

PROYECTO

Implementar en C++ un simulador de memoria cache que permita seleccionar entre los siguientes emplazamientos: Correspondencia Directa y Asociativa por Conjuntos. Más específicamente, las opciones de configuración deberán ser cargadas desde un archivo de configuración (conf.in) y la secuencia de direcciones de bloque que se van a simular desde un archivo de trazas (traz.in).

Formato del Archivo de Configuración (conf.in)

Este archivo debe contener 4 líneas en el siguiente orden:

1. Tamaño de Cache: Tamaño total para datos en Cache. Se debe especificar en bytes y ser una potencia de 2. Por ejemplo, un valor de 512 implica una cache de 512 bytes.
2. Emplazamiento: Un valor de "0" indica correspondencia directa y un valor de "1" indica asociativa por conjuntos.
3. Tamaño de Línea: Contiene el tamaño de línea para la cache en bytes. Siempre será una potencia de 2 (por ejemplo 1, 2, 4, 8, etc).
4. Política de Reemplazo: Un valor de "0" indica LRU y un valor de "1" para:

Cédula de Identidad que termina en:	Política de Reemplazo
0-1	MRU
2-3	LFU
4-5	MFU
6-7	FIFO
8-9	Reemplazo Aleatorio

Formato del Archivo de Trazas (traz.in)

Este archivo debe contener n líneas donde cada una contiene una dirección de bloque. Por ejemplo, para la siguiente secuencia de direcciones de bloque: 0,8,0,6,8; se requiere el siguiente archivo traz.in de 5 líneas:

0
8
0
6
8

Resultados

El programa deberá escribir un archivo de salida(resu.out) con la siguiente información:

Dirección del Bloque de Memoria Accedido	¿Acierto o Fallo?	Contenido de los bloques de cache después de cada acceso
--	-------------------	--

, y la frecuencia de aciertos y la frecuencia de fallos. Junto con el programa se deberá entregar un informe con el Análisis y Discusión de los Resultados así como los casos de prueba utilizados. Se recomienda utilizar el sistema de composición Latex.

La presentación se realizará a través de un video con duración comprendida entre 5 a 10 minutos de duración. La motivación del video es mostrar la importancia del simulador de memoria cache implementado así como las técnicas algorítmicas utilizadas a través de la exploración detallada del código fuente y su compilación y puesta en marcha.

Ambiente de Desarrollo

Se debe utilizar C++11 o superior (C++14-C++17-C++20). Se puede utilizar STL y BOOST. El simulador debe compilar y ejecutarse en Sistema Operativo basado en GNU Linux utilizando GNU Compiler Collection. Se debe incluir un archivo makefile.

Modalidad de Trabajo: Individual.

Forma de Entrega: Correo Electrónico. El Asunto del Correo Electrónico debe ser: <Cedula, Apellido y Nombre del Estudiante> Proyecto. Los archivos adjuntos a la entrega deben ser almacenados en un directorio comprimido en formato *.zip y etiquetado de la siguiente manera: <Cedula, Apellido y Nombre del Estudiante> Proyecto. Se recomienda utilizar Github.

Fecha de Entrega: 15/01.

ÉXITO!!!