Práctica 10

EL SISTEMA DE MEMORIA CACHE EN EL MIPS R2000
LA CACHE DE CÓDIGO

1. ► ¿Cuántos elementos tienen los vectores del programa? ¿Cuántos bytes ocupa cada elemento?

   Tiene 7 elementos de 2 bytes de A y 1 elemento de 4 bytes para B 1 elemento para K y otro para dim ambos de 2 bytes

2. ► Complete la siguiente información del segmento de datos. Utilice el sistema hexadecimal para expresar las direcciones de memoria (haga igual a lo largo de toda la práctica).

Dirección inicial del vector A	0x10000000
Bytes ocupados por el vector A	14
Dirección inicial del vector B	0x10001000
Bytes ocupados por el vector B	4
Dirección de la variable k	.data 0x1000A030
Dirección de la variable `dim`	.data 0x1000A034

3. ► Complete la siguiente información del segmento de código.

Dirección de la primera instrucción	0x00400000
Dirección de la última instrucción	0x00400024
Número de instrucciones del programa	21
Bytes ocupados por el código del programa (instrucciones)	42

4. ► Determine el número de accesos al sistema de memoria del programa. Estos valores son muy importantes porque nos servirán más tarde para conocer qué número de accesos del total son servidos por la memoria cache, esto es, podremos distinguir entre accesos que son aciertos y accesos que son fallos.

Accesos al segmento de datos	77
Accesos al segmento de código	18

5. ► Teniendo en cuenta las características anteriores, indique cuántas líneas hay en la memoria cache.

   Nº Lineas = Tamaño_cache/ tamaño de bloque

   128/4 =32

   Nº Conjuntos = NºLineas / vias = 32

6. ► Indique cuál será la interpretación que esta memoria cache hará de las direcciones que reciba (campos de etiqueta, línea y desplazamiento).

2 campo dezplazamiento , 5 campo lineas , 32-5-2 =25 para campo etiqueta

7. ► La instrucción del programa `jal sax` está almacenada en la dirección `0x0040001C` del segmento de datos. Indique en qué línea de la cache se ubicará y con qué etiqueta.

linea = 3 etiqueta = 0x0008000

8. ► Calcule, para este caso, cuántos bits de control se almacenan por línea. Así mismo, calcule el volumen del directorio, esto es, el número total de bits de control contenidos en la memoria cache de código.

Bits de control por línea	26
Volumen del directorio (bytes)	26x32 =836

etiqueta + bit de validez + bit modificado +0 LRU (Directa) = 25+1+1

9. ► Cargue el *programa original* y ejecútelo mediante la opción F10 (paso a paso) para poder seguir con detalle el efecto sobre la memoria cache de código. Complete la siguiente tabla:

Accesos al segmento de código	77
Aciertos	56
Fallos	21
Tasa de aciertos (H)	0.7272

10. ► Confirme que la instrucción `jal sax` se almacena en la línea prevista y con la etiqueta calculada anteriormente.

11. ► Determine el tiempo medio de acceso al segmento de código experimentado por el programa.

10*0.7272 + (1-0.7272)*100 = 34 ns

12. ► Use el simulador y configure la memoria cache de código con un tamaño de bloque de 16 bytes y manteniendo el resto de parámetros como estaban. Cargue y ejecute ahora el *programa original* y complete la siguiente tabla:

Accesos al segmento de código	77
Aciertos	71
Fallos	6
Tasa de aciertos (H)	0.92

13. ► Como se aprecia, el número de fallos se ha visto reducido de forma considerable. ¿Cuál es la razón?

Aprovechando el hecho bloques de instrucciones consecutivas pueden ser referenciadas utilizamos en principoi de localidad para traer a la cache bloques mas grandes de instruccioes con las siguientes instrucciones que pueden ser referenciadas en las proximas lineas