Trabajo de Laboratorio Número 2

Arquitectura de Computadoras 20/05/2026

Instrucciones

- Los ejercicios propuestos en el presente trabajo práctico pueden ser resueltos en forma individual o grupal.
- El planteo de la solución debe realizarse basándose en lo aprendido en las clases teóricas.
- Puede utilizar las clases de consulta para consultar ejercicios de enunciado y verificar la validez de la solución obtenida.
- Posteriormente, se tomará una evaluación con ejercicios similares a los de este práctico, la cual deberá ser resuelta en forma individual. La fecha de la misma se encuentra en el sitio web de la materia.

Objetivo

El objetivo del laboratorio es enfatizar los conceptos enseñados en clase sobre el funcionamiento de las memorias caché. Para ello se utilizará el simulador CachéSim 2.0, desarrollado en nuestra Facultad, que permite analizar el comportamiento de estos dispositivos, así como también las distintas alternativas de diseño: grado de asociatividad, tamaño del bloque, algoritmo de reemplazo, etc. Debido a su carácter didáctico, el simulador sólo trabaja con un solo nivel de caché.

Conceptos involucrados

- Aritmética de cálculo de direcciones en distintas configuraciones de memoria caché.
- Ubicación de un bloque de memoria en distintas configuraciones de memoria caché.
- Relación entre el código ensamblador de un programa y los accesos a memoria que genera el mismo. Generación de trazas de memoria.
- Distintos tipos de fallos de acuerdo a “las 3 C”: Inevitables, Capacidad y Conflicto.
- Impacto del programa en la performance de la memoria caché.

Problemas propuestos

1) *Este ejercicio y el siguiente buscan demostrar cómo es la aritmética del cálculo de direcciones, procurando que luego puedan realizarla sin ayuda del simulador. Es fundamental realizar primero los cálculos a mano, y luego verificarlos con el simulador.*

   En un caché de mapeo directo de 512 bytes y 16 palabras de 32 bits por bloque, y utilizando la traza de memoria **Traza (19999 entradas)** provista por el simulador, se pide que:
   a) Indique cómo se descomponen las direcciones, luego verifique la misma con el simulador.
   b) Observando los cinco primeros accesos de la traza, indique a qué bloque corresponden, y si son aciertos (HIT) o fallos (MISS). Verifique sus cálculos con el simulador.
   c) Repita el apartado **a** pero utilizando ahora un caché asociativo por conjuntos de dos vías y algoritmo LRU (tamaño y cantidad de palabras por bloque iguales al apartado a).

2) Sea un caché que tiene 4K bloques de 4 palabras de 32 bits, que es accedido con direcciones de 32 bits. Para cada grado de asociatividad indicado más abajo, se pide que calcule el tamaño del índice, el tamaño de la etiqueta y la cantidad total de bits usados en etiquetas.
   a) Un caché de mapeo directo.
   b) Un caché asociativo por conjuntos de 2 vías.
   c) Un caché asociativo por conjuntos de 4 vías.
   d) Un caché asociativo puro.

3) *Este ejercicio sirve para comprender cómo es la lógica utilizada por una memoria caché para ubicar los bloques de datos, y cómo esta ubicación depende de su organización. Es muy importante realizar primero los cálculos a mano, y luego verificarlos con el simulador.*

   Dada la siguiente secuencia de direcciones de acceso a memoria (en decimal) de un sistema de 32 bits con direccionamiento a bytes: 4, 16, 32, 20, 80, 68, 76, 224, 36, 44, 16, 172,

20, 24, 36, 68, se pide que indique para cada una de las memorias caché que se indican a continuación si cada acceso produce un fallo o un acierto, y que muestre el contenido final del caché.

a) Un caché de mapeo directo de 16 bloques de una palabra.
b) Un caché de mapeo directo de 16 palabras en total, pero con bloques de 4 palabras.
c) Un caché asociativo por conjuntos de 2 vías, con bloques de una palabra y 16 palabras en total, cuyo algoritmo de reemplazo es LRU.
d) Un caché asociativo puro, con bloques de una palabra y 16 palabras en total, cuyo algoritmo de reemplazo es LRU.
e) Un caché asociativo puro, con bloques de 4 palabras y 16 palabras en total, cuyo algoritmo de reemplazo es LRU.

