caché. Resolvemos el fallo de caché expulsando del conjunto 0 la línea 0, puesto que lo indica así el bit LRU, y sustituyéndola por la nueva línea solicitada. El bit LRU para entonces apunta a la línea 1, ya que ahora será la menos recientemente referenciada.

La figura 4 muestra el estado del sistema tras completar el segundo acceso.

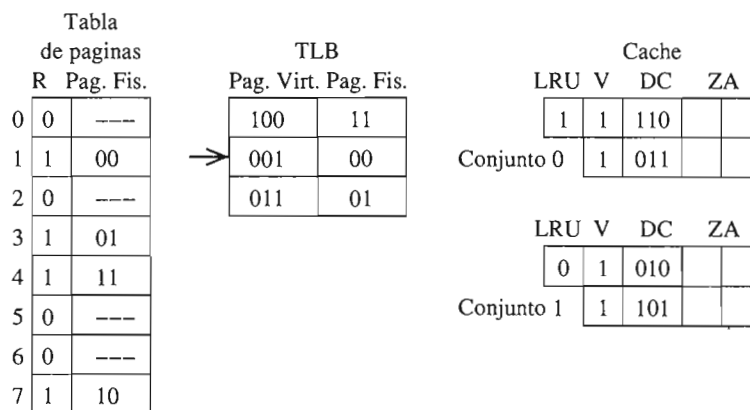


Figura 4: Tabla de páginas, TLB y estado de la caché en el problema 3.4 tras el segundo acceso (21).

c) Acceso 1C:

En binario usando 6 bits (tamaño de la dirección virtual), $1C_{16} = 011100_2$. Dado que las direcciones virtuales tienen 3 bits de número de página virtual y otros tres de desplazamiento dentro de la misma, este acceso afecta la página virtual 011 con desplazamiento 100.

La TLB muestra que la página virtual 011 está ubicada en la página física 01, con lo que la dirección física para este acceso es 01100. Procedemos a acceder a la caché con esta dirección,

descomponiéndola para ello en los siguientes campos:

Etiqueta	Indice	Δ
011	0	0

Podemos apreciar que se trata de un acierto en la caché, puesto que en el conjunto 0, señalado por el campo Índice de la dirección, encontramos una línea con la etiqueta 011 solicitada, en concreto la línea 1. Debe notarse que en el momento de efectuarse el acceso el bit LRU del conjunto señalaba precisamente a esta línea como la menos recientemente referenciada del conjunto. Dado que acabamos de accederla, ahora la línea menos recientemente referenciada del conjunto pasa a ser la línea 0, con lo que debemos modificar el bit LRU del conjunto 0. Así pues, tras este acceso el estado de la tabla de páginas, TLB y caché es el mostrado en la figura 5.

R	Pag.	Fis.
0	0	---
1	1	00
2	0	---
3	1	01
4	1	11
5	0	---
6	0	---
7	1	10

Pag. Virt.	Pag. Fis.
100	11
001	00
011	01

LRU	V	DC	ZA
0	1	110	
1	1	011	

LRU	V	DC	ZA
0	1	010	
1	1	101	

Figura 5: Tabla de páginas, TLB y estado de la caché en el problema 3.4 tras el tercer acceso (1C).

d) Acceso 0C:

En binario usando 6 bits (tamaño de la dirección virtual), $0C_{16} = 001100_2$. Dado que las direcciones virtuales tienen 3 bits de número de página virtual y otros tres de desplazamiento dentro de la misma, este acceso afecta la página virtual 001 con desplazamiento 100.

La TLB indica que la página virtual 001 ha sido ubicada en la página física 00, con lo que el proceso de traducción de dirección virtual a dirección física termina con éxito, siendo la dirección física asociada la 00100, resultante de concatenar el número de página física hallado con el desplazamiento dentro de la página, que es el mismo en el espacio virtual y en el espacio físico. Procedemos pues, a acceder a la caché descomponiendo la dirección física de la siguiente forma:

Etiqueta	Indice	Δ
001	0	0

Dado que ninguna línea del conjunto 0 tiene la etiqueta 001, nos encontramos ante un fallo de caché. El bit LRU indica que debemos expulsar la línea 0, y reemplazarla por la solicitada. Tras el reemplazo, el bit LRU se actualizará para indicar que ahora la línea menos recientemente referenciada del conjunto en la línea uno. Por tanto, tras este acceso el estado del sistema es el que muestra la figura 6.

e) Acceso 15:

En binario usando 6 bits (tamaño de la dirección virtual), $15_{16} = 010101_2$. Dado que las direcciones virtuales tienen 3 bits de número de página virtual y otros tres de desplazamiento dentro de la misma, este acceso afecta la página virtual 010 con desplazamiento 101.

Tabla de páginas			TLB		Cache			
R	Pag.	Fis.	Pag. Virt.	Pag. Fis.	LRU	V	DC	ZA
0	0	---	100	11	1	1	001	
1	1	00	001	00	1	1	011	
2	0	---	011	01				
3	1	01						
4	1	11						
5	0	---						
6	0	---						
7	1	10						

Figura 6: Tabla de páginas, TLB y estado de la caché en el problema 3.4 tras el cuarto acceso (0C).

