

ORGANIZACIÓN DE COMPUTADORES

LABORATORIO 3

Profesores: Néstor González & Leonel Medina

Ayudantes: Matías Fuentes & Ricardo Álvarez

CAPÍTULO 1. CONTEXTO

Como lo confirma la teoría vista en cátedra, uno de los factores que inciden en el rendimiento de un programa es la cantidad de accesos a memoria que debe realizar el procesador para obtener datos en tiempo de ejecución. Para evitar el buscar datos en la memoria principal se utilizan memorias de tipo caché, las cuales son de un tamaño menor y se ubican de forma cercana al procesador. Por esta razón es que permiten una obtención de datos considerablemente más rápida de lo que sería un acceso a la memoria principal. Sin embargo, debido la capacidad de estas memorias, no pueden contener todos los datos existentes en la memoria principal, por lo que estos datos deben ser ingresados por bloques a una dirección determinada de la caché, y en caso que dicha dirección esté ocupada, se debe reemplazar el contenido anterior por el que está ingresando a la caché.

Para identificar la ubicación del bloque en caché existe un índice, el cual dependiendo de la configuración del caché (si es mapeo directo, full-asociativo o de n-vías), determinará la posición en la que el dato consultado se buscará. En caso de que sea necesario realizar un reemplazo de bloque en el índice consultado porque no se pueden ingresar más datos, se aplican las políticas de reemplazo que el cache en cuestión utiliza. Entre estas políticas se encuentran MRU, LRU, LIFO y FIFO.

Cabe preguntarse según lo anterior, ¿qué configuración de caché utilizar?, ¿Cual me entregará una mayor tasa de hit?, ¿podría darse que configuraciones muy distintas se lleguen a comportar de forma similar?.

CAPÍTULO 2. INSTRUCCIONES

En el presente laboratorio se aborda la pregunta: ¿Qué configuración de caché es mas eficiente para un determinado conjunto de datos?

Para esto debe recibir una lista de direcciones de memoria como archivos de entrada, datos que serán representados por números enteros separados por un salto de línea y deben ser consultados a la memoria. Cabe aclarar que el tamaño de palabra corresponde a 32 bits.

Además de los datos, el programa debe recibir el nombre del archivo con los datos, la cantidad de palabras por bloque y el tamaño de la memoria caché mediante la función *getopt* de la librería *unistd.h*, donde los parámetros a recibir serán.

Parámetro	Bandera	Valores	Descripción
Nombre Archivo de Datos	-n	texto.txt	Nombre de archivo con extensión <i>.txt</i>
Tamaño Datos	-m	1 - n	El tamaño de los datos de la memoria cache en bits
Cantidad de palabras	-p	1 - p	Cantidad de palabras por bloque en caché

Cuadro 2.1: Parámetros a recibir mediante función *getopt*.

Por ejemplo, suponiendo que estamos buscando la mejor configuración para una caché de 2 palabras por bloque, con un tamaño de 4KB, y con un nombre de archivo **entrada.txt**, entonces la línea de ejecución de este programa sería:

`./nombreLab3 -n entrada.txt -m 4000 -p 2`

Dados estos datos, se necesita que responda que tipo de caché y configuración son las más adecuadas para un determinado conjunto de datos, esto es, cuando la tasa de hit es la mayor posible.

Vale notar que la librería *unistd.h* se encuentra solamente disponible para sistemas operativos Unix, por lo que en caso de utilizar Windows, la entrada deberá seguir la misma estructura pero sin las flags, teniendo que obtener estos datos directamente a partir del arreglo argv.

`./nombreLab3 entrada.txt 4000 2`

Debe especificarse en las herramientas usadas en el informe el sistema operativo ocupado. Los ayudantes recomiendan el uso de algún sistema operativo Unix, ya que este es usado en la revisión y en ramos siguientes a Organización de Computadores es obligatorio el uso de éstos.

Para el resultado se deben crear dos archivos de salida. El primero debe mostrar la mejor configuración de la caché, esto es, el grado de asociatividad de la caché, la política de reemplazo, y las estadísticas obtenidas (tasa de hit y de miss). La mejor configuración es aquella con mayor cantidad de **hits**, en caso de ser más de una, se deben mostrar todos los casos.

El segundo archivo de salida debe contener el estado de la caché después de la última consulta realizada.

2.1 EXIGENCIAS

- El programa debe estar escrito en C.
- El programa y el informe deben ser entregados como una carpeta comprimida en formato `zip`.
- El programa debe ser entregado en su código fuente junto a un archivo `makefile` para su compilación.
- El programa debe cumplir con un mínimo de calidad de software (funciones separadas y nombres tanto de funciones como de variables, de fácil comprensión).
- El informe escrito no debe exceder 10 páginas de texto escrito, sin considerar portada ni índice, en caso contrario, por cada página extra, se descontará 5 décimas a la nota final.
- El informe escrito debe ser entregado en formato PDF, por lo que puede ser desarrollado en LaTeX, Microsoft Word, OpenOffice Writer, etc.

2.2 RECOMENDACIONES

- Pueden usar estructuras de datos para almacenar la información de los registros y señales de control.
- Utilizar la plantilla de LaTeX disponible en el Moodle del curso.
- Consultar a los ayudantes y en Moodle del curso.
- Se recomienda el uso de sistemas operativos basados en Unix para evitar problemas, ya que la revisión es principalmente en estos, además de que en cursos posteriores el uso de estos es obligatorio (Sistope).

2.3 DESCUENTOS

- Si el programa no compila será evaluado con nota mínima.

- Por cada exigencia no cumplida, se descontarán 5 décimas a la nota, a excepción las que ya mencionan su descuento.
- Por cada dos faltas ortográficas o gramaticales en el informe, se descontará una décima a la nota.
- Por cada falta de formato en el informe, se descontará una décima a la nota.

2.4 EVALUACIÓN

- La nota del laboratorio será el promedio aritmético del código fuente con el informe, y en caso de obtener una nota inferior a 4 en alguna de las dos calificaciones, se evaluará con la menor nota.
- En caso que no se entregue alguno de los dos, se evaluará con la nota mínima.
- Este laboratorio debe ser entregado el día 20 de agosto del año 2018, hasta las 23:55 hrs. La no entrega o entrega tardía se evaluará con la nota mínima. Bajo ningún concepto se postergará la entrega de este laboratorio.
- En caso de dudas o problemas en el desarrollo, comunicarse con su ayudante de laboratorio.