5.8.5 [10]<5.3> Es posible tener una jerarquía de memoria con más de dos niveles de cache. Dado el procesador anterior con una cache de segundo nivel de correspondencia directa, se desea añadir un tercer nivel de cache con un tiempo de acceso de 50 ciclos y que reduce la frecuencia de fallos global al 1.3%. ¿Se mejorarían las prestaciones? En general, ¿cuáles son las ventajas y desventajas de añadir un tercer nivel de cache?

5.8.6 [20]<5.3> En procesadores antiguos, como el Pentium o el Alpha 21264, el segundo nivel cache era externo (situado en un chip diferente) al procesador principal y al primer nivel de cache. Esto permitía disponer de caches de segundo nivel de gran capacidad pero la latencia de acceso era mucho mayor y el ancho de banda típicamente era menor debido a que su frecuencia de reloj era más baja. Suponga una cache de segundo nivel de 512 KB externa y con una frecuencia de fallos global del 4%. Si cada 512 KB adicionales hiciesen disminuir la frecuencia de fallos global en un 0.7% y la cache tuviese un tiempo de acceso total de 50 ciclos, ¿qué tamaño debería tener la cache para igualar las prestaciones de la cache de segundo nivel de correspondencia directa con las características de la tabla? ¿Y de la cache asociativa con conjuntos de ocho vías?

Ejercicio 5.9

En sistemas de altas prestaciones, como por ejemplo el índice de árboles-B de una base de datos, el tamaño de página se determina principalmente en función del tamaño de los datos y las prestaciones del disco. Suponga que, en promedio, la página de índice de árboles-B está llena al 70% con entradas de tamaño fijo. La utilidad de la página es su profundidad de árbol-B, definida como el $\log_2$ (entradas). La siguiente tabla muestra, para entradas de 16 bytes y un disco de hace 10 años con una latencia de 10 ms y un ritmo de transferencia de 10 MB/s, que el tamaño de página óptimo es 16K

Tamaño de página (KB)	Utilidad de la página o profundidad del árbol-B (número de accesos a disco guardados)	Coste de acceso del índice de la página	Utilidad/coste
2	6.49 (o log2(2048/16x0.7))	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

5.9.1 [10]<5.4> ¿Cuál es el mejor tamaño de página si las entradas son de 128 bytes?

5.9.2 [10]<5.4> Basándose en el problema 5.9.1, ¿cuál es el mejor tamaño de página si las páginas están llenas al 50%?

5.9.3 [20]<5.4> Basándose en el problema 5.9.2, ¿cuál es el mejor tamaño de página si se utiliza un disco moderno con una latencia de 3 ms y un ritmo de trans-

ferencia de 100 MB/s? Explique por qué los servidores futuros probablemente tendrán páginas de mayor tamaño.

El número de accesos al disco se puede reducir si se guarda en DRAM las páginas "usadas frecuentemente" (páginas "calientes"), pero ¿cómo se determina el significado exacto de páginas "usadas frecuentemente" en un sistema? Normalmente se utiliza la relación de coste entre accesos a DRAM y disco para cuantificar el umbral de tiempo de reuso de las páginas calientes. El coste de un acceso a disco es $Disco/accesos_por_segundo mientras que el coste de mantener una página en DRAM es $DRAM_MB/tamaño_de_página. Los costes típicos de DRAM y disco y los tamaños de páginas de bases de datos típicos en varios años se muestran en la tabla:

Año	Coste de DRAM ($/MB)	Tamaño de página (KB)	Coste de disco ($/disco)	Frecuencia de acceso a disco (accesos/seg)
1987	5000	1	15 000	15
1997	15	8	2000	64
2007	0.05	64	80	83

5.9.4 [10]<5.1, 5.4> ¿Cuáles son los umbrales de tiempo de reutilización para esas tres generaciones de la tecnología?

5.9.5 [10]<5.4> ¿Cuáles son los umbrales de tiempo de reutilización si se hubiese mantenido un tamaño de página de 4K? ¿Cuál es la tendencia?

5.9.6 [20]<5.4> ¿Qué otros factores pueden cambiarse para seguir utilizando el tamaño de página de 4K (evitando así reescrituras software)? Discuta sus opciones de uso con las tendencias actuales de tecnología y costes.

Ejercicio 5.10

Como se ha descrito en la sección 5.4, la memoria virtual utiliza una tabla de páginas para rastrear la traducción de direcciones virtuales a direcciones físicas. Este ejercicio muestra cómo actualizar esta tabla a medida que se accede a la direcciones. La siguiente tabla contiene una secuencia de direcciones virtuales. Suponga páginas de 4 KB, una TLB totalmente asociativa de cuatro entradas y una estrategia de reemplazo LRU verdadero. Incremente el siguiente mayor número de página, en caso de que la página se tenga que traer desde el disco.

