3

La jerarquía de memoria

Objetivos

El objetivo de los problemas de este apartado es mostrar cómo el empleo de una jerarquía de memoria permite conseguir un sistema de memoria de gran capacidad, gran velocidad y bajo coste, todo ello sustentado por el principio de localidad.

Los problemas muestran una visión simplificada de la jerarquía. Por ejemplo, se supone que el tiempo de lectura en caso de acierto en un nivel cualquiera de la jerarquía es despreciable frente al tiempo de lectura en caso de fallo. Así, por ejemplo, cuando se produce un fallo de caché, el tiempo necesario para llevar el dato de la caché a la CPU se desprecia frente al tiempo necesario para llevar el bloque correspondiente de la memoria principal a la caché. El mismo razonamiento se aplica cuando se produce un fallo en memoria principal y es necesario llevar el dato desde disco a memoria principal y posteriormente a memoria caché.

Cuando se aborda el nivel de la jerarquía que representa el disco, puede llamar la atención el hecho de que el tiempo de acceso a los bloques en que se organiza no dependa del tamaño de los mismos, lo cual es cierto en la práctica para bloques pequeños. En el acceso al disco lo más costoso es acceder al primer byte del bloque, siendo el tiempo de acceso a los demás bytes del bloque despreciable. En el caso del disco duro se debe a las enormes latencias mecánicas que intervienen en el acceso al disco.

3.1. La jerarquía de memoria

Problema 1.

Las características de una jerarquía de memoria son las expresadas en la tabla 3.1 en la página siguiente. En dicha jerarquía el tamaño del bloque de caché es de 64 bytes, transfiriéndose un byte en cada acceso a caché.

A partir de esta información responder:

❑ **1.1** ¿Cuál es el coste del sistema de memoria descrito?

Nivel	t_{acc}	**Tamaño**	**Precio**	**Tasa aciertos**
Memoria caché (MC)	0.5 ns	8 MiB	15 euros/MiB	0.99
Memoria principal (MP)	3 ns	8 GiB	0.01 euros/MiB	0.99999
Disco magnético (MD)	10 ms	16 GiB	0.0001 euros/MiB	1

Tabla 3.1: Características de la jerarquía de memoria

❑ **1.2** ¿Cuál es el tiempo medio de lectura, considerando solamente los dos niveles más próximos al procesador, es decir, suponiendo que la tasa de aciertos de memoria principal fuese del 100 %? Responde en nanosegundos.

❑ **1.3** ¿Cuál es el tiempo medio de lectura de la jerarquía considerando todos sus niveles? Responde en nanosegundos.

❑ **1.4** ¿Qué crees que ocurrirá con la tasa de aciertos de caché si aumentamos el tamaño de la caché? En el caso ideal en el cual la tasa de aciertos alcanzase el máximo posible, ¿cuál sería el tiempo medio de lectura de memoria?

❑ **1.5** ¿Crees que es posible aumentar el tamaño de la caché indefinidamente para mejorar el rendimiento?

❑ **1.6** ¿Qué crees que ocurrirá con la tasa de aciertos de caché si reducimos el tamaño de la caché? Suponiendo que la tasa de aciertos de caché pasase a ser del 75 %, ¿cuál sería ahora el tiempo medio de lectura de memoria suponiendo que la tasa de aciertos de memoria principal es del 100 %? ¿Sería interesante en este caso disponer de memoria caché?

❑ **1.7** ¿Qué crees que ocurriría con el número de accesos a disco por unidad de tiempo si aumentasemos el tamaño de la memoria principal? ¿Qué ocurriría con la velocidad de lectura de la jerarquía en ese caso?

Problema 2.

Se pretende construir un sistema de memoria para un PC utilizando una jerarquía formada por una memoria caché, la memoria principal y el disco. Las características de cada uno de los niveles son las siguientes:

- Caché. El tiempo medio de acceso, t_c, a una palabra de 64 bits es de 0.5 ns.
- Memoria principal. El tiempo medio de acceso, t_p, a una palabra de 64 bits es de 15 ns.
- Disco. La lectura de un bloque de disco, t_d, de cualquier tamaño entre 1 byte y 10 KiB requiere de media unos 8 ms.

Se conoce también:

- La tasa de aciertos en memoria caché, A_c, es del $98.5\,\%$.
- La tasa de aciertos en memoria principal, A_p, es del $99.995\,\%$.
- El tamaño del bloque de caché es de 16 palabras de 64 bits.
- La escritura de una posición de memoria implica la escritura simultánea en caché y en memoria principal para mantener actualizada la información almacenada en ambos niveles.

A partir de esta información responde a las siguientes preguntas:

❑ **2.1** ¿Cuál es el tiempo medio de lectura para esta jerarquía, es decir, t_{cpd}? Responde en nanosegundos.

❑ **2.2** Si la CPU realiza una operación de escritura sobre una palabra de memoria cacheada, ¿cuál es el tiempo necesario para completar la operación? Responde en nanosegundos.

❑ **2.3** ¿Existe alguna diferencia en el tiempo necesario para completar la operación de escritura, en el caso de que la palabra de memoria a modificar no estuviera en caché? Si existe diferencia, cuantifícala.

Problema 3.

Un computador dispone de una memoria caché y una memoria principal con las siguientes características: tiempo de acceso a una palabra de caché, t_c, de 0.5 ns y tiempo de acceso a una palabra de memoria principal, t_p, de 10 ns. Se utilizan bloques de caché de 4 palabras.

Se conoce además:

- Cada acceso de escritura por parte de la CPU implica que el dato se escribe simultáneamente en memoria caché y memoria principal sin necesidad de traer un bloque a la caché.
- La memoria caché tiene una tasa de aciertos, A_c, del $98.5\,\%$.
- De cada 100 accesos por parte de la CPU, 75 son para realizar una operación de lectura y el resto para realizar una operación de escritura.

