

LABORATORIO 05
Distribución de cargas entre hilos en C y OpenMP
Vianka Vanessa Castro Ordoñez 23201

Parte 01

Tabla I. Definiciones de términos

Término	Significado
Private	Cláusula en donde tiene la copia para cada uno de los hilos. De esta manera cada hilo tiene su propia copia privada en la cual pueden realizar cualquier acción.
Shared	Es una sección en donde todos los hilos son capaces de modificar la misma variable.
Firstprivate	Tiene el mismo comportamiento que private pero en este caso se realiza una copia a la variable original que se mantiene consistente que es la que entra a la región paralela.
barrier	Una barrera de todos los hilos en la cual sincroniza a todos los hilos. Todos esperan a que todos lleguen antes de continuar con la siguiente sección del código.
critical	Realiza una organización de hilos para que todos puedan trabajar una sección pero cada hilo trabaja uno a la vez.
atomic	Previene interrupciones de otros hilos en la cláusula. Es similar a critical pero es mejor para proteger la parte de código de culturizar error, principalmente en operaciones simples .

Parte 02

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.

Parte 03

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.

Parte 04

Escribe un programa en C que tenga un ciclo for donde se modifiquen dos variables de manera paralela usando `#pragma omp parallel for`.

- a. Usa la cláusula `shared` para gestionar el acceso a la variable1 dentro del ciclo.
- b. Usa la cláusula `private` para gestionar el acceso a la variable2 dentro del ciclo.
- c. Prueba con ambas cláusulas y explica las diferencias observadas en los resultados.

[Enlace al .cpp](#)

Parte 05

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.