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

TLB

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

Tabla de páginas

Válido	Página física o en disco
1	5
0	Disco
0	Disco
1	6
1	9
1	11
0	Disco
1	4
0	Disco
0	Disco
1	3
1	12

5.10.1 [10]<5.4> Dadas las referencia de la tabla y el estado inicial de la TLB y la tabla de páginas mostrados, determine el estado final del sistema. Indique además para cada referencia de memoria si es un acierto de TLB, un acierto en la tabla de páginas o un fallo de página.

5.10.2 [15]<5.4> Repita el problema 5.10.1 utilizando páginas de 16 KB en lugar de 4KB. ¿Cuáles son las ventajas de tener un mayor tamaño de página? ¿Cuáles son las desventajas?

5.10.3 [15]<5.3, 5.4> Muestre los contenidos finales de la TLB si fuese una TLB asociativa por conjuntos de dos vías. Muestre los contenidos también en caso de una TLB de correspondencia directa. Discuta la importancia de disponer de una TLB en aplicaciones de altas prestaciones. ¿Cómo se manejarían los accesos a memoria virtual si no hubiese TLB?

Hay varios parámetros que influyen en el tamaño de la tabla de páginas. En la tabla se muestran varios parámetros clave para la tabla de páginas:

	Tamaño de la dirección virtual	Tamaño de página	Tamaño de las entradas de la tabla de páginas
a.	32 bits	4 KB	4 bytes
b.	64 bits	16 KB	8 bytes

5.10.4 [5]<5.4> Determine el tamaño total de la tabla de páginas para un sistema que ejecuta cinco aplicaciones que utilizan la mitad de la memoria disponible.

5.10.5 [10]<5.4> Determine el tamaño total de la tabla de páginas para un sistema que ejecuta cinco aplicaciones que utilizan la mitad de la memoria disponible, suponiendo que se dispone de una tabla de páginas de dos niveles con 256 entradas. Suponga que cada entrada de la tabla de páginas principal es de 6 bytes. Calcule la cantidad de memoria mínima y máxima necesaria.

5.10.6 [10]<5.4> Un diseñador de caches quiere aumentar el tamaño de una cache de 4 KB indexada virtualmente y etiquetada físicamente. Dados los tamaños de página de la tabla, ¿es posible construir una cache de correspondencia directa de 16 KB, suponiendo dos palabras por bloque? ¿Cómo debería aumentarse el tamaño de los datos de la cache?

Ejercicio 5.11

En este ejercicio analizamos las optimizaciones espacio/tiempo de las tablas de páginas. La siguiente tabla muestra varios parámetros de un sistema de memoria virtual.

	Dirección virtual (bits)	Tamaño de la DRAM física	Tamaño de página	Tamaño PTE (byte)
a.	32	4 GB	8 KB	4
b.	64	16 GB	4 KB	8

5.11.1 [10]<5.4> Para una tabla de páginas en un único nivel, ¿cuántas entradas de la tabla de páginas (PTE) se necesitan? ¿Cuánta memoria física es necesaria para almacenar la tabla de páginas?

5.11.2 [10]<5.4> La utilización de una página de tablas en varios niveles permite mantener en memoria física sólo las PTE activas, reduciendo así las necesidades de memoria de la tabla de páginas. ¿Cuántos niveles de tablas de páginas se necesitarán en este caso? ¿Cuántos accesos a memoria serán necesarios para la traducción de la dirección si se produce un fallo de TLB?

5.11.3 [15]<5.4> Se puede utilizar una tabla invertida de páginas para optimizar el espacio y el tiempo. ¿Cuántas PTE son necesarias para almacenar la tabla de páginas? Suponiendo una implementación con funciones almohadilla, ¿cuáles son el caso común y el peor caso del número de referencias a memoria necesarios para procesar un fallo de TLB?

La siguiente tabla muestra los contenidos de una TLB de cuatro entradas.

Entrada	Válida	Dirección virtual de la página	Modificada	Protección	Dirección física de la página
1	1	140	1	RW	30
2	0	40	0	RX	34
3	1	200	1	RO	32
4	1	280	0	RW	31

5.11.4 [5]<5.4> ¿En qué casos el bit de validez de la entrada 2 será 0?

5.11.5 [5]<5.4> ¿Qué ocurre cuando una instrucción escribe en la dirección virtual 30? ¿En qué caso una TLB manejada por software es más rápida que una TLB en hardware?

5.11.6 [5]<5.4> ¿Qué ocurre cuando una instrucción escribe en la dirección virtual xxx?

