



24-8-2024

Laboratorio 5

JUAREZ HERRERA, DANIEL OSWALDO 23709

Programación de Microprocesadores sec. 10



1. (18 pts.) Explica con tus propias palabras los siguientes términos:

a) Private

Una variable declarada como private significa que cada hilo tendrá su propia copia independiente de esa variable. Lo que un hilo haga con su copia no afectará a las copias de los demás hilos.

b) Shared

Con esta todos los hilos pueden acceder y modificar la misma variable.

c) Firstprivate

Es decir, cada hilo comienza con una copia de la variable como estaba al principio, pero luego puede modificarla libremente sin afectar a los demás.

d) Barrier

Se utiliza para sincronizar los hilos, asegurando que todos hayan completado una sección de código antes de que alguno pueda continuar.

e) Critical

La directiva critical garantiza que solo un hilo a la vez pueda ejecutar un bloque de código específico.

f) Atomic

garantiza que una operación específica se realice de forma indivisible, evitando que otros hilos interfieran y corrompan el resultado.

2. (12 pts.) Escribe un programa en C que calcule la suma de los primeros N números naturales utilizando un ciclo **for paralelo**. Utiliza la cláusula **reduction con +** para acumular la suma en una variable compartida.

a) Define N como una constante grande, por ejemplo, N = 1000000.

b) Usa `omp_get_wtime()` para medir los tiempos de ejecución.

Nombre del programa: sumaNnum.c

3. (15 pts.) Escribe un programa en C que ejecute tres funciones diferentes en paralelo usando la **directiva #pragma omp sections**. Cada sección debe ejecutar una función distinta, por ejemplo, una que calcule el factorial de un número, otra que genere la serie de Fibonacci, y otra que encuentre el máximo en un arreglo, operaciones matemáticas no simples. Asegúrate de que cada función sea independiente y no tenga dependencias con las otras.

Nombre del programa: tresFunciones.c

4. **(15 pts.)** Escribe un programa en C que tenga un ciclo for donde se modifiquen dos variables de manera paralela usando `#pragma omp parallel for`.
- Usa la cláusula `shared` para gestionar el acceso a la variable1 dentro del ciclo.
 - Usa la cláusula `private` para gestionar el acceso a la variable2 dentro del ciclo.
 - Prueba con ambas cláusulas y explica las diferencias observadas en los resultados.

Nombre del programa: dosVariables.c

(30 pts.) Analiza el código en el programa Ejercicio_5A.c, que contiene un programa secuencial. Indica cuántas veces aparece un valor key en el vector a. Escribe una versión paralela en OpenMP utilizando una descomposición de tareas **recursiva**, en la cual se generen tantas tareas como hilos.

Nombre del programa: Ejercicio_5.c

Referencias

- OpenMP Architecture Review Board. (2020). *OpenMP Application Programming Interface Version 5.1*. <https://www.openmp.org/wp-content/uploads/OpenMP-API-Specification-5.1.pdf>
- Chandra, R., Dagum, L., Menon, R., & Perumalla, J. (2000). *Parallel Programming in OpenMP*. Morgan Kaufmann.