❑ **3.1** ¿Cuál es el tiempo medio de acceso para realizar una operación de lectura en la jerarquía descrita? Responde en nanosegundos.

❑ **3.2** Considerando las probabilidades de lectura y escritura anteriores, ¿cuál es el tiempo medio de acceso a la jerarquía descrita? Responde en nanosegundos. Asume una estrategia *no write allocate*.

Problema 4.

En algunas jerarquías de memoria el funcionamiento durante una operación de escritura es diferente al funcionamiento durante una operación de lectura. La escritura en un nivel de la jerarquía se replica en el siguiente, tal como ocurre en las cachés de tipo *write-through*[1].

En este tipo de cachés, cuando se produce un acierto de caché de escritura, la escritura se lleva a cabo simultáneamente en memoria principal y caché, por lo que el tiempo de escritura coincide con el de la memoria principal. Cuando se produce un fallo de caché de escritura puede implementarse una de estas dos estrategias:

[1] Las trataremos en detalle en el tema de caché.

- *No write allocate*. Se escribe el dato en la memoria principal.
- *Write allocate*. Se copia a la caché el bloque de memoria principal que contiene la dirección de escritura. A continuación se lleva a cabo de forma simultánea la escritura del dato tanto en memoria caché como en memoria principal.

Debemos tener en cuenta que empleando escritura *write-through* el tiempo medio de acceso a memoria será diferente entre lecturas y escrituras. Además, en ambos casos dependerá de la estrategia *write allocate*, o *no write allocate*, empleada. Veámoslo con un ejemplo.

Las características de cada uno de los niveles de una jerarquía de memoria son las siguientes:

- Caché. Tipo *write through*. El tiempo medio de acceso, t_c, a una palabra de 64 bits es de 0.5 ns.
- Memoria principal. El tiempo medio de acceso, t_p, a una palabra de 64 bits es de 15 ns.
- Disco. La lectura de un bloque de disco, t_d, de cualquier tamaño entre 1 byte y 10 KiB requiere de media unos 8 ms.

Se conoce también:

- La tasa de aciertos en memoria caché, A_c, es del $98.5\,\%$ empleando la configuración (a) *no write allocate* y del $99\,\%$ empleando la configuración (b) *write allocate*.
- La tasa de aciertos en memoria principal, A_p, es del $99.995\,\%$.
- El tamaño del bloque de caché es de 16 palabras de 64 bits y el de disco de 4 KiB.
- El 70 % de los accesos a memoria son de lectura y el 30 % de escritura.

A partir de esta información debes responder a las siguientes preguntas:

❑ **4.1** ¿Cuál es el tiempo medio de lectura para esta jerarquía, es decir, tr_{cpd}, en cada una de las dos configuraciones? Responde en nanosegundos.

❑ **4.2** ¿Cuál es el tiempo medio de escritura para esta jerarquía, es decir, tw_{cpd}, en cada una de las dos configuraciones? Responde en nanosegundos.

❑ **4.3** ¿En vista del resultado del apartado anterior, podríamos afirmar que la estrategia *no write allocate* es la que proporcionaría un mayor rendimiento? ¿Por qué?

❑ **4.4** ¿Cuál es el tiempo medio de acceso para esta jerarquía, es decir, t_{cpd}, en cada una de las dos configuraciones? Responde en nanosegundos.

❑ **4.5** ¿Qué configuración proporcionaría un mayor rendimiento en nuestro ejemplo? ¿Por qué?

Problema 5.

Aparte del tamaño de la memoria principal y de la caché, hay un factor que influye drásticamente en la velocidad de acceso a la jerarquía de memoria. ¿Cuál es dicho factor?

Problema 6.

Indica cuál o cuáles de las siguientes afirmaciones son ciertas. Si crees que ninguna lo es, responde NINGUNA.

A) El tamaño del bloque que intercambian los niveles de la jerarquía de memoria suele ser menor cuanto más nos alejamos de la CPU.

B) Habitualmente, la organización de la memoria en una jerarquía busca principalmente obtener una memoria de alta velocidad a bajo coste, siendo menos importante la capacidad.

C) El incremento del tamaño del disco tiene un efecto notable sobre el rendimiento de la jerarquía de memoria.

D) El empleo combinado de variables globales y locales dentro de un procedimiento en C favorece la localidad del programa.

E) En los computadores antiguos no existía memoria caché, dado que la velocidad de la memoria principal era la misma que la velocidad de la CPU.

3.2. La memoria caché

Problema 7.

Actualmente, la caché del sistema suele implementarse dentro del mismo chip del procesador, lo que mejora sensiblemente la velocidad de acceso del procesador a la caché y, por tanto, incrementa en gran medida el rendimiento del computador en su conjunto.

Para la fabricación del procesador y la memoria caché se emplean transistores integrados dentro del chip. A medida que mejora la tecnología de fabricación se pueden integrar más transistores dentro del chip, con lo que pueden utilizarse cachés más grandes. Sin embargo, independientemente de la tecnología de fabricación, en la práctica, la implementación de un bit de caché requiere 6 transistores aproximadamente.

Se tiene el procesador Intel© Core i7-3770T, con tecnología de 22 nm, núcleo Ivy Bridge y con un coste de unos 300 euros, el cual integra 1400 millones de transistores distribuidos entre cuatro núcleos, periféricos integrados y tres niveles de caché:

- 4×64 KiB de caché L1.
- 4×256 KiB de caché L2.
- 8 MiB de caché L3.

❑ **7.1** ¿Qué porcentaje de transistores ocupan aproximadamente las celdas de la caché?

Suponiendo que el coste achacable a la caché es proporcional al número de transistores requerido,

❑ **7.2** ¿qué precio tiene aproximadamente la caché de ese procesador?

