



Casa abierta al tiempo

UNIVERSIDAD AUTÓNOMA METROPOLITANA

Unidad Iztapalapa

Tarea 1. Calibración

Autor

Martínez Buenrostro Jorge Rafael

2203040824

Profesor

Miguel Ángel Pizaña López

6 de agosto de 2024

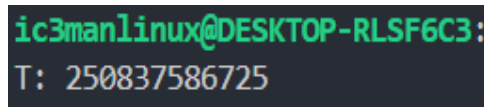
Descripción del problema

En clase vimos el modelo RAM el cual con cuatro supuestos nos ayuda a analizar de una forma más sencilla el orden del tiempo de ejecución de un algoritmo. Uno de estos supuestos es que cada operación elemental: la suma, la resta, la multiplicación, la división, los saltos de línea condicionales, las asignaciones y los paros, toman un tiempo constante para ejecutarse.

Sin embargo, esto no se puede cumplir para computadora debido a que cada procesador es diferente. Dentro de estas diferencias podemos resaltar la velocidad de reloj del procesador, la cantidad de núcleos y la forma en la que paralelizan las tareas. Además de las diferencias en el procesador podemos destacar la cantidad de RAM de cada computadora y el SO de la computadora. Todos estos factores juntos hacen que la cantidad de tiempo que toma en ejecutarse una operación elemental sea diferente en cada computadora.

Programa

El programa de la tarea repite 10^9 o mil millones de veces un ciclo for. Antes de iniciar el ciclo se guarda el tiempo del CPU dentro de una variable, después se realiza el ciclo, para después guardar el tiempo del CPU. Al final se calcula la diferencia en nanosegundos entre los tiempos de inicio y de fin. Esta diferencia se muestra en la terminal como se puede ver en la siguiente figura



```
ic3manlinux@DESKTOP-RLSF6C3:
T: 250837586725
```

Figura 1. Resultado de la ejecución del programa
Calibración

Ahora debemos convertir 250837586725 nanosegundos a un número más fácil de comprender. Esto se puede lograr de la siguiente manera:

$$\text{segundos} = \frac{250837586725}{10^9} = 250.8 \text{ segundos}$$

Ahora 250.8 segundos no es muy útil, vamos a pasarlo a minutos. Esta conversión nos da aproximadamente 4 minutos y 10 segundos. Este tiempo es muy parecido al tiempo promedio que tarda el metro de la CDMX en recorrer la distancia entre dos estaciones.

Descripción del sistema

- ⇒ **SO** – Ubuntu
- ⇒ **Versión** – 20.04-6 LTS
- ⇒ **Número de procesadores** – 1
- ⇒ **Número de núcleos de cada procesador** – 2
- ⇒ **Número de hilos por núcleo** – 2
- ⇒ **Frecuencia de operación** – 2711.998 MHz
- ⇒ **Cantidad de memoria RAM** – 12
- ⇒ **Cache L1d** – 64 KiB
- ⇒ **Cache L1i** – 64 KiB

- ⇒ **Cache L2** – 512 KiB
- ⇒ **Cache L3** – 3 MiB
- ⇒ **Cantidad de RAM** – 3.5 Gi
- ⇒ **Cantidad de SWAP** – 1.0Gi

Conclusiones

Recordando el tiempo obtenido en nanosegundos 250837586725. Ahora veremos cuanto tiempo tarda una operación; hay que recordar que el programa hacía mil millones de operaciones. Dividiendo obtenemos que una operación elemental tarda 2.5083×10^{-7} segundos.

La principal dificultad que tuve al iniciar esta práctica fue el obtener los detalles del SO y del procesador ya que uso un subsistema de Linux en Windows 10. Nunca había tenido que buscar esta información; sin embargo, resultó ser más sencillo de lo que esperaba conseguir todos y cada uno de los datos solicitados.

Para obtener la información utilicé tres instrucciones en la consola; la primera fue *lsb_release -a* con la que pude obtener la información del sistema operativo, la segunda fue *lscpu* con la que pude obtener toda la información del CPU, y por último *free -h* con la que obtuve la información de la RAM.

Lo que me llevo de esta tarea es la premisa de esta. Aunque la frecuencia y la cantidad de RAM de una computadora no son conceptos nuevos para mí. Me abrió los ojos para poder darme cuenta de la forma en la que estos conceptos afectan el tiempo de ejecución de los algoritmos.