

HOJA DE EJERCICIOS

PRÁCTICA 6: JERARQUÍA DE MEMORIA II

ESTA SESIÓN NO TIENE EVALUACIÓN, PERO SE CONTABILIZARÁ LA ASISTENCIA MEDIANTE UNA HOJA DE FIRMAS.

DINÁMICA DE LA SESIÓN:

En esta sesión, se pretende afianzar los conocimientos acerca del funcionamiento de la jerarquía de memoria mediante la realización de ejercicios a mano en los que se medirá el tiempo medio de acceso al sistema de memoria en base al programa o la traza de accesos ejecutada.

Así pues, esta hoja de ejercicios se divide en dos partes:

- Parte 1 – Ejercicio a realizar por el profesor en pizarra.
- Parte 2 – Ejercicio a realizar por el alumno a mano.

PARTE 1: EJERCICIO DEL PROFESOR

Partimos del siguiente código en alto nivel:

```
int i, j, x[3], y[3][2]={...};
for(i=0; i<3; i++)
{
    for(j=0; j<2; j++)
    {
        x[i] = x[i]+y[i][j]; //primero se evalúa la parte derecha de la asignación
    }                       // y, dentro de ella, de izquierda a derecha
}
```

Dispone de un sistema con las siguientes características:

- Procesador de 32 bits de palabra.
- Memoria Principal máxima direccionable de 4KBytes, con tamaño de bloque de 8Bytes.
- Memoria Caché de 32Bytes de tamaño para datos, CB-WA, Mapeado Directo.

Disposición inicial de la Memoria Principal:

Memoria Principal		
B0 (Dir. 0x000)	x[0]	x[1]
B1 (Dir. 0x008)	x[2]	y[0][0]
B2 (Dir. 0x010)	y[0][1]	y[1][0]
B3 (Dir. 0x018)	y[1][1]	y[2][0]
B4 (Dir. 0x020)	y[2][1]	
...		
B511 (Dir. 0xFF8)		

- a)** Represente gráficamente: la MC (con todos sus campos y tamaños), la interpretación de la dirección de MP y de MC. Indique los cálculos realizados.

ac-us.es

b) Rellene la siguiente tabla de accesos a los datos del programa anterior (si algún campo no procede, indíquelo con “-“, no lo deje en blanco). Recuerde la disposición de la Memoria Principal indicada al inicio. *Nota: rellene por orden de acceso a memoria según el código de alto nivel.*

<i>Elemento</i>	<i>Tipo de Acceso</i>	<i>Dirección Física</i>	<i>Línea_{MC}</i>	<i>A/F y tipo fallo</i>	<i>Bloque_{MP} accedido</i>	<i>Bloque_{MP} sustituido</i>
x[0]	Lectura	0x000			0	
y[0][0]	Lectura	0x00C			1	
x[0]	Escritura	0x000			0	
x[0]	Lectura	0x000			0	
y[0][1]	Lectura	0x010			2	
x[0]	Escritura	0x000			0	
x[1]	Lectura	0x004			0	
y[1][0]	Lectura	0x014			2	
x[1]	Escritura	0x004			0	
x[1]	Lectura	0x004			0	
y[1][1]	Lectura	0x018			3	
x[1]	Escritura	0x004			0	
x[2]	Lectura	0x008			1	
y[2][0]	Lectura	0x01C			3	
x[2]	Escritura	0x008			1	
x[2]	Lectura	0x008			1	
y[2][1]	Lectura	0x020			4	
x[2]	Escritura	0x008			1	

c) Represente cómo quedaría rellena la MC al finalizar la ejecución del programa y la tasa de fallos resultante.

PARTE 2: EJERCICIOS A REALIZAR POR EL ALUMNO

SE RECOMIENDA hacer uso de lápiz en las tablas para evitar tachones en caso de alguna errata.

Ejercicio 1: Se ejecuta el siguiente programa en su computador:

```
for(r=0; r<2; r++)
{
    for(i=0; i<5; i++)
    {
        x[i] = y[4-i];
    }
}
```

- a) Rellene la tabla de accesos situada al final de esta página e indique cómo quedaría finalmente su memoria caché (en el recuadro indicado a continuación), si inicialmente su sistema está dispuesto de la siguiente manera y suponemos que se ejecuta EXCLUSIVAMENTE la primera iteración del bucle externo (r=0):

	Memoria Principal	
B0(0x000)		
B1(0x008)	x[0]	x[1]
B2(0x010)	x[2]	x[3]
B3(0x018)	x[4]	y[0]
B4(0x020)	y[1]	y[2]
B5(0x028)	y[3]	y[4]
...		
BN		

	Memoria Caché (MD, CB-WA, LRU)			
	V	D	Etiqueta	Bloque _{MP}
L0	1	0	8	B4
L1	1	1	2	B5 B1' B5 B1'
L2	1	1	4	B2''
L3	1	1	6	B3'