4) *Este ejercicio muestra cómo se relacionan nuestros programas escritos en lenguaje ensamblador, estudiados en el laboratorio anterior, con los accesos a memoria, desde una mirada más global. Para ello se busca demostrar cómo se genera una traza de accesos a memoria a partir de un código ensamblador.*

Considerando el siguiente código escrito en lenguaje ensamblador RV32I del procesador RISC-V resuelva los siguientes apartados.

```
        .data 0x10000000
vector: .word 4,8,15,16,23,42

        .text 0x00000000
main:   addi t4, zero, 6        # t4 cantidad de elementos del vector.
        addi t3, zero, 23       # t3 escalar que se suma al vector.
        la t0, vector           # t0 base del vector.
lazo:   lw t1, 0(t0)
        add t5, t1, t3
        sw t5, 24(t0)
        addi t4, t4, -1
        addi t0, t0, 4
        bne t4, zero, lazo
```

a) Escriba la traza de accesos a memoria resultante para la ejecución del programa considerando una sola iteración del mismo. Clasifique cada acceso a memoria según corresponda a lectura de instrucciones, lectura de datos o escritura de datos. Además indique para cada acceso a cuál instrucción corresponde.
b) Escriba la traza de accesos a memoria considerando la ejecución de dos iteraciones del lazo. Clasifique cada acceso según corresponda a lectura de instrucciones, lectura de datos o escritura de datos. Además indique para cada acceso a cuál instrucción corresponde.
c) Teniendo en cuenta los resultados de los apartados anteriores genere un archivo de traza llamado **trazaSuma.prg** que contenga los accesos a memoria correspondientes a la ejecución completa del programa.
d) Simule la traza **trazaSuma.prg** obtenida en el apartado anterior utilizando el programa CachéSim. Analice el funcionamiento paso a paso de la memoria caché, buscando las diferencias en cantidad y tipo de fallos para un caché de mapeo directo versus uno asociativo por conjuntos de 4 vías, ambos con el mismo tamaño de palabra y total.
   - Tamaño Caché: 256 Bytes
   - Tamaño de la palabra: 32 bits
   - Palabras por bloque: 4
   - Cantidad de vías: Mapeo Directo / 4 vías LRU

   Concluya en pocas palabras a qué se deben las diferencias encontradas en el funcionamiento de ambos cachés.

5) *Este ejercicio busca demostrar cómo diferenciar los tipos de fallos de acuerdo al modelo de "las 3 C" (inevitables, capacidad y conflicto), remarcando que esta clasificación surge de un modelo simple que presenta sus limitaciones.*

a) Considere que usted dispone de un caché infinitamente grande y además asociativo puro, es decir un caché "ideal". ¿Qué tipo de fallos esperaría encontrar? ¿Existe algún tipo de fallo que nunca aparecería en este caché ideal? Justifique su respuesta.

b) Para simular nuestro caché “infinitamente grande” utilizamos un caché de 4 MB (máximo tamaño permitido en el simulador) y para aproximarnos a una caché asociativa pura, seleccionamos una asociatividad de 8 vías. Ejecute la traza **swm.prg**, provista con el simulador con la siguiente configuración de caché y luego responda las preguntas:
   - Tamaño Caché: 4 MB
   - Cantidad de vías: 8 vías
   - Tamaño de la palabra: 64 bits
   - Palabras por bloque: 8

   i) ¿Cuál es la tasa de fallos resultante?
   ii) ¿Qué cantidad de fallos de cada tipo obtiene?
   iii) ¿Concuerda este resultado con su respuesta en el apartado a? ¿Por qué?

c) Realice una nueva simulación utilizando la misma traza que en el apartado anterior y una caché idéntica pero con la cantidad de palabras por bloque cuadruplicada. ¿Observa algún cambio en la tasa de fallos obtenida? ¿Cómo se ve afectada la cantidad de fallos de cada tipo obtenida? ¿A qué se debe este cambio?

d) Considerando sus resultados en los aparatados anteriores, ¿puede afirmar que a mayor cantidad de palabras por bloque obtenemos menor tasa de fallos? ¿Tiene esta afirmación algún límite o se cumple indefinidamente para cualquier configuración de caché?