Problema 8.

Un computador tiene un espacio de direcciones de memoria de 4 MiB y está dotado de una caché de 64 KiB. La memoria se direcciona al byte y la caché intercambia bloques de 16 bytes con la memoria principal.

A partir de esta información responde a las siguientes preguntas:

❑ **8.1** ¿De cuántas líneas de caché disponemos?

❑ **8.2** ¿En cuántos bloques se divide la memoria principal?

❑ **8.3** ¿Cuál es el tamaño, en bits, de la dirección de memoria que utiliza este computador?

La dirección anterior se reparte en varios campos dependiendo de la estrategia de correspondencia que se utilice en la caché. En todos los casos siguientes, indica los campos en los que se divide la dirección de memoria.

❑ **8.4** Estrategia de correspondencia totalmente asociativa.

❑ **8.5** Estrategia de correspondencia directa.

❑ **8.6** Estrategia de correspondencia asociativa por conjuntos de 4 vías.

Problema 9.

A partir de la CPU teórica se plantea desarrollar un sistema de memoria caché. Como es sabido, esta CPU tiene un espacio de direcciones de memoria de 64 Kipalabras de 16 bits cada una. La memoria caché que se plantea utilizará una estrategia de correspondencia asociativa por conjuntos. Bajo esta configuración la línea de caché será de 8 palabras y el campo etiqueta de 6 bits. También se sabe que si la misma memoria caché utilizara una estrategia de correspondencia directa, el campo etiqueta sería entonces de 4 bits.

❑ **9.1** ¿En cuántos conjuntos está dividida la memoria caché?

❑ **9.2** ¿Cuál es el número de líneas de la memoria caché?

❑ **9.3** ¿Qué tamaño en bytes tiene la memoria caché?

Problema 10.

Se conocen las siguientes características de un computador:

- Dispone de un espacio direccionable de 1 MiB, con direccionamiento al byte.
- Utiliza una memoria caché unificada con estrategia de correspondencia directa, de 4 KiB de capacidad y un tamaño de línea de 32 bytes.
- Los tiempos de acceso a los niveles de memoria son: 4 nanosegundos para el nivel de memoria caché y 20 nanosegundos para el nivel de memoria principal. En caso de un fallo de caché, se considera únicamente el tiempo de acceso correspondiente al nivel de memoria principal.
- En cada acceso a memoria se leen 32 bits.

Dadas las características del computador anterior, se pide contestar a las siguientes preguntas:

❑ **10.1** ¿Cuántos bits se utilizan para codificar la dirección en este computador?

❑ **10.2** ¿Cómo se reparten los bits que codifican la dirección entre los distintos campos que utiliza la memoria caché?

❑ **10.3** Considerando la caché inicialmente vacía, la CPU accede de forma consecutiva a las siguientes direcciones de memoria: A5234h, A8230h, A823Fh y FF01Ah. ¿Qué direcciones provocan fallo de caché y por qué motivo?

❑ **10.4** ¿Cuánto tiempo será necesario emplear para realizar los accesos a las direcciones de memoria del apartado anterior?

❑ **10.5** Se vuelve a considerar la memoria caché en las condiciones iniciales, ¿cuál es la tasa de aciertos que se obtiene al ejecutar el siguiente bucle de programa?

```
1 for (i = 0; i < 1024; i++)
2 {
3     datos[i] = 0;
4 }
```

Se sabe también que `datos` es un array de números enteros, cada uno de los cuales utiliza 4 posiciones de memoria para almacenarse, los elementos del array se almacenan de forma consecutiva a partir de la dirección 2A520h. Se supone también que tanto para las constantes como para el índice del bucle no es necesario acceder a memoria, pues previamente ya se han cargado en registros de la CPU.

❑ **10.6** Considerando nuevamente la memoria caché en las condiciones iniciales, ¿cuál es la tasa de aciertos que se obtiene al ejecutar el siguiente bucle de programa?

```
1 for (i = 0; i < 100; i++)
2 {
3     a[i] = 0;
4     b[i] = i;
5 }
```

Se sabe también que tanto `a` como `b` son arrays de números enteros. Los elementos del array `a` se almacenan de forma consecutiva a partir de la dirección 10220h y los del `b` lo hacen a partir de la dirección 00220h. Se supone también que tanto para las constantes como para el índice del bucle no es necesario acceder a memoria, pues previamente ya se han cargado en registros de la CPU.

❑ **10.7** ¿Cómo se podría mejorar la tasa de aciertos obtenida en el apartado anterior?

Problema 11.

Suponemos un computador con las siguientes características:

- Bus de direcciones de 16 líneas.
- El sistema de memoria utiliza palabras de 16 bits.
- Está dotado de una memoria caché de 64 bytes, organizada en líneas de 8 palabras y correspondencia totalmente asociativa.
- La estrategia de reemplazo es LRU.

Nº	**Dir.**	**L/E**	**Nº**	**Dir.**	**L/E**
1	15C0h	L	**9**	208Fh	E
2	15C1h	L	**10**	2090h	L
3	15C2h	L	**11**	15D9h	L
4	15C9h	L	**12**	405Ch	E
5	2080h	E	**13**	15D6h	L
6	2081h	E	**14**	15D7h	L
7	15CAh	L	**15**	6579h	L
8	15D8h	L	**16**	657Ah	L

Tabla 3.2: Accesos a la caché del problema 11.

- La estrategia de escritura es *write-back*.

Sobre este computador se va a ejecutar un programa cuyo patrón de accesos a memoria se muestra en la tabla 3.2. En dicha tabla, `Dir` representa la dirección a la que se accede y `L/E` representa el tipo de operación (Lectura o Escritura) que se realiza.

Si suponemos que la memoria caché está inicialmente vacía. Responde a las siguientes preguntas:

❑ **11.1** ¿De cuántas líneas dispone la memoria caché descrita?

