TEMA 4 - JERARQUÍAS DE MEMORIA. Problemas

4.1. En este ejercicio vamos a considerar las jerarquías de memoria en el contexto de un par de aplicaciones:

Caso a: búsqueda en la web.

Caso b: banca on-line.

i) Suponiendo que tanto cliente como servidor están involucrados en el proceso, en primer lugar indique quién es el cliente y el servidor. ¿Dónde pueden colocarse cachés para acelerar los procesos?

ii) Diseñe una jerarquía de memoria para el sistema. Indique cuál sería el tamaño típico y la latencia en los diversos niveles de la misma. ¿Cuál es la relación entre tamaño de caché y latencia de acceso?

iii) ¿Cuáles son las unidades de transferencia de datos entre los niveles de las jerarquías? ¿Cuál es la relación que existe entre ubicación de los datos, tamaño de los datos y latencia de la transferencia?

iv) El ancho de banda de comunicación y el de procesamiento del servidor son dos importantes factores a considerar cuando se diseña una jerarquía de memoria. ¿Qué anchos de banda pueden ser el factor limitador en el contexto de este ejercicio? ¿Cómo mejorarlo y cuál sería el coste?

4.2. En este ejercicio trataremos de profundizar sobre las propiedades de localidad en memoria de la computación matricial. El siguiente código está escrito en lenguaje C, y los elementos que están en la misma fila de una matriz están almacenados de forma contigua.

Caso a

```
for (I=0; I<8000; I++)
     for (J=0; J<8; J++)
          A[I][J] = B[J][0] + A[J][I];
```

Caso b

```
for (J=0; J<8; J++)
     for (I=0; I<8000; I++)
          A[I][J] = B[J][0] + A[J][I];
```

i) ¿Cuántos enteros de 32 bits pueden almacenarse dentro de una línea de caché de 16 bytes?

ii) ¿Qué referencias exhiben localidad temporal y qué variables involucran?

iii) ¿Qué referencias exhiben localidad espacial y qué variables involucran?

4.3. La localidad se ve afectada tanto por el orden de las referencias como por la organización de la estructura de datos (*data layout*). La misma computación anterior puede escribirse también en MatLab, como aparece más abajo, que difiere de C en que se almacenan de forma contigua los elementos de una misma columna.

Caso a

```
for I = 1:8000
      for J = 1:8
            A(I,J) = B(J,0) + A(J,I);
      end
end
```

Caso b

```
for J = 1:8
      for I = 1:8000
            A(I,J) = B(J,0) + A(J,I);
      end
end
```

i) ¿Cuántas líneas de caché de 16 bytes son necesarias para almacenar todos los elementos de 32 bits de las matrices que son referenciadas?

ii) ¿Qué referencias a variables exhiben localidad temporal?

iii) ¿Qué referencias a variables exhiben localidad espacial?

4.4. Las cachés son importantes para proporcionar una jerarquía de memoria de alto rendimiento a los procesadores. Debajo aparece una lista de referencias a memoria de direcciones de 32 bits, dadas como direcciones de palabra.

Caso a	1, 134, 212, 1, 135, 213, 162, 161, 2, 44, 41, 221
Caso b	6, 214, 175, 214, 6, 84, 65, 174, 64, 105, 85, 215

i) Para cada una de esas referencias identifique la dirección binaria, la etiqueta y el índice supuesto una caché de correspondencia directa con 16 bloques de una palabra. Indique además si cada referencia es un acierto o un fallo, suponiendo que la caché está vacía inicialmente.

ii) Para cada una de esas referencias, identifique la dirección binaria, la etiqueta y el índice dada una caché de correspondencia directa con bloques de dos palabras y un tamaño total de 8 bloques. Además, indique si cada referencia es un acierto o un fallo suponiendo que la caché está inicialmente vacía.

iii) Se le pide que elija el diseño de caché optimizado para las referencias dadas. Hay tres diseños posibles de caché de correspondencia directa, todos ellos con un total de ocho palabras de datos: C1 tiene bloques de una palabra, C2 tiene bloques de dos palabras, y C3 tiene bloques de cuatro palabras. En términos de tasa de fallos, ¿qué diseño de caché es el mejor? Si el tiempo de detención por fallo es 25 ciclos, y C1 tiene un tiempo de acceso de 2 ciclos, C2 lo tienen de 3 ciclos y C3 de 5 ciclos ¿cuál es el mejor diseño de caché?

4.5. Hay diferentes parámetros de diseño que son importantes para el rendimiento global de una caché. La tabla de abajo muestra los parámetros para dos diseños de caché diferentes.

