Universidad del Valle de Guatemala
Facultad de Ingeniería
Departamento de Ciencias de la Computación
CC3086 Programación de microprocesadores

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: es la forma que ponemos las variables privadas para cada hilo, haciendo que los cambios de la variable en un hilo no afecta la misma variable en el resto de hilos.
   b) shared: es la forma que una variable se comparte entre los hilos, encontrá de private, los cambios de la variable en un hilo se notan en todos los hilos.
   c) firstprivate: es un private, pero el valor que se utiliza es uno afuera del paralelo.
   d) barrier: hace que todos los hilos se sincronizan hasta que lleguen a un punto especificado o barrera.
   e) critical: es la forma para que una sección ejecuta el código cada un hilo a la vez, pero contra atomic, son operaciones más complejas.
   f) atomic: parecido a critical, ejecuta el código en cada hilo a la vez, pero las operaciones más básicas y simples.

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. Cuando la variable es shared, se puede notar que el valor si varia, por hilo, pero no es uno que se mantiene en el hilo, mientras que private, la variable dos es única por hilo.

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.

   El valor key que aparece al terminar el proceso es 8 secuencialmente.