Universidad Central de Venezuela

Facultad de Ciencias

Escuela de Computación

Organización y Estructura del Computador 1

Proyecto #3 (Simulación de caché)

Resumen:

En este proyecto, usted deberá implementar una simulación de una memoria principal y una caché de nivel 1. El objetivo es comprender tanto el funcionamiento de la caché como su estructura. Se debe simular tanto el movimiento de líneas de caché como la lectura y escritura de datos en estos dos niveles de la jerarquía de memoria.

Especificaciones:

Parte I (Recepción de parámetros)

Usted debe iniciar el proyecto analizando y entendiendo la estructura básica de la caché y la memoria principal. Recuerde que ambos niveles de memoria pueden ser simulados a través de un array lineal de datos. El programa debe recibir tres parámetros:

./sim <MP> <L1> <Bloque>

Donde:

MP hace referencia al número de bits necesarios para direccionar la memoria principal.

LI hace referencia al número de bits necesarios para direccionar la cache de nivel 1.

Bloque hace referencia al número de bits necesarios para direccionar el bloque.

Ejemplo de ejecución:

./sim 24 19 4

Memoria principal de 16MB

Caché L1 de 512KB

Bloque de 16Bytes

A partir de los argumentos que suministran en la línea de comandos, usted deberá estructurar tanto su memoria principal como su memoria caché definiendo las unidades necesarias para el almacenamiento y movimiento de los datos.

Parte II (Carga de datos)

El estado inicial de la memoria principal está guardado en un archivo de texto ("main.txt"), es decir, usted deberá cargar dicha información en su programa de simulación. El archivo es una secuencia de caracteres. Asuma que cada vez que el programa se ejecuta, la caché está inicialmente vacía.

Parte III (Recepción y ejecución de instrucciones)

El programa deberá leer un archivo de texto llamado "entrada.txt" que contiene las instrucciones de lectura o escritura que referencian los datos en memoria principal. Se deben ejecutar de manera secuencial hasta que se encuentre la instrucción UP. Se definirán tres instrucciones con las cuales se conformará el programa. El formato de correspondencia que usted usará es correspondencia directa. Las instrucciones son las siguientes:

RD (Operación de lectura)

Estructura:

RD <Dirección>

Ejemplo:

RD 000000000000000000000001

La operación RD solicita un byte de la memoria. Si dicho byte se encuentra en caché, habrá un acierto de caché y se escribirá en el archivo salida.txt el mensaje <ACIERTO>, en caso contrario deberá traer el bloque correspondiente desde memoria principal, actualizar la línea de caché adecuada, y debido al fallo de caché se escribirá en el archivo salida.txt el mensaje <FALLO>. Recuerde que debe asegurarse de que cada línea de caché sea válida y de que las etiquetas de la correspondencia y la línea coincidan para determinar que hay un acierto (o fallo) de caché.

WR (Operación de escritura)

Estructura:

WR <Dirección> <Caracter>

Ejemplo:

WR 000000000000000000000001 a

La operación WR escribirá un caracter en la dirección física especificada en la memoria principal y en la memoria cache. Tome en cuenta que se usará escritura directa.

Suponiendo una memoria principal estructurada de la siguiente manera

Bloque	Datos
0	0000000000
1	1111111111
2	2222222222
3	3333333333

Al ejecutar la operación WR 0001 5, el estado final de la memoria sería:

Bloque	Datos
0	0500000000
1	1111111111
2	2222222222
3	3333333333

UP (Operación de actualización)

Estructura:

UP

Ejemplo:

UP

La operación UP guardará el estado de la memoria principal y la caché en sus respectivos archivos. Luego de realizar lo especificado, escribirá un mensaje en el archivo salida.txt y finalizará el programa:

OK

Consideraciones:

Usted debe asumir que el tamaño del bloque es igual al tamaño de la línea en la realización de este proyecto.

El direccionamiento será por byte, por lo que las operaciones de RD y WR leerán o escribirán un byte, esto quiere decir que la palabra es de 1 byte

Antes de definir las estructuras, estudie la organización de la caché de modo que pueda entender la manera en que se realiza el movimiento de datos.

Recuerde que el formato de correspondencia directa es:

| Etiqueta | Linea | Palabra |

Use la siguiente organización como referencia:

Bloque

| Etiqueta | Linea | Palabra |

Por lo que puede tomar en cuenta que Bloque puede referenciar a un bloque de memoria principal.

Importante:

El proyecto debe ser realizado en grupos de dos (2) personas. (Pueden conformarse de diferentes secciones de teoría).

El programa debe ser realizado en lenguaje C, y el nombre del código fuente debe ser "sim.c" (Necesariamente, el nombre del archivo fuente debe ser el especificado).

Debe ser enviado un documento en formato PDF donde se explique la manera en la que usted estructuró su cache y de qué manera se realizan las operaciones. Dicho archivo debe tener el nombre de "análisis.pdf".

El entregable será en un archivo comprimido el código fuente. Dicho archivo seguirá el siguiente formato para su nombre:

Apellido1Apellido2_P1

La fecha de entrega será el día 4/07/18. El sitio de entrega se darán a conocer en las fechas cercanas a la establecida.

Los proyectos entregados después de la fecha estipulada no serán corregidos.

Las copias obtendrán la calificación mínima.

Ejemplo de prueba:

Archivo: main.txt 1111111111111111222222222222222233333333333333334444444444444445 5555555555555556666666666666666677777777777777778888888888888888
Ejecución: ./sim 7 5 4 RD 0000000 RD 1000001 WR 1110000 x WR 1111111 f RD 1110000 RD 1011010 WR 1010001 r WR 1001010 g RD 1010111 RD 1010111 RD 1001010 RD 1001010 UP

Salida (Pantalla): FALLO FALLO ACIERTO FALLO ACIERTO ACIERTO ACIERTO ACIERTO OK
Estado final (main.txt) 1111111111111111222222222222222233333333333333334444444444444445 555555555g555556r666666666666667777777777777777x88888888888888f
Estado final (cache_l1.txt) 5555555555g555556r66666666666666

GDOECI