

Práctica 2. Introducción a OpenMP.

Objetivos de la práctica:

- Adquirir la capacidad de programar en lenguaje C con la librería OpenMP.
- Adquirir la capacidad de resolución de problemas paralelos sencillos.
- Comprender el uso y funcionamiento de las directivas de la librería OpenMP.
- Comprender cómo funciona un programa multihilo.
- Adquirir la capacidad de paralelizar un programa.

1. Introducción

Como hemos visto en clase, para poder paralelizar un programa con la librería OpenMP añadiremos la librería en el código fuente:

```
#include<omp.h>
```

Para compilar un programa que contenga directivas de OpenMP el comando es:

```
gcc -fopenmp -o salida fuente.c
```

Y luego se ejecuta normalmente:

```
./salida
```

De las directivas vistas en clase, en esta práctica vamos a trabajar con la directiva **sections** la cual ejecuta cada sección indicada con la directiva **section** con un hilo distinto. Hay dos opciones para usarla:

- Lanzar la directiva **parallel**, dentro la directiva **sections** y especificar dentro cada sección con la directiva **section**.

- Ejemplo:

```
#pragma omp parallel
{
    #pragma omp sections
    {
        #pragma omp section
        {
        }
        #pragma omp section
        {
        }
    }
}
```

- Englobar la directiva `parallel` y la directiva `sections` con la directiva `parallel sections`:

- Ejemplo:

```
#pragma omp parallel sections
{
    #pragma omp section
    {
    }
    #pragma omp section
    {
    }
}
```

2. Ejercicios propuestos

2.1. Ejercicio 1

- Realice un programa `secciones.c` en lenguaje C que tenga dos secciones que se ejecuten de forma paralela. Primero estas secciones paralelas inicializarán una matriz de tamaño $n \times n$ cada una (como resultado se tendrán dos matrices, A y B, inicializadas). Una vez inicializadas, habrá dos secciones que se ejecuten de forma paralela: en la primera sección se calculará el producto de las dos matrices $A \times B$ guardando el resultado en una matriz C y en la segunda se calculará la suma de las dos matrices $A+B$, guardando el resultado en una matriz D. Mide tiempos para distintos tamaños de las matrices A y B, ¿qué observas? NOTA: las matrices son cuadradas y están inicializadas con números aleatorios entre 1 y 10.

2.2. Ejercicio 2

- Modifique el programa anterior de forma que ahora se usen punteros de tamaño $n \times n$ para las matrices. ¿Qué ocurre con los tiempos? ¿A igualdad de tamaño qué versión es más rápida? ¿Qué ocurre con los tamaños de las matrices? ¿Admiten un mayor tamaño?