❑ **11.2** ¿Qué direcciones producen fallos de caché durante la ejecución de este programa en la configuración de memoria descrita?

❑ **11.3** ¿Qué porcentaje de aciertos tiene el programa anterior con dicha configuración de caché?

❑ **11.4** ¿Qué bloque o bloques de memoria principal resultan reemplazados en el esquema de funcionamiento descrito? Indica su número en hexadecimal. Contesta NINGUNO, si no es necesario realizar reemplazo.

❑ **11.5** ¿Qué bloque o bloques de memoria principal han de ser actualizados en el esquema de funcionamiento descrito? Indica su número en hexadecimal. Contesta NINGUNO, si no es necesario realizar actualización.

Nº	Dir.	L/E	C/D	Nº	Dir.	L/E	C/D
1	400h	L	C	**9**	409h	L	C
2	148h	L	D	**10**	150h	L	D
3	401h	L	C	**11**	160h	E	D
4	402h	L	C	**12**	40Ah	L	C
5	149h	E	D	**13**	361h	L	D
6	40Ah	L	C	**14**	414h	L	C
7	40Bh	L	C	**15**	362h	L	D
8	1BDh	E	D	**16**	363h	E	D

Tabla 3.3: Accesos a la cache del problema 12.

Problema 12.

Consideremos una memoria cache de un nivel unificada con correspondencia directa, inicialmente vacía, formada por 8 líneas de 4 bytes cada uno, y direccionamiento al byte. La estrategia de escritura es *write-back*. El campo etiqueta de la cache es de 7 bits.

En la tabla 3.3 se muestra la traza del programa que emplearemos, en la que para cada acceso se indica la dirección, si el acceso es de lectura (L) o escritura (E) y si se accede a código (C) o datos (D).

A partir de toda la información anterior responder a las siguientes preguntas:

❑ **12.1** ¿Cuál es el tamaño del espacio de direcciones, expresado en bytes, de este sistema de memoria?

❑ **12.2** ¿Cuál es el tamaño, expresado en bytes, de la memoria cache del sistema?

❑ **12.3** ¿Qué direcciones de memoria generan fallos de cache durante la ejecución del programa?

❑ **12.4** ¿Cuál es el porcentaje de aciertos en memoria cache para la ejecución de este programa?

❑ **12.5** ¿Qué bloque o bloques de memoria principal resultan reemplazados durante la ejecución del programa? Responder en hexadecimal. Si crees que ninguno resulta reemplazado, responde NINGUNO.

❑ **12.6** ¿Qué bloque o bloques de memoria principal se actualizan con datos de la cache durante la ejecución del programa? Si crees que ninguno resulta actualizado, responde NINGUNO.

❑ **12.7** Se conoce que el tiempo de acceso a memoria caché es de 2ns y que el tiempo de acceso a un byte de memoria principal es 10ns. ¿Cuál es el tiempo empleado en los accesos de la traza anterior?

La cache anterior se reorganiza dividiéndola a la mitad en una cache de código y una cache de datos, manteniéndose todas las demás características de la cache.

❑ **12.8** ¿Qué bloque o bloques de memoria principal resultan ahora reemplazados? Responder en hexadecimal. Si crees que ninguno resulta reemplazado, responde NINGUNO.

❑ **12.9** Qué bloques de memoria principal se actualizan con datos de la cache durante la ejecución del programa? Responder en hexadecimal. Si crees que ninguno resulta actualizado, responde NINGUNO.

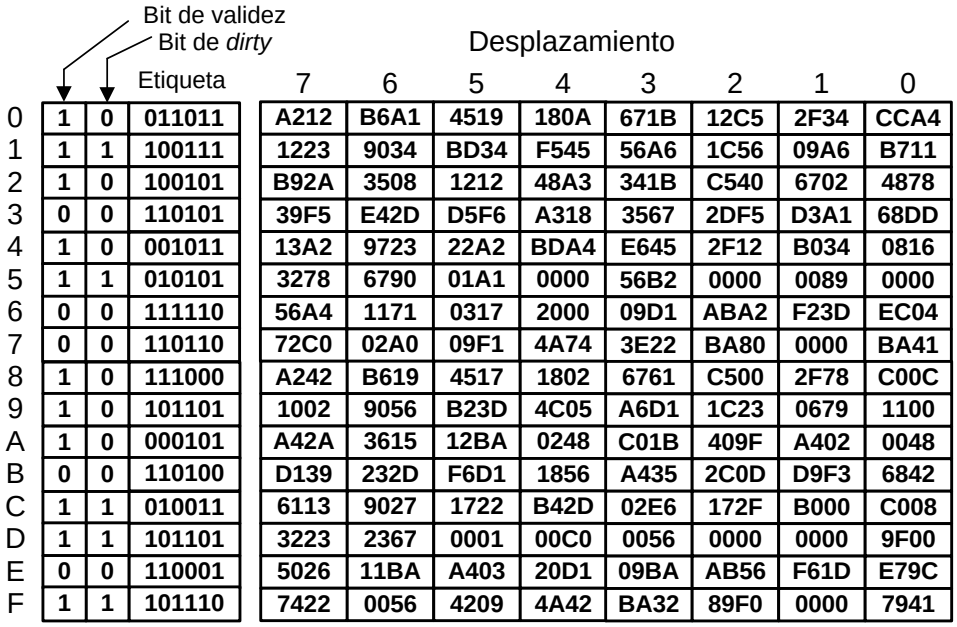
Problema 13.

Se considera el sistema de memoria de un computador cuya memoria caché, con estrategia de correspondencia directa, se muestra en la figura 3.1. Sobre este sistema de memoria se plantean una serie de preguntas.

❑ **13.1** ¿Qué estrategia de escritura se emplea?

