

Simulador de Memoria Caché

Objetivo

Se trata de realizar un simulador de memoria caché donde podamos simular diferentes configuraciones y los efectos de utilizar reemplazo Aleatorio o LRU:

Tamaño de línea: tamaño en bytes potencia de dos.

Número de conjuntos: número entero potencia de dos.

Asociatividad: número entero mayor o igual que 1.

Reemplazo: 0: Aleatorio, 1: LRU.

El simulador leerá la configuración desde un fichero “config.txt” de este estilo.

```
Tlin: 64
Ncon: 1024
Asoc: 6
Reem: 1
```

El simulador ha de calcular la **tasa de fallos** que produce la caché al acceder a memoria siguiendo una traza de accesos almacenada en “traza.txt”, cuyo formato será de este estilo:

```
0x42ad34f1 0x42ad5534 0x12345678 ...
```

donde la traza estará formada por las direcciones en formato hexadecimal a las que supuestamente accede el procesador (“0x” significa que el conjunto de caracteres es un número hexadecimal).

Lectura y escritura de ficheros

Para leer un número en hexadecimal de un fichero podemos utilizar la función “fscanf”:

```
unsigned long int n;
FILE *f;
fscanf(f, "%lx", &n);
```

Generaremos nuestros propios ficheros de traza haciendo que los programas escriban las direcciones utilizadas mediante la función “fprintf”. Véase el ejemplo (*):

```
double v[MAX], res;
FILE *f;
f=fopen("traza.txt", "w");
for (i=0 ; i<MAX ; i++) {
    res=res+v[i];
    fprintf(f, "%p ", &v[i]);
}
```

(*) esto es un ejemplo concreto que guarda la traza de acceso al vector *v*, el programa deberá calcular la tasa de fallos para cualquier traza generada a partir de cualquier código, NO únicamente para este ejemplo.

Información de utilidad para abrir y cerrar ficheros en C:

https://es.wikibooks.org/wiki/Programaci%C3%B3n_en_C/Manejo_de_archivos

Cálculo del valor de un conjunto de bits

Para calcular el valor que representa un conjunto de bits consecutivos pertenecientes a una dirección, utilizaremos la función “rangoBits” que podemos encontrar en el fichero “SimuladorCache.c”.

Esta función calcula el valor entre dos bits (“bitMenor” y “bitMayor”) ambos inclusive y nos servirá para poder calcular el número de línea de MP a la que pertenece la dirección y el conjunto de caché donde ira alojada dicha línea.

Los parámetros son: el valor de la dirección, el bit de menor peso y el bit de mayor peso del conjunto a considerar. *Es requisito imprescindible el uso de dicha función.*

Formato de entrega

Un fichero “Gxx-SIMU-DosApellidosA-DosApellidosB.zip” que contenga el fichero “SimuladorCache.c” bien organizado y documentado, explicando entre comentarios las diferentes funciones y/o partes del código.