



Ejercicios de memoria caché

Ejercicio 1

Sea un computador con una caché de nivel 1 (L1) y memoria principal con las siguientes latencias:

- Acierto de caché: 1 ciclo.
- Fallo de caché: 100 ciclos.
- Acceso a memoria principal con caché desactivada: 90 ciclos.

Ejercicio 1.1:

Si la tasa de fallos global es del 5%, ¿cuál será el tiempo medio de acceso a memoria?

$$T = (1-m) * t(\text{acierto}) + m * t(\text{fallo}) = 0.95 * 1 + 0.05 * 100 = 5.95 \text{ ciclos}$$

Ejercicio 1.2:

Se tiene un programa que realiza accesos a un array que tiene un tamaño de 128 MB. Si la caché L1 tiene un tamaño de 32 KB ¿Cuál será el tiempo medio de acceso a memoria? Asuma que los accesos al array se realizan de forma totalmente aleatoria usando un generador uniforme de números aleatorios.

Como la probabilidad de acceso es uniforme a todas las posiciones del array, la probabilidad de que una posición esté en la caché se estima como la proporción entre el tamaño de la caché y el tamaño del array.

$$(1-m) = 32 / (128 * 1024) = 2^5 / 2^{17} = 1/2^{12} = 0.000244$$

$$T = (0.000244) * 1 + (1-0.000244) * 100 = 99,98$$

Ejercicio 1.3:

Compare con los resultados con los que se obtendrían si se desactiva la memoria caché. ¿A qué se debe la diferencia?

Con la cachés desactivada el tiempo medio de acceso sería de 90 ciclos. Al activar la caché se obtiene un rendimiento peor, puesto que el patrón de accesos es totalmente aleatorio y no se obtiene ningún beneficio debido a la localidad.

Ejercicio 1.4:

La ganancia (G) obtenida por la caché en caso de acierto es de 89 (90-1). Sin embargo, en caso de fallo la pérdida (P) es de 10 (90-100). Para unos valore arbitrarios de G y P, determine la tasa de fallos más alta que se puede tener sin que los resultados sean peores que desactivando la caché.



Sean $T(a)$ y $T(d)$ los tiempos de acceso medio cuando la memoria caché está activada y desactivada respectivamente.

$$T(\text{acierto}) = T(d) - G$$

$$T(\text{fallo}) = T(d) + P$$

$$T(a) = (1-m) * (T(d) - G) + m * (T(d) + P)$$

El punto de corte (T) será aquel en el que $T(a) = T(d) = T$.

$$T = (1-m) * (T - G) + m * (T + P) = T - G - m * T + m * G + m * T + m * P = T - G + m * G + m * P$$

$$G = m * (G + P)$$

$$m = G / (G + P)$$

Para el caso del apartado a) se tiene:

$$m = 89 / 99 = 0.899$$

Ejercicio 2

Sea un computador con una memoria caché de 512 bytes con bloques de 64 bytes, que tiene una memoria principal de 2 KB. Dicha memoria se puede considerar como una secuencia de 32 bloques de memoria ($M_0, M_1, \dots, M_{31}$) de 64 bytes cada uno.

Ejercicio 2.1

Indique para cada posición de la memoria caché todos los bloques que podrían almacenarse si se trata de una memoria caché que usa correspondencia directa.

Bloque de caché	Bloques de memoria
0	M_0, M_8, M_{16}, M_{24}
1	M_1, M_9, M_{17}, M_{25}
2	...
3	...
4	...
5	...
6	...
7	$M_7, M_{15}, M_{23}, M_{31}$



Ejercicio 2.2

Indique para cada posición de la memoria caché todos los bloques que podrían almacenarse si se trata de una memoria caché que usa correspondencia asociativa por conjuntos de 4 vías.

Bloque de caché	Conjunto	Vía	Bloques de memoria
0	0	0	M0, M2, M4, ..., M30
1	0	1	M0, M2, M4, ..., M30
2	0	2	M0, M2, M4, ..., M30
3	0	3	M0, M2, M4, ..., M30
4	1	0	M1, M3, M5, ..., M31
5	1	1	M1, M3, M5, ..., M31
6	1	2	M1, M3, M5, ..., M31
7	1	3	M1, M3, M5, ..., M31

Ejercicio 2.3

Indique para cada posición de la memoria caché todos los bloques que podrían almacenarse si se trata de una memoria caché que usa correspondencia totalmente asociativa.

Bloque de caché	Bloques de memoria
0	M0, M1, M2,..., M31
1	M0, M1, M2,..., M31
2	M0, M1, M2,..., M31
3	M0, M1, M2,..., M31
4	M0, M1, M2,..., M31
5	M0, M1, M2,..., M31
6	M0, M1, M2,..., M31
7	M0, M1, M2,..., M31

Ejercicio 3

Sea una memoria caché que está organizada físicamente en:

- Una tabla de datos, que contiene los bloques de datos.
- Una tabla de etiquetas.
- Una tabla de remplazo, donde cada entrada contiene la información que el algoritmo de remplazo necesita.