No encontramos ninguna traducción para la página virtual 010 ni en la TLB ni en la tabla de páginas, con lo que este acceso ha generado un fallo de página. Además ahora las cuatro páginas de la memoria física ya están ocupadas, con lo que una página debe ser expulsada para proceder a ubicar la página virtual 010 en memoria física. El algoritmo de reemplazo de las páginas físicas es FIFO, y la primera página accedida fue la 111 según el enunciado. Así pues, la página virtual 111 es expulsada, quedando libre la página física 10 que ocupaba, que es entonces asignada a la página virtual 010 recién solicitada. La nueva asociación debe reflejarse en la TLB, de la que echamos la traducción más antigua, de acuerdo a su algoritmo de reemplazo FIFO. Esta es la entrada que asocia la página virtual 001 con la página física 00, como ya habíamos comentado. Tras el reemplazo la entrada más antigua de la TLB pasa a ser la que asocia la página virtual 011 con la página física 01.

Cuando una página virtual es expulsada de la memoria física todas las líneas de caché que mantienen copias de líneas de memoria principal correspondientes a dicha página deben ser invalidadas. El motivo es que estas líneas dejan de ser copias válidas de porciones de memoria principal, puesto que las posiciones de memoria correspondientes van a ser sobrescritas con otros valores. En nuestro caso eso quiere decir que todas las líneas de la caché asociadas a la página física 10 (donde estaba ubicada la página virtual 111 que expulsamos) deben ser invalidadas. Como el número de página física siempre está en los bits más significativos del campo de etiqueta, para identificar las líneas afectadas basta buscar todas aquellas cuya etiqueta comience por los bits 10. En la figura 6, que muestra el estado del sistema tras la finalización del acceso precedente podemos ver que hay una línea en esta situación: la línea 1 del conjunto 1. Así pues, esta línea es invalidada, lo cual produce a su vez un cambio en el bit LRU del conjunto, pues si bien el bit indicaba que la línea menos recientemente referenciada era la línea 0 del conjunto, no tiene sentido que el bit siga señalando a dicha línea cuando la línea 1 ha sido invalidada y está por tanto vacía y disponible para ser ocupada con cualquier línea nueva que se desee introducir en el conjunto. De esta forma, el bit LRU del conjunto 1 pasa a señalar a la línea recién invalidada.

Tras completar la gestión del fallo de página y ubicar nuestra página virtual 010 en la página física 10 dejada libre por la página expulsada podemos calcular que la dirección física asociada al acceso 15_{16} solicitado por el procesador es la 10101, resultado de concatenar el número de página física (10) donde hemos ubicado la página virtual afectada (010) con el desplazamiento 101. Para acceder a la caché descomponemos la dirección física siguiendo el procedimiento habitual:

Etiqueta	Índice	Δ
101	0	1

Es fácil comprobar que nos encontramos ante un fallo de caché, puesto que el conjunto 0 no incluye ninguna línea con la etiqueta 101. Así pues, procedemos a reemplazar la línea indicada por el bit de LRU de este conjunto (línea 1) con la nueva línea solicitada.

El estado final del sistema tras completar este acceso se encuentra reflejado en la figura 7.

Tabla de páginas			TLB		Cache			
R	Pag.	Fis.	Pag. Virt.	Pag. Fis.	LRU	V	DC	ZA
0	0	---	100	11	0	1	001	
1	1	00	010	10	1	1	101	
2	1	10	011	01				
3	1	01						
4	1	11						
5	0	---						
6	0	---						
7	0	---						

Figura 7: Tabla de páginas, TLB y estado de la caché en el problema 3.4 tras el quinto acceso (15).

f) Acceso 2A:

En binario usando 6 bits (tamaño de la dirección virtual), $2A_{16} = 101010_2$. Dado que las direcciones virtuales tienen 3 bits de número de página virtual y otros tres de desplazamiento dentro de la misma, este acceso afecta la página virtual 101 con desplazamiento 010.

Este acceso genera un fallo de página, puesto que no encontramos ninguna página física asociada a la página virtual 100 ni en la TLB ni en la tabla de páginas. Como la memoria física ya está llena, debe expulsarse la página virtual que lleva más tiempo en la memoria principal (de acuerdo al criterio físico) para poder ubicar la nueva página. El enunciado del problema dice que la primera página virtual accedida fue la 111, que ya ha sido expulsada, y que a continuación se accedió a la página virtual 001, que aún reside en memoria física. Así pues, la página virtual 001 es expulsada, siendo reasignada su página física asociada (00) a la nueva página virtual a ubicar en memoria principal, la página 101. La asociación de la página virtual 101 con la página física 00 debe reflejarse en la TLB, para lo cual expulsamos su entrada más antigua, la que asociaba la página virtual 011 con la página física 01. Tras este reemplazo la entrada más antigua de la TLB pasa a ser la que asocia la página virtual 100 con la página física 11.

Como hemos explicado en el apartado anterior, la expulsión de una página virtual de memoria implica la invalidación de todas las líneas de la caché asociadas a la misma, puesto que dejan de ser copias válidas de valores de memoria principal. En este caso, la página virtual expulsada (001) estaba ubicada en la página física 00, con lo que todas las líneas de la caché procedentes de esa página física deben ser invalidadas. Estas páginas son fácilmente identificables, pues son todas aquellas cuya etiqueta comience por 00. Vemos que sólo hay una línea en la caché de este tipo, la línea 0 del conjunto 0, y el bit de LRU del conjunto ya apunta a esta línea, con lo que no es necesario modificarlo.