6) *Este ejercicio continúa con el análisis de los distintos tipos de fallos de acuerdo al modelo de “las 3 C”. En este caso se analiza cómo se incrementa el número de fallos a medida que se disminuye la capacidad del caché, identificando el tipo de fallos con mayor incidencia al realizar estas modificaciones.*

   a) Suponga que a la memoria caché ideal del punto anterior (capacidad infinita y asociatividad pura), sólo le mantenemos su asociatividad pura y comenzamos a disminuirle gradualmente su tamaño. A medida que se disminuye el tamaño de la caché, ¿qué nuevos tipos de fallos aparecerán? ¿Pueden estar presentes los tres tipos de fallos? Justifique sus respuestas.

   b) Utilizando la traza **swm.prg**, simule ahora el siguiente caché:
      - Tamaño Caché: 64KiB
      - Cantidad de vías: 8 vías
      - Tamaño de la palabra: 32 bits
      - Palabras por bloque: 8

      i) ¿Cuál es la tasa de fallos resultante?
      ii) ¿Qué cantidad de fallos de cada tipo obtiene?

   c) Repita la simulación pero modificando el tamaño del caché a 16KiB, 4KiB, 1KiB ¿Qué tipo de fallos presenta la mayor variación? ¿Concuerda este resultado con su respuesta en el apartado a? Justifique sus respuestas.

7) *Este es el ejercicio final para el análisis del modelo de “las 3 C”, y pretende demostrar como varía la cantidad de fallos de acuerdo al grado de asociatividad de la memoria caché y qué tipo de conflictos genera esta variación.*

   a) A la memoria ideal original modificada en el punto anterior estableciendo una capacidad finita, le quitaremos la asociatividad pura y comenzaremos a disminuirle la asociatividad. ¿Qué efecto prevé que tendrá esta modificación sobre la tasa de fallos? ¿Cuál será el tipo de fallos más influyente en este cambio?

   b) Utilizando una vez más la misma traza **swm.prg** manteniendo fijo el tamaño del caché, el tamaño de la palabra y las palabras por bloque, simule el comportamiento de la memoria para una asociatividad de 8 vías, 4 vías, 2 vías y de mapeo directo.
      - Tamaño Caché: 32 KiB
      - Tamaño de la palabra: 32 bits
      - Palabras por bloque: 8

      i) ¿Cómo se modifica la tasa de fallos a medida que se disminuye la asociatividad del caché?
      ii) ¿Qué fallos tienen un mayor peso en estas variaciones?

iii) ¿Concuerda este resultado con su respuesta al apartado a? Justifique sus respuestas.

8) *Este ejercicio busca analizar los diferentes tipos de fallos de acuerdo al modelo de "las 3 C" (inevitables (compulsory), capacidad y conflicto), y como influyen tanto el tamaño del caché como el grado de asociatividad. Para eso utilizaremos el simulador con la traza* ***Cexp.prg*** *con las siguientes configuraciones de caché:*

a) Caché infinitamente grande y asociativo puro, es decir un caché "ideal", con la cantidad de palabras por bloque variable.
   - Tamaño Caché: 4 MB
   - Cantidad de vías: 8 vías
   - Palabras por bloque: 8, 16, 32, 64
   - Tamaño de la palabra: 64 bits
   - Algoritmo de Reemplazo: LRU (Least Recently Used)

b) Caché con tamaño variable, partiremos de la siguinte configuración y luego se disminuirá gradualmete su tamaño.
   - Tamaño Caché: 64KiB, 32KiB, 16KiB, 4KiB y 512B
   - Cantidad de vías: 8 vías
   - Palabras por bloque: 8
   - Tamaño de la palabra: 64 bits
   - Algoritmo de Reemplazo: LRU (Least Recently Used)

c) Caché con grado de asociatividad variable, partimos de la siguinte configuración y luego se modificará su grado de asociatividad.
   - Tamaño Caché: 32 KiB
   - Cantidad de vías: 8 vías, 4 vías, 2 vías y de mapeo directo
   - Palabras por bloque: 8
   - Tamaño de la palabra: 64 bits
   - Algoritmo de Reemplazo: LRU (Least Recently Used)