Al ser la memoria asociativa por conjuntos cada uno de estas tablas se organiza de forma física en sub-tablas, con una sub-tabla para cada vía de la caché.

Las tablas de remplazo se acceden:

- LRU: Una vez por acceso.



- FIFO: Una vez por fallo.
- Aleatoria: No se accede a las tablas de remplazo.

Suponga los siguientes consumos (expresados en unidades de potencia) por acceso a las tablas:

- Tabla de datos: 20.
- Tabla de etiquetas: 5.
- Resto: 1.

Se desea estimar los consumos de potencia para varias configuraciones asumiendo una caché asociativa de cuatro vías, sin considerar el consumo de la memoria principal. Compare los resultados para las tres políticas LRU, FIFO y aleatoria.

1. Un acierto de caché. Todas las tablas se leen simultáneamente.
2. Un fallo de caché.
3. Un acierto de caché. El acceso a la caché se divide en dos ciclos. En el primer ciclo se accede a todas las tablas de etiquetas. En el segundo ciclo se accede, solo la sub-tabla que generó un acierto se accede.
4. Repita el caso (3) para un fallo de caché. No hay accesos a la tabla de datos en el segundo ciclo.
5. Repita el caso (3) asumiendo que se añade lógica para predecir la vía de la caché que se accederá. En el primer ciclo solamente se accede a la sub-tabla de la vía predicha. Un acierto de vía implica un acierto de caché. Un fallo de vía fuerza a examinar todas las sub-tablas de de etiquetas en el segundo ciclo. En caso de acierto de vía solamente la sub-tabla de datos que generó acierto, se accede el segundo ciclo. Asuma que ocurre un acierto de vía.
6. Repita el caso (5) asumiendo que el predictor de vía falló. Cuando falla, el predictor de vía añade un ciclo extra en el que accede a todas las sub-tablas de etiquetas. Asuma un acierto de lectura de caché.
7. Repita el caso (6) asumiendo un fallo de lectura de caché.
8. Use los apartados (5), (6) y (7) para el caso general en el que la carga de trabajo tiene:
 - Tasa de fallos del predictor de vía: 5%.
 - Tasa de fallos de caché: 3%

Ejercicio 3.1

En todos los casos se necesita acceder a las tablas de las cuatro vías.

- LRU: En cada vía se necesita acceder a las 3 sub-tablas.
 - $Potencia = 4 * (20+5+1) = 104 \text{ UP}$
- FIFO: En cada vía se necesita acceder a las sub-tablas de datos y etiquetas. No se accede a la sub-tabla de remplazo por tratarse de un acierto.



- $\text{Potencia} = 4 * (20+5) = 100 \text{ UP}$
- Aleatoria: En cada vía se necesita acceder a las sub-tablas de datos y etiquetas. No se accede a la sub-tabla de remplazo.
 - $\text{Potencia} = 4 * (20+5) = 100 \text{ UP}$

Ejercicio 3.2

En todos los casos se necesita acceder a las tablas de las cuatro vías.

- LRU: En cada vía se necesita acceder a las 3 sub-tablas.
 - $\text{Potencia} = 4 * (20+5+1) = 104 \text{ UP}$
- FIFO: En cada vía se necesita acceder a las sub-tablas de datos y etiquetas. Se accede a la sub-tabla de remplazo solamente en una vía.
 - $\text{Potencia} = 4 * (20+5) + 1 = 101 \text{ UP}$
- Aleatoria: En cada vía se necesita acceder a las sub-tablas de datos y etiquetas. No se accede a la sub-tabla de remplazo.
 - $\text{Potencia} = 4 * (20+5) = 100 \text{ UP}$

Ejercicio 3.3

- LRU: En el primer ciclo se accede a las sub-tablas de etiquetas y a las de remplazo (se accede a las de remplazo sin importar si hay fallo o acierto). En el segundo ciclo se accede a la sub-tabla de datos de la vía seleccionada.
 - $\text{Potencia} = 4 * (5+1) + 20 = 44 \text{ UP}$
- FIFO: No hay acceso a las sub-tablas de remplazo en caso de acierto.
 - $\text{Potencia} = 4 * 5 + 20 = 40 \text{ UP}$
- Aleatoria: No hay acceso a las sub-tablas de remplazo en ningún caso.
 - $\text{Potencia} = 4 * 5 + 20 = 40 \text{ UP}$

Ejercicio 3.4

- LRU:
 - $\text{Potencia: } 4 * (5+1) = 24 \text{ UP}$
- FIFO:
 - $\text{Potencia: } 4 * 5 = 20 \text{ UP}$
- Aleatoria:
 - $\text{Potencia: } 4 * 5 = 20 \text{ UP}$