Tras completar la gestión del fallo de página y ubicar nuestra página virtual 101 en la página física 00 dejada libre por la página expulsada podemos calcular que la dirección física asociada al acceso $2A_{16}$ solicitado por el procesador es la 00010, resultado de concatenar el número de página física (00) donde hemos ubicado la página virtual afectada (101) con el desplazamiento 010. Para acceder a la caché descomponemos la dirección física siguiendo el procedimiento habitual:

Etiqueta	Indice	Δ
000	1	0

Como de esperar, puesto que la página acaba de ser traída del almacenamiento secundario, el conjunto 1 no contiene ninguna línea con esta etiqueta, con lo que la línea indicada por su bit LRU (1) debe ser reemplazada por la línea solicitada. Realmente no hay tal reemplazo porque la línea afectada de la caché es una línea inválida; simplemente la sobrescribimos pues con la línea de etiqueta 000, reactivando el bit de validez. El bit LRU del conjunto se modifica para indicar que ahora la línea menos recientemente referenciada del conjunto es la línea 0.

La figura 8 muestra el estado final de la tabla de páginas, la TLB y la caché tras completar este acceso.

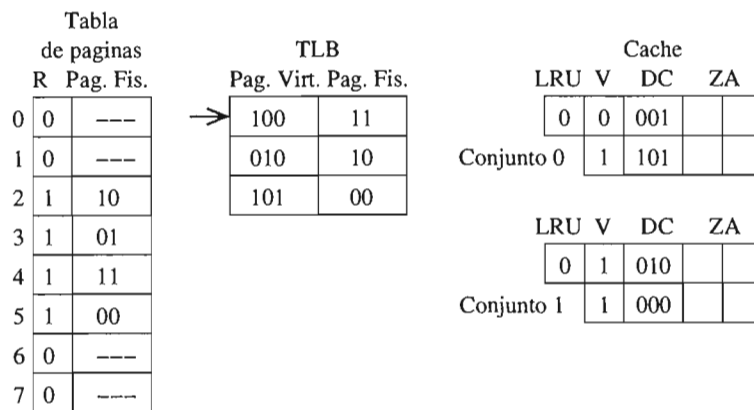


Figura 8: Tabla de páginas, TLB y estado de la caché en el problema 3.4 tras el sexto acceso (2A).

- 3.5** Un computador tiene un espacio virtual con paginación de 16MB y una dirección física de 16 bits. La memoria es direccionable a nivel de byte, y la página física 0 está reservada para almacenar la tabla de páginas, que ocupa exactamente una página, y tiene un solo nivel. Las entradas de la tabla de páginas tienen 32 bits, con la siguiente estructura:

bit Residencia	bit Acceso	Y bits reservados	4 bits contador LFU	X bits núm. PF
----------------	------------	-------------------	---------------------	----------------

Además, las últimas dos páginas físicas de memoria están reservadas para tareas del S.O. y tampoco pueden ubicarse páginas del proceso del usuario en ellas. El algoritmo de reemplazo de páginas es el LFU optimizado explicado en clase, usando un bit de acceso que es revisado cada cuatro referencias para actualizar el contador LFU. Cuando una página se carga por primera vez, el contador LFU se inicializa a 8 y el bit de acceso se pone a cero. Cuando hay varias páginas con el mismo valor mínimo del contador LFU, aquellas con el bit de acceso inactivo tienen la prioridad para ser expulsadas. De haber varias páginas en esa situación, se expulsa la que tiene el número de página virtual más bajo. El sistema tiene también una TLB de cuatro entradas con algoritmo de reemplazo LRU.

- ¿Cuál es el tamaño de página del sistema?
- ¿Cuál es el número de páginas virtuales del sistema?
- Suponiendo que inicialmente la memoria y la TLB están vacías, muestra la evolución de la TLB, la memoria, y la tabla de páginas ante la siguiente secuencia de accesos a páginas virtuales (es decir, este es el número de página virtual accedida, sin el desplazamiento): 6, 7, 2, 6, 5, 8, 7, 9, 9, 4, 8, 5. En las entradas de la tabla de páginas muestra sólo los campos

significativos y en binario; y mientras no sea necesario expulsar páginas para ubicar las nuevas, asigna las páginas físicas en orden creciente (primero la 1, luego la 2, etc.).

Solución

Planteamiento:

$$|V| = 16 \text{ MBytes} \implies 24 \text{ bits de dir. virtual.}$$

$$16 \text{ bits de dir. física} \implies |M| = 64 \text{ KBytes.}$$

Resolución:

- a) Las entradas de la tabla de páginas constan de 32 bits (4 bytes) y el sistema tiene paginación con un solo nivel, habiendo pues una única tabla de páginas, que según el enunciado ocupa justo una página. Así pues, sabemos que

$$\begin{aligned} \text{Tamaño TP} &= \text{Núm. Entradas} \times \text{Tamaño entrada} = \text{Núm. Págs. Virtuales} \times \text{Tamaño entrada} = \\ &= \frac{2^{24} \text{ bytes de espacio virtual}}{2^p \text{ bytes/página}} \times 4 \text{ bytes/entrada} = 2^p \text{ bytes de tabla de páginas} \end{aligned}$$

de donde podemos despejar

$$\frac{2^{26}}{2^p} = 2^p,$$

$$2^{26} = 2^{2p}$$

de lo cual se deduce que $p = 13$, es decir, el tamaño de la página es $2^p = 2^{13} = 8 \text{ KBytes}$.

