Universidad del Valle de Guatemala
Facultad de Ingeniería
Departamento de Ciencias de la Computación
CC3086 Programación de microprocesadores
Diego José López Campos 23242

Ciclo 2 de 2,024
Temario A

Laboratorio 05

Competencias para desarrollar
Distribuir la carga de trabajo entre hilos utilizando programación en C y OpenMP.

Instrucciones
Esta actividad se realizará individualmente. Al finalizar los períodos de laboratorio o clase, deberá entregar este archivo en formato PDF y los archivos .c en la actividad correspondiente en Canvas.

1. **(18 pts.)** Explica con tus propias palabras los siguientes términos:
   a) private: Dentro de una zona paralelizable crea una copia local o privada de las variables designadas en cada hilo en esta zona.
   b) shared: Especifica que en una zona paralelizable las variables designadas son compartidas.
   c) firstprivate: Las variables designadas se crean de manera local o privada para cada subproceso pero que debe inicializarse con el valor de la variable.
   d) barrier: Define un punto donde se sincronizará donde los hilos de la zona paralelizable no harán ninguna ejecución hasta que todos los hilos estén libres.
   e) critical: Define una sección de código que debe ser ejecutada por un hilo a la vez.
   f) atomic: Se utiliza para acceder a variables de manera segura, asegurando que solo se acceda o modifique un solo hilo para evitar condiciones de carrera.

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.

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.

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.
   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.
      i. **Al modificar de la misma manera cada una de las variables se puede ver que cada hilo comparte la variable x por lo que su valor se mantiene en cada hilo, sin embargo y es privado, es decir cada hilo hace su copia de y, por lo que al finalizar el ciclo, se destruyen estos hilos y se regresa al hilo original donde y sigue siendo 0.**

5. **(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.

6. REFLEXIÓN DE LABORATORIO: se habilitará en una actividad independiente.