

Práctica optativa: Simulador de una cache simple

En la memoria cache (MC) se cargan los bloques que más va a utilizar el procesador. Se dice que una referencia a memoria (rd / wr) es un acierto si el bloque está en la MC y un fallo si no lo está. En ese caso, habrá que traer el bloque de la memoria principal (MP). Como la MC es mucho más pequeña que la MP, hay que decidir dónde cargar los bloques. Hay varias opciones para ello, pero se puede decir, en general, que el espacio que hay en la cache se divide en conjuntos, que la asignación de los conjuntos es directa y que, dentro del conjunto, se puede elegir cualquier posición. Si los conjuntos son de un único bloque, estamos ante una correspondencia **directa**, es decir, a cada bloque de MP le corresponde una posición fija en la MC. En el otro extremo está la correspondencia **totalmente asociativa**, donde hay un único conjunto que incluye todas las posiciones de la MC. La correspondencia **asociativa por conjuntos** es el caso intermedio entre las otras dos. Habitualmente, se utilizan dos políticas de reemplazo para decidir que bloque se saca de la MC cuando necesitamos espacio: FIFO y LRU.

Escribe un programa que simule el funcionamiento de una memoria cache (utiliza el lenguaje de programación que quieras). La cache será de 8 bloques y los conjuntos pueden ser de 1 bloque (directa, cada conjunto un bloque), de 2, de 4 o de 8 (totalmente asociativa, un único conjunto). La política de escritura será *write-back*.

El programa solicitará como **entrada** estos datos:

<i>Tamaño de palabra</i>	4 u 8 bytes
<i>Tamaño de bloque</i>	32 o 64 bytes
<i>Tamaño de conjunto</i>	1 (directa), 2, 4, u 8 (totalmente asociativa)
<i>Política de reemplazo</i>	FIFO o LRU
<i>Dirección</i>	Una dirección de memoria (al byte)
<i>Operación</i>	lectura (ld, 0) o escritura (st, 1)

Y el **resultado** será:

Interpretación de la dirección: palabra, bloque, conjunto, tag...
Acierto o fallo
Si ha habido reemplazo o no
Tiempo de acceso

Tiempos de acceso: $T_{cm} = 2c.$; $T_{mp} = 21c.$; $T_{buf} = 1c.$ Por ejemplo, un acierto en al cache: $2c.$; un fallo sin reemplazo: $2 + 21 \times 1 = 30 c.$

Puedes utilizar una matriz de 8x5 para representar la MC. Cada fila es un bloque y las columnas son:

OCUP (1/0), MODIF (1/0), TAG (etiq. de bloque búsquedas) y REEM (info. de la política de reemplazo).

Al final del documento tienes un ejemplo de funcionamiento de un opción concreta.

Con esta práctica se puede conseguir un punto extra para la nota final dependiendo de lo que se haga:

- (a) *Interpretación de la dirección.* Es muy simple: 0,2 puntos
- (b) *Funcionamiento de la cache y actualización de los datos y de la información de control:*
 - estructura de la cache, una o más (directa, asociativa...): hasta 0,35 puntos
 - algoritmos de reemplazo (no, uno, dos): hasta 0,3 puntos
 - tasa de acierto, tiempos...: 0,15 puntos

El programa se podrá realizar en parejas o individualmente. Para evaluarlo habrá que hacer una ejecución ante el profesor fuera de las horas de clase.

Tiempo estimado para llevarlo a cabo: 2-3 horas para el diseño + 4-6 horas para el código y las pruebas.

PLAZO DE ENTREGA: 4 de octubre.

Un ejemplo de entrada/salida para el simulador

```
$ cachesim
```

```
Tam. palabra (bytes): 4 - 8 >4
Tam. bloque (bytes): 32 - 64 >32
Tam. conjunto (bloques): 1-2-4-8 >2
Pol. remplazo: 0(FIFO) - 1(LRU) >1
```

```
Direccion (byte) (-1 salir) >1123
Load (0) / Store (1) >1
```

```
>Direccion: 1123 - Palabra: 280 - Bloque: 35 (palabras 280 - 287)
>Conjunto: 3 - Tag: 8
>FALLO de CACHE
```

```
>Tiempo de acceso: busqueda cache, 2 -- transferir bloque (M>C o C>M), 21+7
>T_acc: 30 ciclos
```

ocup	mod	tag	rem		bloque
0	0	0	0		---
0	0	0	0		---
0	0	0	0		---
0	0	0	0		---
0	0	0	0		---
1	1	8	0		b35
0	0	0	0		---

```
Direccion (byte) (-1 salir) > ...
```

Las direcciones pueden darse una a una por teclado, o en un fichero. Al finalizar la simulación, hay que calcular la tasa de aciertos y el tiempo total. Por ejemplo,

```
Referencias: 4 -- Aciertos: 2 -- Tasa de aciertos, h = 0.50
Tiempo total = 64 ciclos
```