- b) Sabemos que

$$\text{Tamaño Espacio Virtual} = \text{Núm. Págs. Virtuales} \times \text{Tamaño de la Página}$$

sustituyendo,

$$2^{24} \text{ bytes} = \text{Núm. Págs. Virtuales} \times 2^{13} \text{ bytes/página}$$

$$\text{Núm. Págs. Virtuales} = \frac{2^{24}}{2^{13}} = 2^{11} \text{ páginas}$$

- c) Para resolver este apartado del problema necesitamos saber cuántas páginas físicas caben en la memoria principal, pues de ello dependerá el momento en que no cabrán más páginas virtuales y será necesario comenzar a hacer reemplazos. Otro motivo por el que debemos despejar esta incógnita es saber el valor de X , el número de bits de número de página física que debemos representar en las entradas de la tabla de páginas y la TLB. En el planteamiento del problema dedujimos que la memoria física constaba de un total de 64 KBytes, habida cuenta de que según el enunciado las direcciones físicas del sistema constan de 16 bits. Por otro lado, en el primer apartado del problema calculamos que el tamaño de la página es de 8 KBytes. Así pues, es directo deducir que el sistema consta de un total de

$$\frac{2^{16} \text{ bytes de espacio físico}}{2^{13} \text{ bytes/página}} = 2^3 \text{ páginas físicas}$$

con lo que el número de bits X necesario para indexarlas es $X = \log_2(2^3) = 3$ bits.

Podemos recordar en este punto que el enunciado dice que la página física 0 está ocupada por la tabla de páginas, que ocupa precisamente una página; y que las dos últimas páginas físicas (la 6 y la 7 en nuestro caso, dado que hay 8) están reservadas para tareas del S.O. y tampoco pueden ubicarse páginas del proceso del usuario en ellas. Así pues, iremos ubicando páginas virtuales a medida que vayan siendo accedidas en las páginas físicas de la 1 a la 5, y cuando estas cinco páginas estén ocupadas y sea necesario traer nuevas páginas virtuales a memoria principal, será necesario comenzar a expulsar páginas de memoria física para hacerles sitio. Veamos, pues, la evolución de la ubicación de páginas virtuales en páginas físicas, la tabla de páginas y la TLB ante la secuencia de accesos a páginas virtuales propuesta por el problema. Dicha evolución está representada paso a paso detalladamente en las figuras 9 y 10, que comentamos a continuación:

1) Acceso 1: Página virtual 6.

El enunciado dice que mientras no sea necesario expulsar páginas para ubicar las nuevas, las páginas físicas deben asignarse en orden creciente comenzando por la 1. Así pues, ubicamos la página virtual 6 en la página física 1, restando esta ubicación tanto en nuestro dibujo de la memoria física dividida en páginas como en la TLB y en la entrada afectada de la tabla de páginas (entrada 6). En la entrada puede apreciarse que hemos activado el bit más significativo de la entrada, ya que es el bit de residencia de la página. Por otro lado, hemos inicializado el contador LFU de la página a 8 (1000_2) y el bit de acceso a 0, tal como indica el enunciado. En cuanto a los 3 bits del campo de página física, adoptan el valor $(001)_2 = 1$, indicando la página física donde hemos ubicado la página virtual 6.

2) Acceso 2: Página virtual 7.

Siguiendo las reglas antes comentadas, la página virtual 3 debe ubicarse en la página física 2. En la TLB hemos introducido la nueva entrada encima de la primera para indicar que es más reciente. Siguiendo este sistema, cuando debamos efectuar un reemplazo en la TLB sabremos que la entrada menos recientemente usada, que deberá ser la expulsada, será aquella que se encuentra en la parte inferior de nuestra representación.

3) Acceso 3: Página virtual 2.

La página virtual 2 da lugar a un nuevo fallo de página y debe por tanto ser ubicada en memoria física, en concreto en la siguiente página física disponible, que en este caso es la página 3.

4) Acceso 4: Página virtual 6.

Por primera vez un acceso a una página no resulta en un fallo de página. El sistema encuentra en la TLB la entrada que asocia la página virtual 6 con la página física 1. Por tanto esta entrada se convierte en la más recientemente usada, con lo que la traemos a la parte superior de nuestra representación de la TLB, bajando mientras mantienen su orden relativo entre sí, las otras dos entradas ocupadas actualmente en la TLB.

En cuanto a la tabla de páginas, ésta se actualiza activando el bit de acceso de la página 6, lo cual indica que dicha página ha vuelto a ser accedida desde la última revisión de los contadores LFU. Por cierto, que al ser éste el cuarto acceso de nuestra secuencia, es preciso hacer una revisión de estos contadores, puesto que el enunciado indica que dicha revisión tiene lugar cada cuatro accesos. En esta revisión, de acuerdo al funcionamiento del algoritmo LFU optimizado, aquellas entradas de la tabla de páginas cuyo bit de acceso es 0 ven decrementado el valor de su contador, en tanto que el contador de aquellas que lo tienen activado es incrementado, al tiempo que se les desactiva dicho bit. En la parte inferior de la primera columna de la figura 9 vemos el estado de la tabla de páginas tras aplicar este algoritmo a la tabla que teníamos tras completarse el cuarto

acceso.