Nº Acceso	Elemento	Tipo de Acceso	Bloque _{MP}	Línea _{MC}	A/F y tipo	Bloque _{MP} sustituido
1	y[4]	Lectura	B5	L1	Fallo Forzoso	
2	x[0]	Escritura	B1	L1	Fallo Forzoso	B5
3	y[3]	Lectura	B5	L1	Fallo Conflicto	B1
4	x[1]	Escritura	B1	L1	Fallo Conflicto	B5
5	y[2]	Lectura	B4	L0	Fallo Forzoso	
6	x[2]	Escritura	B2	L2	Fallo Forzoso	
7	y[1]	Lectura	B4	L0	Acierto	
8	x[3]	Escritura	B2	L2	Acierto	
9	y[0]	Lectura	B3	L3	Fallo Forzoso	
10	x[4]	Escritura	B3	L3	Acierto	

b) Indique, a continuación, el número de fallos de cada tipo para cada una de las iteraciones del bucle externo:

	Primera iteración (r=0)	Segunda iteración (r=1)
Fallos forzosos	5	0
Fallos por conflicto	2	0
Fallos de capacidad	0	4

Para r=1:

	Memoria Principal	
B0(0x000)		
B1(0x008)	x[0]	x[1]
B2(0x010)	x[2]	x[3]
B3(0x018)	x[4]	y[0]
B4(0x020)	y[1]	y[2]
B5(0x028)	y[3]	y[4]
...		
BN		

	V	D	Etiqueta	Bloque _{MP}
L0	1	0	8	B4
L1	1	1	2	B5 B1' B5 B1' B5 B1' B5 B1'
L2	1	1	4	B2''''
L3	1	1	6	B3''

Nº Acceso	Elemento	Tipo de Acceso	Bloque _{MP}	Línea _{MC}	A/F y tipo	Bloque _{MP} sustituido
1	y[4]	Lectura	B5	L1	Fallo por Capacidad	B1
2	x[0]	Escritura	B1	L1	Fallo por Capacidad	B5
3	y[3]	Lectura	B5	L1	Fallo por Capacidad	B1
4	x[1]	Escritura	B1	L1	Fallo por Capacidad	B5
5	y[2]	Lectura	B4	L0	Acierto	
6	x[2]	Escritura	B2	L2	Acierto	
7	y[1]	Lectura	B4	L0	Acierto	
8	x[3]	Escritura	B2	L2	Acierto	
9	y[0]	Lectura	B3	L3	Acierto	
10	x[4]	Escritura	B3	L3	Acierto	

c) Calcule la tasa de fallos (miss rate) de la ejecución del programa anterior al completo.

Tasa de fallos = N° Fallos/ N° accesos = $11/20 = 0.55 \Rightarrow 55\%$ de Fallos.

Ejercicio 2: Posee un procesador de 32 bits de palabra con las siguientes características del sistema de memoria:

- Memoria Principal máxima direccionable de 64KBytes, con tamaño de bloque de 16Bytes.
- Memoria Caché de 128Bytes de tamaño para datos, WT-NWA, Asociativa por Conjuntos de 2 vías, LRU.

Se ejecuta el siguiente código en su procesador:

```
int i, x[16]={...}, y[16]={...} , z[16]={...}; //1 "int" equivale a 1 palabra
for(i=0; i<16; i=i+zancada)
    x[i] = y[i]+z[i]; //primero se evalúa la parte derecha de la asignación
                    // y, dentro de ella, de izquierda a derecha
```

Donde: "x" se ubica a partir del bloque 16 de MP (inclusive), "y" y "z" se encuentran consecutivos a "x".

- a) Represente gráficamente: la MP, la MC (con todos sus campos y tamaños), la interpretación de la dirección de MP y de MC. Indique los cálculos realizados. Incluya el contenido de la MP (en las posiciones correspondientes) contabilizando únicamente los vectores.

Procesador 32 bits

MP 64 KBytes = 2^{16} Bytes

TBloq 16 Bytes = $2^4 \Rightarrow$ **4 bits de desplazamiento.**

Mem Caché 128 Bytes = 2^7 Bytes

Numero de líneas = $T_{mc}/T_{bloq} = 2^7/2^4 = 2^3 = 8$ líneas / 2 vías = 4 conjuntos por vía = 2^2 , por tanto **índice = 2 bits**

Etiqueta = $T_{mp} - \text{índice} - \text{desplazamiento} = 16 - 2 - 4 = 10$ bits

Tamaño de bloque = 16 Bytes * 8 bits/Bytes = **128 bits**

Memoria Caché

No tendremos el bit *dirty* con *Write Through*.

		V (1 bit)	ETIQUETA (10 bits)	BLOQUE (128 bits)
V0	C0			
	C1			
	C2			

	C3			
V1	C0			
	C1			
	C2			
	C3			

Memoria Principal

32 bits de palabra = 4 bytes por palabra | int ocupa un palabra y el tamaño de bloque es 16 Bytes, por tanto habrá 4 columnas por fila.