Ejercicio 5.12

En este ejercicio analizaremos cómo influyen las estrategias de reemplazo en frecuencia de fallos. Suponga una cache asociativa por conjuntos de dos vías con cuatro bloques. Podría serle de utilidad hacer una tabla como la de la página 483 para resolver los problemas de este ejercicio, como se muestra a continuación para la secuencia de direcciones "0, 1, 2, 3, 4".

Dirección de memoria del bloque accedido	Fallo/ acierto	Bloque expulsado	Contenido de los bloques de cache después de la referencia			
			Conjunto 0	Conjunto 0	Conjunto 1	Conjunto 1
0	fallo		Mem[0]			
1	fallo		Mem[0]		Mem[1]	
2	fallo		Mem[0]	Mem[2]	Mem[1]	
3	fallo		Mem[0]	Mem[2]	Mem[1]	Mem[3]
4	fallo	0	Mem[4]	Mem[2]	Mem[1]	Mem[3]
...						

La siguiente tabla muestra la secuencia de direcciones:

	Secuencia de direcciones
a.	0, 2, 4, 0, 2, 4, 0, 2, 4
b.	0, 2, 4, 2, 0, 2, 4, 0, 2

5.12.1 [5]<5.3, 5.5> Suponiendo una estrategia de reemplazo LRU, ¿cuántos aciertos se producirán en esta secuencia de direcciones?

5.12.2 [5]<5.3, 5.5> Suponiendo una estrategia de reemplazo MRU (más recientemente usado), ¿cuántos aciertos se producirán en esta secuencia de direcciones?

5.12.3 [5]<5.3, 5.5> Simule una estrategia de reemplazo aleatorio echando una moneda al aire. Por ejemplo, "cara" significa expulsar el primer bloque del conjunto y "cruz" significa expulsar el segundo bloque del conjunto. ¿Cuántos aciertos se producirán en esta secuencia de direcciones?

5.12.4 [10]<5.3, 5.5> ¿Qué dirección debería ser expulsada en cada reemplazo para maximizar el número de aciertos? ¿Cuántos aciertos se producirán en esta secuencia de direcciones con esta estrategia "óptima"?

5.12.5 [10]<5.3, 5.5> Describa por qué es difícil implementar una estrategia de reemplazo que sea óptima para cualquier secuencia de direcciones.

5.12.6 [10]<5.3, 5.5> Suponga que para cada referencia de memoria se puede decidir si la dirección solicitada se lleva o no a la cache. ¿Qué impacto podría tener en la frecuencia de fallos?

Ejercicios 5.13

Para soportar varias máquinas virtuales se requieren dos niveles de virtualización. Cada máquina virtual controla la traducción de dirección virtual (VA) a dirección física (PA), y el hipervisor traduce la dirección física (PA) de cada máquina virtual a dirección de la máquina (MA). Para acelerar las traducciones se utiliza una técnica software denominada "tabla de páginas acompañante (shadow paging)" que duplica cada tabla de páginas de la máquina virtual en el hipervisor, e intercepta cambios en las traducciones VA a PA para mantener copias consistentes. Para eliminar la complejidad de las tablas de páginas acompañantes, una estrategia hardware, denominada tabla de páginas anidada (o tabla de páginas extendida), soporta de forma explícita dos tipos de tablas de páginas (VA => PA y PA => MA) y puede procesar estas tablas puramente en hardware.

Consideremos la siguiente secuencia de operaciones:

(1) crear proceso; (2) fallo de TLB; (3) fallo de página; (4) cambio de contexto

5.13.1 [10]<5.4, 5.6> Para esta secuencia de operaciones, indique qué ocurriría al utilizar la estrategia de tablas de páginas sombreadas y la estrategia de tabla de páginas anidadas.

5.13.2 [10]<5.4, 5.6> Suponiendo una tabla de páginas en cuatro niveles y basada en el x86 tanto en la tabla de páginas invitada como en la tabla de páginas anidada, ¿cuántas referencias de memoria son necesarias para procesar un fallo de TLB en la tabla de páginas nativa y en la tabla de páginas anidada?

5.13.3 [10]<5.4, 5.6> Entre frecuencia de fallos de TLB, latencia de fallo de TLB, frecuencia de fallos de páginas y latencia del manejador del fallo de página, ¿qué métrica es más importante para una tabla de páginas sombreada? ¿Cuáles son importantes para una tabla de páginas anidada?

La siguiente tabla muestra algunos parámetros de un sistema de paginado sombreado.

Fallos de TLB por cada 1000 instrucciones	Latencia de fallo de TLB de tabla de páginas anidada	Fallos de página por cada 1000 instrucciones	Sobrecoste de fallos de página sombreado
0.2	200 ciclos	0.001	30 000 ciclos

5.13.4 [10]<5.6> Para un programa de prueba con una ejecución nativa con CPI = 1, ¿cuál es el CPI con tablas de páginas sombreadas con respecto al CPI con tablas de páginas anidadas (suponiendo sólo sobrecoste de virtualización de tablas de páginas)?