	Tamaño de la caché (espacio de datos) (KB)	Tamaño del bloque de la caché (palabras)	Tiempo de acceso a la caché (ciclos)
Caso a	64	1	1
Caso b	64	2	2

Calcule el número total de bits que se requiere para construir la caché indicada en la tabla, asumiendo que las direcciones son de 32 bits. Dado el tamaño total, encuentre el tamaño total de la caché que más se aproxime a ella con correspondencia directa y bloques de caché de 16 palabras y que tenga un tamaño total mayor o igual que el propuesto. Explique por qué la segunda caché, a pesar de tener un espacio de datos mayor puede proporcionar un menor rendimiento que la primera.

4.6. La fórmula mostrada en la transparencia 4.7 muestra el método típico de indexación de una caché de correspondencia directa, en concreto (Dirección de bloque) módulo (Número de bloques en la caché). Suponiendo una dirección de 32 bits y 1024 bloques en la caché, considere una función de indexación distinta, en concreto (Dirección de bloque [31:27] XOR Dirección de bloque [26:22]). ¿Es posible utilizar esto para indexar una caché de correspondencia directa? Si lo es, explique el porqué y explique cualquier cambio que pudiese ser necesario hacer a la caché. Y si no es posible, explique el porqué.

4.7. Para el diseño de una caché de correspondencia directa con direcciones de 32 bits, se utilizan los siguientes bits de dirección para acceder a la caché:

	Etiqueta	Índice	Desplazamiento
Caso a	31-10	9-4	3-0
Caso b	31-12	11-5	4-0

i) ¿Cuál es el tamaño de la línea de caché, expresado en palabras?
ii) ¿Cuántas entradas tiene la caché?
iii) ¿Cuál es la razón entre el número total de bits requerido para construir esta implementación de caché y el número de los bits de almacenamiento de datos?
iv) Comenzando en el momento del encendido, se registran las siguientes referencias a palabra de caché, expresadas por medio de la dirección a la que se accede: 0, 4, 16, 132, 232, 160, 1024, 30, 140, 3100, 180, 2180. Deduzca cuántos bloques son reemplazados, cuál es la tasa de aciertos y cuál es el estado final de la caché. Para expresar esto último construya una tabla con entradas de la forma <índice, etiqueta, dato>, con una entrada para cada dirección involucrada.

4.8. El tamaño del bloque de caché (B) puede afectar tanto a la tasa de fallos como a la latencia por fallo. Suponiendo una tasa de fallos como la que se muestra en la tabla siguiente y una máquina de 1 CPI con un promedio de 1,35 referencias (incluidas las de datos e instrucciones) a memoria por instrucción, trataremos de encontrar el tamaño de bloque óptimo suponiendo que se tienen las siguientes tasas de fallo para los diversos tamaños de bloques:

	8 bytes	16 bytes	32 bytes	64 bytes	128 bytes
Caso a	8 %	3 %	1,8 %	1,5 %	2 %
Caso b	4 %	4 %	3,0 %	1,5 %	2 %

i) ¿Cuál es el tamaño de bloque óptimo para una penalización por fallo de 20xB ciclos?
ii) ¿Cuál es el tamaño de bloque óptimo para una penalización por fallo de (24+B) ciclos?
iii) Si la penalización por fallo fuera constante (esto es, que no dependiera del tamaño B): ¿cuál sería el tamaño de bloque óptimo?

4.9. En este ejercicio analizaremos de qué formas diversas afecta la capacidad de la caché al rendimiento total. Supóngase que un acceso a memoria principal tarda 70 ns y que el 36 % de todas las instrucciones son de acceso a memoria. Vamos a considerar sólo los accesos a la caché de datos. En la siguiente tabla se muestran los datos que caracterizan a las cachés L1 de dos procesadores, P1 y P2 respectivamente.

		Tamaño L1 (KB)	Tasa de fallos de L1	Tiempo de acierto en L1 (ns)
Caso a	P1	1	11,4 %	0,62
	P2	2	8,0 %	0,66
Caso b	P1	8	4,3 %	0,96
	P2	16	3,4 %	1,08

i) Suponiendo que el tiempo de acierto en L1 determina el tiempo de ciclo para P1 y P2, ¿cuáles son sus respectivas frecuencias de reloj?
ii) ¿Cuál es el AMAT para cada uno de los procesadores?
iii) Suponiendo un CPI base de 1.0 ¿cuál es el CPI total para cada procesador P1 y P2? ¿qué procesador es el más rápido?