d) Analizando las configuraciónes a, b y c con sus variaciones responda:
   - ¿Cómo se ve afectada la cantidad de fallos de cada tipo, en las diferentes configuraciones?
   - ¿Cómo es la tasa de fallos en las diferentes configuraciones?
   - ¿En que configuración detecta un mayor incremento en la tasa de fallos resultante?
   - ¿Qué configuración presenta la mayor variación en la taza de fallos inevitables?
   - ¿Qué configuración presenta la mayor variación en la taza de fallos por conflicto?
   - ¿Qué configuración presenta la mayor variación en la taza de fallos por capacidad?

9) *En este ejercicio se demuestra cómo el programa puede influir en la performance del sistema de memoria, y cuan dependiente es el mismo del orden en que se accede a los datos. Es fundamental resolver los apartados sin utilizar el simulador y luego verificar los resultados con el mismo.*
Se desea acceder a todos los elementos de una matriz 5 x 5 almacenada en memoria por columnas contiguas como se muestra a continuación (en cada elemento a_{ij} se muestra la dirección de memoria donde está almacenado):

a_{00} [00009B30]	a_{01} [00009B44]	a_{02} [00009B58]	a_{03} [00009B6C]	a_{04} [00009B80]
a_{10} [00009B34]	a_{11} [00009B48]	a_{12} [00009B5C]	a_{13} [00009B70]	a_{14} [00009B84]
a_{20} [00009B38]	a_{21} [00009B4C]	a_{22} [00009B60]	a_{23} [00009B74]	a_{24} [00009B88]
a_{30} [00009B3C]	a_{31} [00009B50]	a_{32} [00009B64]	a_{33} [00009B78]	a_{34} [00009B8C]
a_{40} [00009B40]	a_{41} [00009B54]	a_{42} [00009B68]	a_{43} [00009B7C]	a_{44} [00009B90]

a) Para tal fin, se utiliza el código que se muestra a continuación, con el cual se genera la traza de memoria **PorFilas** disponible en el simulador.

```
for (i = 0; i < 5; i++)
    for (j = 0; j < 5; j++)
        aux = a[i][j];
```

Dada la siguiente configuración de un caché de datos, calcule **sin usar el simulador** la tasa de fallos que generaría este código:

- Tamaño Caché: 64 bytes
- Tamaño de la palabra: 32 bits
- Palabras por bloque: 4
- Cantidad de vías: 4 vías

Compruebe sus resultados simulando la traza y registre los resultados más relevantes arrojados por el simulador.

b) Se decide comparar los resultados obtenidos en el apartado anterior, recorriendo en está oportunidad la matriz de acuerdo al nuevo código que se muestra a continuación, el cual genera la traza de memoria **PorColumnas** (adjunta con el enunciado).

```
for (i = 0; i < 5; i++)
    for (j = 0; j < 5; j++)
        aux = a[j][i];
```

Utilizando la misma configuración de caché del apartado anterior, calcule **sin usar el simulador** la tasa de fallos que generaría este código. Luego compruebe sus resultados simulando la traza y registre los resultados más relevantes arrojados por el simulador.

c) Utilizando los resultados obtenidos en los apartados anteriores concluya si existe alguna diferencia cuando se modifica el algoritmo utilizado para recorrer la matriz, teniendo en cuenta que los datos se encuentran alojados en las mismas direcciones de memoria y que el caché utilizado es exactamente el mismo. Describa brevemente a qué se debe esta diferencia.

d) ¿Considera usted que sería posible obtener los mismos resultados pero sin realizar ninguna modificación en el código? ¿Qué cambios debería realizar?

