



**SISTEMAS OPERATIVOS**

**VERSIÓN PARALELA**

**ALGORITMO DE MULTIPLICACIÓN DE MATRICES**

**AUTOR**

**JUAN FELIPE GALVIS VARGAS**

**PROFESOR**

**JOHN JAIRO CORREDOR FRANCO**

**13 DE ABRIL DE 2024**

**BOGOTÁ D.C**

La implementación del algoritmo de multiplicación de matrices en el ejercicio del paralelismo es una práctica esencial para comprender el uso de los hilos y aprovechar al máximo los núcleos de un computador evitando de esta manera la ociosidad, lo cual se refiere al tiempo durante el cual los recursos de la CPU no se utilizan de manera efectiva debido a la falta de procesos a ejecutar. Al distribuir la carga de trabajo entre múltiples hilos, se puede mantener ocupados los núcleos, maximizando así el rendimiento y acelerando la ejecución de procesos.

La multiplicación de matrices consiste en obtener el producto de una fila de la primera matriz por una columna de la segunda matriz y luego sumar sus resultados. Desde el punto de vista de un computador, las matrices se almacenan en la memoria como arreglos unidimensionales de manera contigua. Esta organización en memoria permite un acceso eficiente a los elementos de la matriz y facilita la distribución del trabajo entre los hilos, los cuales son una secuencia de instrucciones que se ejecutan de manera independiente con otros hilos dentro de un mismo proceso y se distribuyen en la CPU.

Con el fin de relacionar el uso de hilos con el algoritmo para multiplicar matrices, se divide la primera matriz en la cantidad de hilos que se quieran. Para esto, se crea un vector de hilos que almacena los identificadores de los hilos que se crean, y de esta manera cada hilo se encarga de cierta parte de la primera matriz que multiplica a la segunda. Estos hilos van a tener un inicio y un fin que está determinado por el tamaño de la matriz y la cantidad de hilos. Por lo tanto, cada hilo multiplica una parte específica de la matriz y así se logra dividir los procesos en los diferentes núcleos de la CPU.

El ejercicio práctico comienza en reservar el espacio de memoria que se utiliza en todo el programa por medio de un arreglo estático, seguidamente, en la función main, se reciben los argumentos de entrada tales como N y Th que representan el tamaño de la matriz cuadrada y los hilos respectivamente. Se establece que el máximo tamaño que tendrán las matrices es menor a 10 para evitar tener problemas innecesarios con la memoria.

Las matrices cuentan con apuntadores que iteran sobre las filas y columnas para asignar sus valores mediante la función para inicializarlas e imprimirlas. Sin embargo, lo más importante es la función para multiplicar las matrices, la cual recibe un argumento que contiene el tamaño de la matriz y los apuntadores a las matrices. En la función se calcula el rango de filas que cada hilo debe procesar y se asigna su rango de filas en función de su tamaño y la cantidad de hilos (N y Th). Esto se realiza mediante el cálculo de los índices de inicio y fin de las filas que corresponderán a cada hilo. Además, se verifica si el hilo actual es el último, para asegurar que todas las filas sean multiplicadas, luego, se procede a iterar sobre estas filas y las columnas de las matrices con apuntadores creados en la función (pA y pB), y con la ayuda de la variable SumTemp que funciona como un auxiliar se guarda la suma. y finalmente asignar el resultado en la matriz resultado.

En la función principal se crea el vector de hilos anteriormente mencionado, y se inicia un ciclo for que recorre todos los hilos que se van a crear. En cada iteración de este ciclo, se crea un nuevo hilo utilizando la función `pthread_create`. Se asigna al hilo actual un identificador único, que se utiliza para indicar qué parte de la matriz debe multiplicar, y se envía a la función de multiplicar matrices. Después de crear todos los hilos, el programa espera a que todos los hilos terminen su ejecución antes de continuar. Esto se hace utilizando la función `pthread_join`, que espera a que el hilo especificado finalice su ejecución. Se repite este proceso para cada hilo creado, asegurándose de que el programa no continúe hasta que todos los hilos hayan completado su tarea. Finalmente, se hace la impresión de la matriz resultado que llega desde la función que realiza la multiplicación, terminando ahí la implementación del algoritmo para multiplicar matrices haciendo uso de hilos.