4.10. En este ejercicio se examina el impacto de utilizar diferentes diseños de caché, en concreto comparando las cachés asociativas con las cachés de correspondencia directa. Para este ejercicio utilice las secuencias de direcciones que aparecen en el ejercicio 4.4.

i) Muestre el contenido final de la caché para una caché 3-asociativa con bloques de 2 palabras y un tamaño total de 24 palabras. Utilice reemplazo LRU. Para cada referencia identifique los bits de índice, los bits de etiqueta, los bits de desplazamiento dentro del bloque y si se trata de un acierto o de un fallo.

ii) Muestre el contenido final de la caché para una totalmente asociativa con bloques de una palabra y un tamaño total de 8 palabras. Utilice reemplazo LRU. Para cada referencia identifique los bits de índice, los bits de etiqueta y si se trata de un acierto o de un fallo.

iii) ¿Cuál es la tasa de fallos para una caché totalmente asociativa con bloques de dos palabras y un tamaño total de ocho palabras, utilizando reemplazo LRU? ¿Cuál es la tasa de fallos utilizando reemplazo MRU (*Most Recently Used*, más recientemente utilizado)? Y finalmente, ¿cuál es la mejor posible tasa de fallos para esta caché, dada cualquier política de reemplazo?

4.11. Supóngase una caché asociativa por conjuntos de dos vías con cuatro bloques. Le resultará útil dibujar una tabla para resolver este ejercicio, tal como la siguiente, con la secuencia de direcciones de bloque: 0, 1, 2, 3, 4.

Dirección del bloque al que se accede	Acierto o fallo	Bloque expulsado de la caché	Contenidos de los bloques de caché después de esa referencia			
			Conjunto 0		Conjunto 1	
			Bloque 0	Bloque 1	Bloque 0	Bloque 1
0	F		mem[0]			
1	F		mem[0]		mem[1]	
2	F		mem[0]	mem[2]	mem[1]	
3	F		mem[0]	mem[2]	mem[1]	mem[3]
4	F	0	mem[4]	mem[2]	mem[1]	mem[3]

A continuación se muestran secuencias de direcciones referenciadas (son direcciones a bloques de memoria):

Caso a: 0, 2, 4, 0, 2, 4, 0, 2, 4

Caso b: 0, 2, 4, 2, 0, 2, 4, 0, 2

i) Suponiendo una política de reemplazo LRU, ¿cuántos aciertos presenta esta secuencia de direcciones?

ii) Suponiendo una política de reemplazo MRU ("Más Recientemente Utilizado"), ¿cuántos aciertos presentan estas secuencias de direcciones?

iii) Simule una política de reemplazo aleatorio lanzando una moneda. Por ejemplo, "cara" significa expulsar el primer bloque del conjunto y "cruz" significa expulsar el segundo bloque del conjunto. ¿Cuántos aciertos presenta esta secuencia de direcciones?

iv) ¿Qué bloques deberían expulsarse en cada reemplazo para maximizar el número de aciertos? ¿Cuántos aciertos se consiguen con esta secuencia si sigue esta política "óptima"?

v) Describa por qué es difícil implementar una caché con una política de reemplazo que sea óptima para todas las secuencias de direcciones.

vi) Suponga que usted pudiese decidir sobre cada referencia a memoria si usted desea que la dirección pedida sea o no pasada a través de la caché. ¿Qué impacto podría tener esto sobre la tasa de fallos?

4.12. En este ejercicio vamos a tratar de estudiar el efecto de añadir una caché L2 a un procesador, para tratar de paliar los inconvenientes de la limitada capacidad de L1. Considere los datos del procesador P1 y su caché L1 del ejercicio 4.9. En la tabla siguiente se muestra la tasa de fallos local de la L2 utilizada.

	Tamaño de L2 (KB)	Tasa de fallos de L2	Tiempo de acierto en L2 (ns)
Caso a	512	98 %	3,22
Caso b	4096	73 %	11,48

i) ¿Cuál es el AMAT para P1 al añadir una caché L2? ¿Es mejor o peor que sin añadirla?

ii) Suponiendo un CPI base de 1,0 ¿cuál es el CPI total para P1 cuando se añade una caché L2?

iii) ¿Qué procesador es ahora más rápido, el P1 con su caché L2 o el P2 sólo con caché L1 del ejercicio 4.9? Si P1 es más rápido, ¿qué tasa de fallos necesitaría tener P2 en su caché L1 para alcanzar el rendimiento de P1? Si P2 es más rápido, ¿qué tasa de fallos debería tener P1 en su caché L1 para alcanzar el rendimiento de P2?