5) Acceso 5: Página virtual 5.

No se encuentra una traducción para la página virtual 5 ni en la TLB ni en la tabla de páginas. Así pues, nos encontramos ante un nuevo fallo de página que se resuelve ubicando dicha página en la siguiente página física libre, la 4. La TLB se actualizan mostrando la nueva asociación como su entrada más reciente y la entrada 5 de la tabla de páginas refleja también la ubicación de la página y los valores iniciales del bit de acceso y contador LFU de acuerdo al enunciado del problema, activándose además el bit de residencia de la misma.

6) Acceso 6: Página virtual 8.

Tampoco hay ninguna traducción para la página virtual 8, puesto que no reside en memoria física. Se la ubica en la siguiente página física disponible, que es la 5, y siguiendo el procedimiento habitual reflejamos esta ubicación tanto en la TLB como en la tabla de páginas. Como novedad, debemos expulsar por primera vez una entrada de la TLB, pues sus cuatro entradas ya están ocupadas. Gracias a nuestra representación, que mantiene ordenadas de arriba a abajo las entradas en función de la última vez que se las ha accedido, sabemos que la entrada a expulsar es la inferior de acuerdo al algoritmo LRU, que es la que asocia la página virtual 7 con la página física 2. Tras la expulsión, las otras tres entradas se desplazan hacia abajo, y la nueva asociación se inserta en la entrada superior, al ser la más recientemente usada.

7) Acceso 7: Página virtual 7.

En la TLB no encontramos ninguna traducción para la página virtual 7, pero sí la hallamos en la tabla de páginas en memoria física. Así pues, se producen dos cambios en el sistema. Por un lado, la tabla de páginas se actualiza activando el bit de acceso de la entrada de esta página, ya que ha sido accedida tras la última revisión de los contadores LFU. Por otro lado, como la TLB debe mantener las traducciones para las páginas más recientemente usadas, procedemos a expulsar su entrada más antigua, que asocia la página virtual 2 con la página física 3 para hacer sitio para la traducción de la página virtual 7 a la página física 2.

8) Acceso 8: Página virtual 9.

Este acceso resulta en un fallo de página, pues ni en la TLB ni en la tabla de páginas encontramos una traducción para la página virtual 9. Por otro lado, como razonamos al inicio de la secuencia de accesos, de acuerdo al enunciado del problema sólo disponemos de las páginas físicas entre la 1 y la 5 (ambas inclusive) para ubicar nuestras páginas de usuario, ya que la página física 0 esta ocupada por la tabla de páginas y las dos últimas páginas están reservadas por el Sistema Operativo. Así pues, como estas 5 páginas están ya ocupadas, es necesario expulsar una página de memoria principal aplicando el algoritmo LFU optimizado. Según este algoritmo debemos expulsar aquella página virtual con el menor valor del contador LFU. El enunciado también indica que en caso de empate en el valor mínimo del contador, las páginas con el bit de acceso inactivo tienen la prioridad para ser expulsadas, y que si hay varias páginas en esta situación, simplemente se expulsa la que tiene el número de página virtual más bajo.

La revisión de la tabla de páginas nos muestra que el valor mínimo del contador en este momento es $(0111)_2 = 7$, valor que comparten las páginas virtuales 2 y 7. Ahora bien, mientras el bit de acceso de la página 7 está activo, el de la página 2 no lo está, con lo que esta última es la seleccionada para ser reemplazada por la página virtual 9 recién accedida. La página virtual 2 estaba ubicada en la página física 3, con lo que ahora colocaremos la página virtual 9 en dicha página física. La nueva traducción es insertada en la parte superior de la TLB, que debe expulsar su entrada inferior para hacerle sitio. Además, en la entrada dos de la tabla de páginas desactivamos el bit de residencia,