5.13.5 [10]<5.6> ¿Qué técnicas pueden utilizarse para reducir el sobrecoste introducido por el sombreado de la tabla de páginas?

5.13.6 [10]<5.6> ¿Qué técnicas pueden utilizarse para reducir el sobrecoste introducido por la tabla de páginas anidada?

Ejercicio 5.14

Una de los mayores impedimentos para la difusión de la máquinas virtuales es el sobrecoste introducido por su ejecución en el sistema nativo. En la tabla se muestran varios parámetros de las prestaciones y del comportamiento.

	CPI base	Accesos privilegiados del SO por cada 1000 instrucciones	Impacto de las llamadas al SO invitado sobre las prestaciones	Impacto de las excepciones al MMV sobre las prestaciones	Accesos de E/S por cada 1000 instrucciones	Tiempo de acceso a E/S (incluye llamada al SO invitado)
a.	2	100	20 ciclos	150 ciclos	20	1000 ciclos
b.	1.5	110	25 ciclos	160 ciclos	10	1000 ciclos

5.14.1 [10]<5.6> Calcule el CPI suponiendo que no hay accesos a E/S. ¿Cuál sería el CPI si se duplica el impacto de la MMV? ¿Y si se reduce a la mitad? Si un desarrollador de software de máquina virtual quiere obtener una degradación en las prestaciones del 10%, ¿cuál sería la mayor penalización posible de las excepciones al MMV?

5.14.2 [10]<5.6> Los accesos de E/S tienen un impacto elevado sobre las prestaciones del sistema. Calcule el CPI de un máquina con las características de la tabla, suponiendo un sistema no virtualizado. Calcule el CPI otra vez suponiendo un sistema virtualizado. ¿Cómo cambiaría el CPI si el número de accesos de E/S se redujese a la mitad? Explique por qué las aplicaciones limitadas por E/S son menos afectadas por la virtualización.

5.14.3 [30]<5.6> Compare y contraste las ideas de memoria virtual y máquina virtual. ¿Cuáles son sus objetivo? ¿Cuáles son los pros y contras de cada una? Señale casos en los que es deseable tener memoria virtual y casos en los que es deseable tener una máquina virtual.

5.14.4 [20]<5.6> En la sección 5.6 se ha discutido la virtualización bajo la suposición de que el sistema virtualizado tiene la misma ISA que el hardware subyacente. Sin embargo, un posible uso de la virtualización es la emulación de una ISA no nativa. Un ejemplo es QEMU, que emula varias ISA, por ejemplo, MIPS, SPARC y PowerPC. Indique algunas de las dificultades de esta clase de virtualización. ¿Es posible que el sistema emulado sea más rápido que la ejecución en la ISA nativa?

Ejercicio 5.15

En este ejercicio se analiza la unidad de control de un controlador de cache para un procesador con un búfer de escritura. Como punto de partida para el diseño de la máquina de estados finitos, se utiliza la máquina de estados finitos de la figura 5.34. Suponga que el controlador es para una cache de correspondencia directa sencilla, como la descrita en la página 530, pero con un búfer de escritura añadido, con capacidad de un bloque.

Recuerde que el objetivo del búfer de escritura es servir como un almacenamiento temporal para que el procesador no tenga que esperar por dos accesos a memoria en el procesamiento de un fallo en un bloque inconsistente. En lugar de escribir en memoria el bloque antes de leer el nuevo bloque, se almacena el bloque inconsistente en el búfer y se comienza de forma inmediata la lectura del nuevo bloque. De este modo, el bloque inconsistente se escribe en memoria mientras el procesador está trabajando.

5.15.1 [10]<5.5, 5.7> ¿Qué ocurriría si el procesador hace una referencia que produce un acierto en la cache mientras se está escribiendo un bloque del búfer de escritura en la memoria?

5.15.2 [10]<5.5, 5.7> ¿Qué ocurriría si el procesador hace una referencia que produce un fallo en la cache mientras se está escribiendo un bloque del búfer de escritura en la memoria?

5.15.3 [10]<5.5, 5.7> Diseñe una máquina de estados finitos que permita el uso del búfer de escritura.

Ejercicio 5.16

La coherencia cache incumbe a cómo acceden varios procesadores a un bloque de cache. La siguiente tabla muestra dos procesadores y sus operaciones de lectura/escritura en palabras diferentes de un bloque de cache X (inicialmente X[0] = X[1] = 0).

	P1	P2
a.	`X[0] ++; X[1] =4;`	`X[0] = 2; X[1] ++;`
b.	`X[0] ++; X[1] += 3;`	`X[0] = 5; X[1] = 2;`