

Laboratorio 3 Organización de Computadores: Cache

Profesores: Felipe Garay, Erika Rosas, Nicolás Hidalgo

Ayudante: Ian Mejias

Email: {nombre}.{apellido}@usach.cl

14 de enero de 2016

Enunciado

El caché es una importante forma de mejorar la velocidad de ejecución de programas, pero es necesario tomar en cuenta la forma en la que los programas acceden a los datos en memoria principal para poder aprovechar todas las capacidades de este componente.

En este laboratorio se usará nuevamente el simulador MARS para analizar como las diferentes formas de acceder a memoria pueden afectar las tasas de hit y miss.

Se deben implementar la función fold ¹ en su versión que opera en un arreglo y la que opera en una lista enlazada (debe entregar dos archivos asm, uno por cada versión). Asuma que los datos con los que trabajar se encuentra en un arreglo en la sección .data. Para el caso de la lista enlazada, debe leer ese arreglo y transformarlo a una lista enlazada pero debe tener en cuenta que esa creación de la estructura no debe ser considerada en sus experimentos con el caché.

Un ejemplo en python, donde fold se llama reduce, sería:

```
l = [1, 2, 3]
def sumar(acomulador, valor):
    return acomulador + valor

inicial = 0

l2 = reduce(sumar, l, inicial)
l2 == 6
```

Es decir, fold/reduce recibe una función, una lista y un valor inicial para el acumulador. Este tipo de funciones que reciben como argumento a otras funciones se llaman “Funciones de alto orden” ² y lo interesante es que reduce abstrae del programador las estructuras de control como while o for y ya no debe preocuparse sobre como se itera, solamente de como se van a combinar los elementos de la lista en un solo valor.

Además, basta con cambiar la función que recibe reduce para cambiar el comportamiento de esta haciendo que sea reusable en otras partes del código.

Por ejemplo si se quisiera calcular la multiplicación de todos los números se puede escribir:

```
l = [1, 2, 3]
def mult(acomulador, valor):
    return acomulador * valor

inicial = 0
```

¹<https://wiki.haskell.org/Fold>

²https://en.wikipedia.org/wiki/Higher-order_function

```
l2 = reduce(mult, l, inicial)
l2 == 6
```

Una implementación en python de reduce podría ser:

```
def reduce2(f, l, init):
    acc = init
    for x in l:
        acc = f(acc, x)
    return acc
```

Note que la función es pasada como argumento y en ningún momento es referenciada por el nombre dentro de reduce2, si no que es una variable llamada f.

El procedimiento del laboratorio se detalla a continuación:

- Implementar las dos versiones de la función fold.
- Preparar una cantidad de datos suficiente para poder sacar conclusiones.
- Proponga una hipótesis sobre que afectaría el desempeño del cache antes de realizar las pruebas.
- Utilizar el simulador de caché de MARS con distintas configuraciones. Trate de encontrar razones por las cuales cambia la tasa de hit y miss con las opciones del cache y las diferencias entre utilizar un arreglo o una lista enlazada.
- Concluya sobre el trabajo realizado y exponga un análisis de los resultados.

Consideraciones

Generales

- Las copias entre compañeros e Internet serán calificadas con la nota mínima.
- El laboratorio es individual.
- Si el laboratorio es entregado con 4 o más días de atraso se considerará que no se ha presentado ningún trabajo.
- El programa corresponde a un 20 % de la nota y el informe un 80 % de la nota de entrega del laboratorio. Note que no puede hacer el informe sin hacer los programas ya que debe basarse en ellos.
- Por cada día de atraso se descuenta 1 punto a la nota general del laboratorio (esto incluye la entrega atrasada de informes). Ej: si el laboratorio se debe entregar a las 23:50 y se entrega a las 23:55 hay un punto de descuento. Si se entrega a las 23:55 del siguiente día hay dos puntos de descuento. Si tiene un 7 como nota de laboratorio (promedio entre informe y programa) entonces tendría un 6.
- Debe entregar en el espacio habilitado en usachvirtual una carpeta comprimida (.zip o .tar.gz o .tar.bz2) con el código fuente del programa en una carpeta llamada "src" y el informe y manual de usuario en pdf. Este archivo debe llamarse: Apellido1_Apellido2.(zip—tar.bz2)
- Cree un simple manual de usuario (sin explicar la instalación) sobre como interactuar con su programa.
- La fecha de entrega es el viernes 4 de marzo a las 23:55 por usachvirtual. No se corregirán laboratorios entregados por otros medios.
- El laboratorio consiste en informe más programa. Si no entrega uno de estos elementos se calificará con la nota mínima.

Programa

- Utilice las convenciones de MIPS para las funciones y registros, de lo contrario perderá décimas (una décima por cada dos convenciones mal utilizadas).
- Divida el programa en funciones utilizando las instrucciones `jal` y `jr`, se descontarán décimas por programas que no sean modulares.
- Recuerde poner comentarios en su programa e indicar los argumentos que reciben las funciones y que es lo que retornan (también indique aquellas funciones que no tienen un valor de retorno).
- El programa debe funcionar en sistemas GNU/Linux. Para que el programa se califique de forma correcta debe poder ejecutarse sin problemas en este sistema operativo, de lo contrario no se va a poder evaluar el programa.
- Ambos programas valen 50 %.

Informe

- El informe debe ser entregado impreso en secretaría con el nombre de del profesor, ayudante y asignatura a más tardar 12 hrs desde la fecha de entrega por usachvirtual. Además debe subir el informe junto con el programa a usachvirtual. Ej: Si el laboratorio se debe entregar a las 23:50 por usachvirtual, entonces el informe debe estar a las 12:00 en secretaría. Se aplicará el mismo criterio que el que aparece en las consideraciones generales para los atrasos.
- El informe debe contener las siguientes secciones (consulte el formato de memoria para más información sobre lo que deben llevar estas secciones):
 1. Introducción: Objetivo generales, objetivos específicos, organización del documento, motivación, problema, herramientas y presentar hipótesis. (10 %)
 2. Marco teórico: Cache y listas enlazadas (20 %)
 3. Desarrollo: Explicar como se construyeron los programas y los experimentos realizados (solamente los resultados, no análisis) (20 %)
 4. Análisis de los resultados (30 %): Explique si su hipótesis fue correcta, además reflexione sobre los resultados anteriores explicando los números obtenidos en los experimentos del desarrollo. ³
 5. Conclusión: Sobre los objetivos planteados en la introducción y la hipótesis. (10 %)
 6. Referencias (10 %)
- Debe utilizar el formato de presentación de memoria disponible en el archivo “Propuesta de normas para presentación del trabajo de titulación de pregrado, Departamento de Ingeniería Informática” que puede encontrar en la página del curso. Se descontará una décima por cada error en el formato.
- Cuide la ortografía. Se descontará una décima por cada falta.
- Incluya referencias en la bibliografía indicando claramente el texto citado. Utilice el formato APA (parte del formato del informe). En caso de encontrar textos que no hayan sido citados se considerará como copia.

³No es correcto decir X aumenta Y veces en el caso Z. Debe explicar porque X aumenta Y veces