



503305 Arquitectura de Computadores Tarea 3

Javier Vidal Valenzuela

27 de noviembre de 2023

1. Introducción

En el presente documento se describe los objetivos y condiciones para la realización del trabajo práctico evaluado del curso Arquitectura de Computadores.

2. Objetivo

Con este ejercicio se persigue que cada alumno reafirme los conceptos sobre optimización de rendimiento de programas usando las herramientas expuestas en el curso Arquitectura de Computadores.

3. Enunciado

El problema a resolver en este trabajo consistirá en escribir diferentes versiones de algunos procedimientos y funciones, calculando en cada caso el rendimiento medido ya sea en tiempo de CPU o en ciclos de reloj. Para lograr este objetivo Ud. debe:

- Implementar, en lenguaje C, una versión no optimizada de cada ejercicio solicitado y medir el tiempo de ejecución (CPU time o ciclos de reloj) sin optimizar.
- Implementar diferentes versiones optimizadas y volver a evaluar el rendimiento.
- Explicar por qué, a nivel conceptual, se producen los resultados.
- Escribir un informe con los experimentos realizados y los resultados obtenidos.

4. Plazos y Entrega

El resultado de esta tarea debe ser enviado por Canvas en el módulo habilitado para el efecto. El plazo de entrega es el viernes 8 de diciembre de 2023 a las 23:59 horas

La entrega consistirá en :

- Un **informe** que contenga:
 - una portada identificado al equipo, sus integrantes y la asignatura;
 - una introducción explicando el propósito de la tarea y los resultados generales obtenidos;
 - un desarrollo en que se debe explicar como se implementaron (a nivel conceptual) las diferentes versiones de los procedimientos, los datos y estructuras de datos usados y los resultados obtenidos.

- las conclusiones
- El(los) **programa(s) fuente(s) y archivo(s) de dato(s)**(si es que los usa) con el código C que resuelven los problemas.
- Un archivo **readme** con las instrucciones y/u órdenes para compilar y ejecutar los programas.
- Todos los archivos anteriores deben venir empaquetados en un archivo ZIP o RAR, llamado EQUIPO_XX.ZIP, donde XX es el número de su equipo.

NO se debe enviar código ejecutable!!!.

5. Problemas

1. Escriba un procedimiento optimizado `combine5` usando loop unrolling con 4 operaciones por ciclo equivalente al procedimiento `combine1` (versión con máximo uso de abstracciones) visto en clases:

```
void combine1(vec_ptr v, data_t *dest) {
    int i;
    *dest = 0;
    for (i = 0; i < vec_length(v); i++) {
        data_t val;
        get_vec_element(v, i, &val);
        *dest = *dest + val;
    }
}
```

considere las estructuras de datos y procedimientos anexos vistos en clases. Agregue en su versión todas las optimizaciones a `combine1` vistas en clases.

2. Realice pruebas para hacer comparaciones entre los procedimientos `combine1` y `combine5`.
3. Construya y pruebe un procedimiento optimizado `combine6` que use loop unrolling con 4 operaciones por ciclo con paralelismo de 4 vías.
4. Un procedimiento que calcula el n-ésimo término de la Serie de Fibonacci es:

```
long long int fibonacci(long long int *n) {
    if(*n==1) return 0;
    if(*n==2) return 1;
    long long int m=*n-1;
    long long int p=*n-2;
    return fibonacci(&m)+fibonacci(&p);
}
```

mida el tiempo de ejecución de este procedimiento.

5. Escriba un procedimiento optimizado equivalente que incluya la eliminación de referencias innecesarias a memoria y loop unrolling con 4 operaciones por ciclo y paralelismo de 4 vías. Calcule los tiempos de ejecución para esta versión.
6. Escriba un procedimiento no optimizado (con máximo uso de abstracciones) que calcule el invertido de un string:

```
void invertir(char *s) {  
    ...  
}
```

es decir, si `s` almacena el string “Arquitectura de Computadores” al terminar de ejecutar el resultado de la ejecución de `invertir` debe ser “serodatuPmoC ed arutCetiurA”. Mida el tiempo de ejecución del procedimiento implementado para diferentes tamaños de strings.

7. Escriba al menos tres versiones optimizadas (versiones con optimización acumulativa) equivalentes a `invertir` y mida sus tiempos de ejecución

6. Pauta de evaluación

Para evaluar el resultado de la tarea se tendrán en cuenta los siguientes aspectos:

- **Puntualidad:** Se debe respetar la fecha y hora de entrega del trabajo. Es importante que Ud. planifique su trabajo para que se cumplan los plazos establecidos para su entrega, incluso teniendo en cuenta posibles inconvenientes que se les pudieran presentar.
- **Presentación:** La presentación del informe debe ser formal, con una estructura que contemple una portada y capítulos separados para cada uno de los aspectos relevantes de la descripción del trabajo. Además, la propuesta debe estar escrita en un computador, sin errores de ortografía ni de redacción.
- **Ejecución de(l/los) programa(s):** Cada alumno se debe asegurar de que su aplicación compile y ejecute correctamente, para ello el código debe estar libre de errores y se deben entregar todos los datos necesarios para procesarlos. Finalmente, es responsabilidad del alumno entregar instrucciones claras que indiquen cómo compilar y ejecutar el programa.
- **Calidad y claridad del informe:** El informe dará cuenta de la comprensión de los conceptos utilizados en sus procesos de optimización, por lo tanto, para cada ejercicio debe quedar claro qué procesos de optimización fueron utilizados, cómo funcionan y qué factores de la ejecución afectan. Utilice todas las herramientas que considere necesarias para explicar esta parte: datos usados, cantidad de datos usados, código assembly de los programas, etc.