



IIC2343 – Arquitectura de Computadores (II/2016)

Tarea 5

Fecha de entrega: jueves 10 de noviembre de 2016 a las 23:59 horas

Ayudante encargado: Patricio Díaz (pndiaz1@uc.cl)

Jerarquía de memoria

En esta tarea tendrán la oportunidad de experimentar con el diseño de una memoria caché, desde el punto de los mecanismos de acceso a datos y las políticas de escritura. Mas específicamente, deberán programar la simulación de una jerarquía de memoria de dos niveles (caché y memoria principal), la cual se ve enfrentada a una serie de peticiones de lectura y escritura.

La tarea desarrollada debe recibir como entrada a través de la consola, el nombre de un archivo de texto que contiene la especificación completa de la jerarquía requerida y de la secuencia de lecturas/escrituras. Como salida, la tarea debe entregar el *hit-rate* y el *write-rate* correspondientes a la secuencia, donde el *write-rate* se define como la cantidad de escrituras realizadas efectivamente en la memoria principal dividida por la cantidad de escrituras totales indicadas en la secuencia.

El archivo de entrada tendrá un formato **{propiedad=valor(es)}**, donde **propiedad** indica el elemento a configurar y **valor** indica el o los valores para el elemento. En particular, las propiedades a configurar y sus posibles valores se indican a continuación:

- **CACHE_SIZE**: indica el tamaño de la memoria caché en función de la cantidad de palabras (bytes).
- **LINE_SIZE**: indica el tamaño de las líneas de la memoria caché en función de la cantidad de palabras (bytes).
- **MEM_SIZE**: indica el tamaño de la memoria principal en función de la cantidad de palabras (bytes).
- **CORR**: indica la función de correspondencia a utilizar, de entre las siguientes: DM (mapeo directo), FA (*fully associative*), 2W (2-way associative), 4W, 8W.
- **SUBS**: indica la política de sustitución a utilizar, de entre las siguiente: FIFO, LRU, LFU.
- **WRITE**: indica la política de escritura a utilizar, de entre las siguiente: WT (*write-through*), WB (*write-back*).
- **SEC**: indica una secuencia ordenada de largo arbitrario de lecturas/escrituras. Los valores serán números que indican la **dirección** de memoria requerida, seguidos de la letra W en caso de tratarse de una escritura.

Un ejemplo de un posible archivo de entrada puede verse a continuación:

```
CACHE_SIZE=8
LINE_SIZE=2
MEM_SIZE=32
CORR=FA
SUBS=FIFO
WRITE=WB
SEC=0,1,2,3W,4,5,6,7,8,9,8,7W,6,5,4,3,2,1,0W
```

Implementación

Para la realización de esta tarea deberán trabajar en el lenguaje Python 3.x, sin ocupar bibliotecas externas que implementen la lógica de la tarea. Sí está permitido el uso de bibliotecas para leer archivos, imprimir en pantalla y manejar parámetros por línea de comandos.

Para ejecutar la tarea, se deberá entregar por línea de comandos el nombre del archivo que contiene los números:

```
$ python tarea.py nombre_archivo.txt
```

Luego de esto, el programa deberá imprimir en pantalla los resultado requeridos.