❑ **13.2** ¿Cuál es el tamaño de la memoria caché expresado en bytes?

❑ **13.3** ¿Cuántos bloques de memoria principal se encuentran pendientes de actualización?



0	1	0	011011	A212	B6A1	4519	180A	671B	12C5	2F34	CCA4
1	1	1	100111	1223	9034	BD34	F545	56A6	1C56	09A6	B711
2	1	0	100101	B92A	3508	1212	48A3	341B	C540	6702	4878
3	0	0	110101	39F5	E42D	D5F6	A318	3567	2DF5	D3A1	68DD
4	1	0	001011	13A2	9723	22A2	BDA4	E645	2F12	B034	0816
5	1	1	010101	3278	6790	01A1	0000	56B2	0000	0089	0000
6	0	0	111110	56A4	1171	0317	2000	09D1	ABA2	F23D	EC04
7	0	0	110110	72C0	02A0	09F1	4A74	3E22	BA80	0000	BA41
8	1	0	111000	A242	B619	4517	1802	6761	C500	2F78	C00C
9	1	0	101101	1002	9056	B23D	4C05	A6D1	1C23	0679	1100
A	1	0	000101	A42A	3615	12BA	0248	C01B	409F	A402	0048
B	0	0	110100	D139	232D	F6D1	1856	A435	2C0D	D9F3	6842
C	1	1	010011	6113	9027	1722	B42D	02E6	172F	B000	C008
D	1	1	101101	3223	2367	0001	00C0	0056	0000	0000	9F00
E	0	0	110001	5026	11BA	A403	20D1	09BA	AB56	F61D	E79C
F	1	1	101110	7422	0056	4209	4A42	BA32	89F0	0000	7941

Figura 3.1: Estado inicial de la caché del problema 13.

❑ **13.4** ¿Cuál es el bloque de memoria principal con dirección más baja que se encuentra cacheado? Responder en hexadecimal o INDEFINIDO si no se puede conocer.

❑ **13.5** ¿Cuántos bloques de memoria principal diferentes pueden ocupar, en diferentes momentos, la línea de memoria cache 2?

❑ **13.6** ¿Podría aplicarse una estrategia de reemplazo LRU a esta caché? Justifica tu respuesta.

❑ **13.7** Cuando la CPU accede para leer la dirección 012Ah, ¿se produce acierto o fallo? ¿A qué línea de caché se accederá? ¿Será reemplazado algún bloque de memoria principal, si lo es cuál? Si es necesario, responder en hexadecimal.

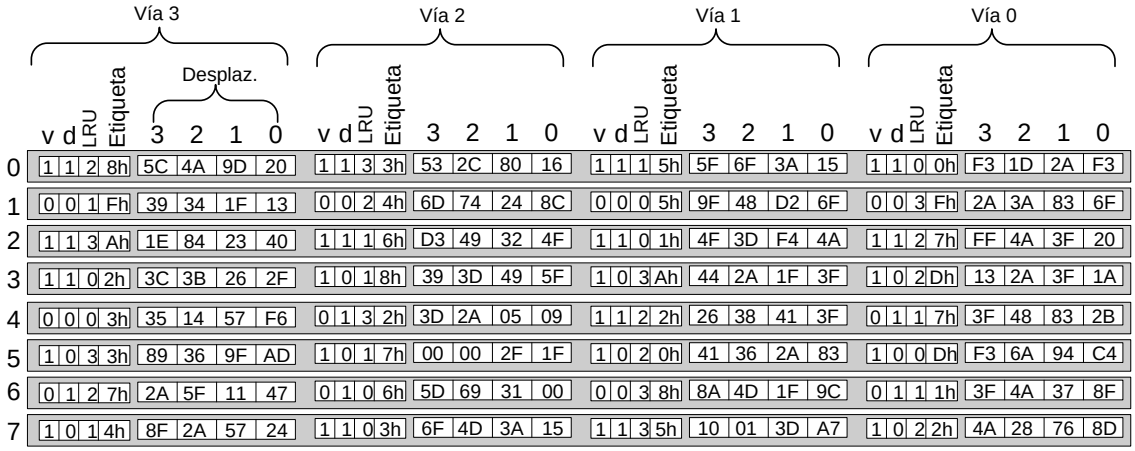


Figura 3.2: Estado inicial de la caché del problema 14.

❑ **13.8** Si un periférico con capacidad DMA accediera a memoria principal para leer el contenido de la dirección 177Fh, ¿sería posible saber el valor que leería? Si fuera posible indica cuál sería ese valor. ¿Le sucedería algo a la memoria caché?

❑ **13.9** Si un periférico con capacidad DMA accediera a memoria principal para escribir en la dirección de memoria 018Dh, ¿qué le sucedería a la memoria caché?

Problema 14.

La figura 3.2 muestra el estado de una caché unificada. Cada línea de caché tiene asociado un bit de validez, *v*, un bit de *dirty*, *d*, dos bits *LRU* y una *Etiqueta* de 4 bits. La línea con menor valor LRU es el que ha sido accedido menos recientemente.

❑ **14.1** ¿Cuántos bloques de memoria principal están almacenados en caché?

❑ **14.2** ¿Qué valor devuelve la caché cuando la CPU trata de leer de la dirección 053h? Indica "Ninguno" si crees que se produce fallo de caché.

❑ **14.3** ¿Cuántos bloques de memoria no son coherentes entre caché y memoria principal?

❑ **14.4** Indica la direción de memoria más alta que podría cachearse en la vía 1 del conjunto 6.

❑ **14.5** Si la CPU escribiese en la dirección 044h, ¿cuál sería el estado de los bits *v*, *d* y *LRU* de la línea de caché correspondiente tras la escritura? Asume que se sigue una política de escritura *write allocate*.

❑ **14.6** Indica una dirección a la que pueda acceder la CPU que cause que el bloque de memoria principal 1Fh sea actualizado en memoria principal.