B0				
----	--	--	--	--

.....

B16	X[0]	X[1]	X[2]	X[3]
B17	X[4]	X[5]	X[6]	X[7]
B18	X[8]	X[9]	X[10]	X[11]
B19	X[12]	X[13]	X[14]	X[15]
B20	Y[0]	Y[1]	Y[2]	Y[3]
B21	Y[4]	Y[5]	Y[6]	Y[7]
B22	Y[8]	Y[9]	Y[10]	Y[11]
B23	Y[12]	Y[13]	Y[14]	Y[15]
B24	Z[0]	Z[1]	Z[2]	Z[3]
B25	Z[4]	Z[5]	Z[6]	Z[7]
B26	Z[8]	Z[9]	Z[10]	Z[11]
B27	Z[12]	Z[13]	Z[14]	Z[15]

.....

BN				
----	--	--	--	--

- b) Rellene la siguiente tabla de accesos a los datos del programa anterior suponiendo **zancada=2** (si algún campo no procede, indíquelo con “-”, no lo deje en blanco). *Nota: rellene por orden de acceso a memoria según el código de alto nivel.*

ACLARACIONES:

- Realice este apartado en paralelo al siguiente (representación final de la MC) para facilitar la obtención del tipo de fallo y bloque sustituido.
- La línea negra resaltada marca el punto a partir del cual se puede intuir el comportamiento de la traza de accesos sin necesidad de seguir haciendo ninguno más (ya que éste es repetitivo). Haga exclusivamente los accesos hasta llegar a dicha marca y continúe con el resto de los ejercicios. **Al final puede retomar la traza.**

Elemento	Tipo de Acceso	Dirección Física	Conjunto _M c	Vía _{MC}	A/F y tipo fallo	Bloque _M P accedido	Bloque _{MP} sustituido
Y[0]	Lectura	B20=> 16*20=320 => 0x140	C0	0	Fallos Forzoso	B20	-
Z[0]	Lectura	0x180	C0	1	Fallo Forzoso	B24	-
X[0]	Escritura	0x100	C0	WT-NWA => no realizan ubicación en escritura	Fallo Forzoso	B16	-
Y[2]	Lectura	0x148	C0	0	Acierto	B20	-
Z[2]	Lectura	0x188	C0	1	Acierto	B24	-
X[2]	Escritura	0x108	C0	-	Fallo Forzoso	B16	-

- c) Represente cómo quedaría rellena la MC al finalizar la ejecución del programa con **zancada=2**.

		V (1 bit)	ETIQUETA (10 bits)	BLOQUE (128 bits)
V0	C0	1	101	B20
	C1	1	101	B21
	C2	1	101	B22
	C3	1	101	B23
V1	C0	1	110	B24
	C1	1	110	B25
	C2	1	110	B26
	C3	1	110	B27

- d) Explique justificadamente, para cada una de las variaciones siguientes, qué implicación tendría sobre el programa anterior y estime el nuevo M_R sin hacer ningún cálculo. Puede hacer en sucio una tabla similar a la rellena en el ejercicio 2-b.

1. La política de escritura de la caché es CB-WA

Con WT-NWA al no alojarse bloques en caché con fallos de escritura, ya que realiza las modificaciones directamente en memoria principal, mantendremos 2 aciertos en escritura cada dos ciclos, por tanto, tendremos:

$$n^{\circ} \text{ aciertos} = 2 * 4 \text{ (8 veces que se repite el bucle / } 2 = 4) = 8$$

$$n^{\circ} \text{ fallos} = 4 * 4 = 16$$

$$\text{total accesos} = 16 + 8 = 24$$

$$\text{miss rate} = 16/24 = 2/3$$

Con la política CB-WA al tener una política de reemplazo de LRU tendríamos que cada bloque se irá sustituyendo de forma que el B20 se sustituirá por B16, B24 por B20, B16 por B24 y así repetidamente con los siguientes accesos en los demás bloques. Por tanto, nos quedaría un miss rate de 1, en los que todos los accesos a memoria serán fallos.

2. zancada = 1

Suponiendo que mantenemos WT-NWA y cambiamos la zancada a 1, los accesos a datos aumentarán por cada bloque el doble. Lo que implica que de los 4 ciclos en los que se accede a cada bloque, el primer ciclo serán todos fallos, y en los 3 ciclos restantes resultará, de los 3 accesos, 2 aciertos y 1 fallo por ciclo. Por tanto, tendríamos:

$$n^{\circ} \text{ aciertos} = 2 \text{ aciertos} * 12 \text{ ciclos} = 28$$

$$n^{\circ} \text{ fallos} = (3 \text{ fallos} * 4 \text{ ciclos}) + (12 \text{ ciclos} * 1 \text{ fallo}) = 28$$

$$\text{total accesos} = 48$$

$$\text{miss rate} = 28/48 = 7/12$$