Para verificar su adecuado funcionamiento, se hace uso de Matrix Reshish (<https://matrix.reshish.com/es/multCalculation.php>) la cual permite hacer multiplicación de matrices. Haciendo la prueba con el algoritmo de una matriz cuadrada de tamaño 4 se tienen los siguientes resultados:

```

0.000000  1.100000  2.200000  3.300000
4.400000  5.500000  6.600000  7.700000
8.800000  9.900000  11.000000 12.100000
13.200000 14.300000 15.400000 16.500000
-----
0.000000  2.200000  4.400000  6.600000
8.800000 11.000000 13.200000 15.400000
17.600000 19.800000 22.000000 24.200000
26.400000 28.600000 30.800000 33.000000
-----
0.000000  1.000000  2.000000  3.000000
4.000000  5.000000  6.000000  7.000000
600.160000 692.120000 784.080000 876.040000
832.480000 963.160000 1093.840000 1224.520000
-----

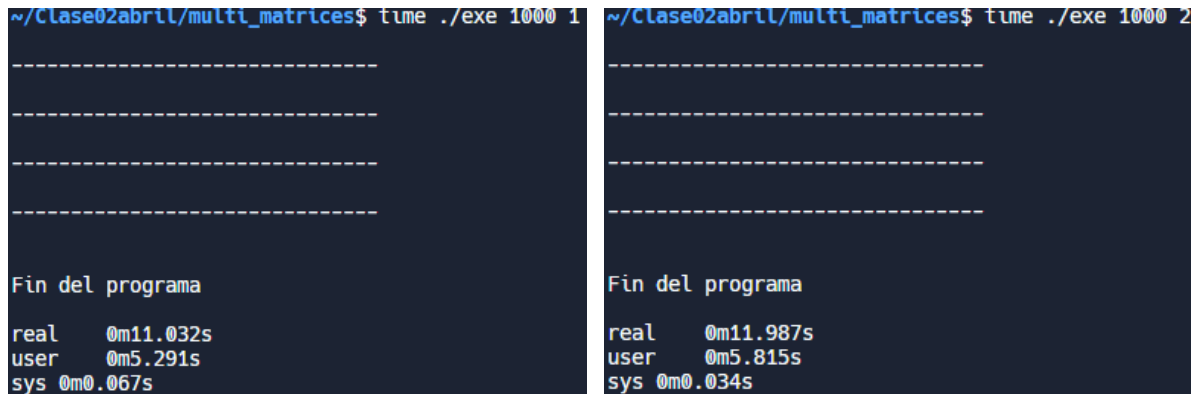
```

En Reshish se obtiene:

	C <sub>1</sub>	C <sub>2</sub>	C <sub>3</sub>	C <sub>4</sub>
1	135.52	150.04	164.56	179.08
2	367.84	421.08	474.32	527.56
3	600.16	692.12	784.08	876.04
4	832.48	963.16	1093.84	1224.52

Lo anterior indica que la multiplicación se está haciendo adecuadamente.

Por otro lado, como el uso de hilos permite dividir el proceso, con el comando `time` seguido del ejecutable (por ejemplo, `time ./exe 1000 1`) se logra ver el tiempo de ejecución del proceso. El comando arroja 3 tiempos: user time, system time y real time que es la suma de user y system. Haciendo pruebas con el algoritmo se obtienen los siguientes resultados:



Configuration	real	user	sys
1 thread (1000 1)	0m11.032s	0m5.291s	0m0.067s
2 threads (1000 2)	0m11.987s	0m5.815s	0m0.034s

Como se puede apreciar, el tiempo del sistema se redujo a la mitad cuando se pasó de 1 a 2 hilos, lo cual nos indica que efectivamente el proceso se divide y tenemos mejor rendimiento tanto en uso como en capacidad de la CPU.

En definitiva, el uso de hilos es muy interesante y permite optimizar y mejorar el rendimiento de los procesos de tal forma que se aprovecha al máximo el entorno tecnológico en el que vivimos hoy en día.