❑ **14.7** La interfaz de un periférico con capacidad de DMA accede al sistema de memoria a tres posiciones de memoria distintas. Indica qué bits de las líneas de caché son modificados así como su valor final para cada uno de los accesos.

Los accesos a memoria son: escritura en 101h, lectura de 1A2h, y lectura de 0BCh.

Problema 15.

La figura 3.3 muestra el estado en un instante dado de una memoria caché dividida. Cada línea de la caché de datos tiene asociados un bit de validez (v), *dirty* (d) y accedido (a). El bit "a" se pone a uno cuando se accede a una línea del conjunto y se pone a cero cuando se accede a la otra línea del conjunto. La estrategia de reemplazo utilizada es de tipo LRU basada en el valor del bit "a". La caché de código carece del bit de *dirty*.

❑ **15.1** ¿Cuál es la estrategia de escritura de la caché?

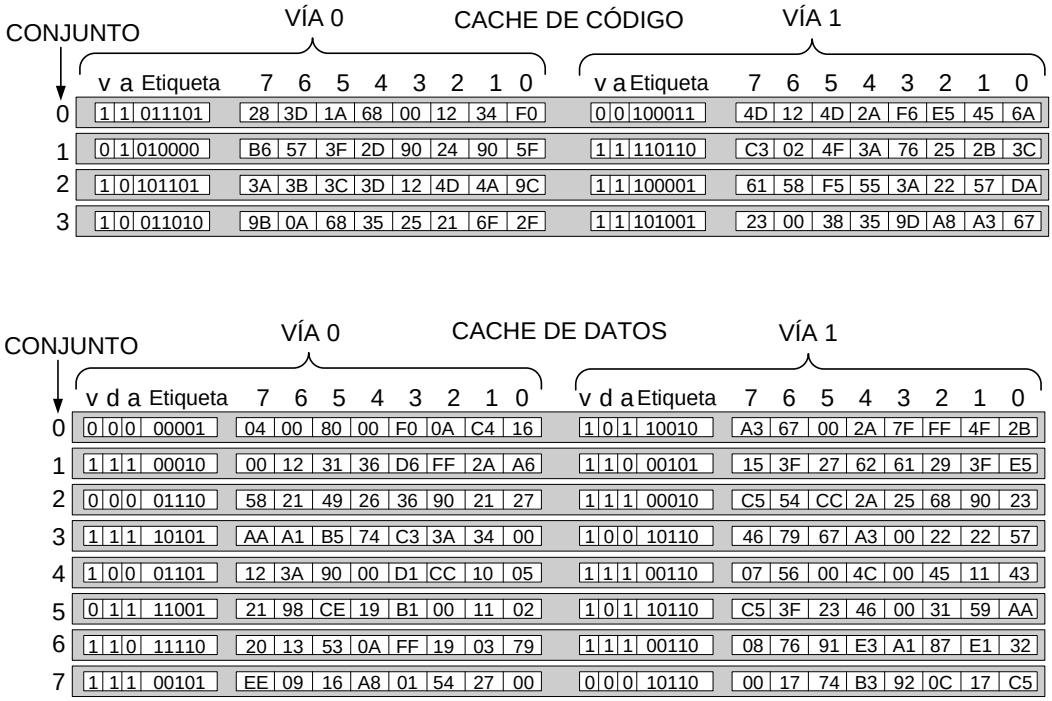


Figura 3.3: Estado inicial de la caché del problema 15.

❑ **15.2** ¿Cuál es el máximo número de bloques de memoria principal que podrían estar cacheados en un momento dado?

❑ **15.3** A partir del estado mostrado en la figura 3.3, ¿cuál es el máximo número de fallos de caché que pueden producirse sin que tenga lugar un reemplazo?

❑ **15.4** A partir del estado mostrado en la figura 3.3, ¿cuántos bloques de memoria principal deben ser actualizados en caso de ser reemplazados?

❑ **15.5** ¿Cuál es la dirección de código más baja que está cacheada? Responder en hexadecimal.

❑ **15.6** ¿Qué bloque o bloques de memoria principal resultan reemplazados cuando la CPU lee una variable flotante de doble precisión (64 bits) ubicada en la dirección 070h? Responde en hexadecimal.

❑ **15.7** Cuando la CPU escribe un dato en la dirección 15Ah se produce un fallo de caché. Si se sigue una política *write allocate* es necesario traer el bloque la caché antes de escribir. Indica la ubicación de la línea de caché en la que se cargará el bloque de memoria principal que contiene la dirección anterior, así como el nuevo estado de la etiqueta y los bits "v", "d" y "a" de la línea correspondiente tras la escritura.

Problema 16.

Un computador dispone de dos procesadores de un núcleo, denominados P1 y P2. Cada procesador incorpora un único nivel de caché, de tamaño 8 KiB, estrategia de correspondencia asociativa por conjuntos de 2 vías y líneas de 16 bytes. El computador dispone además de una memoria principal de tamaño 1 MiB direccionada al byte.

Inicialmente ambas cachés están vacías y los procesadores llevan a cabo la secuencia de accesos a memoria en el orden indicado:

1. P1 lectura de dirección 201A3h.
2. P2 lectura de dirección 8135Bh.
3. P2 lectura de dirección 201A8h.
4. P1 escritura en dirección 201A0h.
5. P2 escritura de dirección 201A8h.
6. P1 lectura de dirección 8135Ch.

❑ **16.1** Para cada uno de los accesos anteriores debe indicarse si se produce acierto de caché, fallo de caché, reemplazo o actualización, así como el nuevo estado MESI de las líneas de caché implicadas los accesos en ambos procesadores.

Problema 17.

Indica cuál o cuáles de las siguientes afirmaciones son ciertas. Si crees que ninguna lo es, responde NINGUNA.