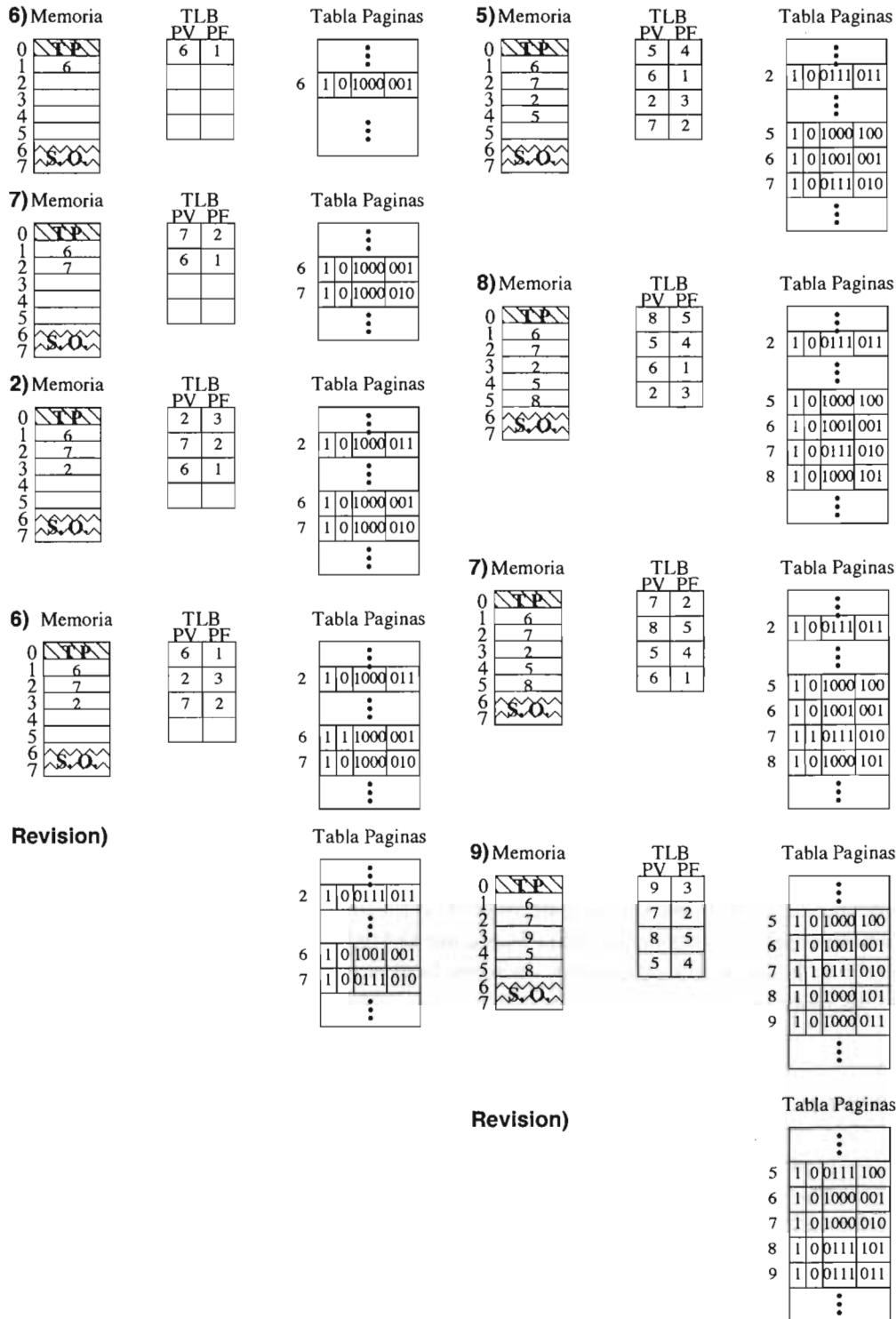


Figura 9: Evolución del sistema del problema 3.5 durante los ocho primeros accesos de su tercer apartado.

e inicializamos la entrada 9 de la tabla con los valores indicados por el enunciado del problema y la información de que se haya en la página física $(011)_2 = 3$ recién liberada. Este acceso constituye el octavo de nuestra serie, con lo que toca hacer una revisión de los contadores LFU, puesto que tienen lugar cada cuatro accesos. La parte inferior de la segunda columna de la figura 9 muestra el estado de la tabla de páginas tras efectuar esta revisión: las entradas cuyo bit de acceso estaba a cero han visto decrementado el valor de su contador, en tanto que aquellas que lo tenían activado, en nuestro caso sólo la página virtual 7, lo han visto incrementado.

9) Acceso 9: Página virtual 9.

El noveno acceso resulta en un acierto de TLB, puesto que su entrada más reciente contiene la traducción para la página virtual 9 que se ha solicitado. La TLB no precisa ya pues ser reordenada, y en la tabla de páginas activamos el bit de acceso para esta página virtual para indicar que ha sido accedida desde la última revisión de los contadores LFU de la tabla de páginas.

10) Acceso 10: Página virtual 4.

Este acceso resulta en un fallo de página, ya que la página virtual 4 no se haya en memoria principal en el momento de intentar accederla, como podemos apreciar en la figura 10. Aplicamos el algoritmo LFU optimizado siguiendo las normas del enunciado del problema y observamos que el valor mínimo del contador LFU en la tabla es $(0111)_2 = 7$, valor que encontramos en las entradas de las páginas virtuales 5, 8 y 9. La página virtual 9 tiene activado su bit de acceso, mientras las otras dos no lo tienen, lo que la descarta para ser expulsada. En esta situación de empate entre las páginas virtuales 5 y 8, el enunciado dice que debe expulsarse aquella con el número de página virtual más bajo, es decir, la 5 en nuestro caso.

Procedemos poniendo a 0 el bit de residencia de la página virtual 5 e invalidando la entrada de la TLB que contiene actualmente su asociación con la página física 4. Una vez expulsada, la página física liberada 4 es asociada a la página virtual 4 que generó el fallo de página y actualizamos tanto la TLB como la tabla de páginas con esta información.

11) Acceso 11: Página virtual 8.

Este acceso resulta en un acierto de TLB, pues una de sus entradas contiene la traducción para la página virtual 8. En esta situación sólo se necesitan hacer dos cambios. Por un lado, el bit de acceso de la entrada correspondiente de la tabla de páginas (entrada 8) se activa para reflejar que la página ha sido accedida desde la última revisión de los contadores LFU. Por otro lado, la entrada asociada a la página virtual 8 pasa a ser la que ha sido accedida más recientemente, con lo que la desplazamos a la parte superior de nuestra representación de la TLB, y bajando en consecuencia las entradas de la parte superior.

12) Acceso 12: Página virtual 5.

