

Proyecto #1

Memoria caché

El rendimiento de un computador no viene solo dado por la velocidad del procesador, sus características o si es único o hay múltiples procesadores. El rendimiento de un computador depende de las características de todos sus componentes: CPU, memoria, unidades de E/S, forma de interconexión entre todos esos componentes, etc.

En este proyecto vamos a estudiar el comportamiento de la memoria, y es que, una característica básica de la memoria en la actualidad, es que cuanto más rápida sea, resultará más pequeña y cara, pero se, en general, se necesitan almacenar muchos datos y programas lo que requiere grandes cantidades de memoria, así que se ha hecho imprescindible encontrar un equilibrio, lograr tener grandes memorias, que, a su vez sean rápidas y baratas.

Como hemos visto en clase, la solución planteada a este problema es utilizar una jerarquía de memorias, en lugar de un solo tipo de memoria. Aprovechando el principio de localidad, se van creando “capas” de memorias, donde las más rápidas y pequeñas almacenan los datos que se considera tienen la mayor probabilidad de ser accedidas, dejando en las memorias más lentas los datos que tienen menor probabilidad de ser requeridos por el procesador.

Este proyecto tiene como finalidad que su equipo desarrolle una herramienta en lenguaje C que le permita simular el comportamiento de una jerarquía de memoria que consiste en una memoria caché y una memoria principal. Este simulador se dividirá en 3 módulos que nos permitirán estudiar el funcionamiento de una jerarquía de memoria desde distintos aspectos.

Como requerimiento previo al desarrollo de las simulaciones, el equipo deberá realizar una investigación de los conceptos a desarrollar en las simulaciones.

Al finalizar, deberán ejecutar los distintos módulos con una serie de datos que se enviarán en los próximos días y realizar un análisis de los resultados obtenidos.

A continuación se describen los tres módulos a desarrollar.

Módulo 1: Función de *mapeo*

Este módulo debe permitir visualizar la forma en la que se dividirá la dirección en memoria de un byte determinado, para que pueda ser ubicado en memoria caché o en memoria principal, según sea el caso. Los parámetros requeridos de entrada son:

- Función de mapeo: que puede ser mapeo directo o mapeo asociativo por conjunto.

- Tamaño, en bytes, de la memoria principal. Opciones: 256KB, 512KB, 1MB, 2MB, 4MB, 8MB, 16MB, 32MB
- Tamaño, en bytes, de la memoria caché . Opciones: 64B, 128B, 256B, 512B, 1KB
- Tamaño, en bytes, de los bloques. Opciones: 2B, 4B, 8B, 16B, 32B
- Tamaño del conjunto, en bloques (para mapeo asociativo por conjunto). Opciones: 2, 4 u 8 bloques.

En base a estos datos se requiere que muestre la partición de la dirección que corresponda.

Ejemplo:

Dados los siguientes datos:

- Mapeo directo
- Memoria principal de 256KB
- Memoria caché de 1KB
- Tamaño del bloque 8B

Resultado:

Etiqueta								Línea							Byte		
17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Módulo 2: Ubicación de Bloques

Este módulo debe permitir visualizar la forma en la que se ubicaría un bloque de memoria en la memoria caché. Específicamente, dados los parámetros de caché y una dirección de memoria (en decimal), el sistema debe indicar el número de línea o de conjunto de caché, según sea el caso, en el que debería incluirse el bloque de memoria. Adicionalmente debe mostrarse la partición de la dirección. Los parámetros de entrada deben ser:

- Función de mapeo: que puede ser mapeo directo o mapeo asociativo por conjunto.
- Tamaño, en bytes, de la memoria principal. Opciones: 32KB, 64KB, 256KB, 512KB, 1MB, 2MB, 4MB, 8MB, 16MB, 32MB
- Tamaño, en bytes, de la memoria caché. Opciones: 64B, 128B, 256B, 512B
- Tamaño, en bytes, del bloque. Opciones: 2B, 4B, 8B, 16B, 32B
- Tamaño del conjunto, en bloques (mapeo asociativo por conjunto). Opciones: 2, 4 u 8 bloques
- Dirección de memoria, en decimal, del byte que se busca

Ejemplos:

Dados los siguientes parámetros de caché:

- Mapeo directo
- Memoria principal de 256KB
- Memoria caché de 128 B
- Tamaño del bloque 8B
- Dirección 219

Resultado:

Etiqueta											Línea				Byte		
0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0	1	1

El bloque se coloca en la línea 11 de la memoria caché.