A) El grado de asociatividad de una memoria caché asociativa por conjuntos se incrementa a medida que se reduce el número de conjuntos de la caché manteniendo el número de líneas constante.

B) Las cachés L1 divididas mejoran el rendimiento con respecto a las unificadas cuando se conectan a una CPU que no implemente ningún tipo de paralelismo a nivel de instrucción.

C) Cuando un periférico con capacidad de DMA accede a una dirección de memoria cacheada para realizar una operación de lectura, y la memoria está conectada a una memoria caché con estrategia de escritura *write-back*, se puede producir un problema de coherencia.

D) En la práctica, la estrategia de correspondencia más usada es la estrategia de correspondencia totalmente asociativa.

E) Una caché dividida, en general, tiene una tasa de aciertos menor que su equivalente unificada.

Problema 18.

Indica cuál o cuáles de las siguientes afirmaciones son ciertas. Si crees que ninguna lo es, responde NINGUNA.

A) La jerarquía de memoria proporciona el mínimo tiempo de acceso cuando la tasa de aciertos en la caché es del 0 %.

B) Para que una línea de caché sea reemplazada durante una operación de lectura de memoria por parte de la CPU, es necesario que la lectura dé lugar a un fallo de caché.

C) En general, la estrategia de escritura *write-back* proporciona un mayor rendimiento en el acceso a memoria que la estrategia de escritura *write-through*.

D) Cuando se transfiere el control de una tarea al sistema operativo o viceversa, es habitual que se incremente el número de fallos de caché que se producen por unidad de tiempo.

E) En los procesadores multinúcleo es habitual que todos los niveles de caché se encuentren replicados en todos los núcleos.

3.3. La memoria virtual

Problema 19.

La gestión de memoria llevada a cabo por la paginación permite incrementar la capacidad del sistema de memoria, pues se emplea el archivo de paginación como el último nivel de la jerarquía de memoria.

❑ **19.1** ¿Cómo se llama la unidad lógica encargada de traducir las direcciones virtuales?

❑ **19.2** ¿En qué dispositivo hardware del computador se encuentra habitualmente integrada?

❑ **19.3** Aparte de incrementar la capacidad del sistema de memoria, ¿qué otros beneficios reporta al computador el empleo de la paginación?

Problema 20.

Indica cuál o cuáles de las siguientes afirmaciones son CIERTAS. Contesta NINGUNA si crees que ninguna lo es.

A) Lo que una tarea escribe en una dirección virtual puede ser leído por otra tarea si emplea la misma dirección virtual.

B) La independencia de los espacios de direcciones virtuales de las tareas se consigue haciendo que las direcciones físicas asociadas a direcciones virtuales de diferentes programas no se solapen en memoria.

C) En general, una dirección virtual tiene asociada una dirección física en memoria principal o una ubicación en el archivo de paginación, o no tiene almacenamiento asignado.

D) El espacio de direcciones virtuales de una tarea puede ser mayor que el espacio de direcciones físicas del sistema.

E) Habitualmente, los computadores disponen de un único espacio de direcciones físicas, en el que se ubican el sistema operativo y las tareas.

F) El archivo de paginación representa el nivel de la jerarquía de memoria más alejado de la CPU.

Problema 21.

Indica cuál o cuáles de las siguientes afirmaciones relativas al TLB son CIERTAS. Contesta NINGUNA si crees que ninguna lo es.

A) El TLB almacena las tablas de páginas de memoria accedidas más recientemente.

B) El TLB es una memoria caché que emplea típicamente una estrategia de correspondencia totalmente asociativa.

C) El TLB suele estar ubicado dentro de la cache de datos.

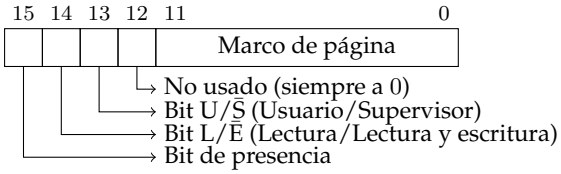
D) Las entradas del TLB marcadas como globales son válidas para todas las tareas del sistema.

E) El TLB no proporciona ninguna mejora de rendimiento si no puede ser accedido de forma simultánea con la memoria caché de propósito general.

F) Tiene poco sentido disponer de TLB cuando el sistema dispone de caché, pues ésta puede emplearse también para almacenar entradas de las tablas de páginas.

Problema 22.

Un computador usa direcciones virtuales de 32 bits, un tamaño de página de 4 KiB y se direcciona al byte. En este computador, cada entrada de la tabla de páginas tiene los siguientes campos:



En un momento dado, las tablas de páginas correspondientes a dos tareas son las mostradas en las figuras adjuntas. Además, el S.O. empleado en este computador establece las páginas de código como sólo lectura y las de datos como lectura y escritura.

Tarea 1

	Pre	L/E	U/S	Marco
0 0050	1	0	1	316h
0 0051	1	0	1	80Bh
0 0052	1	1	1	9CDh
0 0100	0	0	1	Ubic. X
0 0101	1	0	1	1CDh
0 0102	0	0	0	Inválido
A 0001	1	1	0	100h
A 0002	1	1	0	150h
A 0003	0	1	0	Ubic. Z
C 2000	0	0	0	Inválido
C 2001	0	0	0	Inválido

Tarea 2

	Pre	L/E	U/S	Marco
0 0100	1	1	1	025h
0 0151	0	0	1	Ubic. X
0 0152	1	0	1	085h
0 5004	1	0	1	A2Fh
0 5005	0	0	1	Ubic. W
0 5006	0	0	0	Inválido
A 0001	1	1	0	100h
A 0002	1	0	0	150h
A 0003	0	1	0	Ubic. Z
C 2000	0	0	0	Inválido
C 2001	0	0	0	Inválido

Considerando que todas las entradas no mostradas son inválidas, se pide responder a las siguientes preguntas:

❑ **22.1** ¿Cuál es el tamaño del espacio de direcciones físicas de este computador?