El último acceso de la secuencia resulta en un fallo de página, pues como podemos apreciar en la figura 10, la página 5 no reside en memoria principal en el momento en que es solicitada. Aplicando el algoritmo de reemplazo, vemos que el valor mínimo del contador LFU que encontramos en la tabla es $(0111)_2 = 7$, el cual hallamos en las entradas de las páginas virtuales 8 y 9. Ambas páginas tienen activado el bit de acceso, con lo que, al haber empate, de acuerdo al enunciado, expulsamos aquella con el número de página virtual más bajo, es decir, la 8. La expulsión de la página 8 supone la invalidación de la entrada asociada a ella en la TLB así como la puesta a cero del bit de residencia de la entrada correspondiente a esta página en la tabla de páginas.

9) Memoria

0	TP
1	6
2	7
3	9
4	5
5	8
6	SO
7	

TLB	
PV	PF
9	3
7	2
8	5
5	4

Tabla Paginas

5	1	0	0111	100
6	1	0	1000	001
7	1	0	1000	010
8	1	0	0111	101
9	1	1	0111	011

Una vez expulsada la página virtual 8, la página física que ocupaba (página 5) queda disponible para ubicar en ella la página virtual recién solicitada. Nuestro gráfico de la distribución de la memoria principal, de la TLB y de la tabla de páginas reflejan la asociación de la página virtual 5 con la página física 5 liberada por el reemplazo.

Por último, dado que éste es nuestro decimosegundo acceso, toca hacer una revisión de los contadores LFU, dado que tienen lugar cada cuatro accesos. El estado de la tabla de páginas tras efectuar esta revisión se refleja en la parte inferior de la figura 10. Todas las entradas de la tabla de páginas excepto la que está asociada a la página virtual 9 han visto decrementado el valor de su contador, puesto que su bit de acceso estaba a cero. Sólo hemos incrementado el contador de la entrada 9, ya que era la única que tenía activado el bit de acceso.

4) Memoria

0	TP
1	6
2	7
3	9
4	4
5	8
6	SO
7	

TLB	
PV	PF
4	4
9	3
7	2
8	5

Tabla Paginas

4	1	0	1000	100
5	0			
6	1	0	1000	001
7	1	0	1000	010
8	1	0	0111	101
9	1	1	0111	011

8) Memoria

0	TP
1	6
2	7
3	9
4	4
5	8
6	SO
7	

TLB	
PV	PF
8	5
4	4
9	3
7	2

Tabla Paginas

4	1	0	1000	100
5	0			
6	1	0	1000	001
7	1	0	1000	010
8	1	1	0111	101
9	1	1	0111	011

5) Memoria

0	TP
1	6
2	7
3	9
4	4
5	5
6	SO
7	

TLB	
PV	PF
5	5
4	4
9	3
7	2

Tabla Paginas

4	1	0	1000	100
5	1	0	1000	101
6	1	0	1000	001
7	1	0	1000	010
8	0			
9	1	1	0111	011

Revision)

Tabla Paginas

4	1	0	0111	100
5	1	0	0111	101
6	1	0	0111	001
7	1	0	0111	010
8	0			
9	1	0	1000	011

Figura 10: Evolución del sistema del problema 3.5 durante los cuatro últimos accesos de su tercer apartado.

3.6 Un computador tiene un esquema de memoria virtual paginada en dos niveles direccionable a nivel de byte y reservado para un único proceso. Tanto la tabla de páginas del primer nivel como las del segundo tienen la misma estructura: caben exactamente en una página y constan de 128 entradas de 16 bits, siendo el primer bit (el más significativo) el de residencia y los 8 últimos (los menos significativos) el número de página física. Por otro lado, la tabla páginas del primer nivel se encuentra siempre residente en la última página física.

El computador tiene una caché indexada por direcciones físicas con una zona de almacenamiento de 4KB con líneas de 4 bytes en la que cada entrada del directorio caché sólo consta de un bit de validez más la etiqueta de la línea correspondiente. El tamaño total del directorio caché es 1024 bytes.

Sabiendo que el contenido de un byte de la memoria principal se obtiene sumando el número de página física en que reside con su desplazamiento dentro de dicha página y obteniendo el módulo 256, es decir, el byte de la página física X , desplazamiento Y , contiene el valor $(X + Y) \bmod 256$; indica:

- Tamaño del espacio virtual y del espacio físico.
- Dirección física correspondiente a la dirección virtual 415A.
- Indica el tamaño de los campos en que se divide la dirección física desde el punto de vista de esta caché y cuántas líneas tiene cada conjunto de la misma.
- Si esta caché tuviera una tasa de fallos del 10%, un tiempo de acierto de 2 ns. y su tiempo medio de acceso fuera de 5 ns., ¿Cuál sería su penalización de fallo?

Solución:

Planteamiento:

Como cada tabla de páginas cabe en una página del sistema y tienen 128 entradas de 16 bits (2 bytes) cada una \implies Tamaño de página = $128 \times 2 = 256 (2^8)$ bytes.

- Descomposición de la dir. virtual
 - Como el sistema tiene paginación en dos niveles y las tablas de cada nivel tienen 128 entradas, el campo para indexar la tabla de cada nivel requiere $\log_2(128) = 7$ bits.
 - Al ser las páginas de 256 bytes, el campo de desplazamiento dentro de cada página necesitará $\log_2(256) = 8$ bits.

