



**Tecnológico
de Monterrey**

Cálculo del número PI en paralelo usando threads

Implementación de sistemas embebidos avanzados
Grupo 501

07.10.2022

Alumno

Fernando Cerriteño Magaña - A01702790

Profesor

Víctor Rodríguez Bahena

Esta práctica consistió en calcular el valor de pi con el método de MonteCarlo en c usando threads.

Enlace del repositorio en GitHub:

https://github.com/Shedew/parallel-programming-ITESM/blob/main/practice_1/aproxPi.c

Gráfica comparativa del tiempo de ejecución cuando se tienen 1, 2, 4, 8 y 12 threads

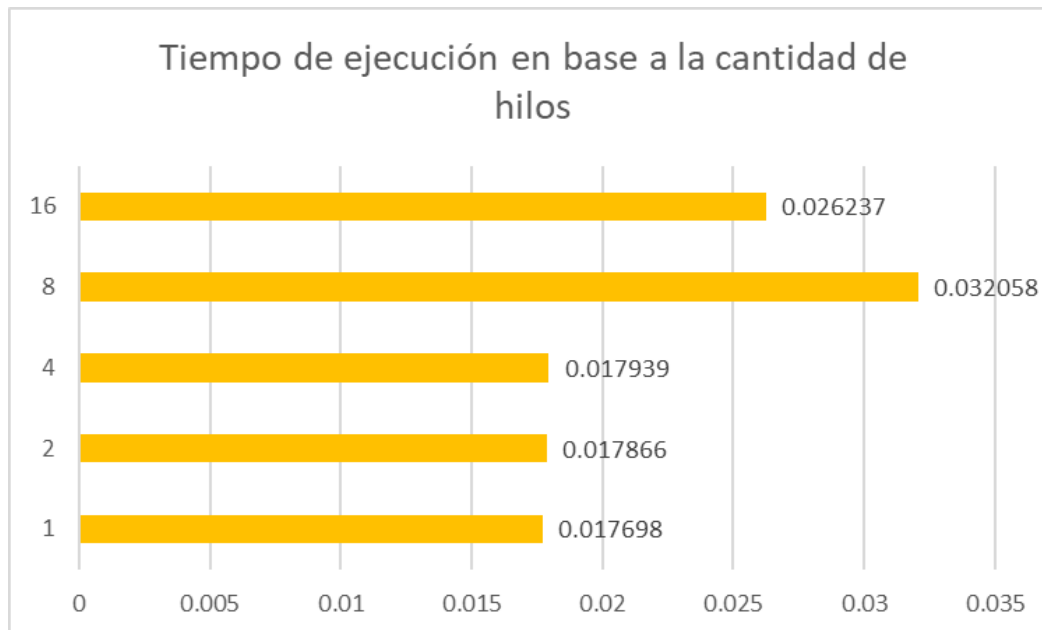


Figura 1. Gráfica comparativa

Grafica usando HTOP de la utilización del CPU

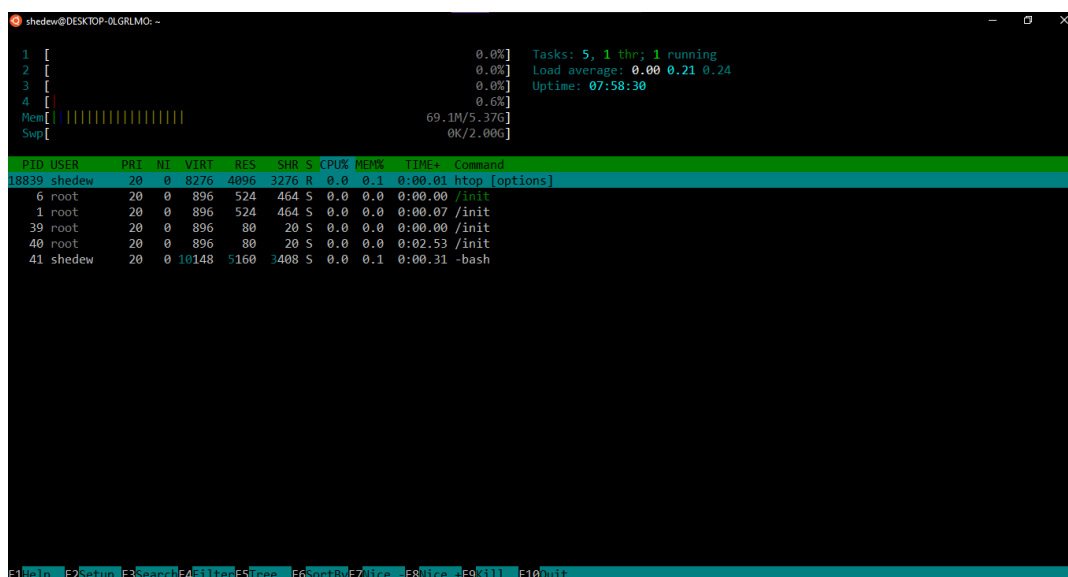


Figura 2. Utilización del CPU

Conclusiones

El método Montecarlo resulta extremadamente eficaz al momento de predecir valores o fenómenos a pesar de ser un método relativamente sencillo en comparación de otros métodos existentes, así mismo, su implementación en C no resulta demasiado compleja, no obstante al momento de usar threads, la composición del código puede resultar un poco confusa, principalmente por el como se tienen que contar los puntos adentro del área del círculo, esto se debe a que el los threads son los que determinan cuantos puntos quedaron en el círculo, por lo que si el conteo llegara a fallar, o si se implementa de forma errónea, el código va a fallar.

De igual forma se puede observar el tiempo de ejecución usando diferentes threads en la figura 1, algo que llama la atención es el como la situación con el mejor tiempo es cuando se cuenta con un solo thread, esto puede ser debido a que el programa no es lo suficientemente complejo para sacarle todo el provecho a los threads. Esto se debe a que en el código se unen los threads para poder determinar cuantos puntos existieron en el círculo, por lo que al incrementar los threads, este calculo se puede demorar un poco más. No obstante, si el código fuera más complejo, una programación en paralelo podría optimizar el tiempo al ejecutar diversas tareas al mismo tiempo.