Ejercicio 3.5

- LRU: En el primer ciclo se accede a la sub-tabla de etiquetas y a la sub-tabla de remplazo. En el segundo ciclo se accede a la sub-tabla de datos.
 - $\text{Potencia: } 5+1+20 = 26 \text{ UP}$
- FIFO: No hay acceso a la sub-tabla de remplazo por tratarse de un acierto.
 - $\text{Potencia: } 5+ 20 = 25 \text{ UP}$



- Aleatoria: No hay acceso a la sub-tabla de remplazo.
 - Potencia: $5 + 20 = 25$ UP

Ejercicio 3.6

- LRU: En el primer ciclo se realiza un acceso a la sub-tabla de etiquetas y a la de remplazo de la vía predicha. En el segundo ciclo se accede a todas las sub-tablas de etiquetas y de remplazo. Se accede a la sub-tabla de datos de la vía seleccionada.
 - Potencia = $5+1 + 4 * (5+1) + 20 = 50$ UP
- FIFO: Al tratarse de un acierto de lectura no se realizan accesos a las sub-tablas de remplazo.
 - Potencia = $5 + 4 * 5 + 20 = 45$ UP
- Aleatoria: No se realizan accesos a las sub-tablas de remplazo.
 - Potencia: $5 + 4 * 5 + 20 = 45$ UP

Ejercicio 3.7

- LRU: En el primer ciclo se realiza un acceso a la sub-tabla de etiquetas y a la de remplazo de la vía predicha. En el segundo ciclo se accede a todas las sub-tablas de etiquetas y de remplazo.
 - Potencia = $5+1 + 4 * (5+1) = 30$ UP
- FIFO: Al tratarse de un fallo de lectura si se realizan un acceso a la sub-tabla de remplazo.
 - Potencia = $5 + 4 * 5 + 1 = 26$ UP
- Aleatoria: No se realizan accesos a las sub-tablas de remplazo.
 - Potencia: $5 + 4 * 5 = 25$ UP

Ejercicio 3.8

Se tienen tres casos a considerar:

- Acierto de vía y acierto de caché: 95%.
- Fallo de vía y fallo de caché: 3%.
- Fallo de vía y acierto de caché: 2%.

	Acuerdo de vía y acierto de caché	Fallo de vía y fallo de caché	Fallo de vía y acierto de caché	Total
Probabilidad	95%	3%	2%	
LRU	26	30	50	26.6
FIFO	25	26	45	25.43
Aleatoria	25	25	45	25,40



Ejercicio 4

Sea una memoria caché de 64 KB con un tamaño de línea de 32 bytes. La caché asigna una línea en fallos de escritura. Si la caché usa post-escritura escribirá la línea completa que esté modificada y necesite remplazarse. La caché está conectada al siguiente nivel de la jerarquía mediante un bus de 64 bits (8 bytes). El número de ciclos de CPU para un acceso de escritura en este bus de B-bytes es $10 + 5 * (\text{INT}[B/8] - 1)$.

Nota: INT aproxima una fracción al entero superior más próximo.

Sea el siguiente segmento de código en C

```
#define PORCION 1
unsigned base = 8 * i;
for (unsigned int j = base; j < base+PORCION; j++) // j está en registro
    data[j] = j;
```

Ejercicio 4.1

En el caso de una caché de escritura inmediata. ¿Cuántos ciclos de CPU se emplean en transferencias de escritura a memoria para todas las iteraciones del bucle j?

Solamente hay una iteración en la que se transfiere una palabra (4 bytes).

$$\text{Ciclos} = 10 + 5 (1-1) = 10$$

Ejercicio 4.2

En el caso de una caché de post-escritura ¿Cuántos ciclos de CPU se emplean en transferencias de escritura a memoria para todas las iteraciones del bucle j?

En este caso se escriben 32 bytes (tamaño de línea).

$$\text{Ciclos} = 10 + 5 * (4-1) = 25 \text{ ciclos}$$

Ejercicio 4.3

¿Cuál sería el resultado del ejercicio 4.1 si se cambia la definición de PORCION a 8?

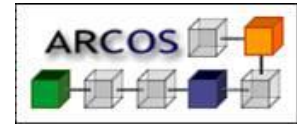
En este caso se realizarían 8 iteraciones y cada una generaría una escritura independiente.

$$\text{Ciclos} = 8 * 10 = 80$$

Ejercicio 4.4

¿Cuál es número mínimo de actualizaciones de array a la misma línea de la caché (antes de remplazarla) que haría que la estrategia de post escritura sea superior?

Para un número n de accesos se tiene:



- Escritura directa: $10 * n$
- Post-escritura: 25

Por tanto en cuanto el número de accesos sea mayor o igual que 3 la estrategia de post-escritura requiere menos ciclos.