Núm PV		Δ
TP1	TP2	
7 bits	7 bits	8 bits

- Descomposición de la dir. física
 - El enunciado dice que las tablas usan 8 bits para representar un número de página física.
 - Al ser la página de 256 bytes, el desplazamiento en la página requiere $\log_2(256) = 8$ bits.

Núm PF	Δ
8 bits	8 bits

Resolución:

- Hemos calculado que el sistema tiene 22 bits de dirección virtual y 16 bits de dirección física. Así pues, este computador tiene un espacio virtual de $|V| = 2^{22}$ bytes = 4 MBytes y un espacio físico de $|F| = 2^{16} = 64$ KBytes.

b) Descompongamos la dirección virtual 415A en los campos que hemos calculado previamente:

TP1	TP2	Δ
$(0000000)_2 = 0_{16}$	$(1000001)_2 = 41_{16}$	$(01011010)_2 = 5A_{16}$

Así pues se lee la entrada 0 de la TP1, que según el enunciado se halla en la última página física del sistema. Ya que el número de página física consta de 8 bits, el sistema tiene 256 páginas físicas, que estarán numeradas de la 0 a la 255. En consecuencia la última página física es la 255 (FF). Para leer la entrada X dentro de una tabla, debemos calcular el desplazamiento correspondiente multiplicando X por 2, ya que cada entrada consta de 2 bytes. Así pues, siendo $X = 0$ en nuestro caso, calculamos que la entrada que buscamos está en la dirección física $FF|00$. El enunciado dice que el byte de la página física X , desplazamiento Y , contiene el valor $(X + Y) \bmod 256$. En consecuencia los contenidos de esta entrada de la tabla de páginas del primer nivel serán

Dirección	FF00	FF01
Contenido	FF	00

Lo primero que habría que comprobar sería, como siempre, el bit de residencia, que es el más significativo de la entrada. Vemos que está activo, con lo que podemos proceder a leer el número de página física donde se encuentra la tabla de páginas del segundo nivel que tenemos que leer. Como este valor radica en los 8 bits menos significativos de la entrada de la tabla de páginas, vemos que es 0 en nuestro caso.

El siguiente paso consiste en indexar la tabla de páginas del segundo nivel que se halla en la página física 0 del computador usando el índice correspondiente que hemos extraído de la dirección virtual. Este índice era $(1000001)_2 = 41_{16}$. De nuevo, al tener cada entrada dos bytes, debemos multiplicar este índice por dos para indexar la tabla. Así llegamos a la conclusión de que la entrada que tenemos que leer estará en la dirección de memoria $00|82$ y sus contenidos, según las normas del enunciado serán:

Dirección	0082	0083
Contenido	82	83

De nuevo, en primer lugar debemos comprobar que el bit de residencia esté activado, lo cual vemos que se verifica. Así pues, los ocho bits menos significativos de la entrada (83_{16}) contienen el número de página física que estamos buscando. Para completar el proceso de traducción sólo resta concatenarle a este número de página física el desplazamiento dentro de la misma, que según habíamos calculado al principio de este apartado es $5A_{16}$. Por tanto, la dirección física correspondiente es la $\boxed{835A}$.

c) La caché tiene una zona de almacenamiento de 4KBytes, y cada línea consta de 4 bytes. Así pues, trivialmente, la caché tiene un total de

$$\text{Núm. líneas} = \frac{4096 \text{ bytes de almacenamiento}}{4 \text{ bytes/línea}} = 1024$$

Por otro lado, el directorio de una caché tiene una entrada por cada línea que ésta almacena. Ya que el directorio consta de 1024 bytes, y hay 1024 líneas, cada entrada constará exactamente de un byte. Según el enunciado esta entrada sólo consta de un bit de validez y el resto corresponde a la etiqueta de la línea. Así pues, las líneas tienen una etiqueta de $8 - 1 = 7$ bits. Además, como las líneas son de 4 bytes, se deriva que el campo de desplazamiento dentro de una línea requerirá de un total de $\log_2(4) = 2$ bits. Finalmente, dando que la caché se indexa por direcciones físicas, y como hemos visto anteriormente, éstas constan de 16 bits, el campo de índice deberá constar de $16 - 7 - 2 = 7$ bits. Así pues, desde el punto de vista de la caché la dirección física se divide de la siguiente forma:

Etiqueta	Indice	Δ
7 bits	7 bits	2 bits

El campo de índice se emplea para buscar el conjunto en el que puede estar guardada la línea que contiene la dirección solicitada. Dado que este campo tiene una longitud de 7 bits, nuestra caché constará de $2^7 = 128$ conjuntos. Hemos calculado que la caché tiene un total de 1024 líneas. Así pues, la respuesta a la pregunta sobre el número de líneas por conjunto será

$$\text{Núm. líneas por conjunto} = \frac{1024 \text{ líneas}}{128 \text{ conjuntos}} = \boxed{8 \text{ líneas}}$$

d) Sabemos que

Tiempo medio de acceso a memoria = Tiempo de acierto + Tasa de Fallos \times Penalización de fallos

donde podemos sustituir

$$5ns = 2ns + 10/100 \times PF$$

con lo que despejamos

$$PF = \frac{(5 - 2)ns}{0,1} = \boxed{30 \text{ ns}}$$