Ejercicio a Desarrollar en Clase Práctica

Problemas 4 del Segundo Parcial de 2008 (acotado)

10) Se tiene un procesador con un caché de datos asociativo puro de 4 bloques de 4 palabras cada uno, cuyo algoritmo de reemplazo es LRU. Sobre este procesador se ejecuta un programa que multiplica una matriz A por un vector B y deja el resultado en el vector C. La matriz A tiene 16x16 palabras, *organizadas por columnas* en posiciones consecutivas de memoria a partir de la dirección 0x10000000. Los vectores B y C tienen 16 palabras cada uno y comienzan en las direcciones 0x10000400 y 0x10000440 respectivamente. El código del programa es el siguiente:

```
for(i = 0; i <= 15; i++) {
     for(j = 0; j <= 15; j++) {
         r1 = A[i, j];
         r2 = B[j];
         r3 = C[i];
         r4 = r1 * r2;
         r5 = r3 + r4;
         C[i] = r5;
     }
}
```

a) Calcular la tasa de fallos que producirá el programa en la caché. Calcule qué fracción de esta tasa se debe a fallos inevitables (compulsivos), de capacidad y de conflicto.
b) Proponga una nueva forma de organizar los datos en memoria para reducir los fallos. Calcule nuevamente la tasa de fallos. Calcule qué fracción de esta tasa se debe a fallos inevitables (compulsivos).
c) Para la organización original de datos, proponga un cambio de programa para bajar la cantidad de fallos a un número similar a (b). Calcule nuevamente la tasa de fallos y la fracción de fallos inevitables (compulsivos).
d) Si existiera la posibilidad de duplicar el tamaño del caché, indique qué convendría duplicar. Justifique su respuesta. Respuesta equivocada produce puntos en contra.
    1) Cantidad de Vías.
    2) Cantidad de Bloques.
    3) Tamaño del bloque.

Soluciones propuestas

1)
   a) . . .
   b) 4 fallos, 1 acierto.
   c) . . .

2) a) Indice 12 bits. Etiqueta 16 bits. Total 64 KiBits.
   b) Indice 11 bits. Etiqueta 17 bits. Total 68 KiBits.
   c) Indice 10 bits. Etiqueta 18 bits. Total 72 KiBits.
   d) No necesita indice. Etiqueta 28 bits. Total 112 KiBits.

3) a) La cantidad de fallos es de 13.
   b) La cantidad de fallos es de 10.
   c) La cantidad de fallos es de 12.
   d) La cantidad de fallos es de 12.
   e) La cantidad de fallos es de 10.

4) Al utilizar una caché de mapeo directo si bien el tamaño de la misma es más que suficiente para alojar todos los datos, gran parte de los accesos son mapeados al mismo sector del caché generándose muchos fallos por conflicto. Por el contrario utilizando una caché de mayor asociatividad se utiliza mejor la capacidad disponible no presentándose fallos por conflicto y manteniéndose solo los fallos inevitables.

5) a) En un caché infinito y asociativo puro los únicos fallos existentes son los fallos inevitables.
   b) No se producen fallos por capacidad ni por conflicto.
   c) Al incrementar el tamaño del bloque aprovecho mejor la localidad espacial y por ende requiero menos accesos para acceder por primera vez a cada uno de los bloques utilizados por el programa. Al mantenerse el tamaño y asociatividad de la caché los fallos por capacidad y por conflicto continúan siendo cero.
   d) Es verdad que al aumentar la cantidad de palabras por bloque logramos una disminución en los fallos inevitables pero si continuamos incrementando este valor comenzaremos a aumentar la penalidad por fallo (más información a transmitir en caso de fallo) y además si tomamos cachés más pequeñas comenzaremos a tener mayor número de fallos por conflictos y también por capacidad.

6) Tasa de fallos resultante: 5,50 % Inevitables: 1099 (100 %) Conflictos: 0 (%) Capacidad: 0 (%)

7) a) Al disminuir la asociatividad esto incrementará la tasa de fallos y esto será provocado principalmente por el aumento de los fallos por conflicto.
   b) Vemos como al reducir la asociatividad se incrementa la tasa de fallos por el incremento de los fallos por conflicto.

8) Los fallos disminuyen al aumentar las palabras por bloque, y se mantienen constantes respecto del tamaño del caché o la cantidad de vías usadas.

9) a) ...
   b) ...
   c) Se altera el funcionamiento por el modo en que están almacenados los datos en memoria.
   d) Si. Modificando la forma en que se almacenan los datos en memoria.