4.13. Continuemos estudiando los sistemas con varios niveles de caché. Considere un procesador con los siguientes parámetros:

	CPI base, sin detenciones por acceso a memoria	Velocidad del procesador (GHz)	Tiempo de acceso a memoria principal (ns)	Tasa de fallos de caché de primer nivel por instrucción (%)	Latencia de la caché de segundo nivel de correspondencia directa (ciclos)	Tasa de fallos global con caché de segundo nivel de correspondencia directa (%)	Latencia de caché de segundo nivel 8-asociativa (ciclos)	Tasa de fallos global con caché de segundo nivel 8-asociativa (%)
Caso a	2,0	3	125	5,0	15	3,0	25	1,8
Caso b	2,0	1	100	4,0	10	4,0	20	1,6

i) Calcular el CPI para el procesador de la tabla utilizando:
   1) Sólo una caché de primer nivel,
   2) Una caché de segundo nivel de correspondencia directa, y
   3) Una caché de segundo nivel 8-asociativa.

   ¿Cómo cambian los valores obtenidos si se duplica el tiempo de acceso a memoria principal? ¿Y si se reduce a la mitad?

ii) Es posible tener una jerarquía de más de dos niveles. Dado el procesador anterior con una caché de segundo nivel de correspondencia directa, un diseñador desea añadir un tercer nivel de caché que tiene un tiempo de acceso de 50 ciclos y que reduce la tasa de fallo global al 1,3 %. ¿Mejoraría esto el rendimiento? En general ¿cuáles son las ventajas y desventajas de añadir una caché de tercer nivel?

iii) En procesadores antiguos tales como el Pentium de Intel o el Alpha 21264, el segundo nivel de caché era externo (alojado en un chip diferente) respecto del procesador y del primer nivel de caché. Aunque esto permitía tener cachés de segundo nivel mayores, la latencia de acceso a la caché era mucho mayor, y el ancho de banda era típicamente menor porque la caché de segundo nivel funcionaba a menor frecuencia. Suponga una caché externa de segundo nivel de 512 KB con una tasa global de fallo del 4 %. Si cada 512 KB adicionales reducen la tasa de fallo global en un 0,7 %, y la caché tiene un tiempo de acceso de 50

ciclos, ¿qué tamaño debería tener la caché L2 para igualar el rendimiento de la caché L2 de correspondencia directa? ¿Y para igualar el de la 8-asociativa?

4.14. Las aplicaciones de los medios de reproducción tales como las que reproducen archivos de vídeo o de audio forman parte de un tipo de cargas de trabajo conocidas como cargas de trabajo de flujo (*streaming*); esto es, se mueven grandes cantidades de datos pero no se reutiliza gran cosa de ellos. Considere una carga de trabajo de flujo que accede a un conjunto de datos de trabajo de 512 KB de forma secuencial con la siguiente secuencia de direcciones: 0, 4, 8, 12, 16, 20, 24, 28, 32,…

i) Suponga una caché de correspondencia directa de 64 KB con líneas de 32 bytes. ¿Cuál es la tasa de fallos para la anterior secuencia de direcciones? ¿Cómo es de sensible esta tasa de fallos al tamaño de la caché o del conjunto de trabajo? ¿Cómo se podrían clasificar los fallos que aparecen en esta carga de trabajo basándose en el modelo 3C?

ii) Recalcule la tasa de fallos cuando el tamaño de la línea de caché es de 16 bytes, 64 bytes y 128 bytes. ¿Qué tipo de localidad está aprovechando esta carga de trabajo?

4.15. Para un sistema de alto rendimiento tal como un árbol binario de indexación de una base de datos, el tamaño de la página viene determinado principalmente por el tamaño de los datos y el rendimiento del disco. Suponga que, por término medio, una página de índice de árbol binario se llena al 70 % con entradas de tamaño fijo. La utilidad de una página es su profundidad en el árbol binario, calculada como $\log_2$(número de entradas). La tabla siguiente muestra que, para entradas de 16 bytes, y un disco de hace 10 años con latencia de 10 ms y una velocidad de transferencia de 10 MB/s, el tamaño óptimo de página es 16 KB.

Tamaño de página (KB)	Utilidad de la página o profundidad en el árbol binario (número de accesos a disco almacenados)	Coste del acceso al índice de página (ms)	Utilidad/ coste
2	6,49 (=$\log_2$(2048/16 x 0.7))	10,2	0,64
4	7,49	10,4	0,72
8	8,49	10,8	0,79
16	9,49	11,6	0,82
32	10,49	13,2	0,79
64	11,49	16,4	0,70
128	12,49	22,8	0,55
256	13,49	35,6	0,38

i) ¿Cuál sería el mejor tamaño de página si las entradas ahora tuviesen 128 bytes?

ii) Basándonos en el apartado anterior, ¿cuál es el mejor tamaño de páginas si las páginas están llenas hasta la mitad?

iii) Basándonos en el apartado anterior, ¿cuál es el mejor tamaño de página si utilizamos un disco moderno con una latencia de 3 ms y una velocidad de transferencia de 100 MB/s? Explique por qué los servidores del futuro probablemente tengan páginas de mayor tamaño.

