

Proyecto Programado II

Profesor: Santiago Núñez Corrales

4 de junio de 2011

1. Introducción

Los *benchmarks sintéticos* son programas diseñados para caracterizar de manera artificial el rendimiento de un computador. Una diferencia con programas reales radica en su tamaño corto, uso de conjuntos específicos de instrucciones y enfoque en un solo componente de la arquitectura. La razón fundamental para desarrollar este tipo de benchmarks es su simplicidad de ser analizados, así como la posibilidad de aislar factores particulares del rendimiento.

Este proyecto programado consiste en la implementación de un benchmark específico para el CPU.

2. Descripción del Proyecto

Desarrollar un benchmark que permitan cuantificar la cantidad de operaciones de CPU por unidad de tiempo por medio de un programa sintético.

1. **Benchmark de CPU.** El benchmark de CPU debe ser un programa no mayor a 500 líneas en código ensamblador que lleve una cuenta de la cantidad de instrucciones ejecutadas. El programa debe recibir como parámetros la cantidad de iteraciones que se ejecutarán y una bandera que indicara si se utilizará un algoritmo que produce cache misses o no (`-cm|-nocm`). Debe elegir un algoritmo cuya complejidad sea al menos $O(n^3)$. Solo se utilizarán instrucciones enteras.
2. **Medición.** Para la medición del tiempo, deberá utilizar el programa de Unix `time` que indicara el tiempo real de la aplicación. Con ello, para cada ejecución del programa podrá calcularse la cantidad de instrucciones enteras por segundo.

2.1. Código Fuente

El código fuente debe compilarse y ejecutarse en un sistema GNU/Linux con el ensamblador GAS, sintaxis AT&T.

2.2. Documentación

La tarea programada debe venir acompañada de documentación externa en formato IEEE Transactions (LaTeX o plantilla para Word) que contenga una descripción del problema, la estrategia de solución implementada y un conjunto de pruebas, así como una copia verbatim (completa) del código fuente de la aplicación. La plantilla de documentación puede descargarse de

<http://www.ieee.org/documents/TRANS-JOUR.doc>. Con respecto a las pruebas, los benchmarks deben ejecutarse al menos 25 veces y con ellos calcular el promedio y la desviación estandar y para $1 \leq n \leq 10$ instancias del programa ejecutándose de forma concurrente (sugerencia: construya un script de Bash que utilice el operador & para invocar programas en serie sin esperar la finalización de uno anterior).

3. Evaluación

1. **Correctitud.** El programa compila, se ejecuta y generera resultados correctos sin errores o advertencias. [50 %]
2. **Completitud.** El programa cumple con la especificación de la tarea programada. [30 %]
3. **Estructura y organización.** El código fuente está apropiadamente documentado; existe una organización lógica en funciones que cumplen objetivos específicos; los nombres de los procedimientos permiten leer fácilmente el código. [10 %]
4. **Diseño y proceso.** La documentación externa es apropiada. El diseño del programa es modular y comprensible. El entregable se envia en un archivo tar.gz. [10 %]

4. Entrega

El proyecto programado deberá entregarse el día 17 de Junio de 2011 al correo electrónico arquitectura.santaclara@gmail.com con hora máxima del servidor 23:59:59.

5. Puntos adicionales

1. Programa de estadísticas. Se construirá un programa que almacene en un archivo de formato texto el nombre del programa ejecutado, la cantidad de instrucciones, el tiempo tomado y la cantidad de instrucciones por unidad de tiempo. El programa debe construirse en C. Debe integrarse el benchmark como una liga ensamblador-C. (30 pts).
2. Benchmark de IO. Utilizando el mismo algoritmo, se construirá un programa en ensamblador que mida la cantidad de instrucciones por segundo en donde al menos en cada iteración se efectúe una lectura y una escritura a disco. (35pts)
3. Benchmark de FPU. Utilizando el mismo algoritmo, se construirá un programa en ensamblador con instrucciones de punto flotante que mida la cantidad de instrucciones por segundo. Debe mostrar el efecto de ejecutar $n < 5$ instancias simultáneas del programa. (35pts)