❑ **22.2** ¿Tienen las tareas alguna página de datos de usuario compartida?

❑ **22.3** ¿Cuál es el valor de la entrada en la tabla de páginas de la página 52h de la tarea 1?

❑ **22.4** Escribe una dirección virtual correspondiente a la tarea 1 y nivel de usuario que produzca fallo de página recuperable.

❑ **22.5** ¿Qué sucederá si la tarea 2 trata de acceder a la dirección A000 104Fh?

❑ **22.6** ¿Cuánto ocupa la tabla de páginas de la tarea 1 en memoria? Responde en bytes.

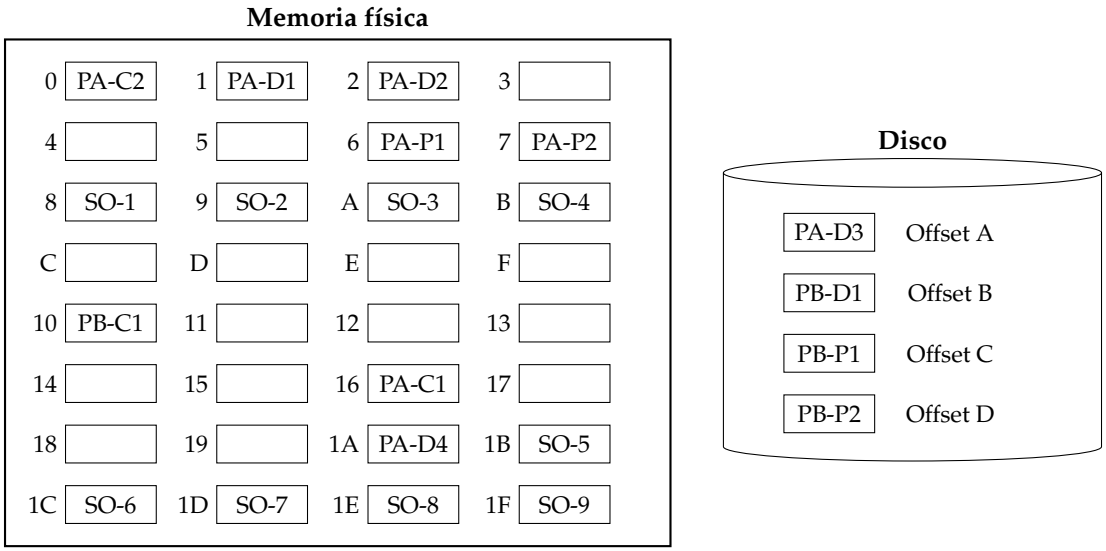
Problema 23.

Se dispone de un computador que utiliza un sistema de memoria virtual paginada con las siguientes características:

- Direcciones virtuales de 20 bits.
- Direcciones físicas de 16 bits.
- Cada página contiene 512 posiciones de memoria de 1 byte cada una.

En este computador está instalado un sencillo sistema operativo multitarea. El único sistema de protección de memoria proporcionado por este sistema operativo es la utilización de tablas de páginas independientes para cada proceso.

En un determinado instante, además del sistema operativo, están en ejecución dos procesos, denominados PA y PB. Su disposición en memoria física (y disco) se muestra en la figura adjunta. Las páginas se nombran utilizando como prefijo el nombre del proceso (PA, PB o SO), seguido de una letra (C, D o P), que indica si la página es de código, datos o pila, y finalmente un número indicando el orden que ocupa la página en la sección del programa a la que pertenece. Así por ejemplo, las páginas PA-D1 y PA-D2 son las dos primeras páginas de la sección de datos del proceso PA, encontrándose ubicadas en el espacio de direcciones virtuales del proceso A primero PA-D1 y después, PA-D2.



Se sabe que la sección de código de todos los programas comienza siempre a partir de la dirección virtual 1A000h y la sección de datos a partir de la dirección virtual 60000h.

La mitad más alta del espacio de direcciones virtuales de cada proceso está siempre reservada para direccionar al sistema operativo, y la sección de pila de los procesos comienza siempre a partir de la dirección virtual más significativa del rango del espacio de direcciones virtuales reservado a procesos de usuario (la mitad más baja).

Todas las páginas virtuales que direccionan al sistema operativo se ubican de forma consecutiva a partir de la primera dirección del espacio de direcciones virtuales reservado al sistema operativo.

❑ **23.1** Rellena la tabla de páginas del proceso B. Esta tabla debe direccionar las páginas que forman las secciones de código datos y pila del proceso B, además de las páginas del sistema operativo

❑ **23.2** En la figura anterior tan sólo se muestra una parte de la tabla de páginas, ya que ésta es muy grande. Suponiendo que cada entrada de la tabla de páginas tiene un tamaño de 1 byte, ¿cuánto ocupa la tabla de páginas del proceso B?

❑ **23.3** Las dos primeras páginas del proceso A (PA-D1 y PA-D2) se utilizan para contener un array de 256 enteros (cada entero ocupa 4 bytes). ¿Qué rango de direcciones físicas se utilizan para almacenar el elemento vector[131]?

❑ **23.4** Se sabe que la primera posición de la página SO-8 contiene información crítica del sistema. Si un usuario malintencionado quisiera borrar el contenido de esa posición para colgar el sistema (aprovechando la carencia de mecanismos de protección de memoria), ¿en qué dirección virtual tendría que escribir?

Núm. pág. virtual	**Presencia**	**Núm. pag. física** **Desp. disco**
0D0h	Sí	10h
300h	No	Offset B

Tabla 3.4: Tabla respuesta para el problema 23.