4.16. Mantener en memoria las páginas "frecuentemente utilizadas" (ó también denominadas "calientes") puede ahorrar accesos a disco, pero ¿cómo determinar el significado exacto de "frecuentemente utilizada" para un determinado sistema? Los ingenieros de datos utilizan el cociente entre el coste de acceso a disco y el coste de DRAM para cuantificar el umbral del tiempo de reutilización para las páginas calientes. El coste de un acceso a disco es (precio del disco en $)/(número de accesos por segundo), mientras que el coste de mantener una página en DRAM es (precio de la DRAM en $ por MB)*(tamaño de la página en MB). En la siguiente tabla se muestran los costes típicos de DRAM y disco y los tamaños de páginas típicos para las bases de datos en distintos momentos:

Año	Coste DRAM ($/MB)	Tamaño de página (KB)	Coste del disco ($/disco)	Velocidad de acceso a disco (accesos/segundo)
1987	5000	1	15000	15
1997	15	8	2000	64
2007	0,05	64	80	83

i) ¿Cuáles son los umbrales de tiempo de reutilización para estas tres generaciones tecnológicas?

ii) ¿Cuáles serían los umbrales de tiempo de reutilización si mantuviésemos el mismo tamaño de página de 4 KB? ¿Cuál es la tendencia que se observa?

4.17. Sabemos que la memoria virtual utiliza una tabla de páginas para seguir la correspondencia entre direcciones virtuales y físicas. En este ejercicio se muestra cómo esta tabla necesita actualizarse a medida que se accede a las direcciones. La tabla siguiente muestra una secuencia de direcciones tal como se la ve en un sistema. Suponga páginas de 4 KB, un TLB de 4 entradas totalmente asociativo, y un reemplazo LRU verdadero. Si la página necesita traerse desde disco, debe alojarse en la página siguiente a la presente con número mayor.

Secuencias de direcciones:

Caso a	4095, 31272, 15789, 15000, 7193, 4096, 8912
Caso b	9452, 30964, 19136, 46502, 38110, 16653, 48480

Estado inicial del TLB:

Válido	Etiqueta	Número de página física
1	11	12
1	7	4
1	3	6
0	4	9

Contenido de la tabla de páginas:

Bit de Validez	1	0	0	1	1	1	0	1	0	0	1	1
Página física o "en disco" (D)	5	D	D	6	9	11	D	4	D	D	3	12

i) Dada la secuencia de direcciones de la tabla, y el estado inicial del TLB y la tabla de páginas, muestre el estado final del sistema. Además indique para cada referencia si se trata de un acierto en el TLB, un acierto en la tabla de páginas o un fallo de página.

ii) Repita el apartado anterior, pero esta vez utilice páginas de 16 KB en lugar de 4 KB. ¿Cuáles podrían ser algunas de las ventajas de tener un mayor tamaño de página? ¿Cuáles serían algunas de las desventajas?

iii) Muestre los contenidos finales del TLB si fuese 2-asociativo. Muestre también los contenidos del TLB si fuese de correspondencia directa. Explique la importancia de tener un TLB para obtener un rendimiento elevado ¿Cómo serían gestionados los accesos a memoria virtual si no hubiera TLB?

4.18. La tabla siguiente muestra el contenido de un TLB de cuatro entradas:

Identificador de entrada	Válido	Página virtual	Modificado	Protección	Página física
1	1	140	1	RW	30
2	0	40	0	RX	34
3	1	200	1	RO	32
4	1	280	0	RW	31

i) ¿Qué podría haber pasado para que el bit de válido de la entrada 2 se pusiese en 0?
ii) ¿Qué ocurre cuando una instrucción escribe sobre la página virtual 30?
iii) ¿Qué ocurre cuando una instrucción intenta escribir sobre la página virtual 200?

4.19. La siguiente tabla muestra varios parámetros de un sistema de memoria virtual:

	Dirección virtual	Tamaño de la DRAM física	Tamaño de página	Tamaño PTE
Caso a	32 bits	4 GB	8 KB	4 bytes
Caso b	64 bits	16 GB	4 KB	8 bytes

¿Cuántas entradas de tabla de página (PTE) se necesitan? ¿Cuánta memoria física es necesaria para almacenar la tabla de páginas?