Ejemplos:

Dados los siguientes parámetros de caché:

- Mapeo asociativo por conjunto
- Memoria principal de 256KB
- Memoria caché de 128 B
- Tamaño del bloque 8B
- Tamaño del conjunto 2
- Dirección 220
-

Mapeo Asociativo por conjunto

Etiqueta												Conjunto			Byte		
0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	0	0

El bloque se coloca en el conjunto 3 de la memoria caché.

Módulo 3: Simulador de reemplazo de bloques

Este módulo emulará el comportamiento de una memoria caché pequeña con mapeo asociativo de conjunto. Dados los parámetros requeridos de configuración de la memoria caché, este simulador deberá mostrar la posición en la memoria caché donde se debe ubicar cada bloque basados en una secuencia de máximo 20 localidades de memoria que serán introducidas por el usuario, indicando si al colocar cada nuevo bloque hubo un acierto (hit) o una falla (miss) y tipo de falla. Los parámetros de caché a considerar son los siguientes:

- Tamaño, en bloques, de la memoria caché. Opciones: 4, 8, 16, 32, 64, 128
- Cantidad de conjuntos. Opciones: 2, 4, 8, 16
- Política de reemplazo. Opciones: LRU, FIFO y RAND
- Secuencia de direcciones

Se asume que el tamaño de cada bloque es de 1 palabra.

El resultado debe mostrar el contenido de la memoria caché después de incluir cada solicitud de acceso a memoria, indicando si fue un éxito o una falla y el tipo de falla. En caso de que la falla implique la sustitución de otro bloque, se debe indicar cual fue sustituido.

Al finalizar deberá mostrar el contenido de la memoria caché, una traza que muestre el resultado de cada solicitud de acceso a memoria y las estadísticas finales de aciertos y fallas (total y por tipo)

Ejemplo:

- Tamaño de la memoria caché: 16 bloques

- Cantidad de conjuntos: 4
- Política de reemplazo: FIFO
- Secuencia de direcciones: 37 21 20 12 59 44 30 24 43 24 16 36 59 57 32 21 53 14 63 17

Resultado final:

# Conjunto	caché			
0	16	36	32	24
1	17	21	57	53
2	30	14		
3	59	43	63	

Traza

37 falla forzosa	21 falla forzosa	20 falla forzosa
12 falla forzosa	59 falla forzosa	44 falla forzosa
30 falla forzosa	24 falla forzosa	43 falla forzosa
24 acierto	16 falla por conflicto(-20)	36 falla por conflicto(-12)
59 acierto	57 falla por conflicto(-37)	32 falla por conflicto(-44)
21 acierto	53 falla forzosa	14 falla forzosa
63 falla forzosa	17 falla forzosa	

Estadísticas

Total de solicitudes de caché: 20

Fallas

Total fallas: 17
 Cantidad fallas forzosas: 13
 Cantidad fallas por capacidad: 0
 Cantidad fallas por conflicto: 4
 Tasa de fallas: 85%

Aciertos:

Total aciertos: 3
 Tasa de aciertos: 15%

Especificaciones del proyecto

- El proyecto se realizará en grupos de 2 personas.(sin excepción)
- La entrega del proyecto debe incluir:
 - Informe:
 - Introducción
 - Marco teórico. Que debe incluir:
 - Funciones de mapeo: cuáles existen y su funcionamiento
 - Políticas de reemplazo de bloques en memoria caché: cuáles existen y su funcionamiento
 - Tipos de fallas: descripción

- Resultados obtenidos de los casos de prueba (captura de pantallas)
 - Análisis de cada caso de prueba
 - Conclusiones
- Programa en C
- No es necesario que los resultados tengan el formato presentado arriba, pero si deben contener toda la información requerida, mostrado de manera legible y entendible.
- Fecha de entrega: lunes 12 de diciembre de 2016 hasta medianoche.
- Luego de la entrega se realizará un interrogatorio al cual deben asistir los 2 integrantes del equipo. El interrogatorio tendrá una duración de 30 minutos. Durante el interrogatorio deberán ejecutar el programa realizado, los estudiantes serán los encargados de proveer de la infraestructura necesaria para la ejecución del programa, bien sea en alguno de los laboratorios de la universidad o su laptop.

Nota: En la evaluación del informe se incluirá la ortografía: se restará 0,25 puntos de la nota final por cada error ortográfico, hasta un máximo de 2 puntos. Se incluyen acentos.