



## **Práctica**

### **Introducción a Open Multiprocessing (Open MP)**

#### **Parte 1**

#### **Objetivos:**

El estudiante se introducirá en una de las plataformas concurrentes para programación multihilo con el fin tener una herramienta que le permita paralelizar algunos programas en arquitecturas de computadora con memoria compartida.

**Al final de la práctica** el estudiante comprenderá el uso de las directivas `parallel` y `for` utilizadas en OpenMP.

#### **Previo**

Investigar lo siguiente

¿Qué es un hilo y un proceso?

¿Cuál es la diferencia entre hilos y procesos?

¿Cuál es la diferencia entre concurrencia y paralelismo?

¿A qué se refiere que una computadora tenga un procesador con arquitectura de memoria compartida?

#### **Antecedentes**

Para la realización de esta práctica se requiere que el estudiante tenga conocimientos sólidos en programación básica en el lenguaje ANSI C y haya cursado las asignaturas de sistemas operativos y arquitectura de computadoras.

#### **Introducción**

En ésta práctica se iniciará con una introducción a una de las plataformas concurrentes para programación multihilo que es OpenMP.

OpenMp no es un lenguaje de programación, se puede decir que es una extensión a algunos lenguajes como lo es el lenguaje C y Fortran. OpenMP es un conjunto de directivas del compilador, funciones que están dentro de una biblioteca y variables de ambiente; en su conjunto es un API que permite a los desarrolladores agregar directivas, usar funciones y modificar variables, para que se pueda producir una versión paralela de un programa sin tener que reconstruir la versión serial.

OpenMP se utiliza principalmente para programar en computadoras que tienen multiprocesadores que tienen memoria compartida y sistemas multicore.

En OpenMP la idea es tener un número de hilos y dividir el trabajo entre ellos, no es necesario crear los hilos ya que utiliza una arquitectura llamada fork/join (o maestro esclavo), donde en alguna parte del código se pueden crear las llamadas regiones paralelas, ahí existen  $n$  hilos y entre ellos se puede dividir el trabajo o el cómputo ( Figura 1). Dentro de la región paralela se pueden usar diferentes directivas o constructores con sus clausulas (si es que las tienen) que permiten plantear la división del trabajo entre los hilos de forma adecuada.

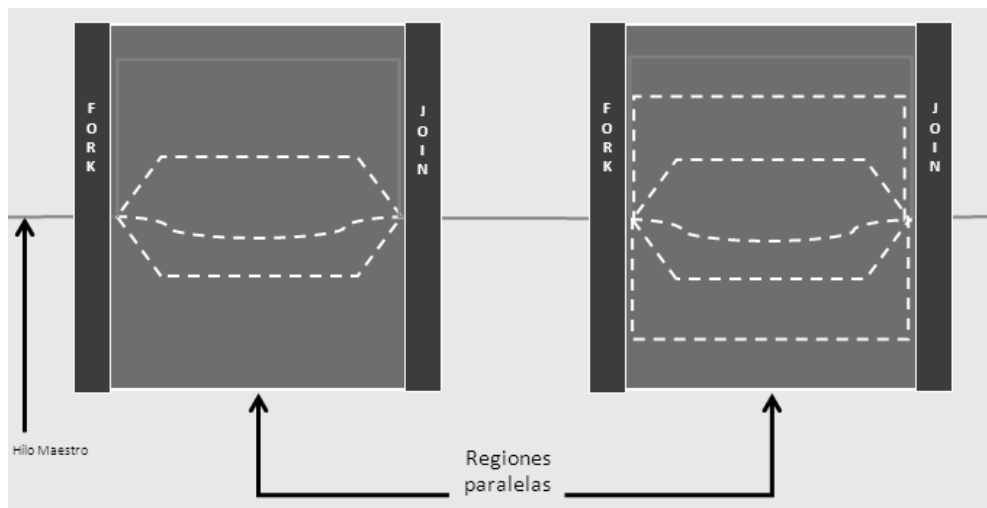


Figura 1

Lo anterior permite crear programas donde se tendrán segmentos de código que se podrán ejecutar en paralelo y otros en forma serial.

### Desarrollo

Lo primero que se hará es crear una región paralela, para ello se necesita saber la sintaxis o formato para agregar directivas del compilador y, es la siguiente:

```
#pragma omp Nombre_directivaConstructor [clausula, ...]
```

Nota: Es posible tener más de una clausula.

Proyecto PAPIME PE104911

Pertinencia de la enseñanza del cómputo paralelo en el currículo de las ingenierías

Práctica - **Introducción a Open Multiprocessing (Open MP)**

Facultad de Ingeniería – UNAM

**Actividad 1:** Compilar el siguiente código (código1) al que se llamará *versión serial* y después agregar la directiva introducir las primeras cuatro líneas del main en una región paralela. Para agregar una región paralela se utiliza el constructor **paralell** de la siguiente manera:

```
#pragma omp parallel
{
    /*bloque de código dentro de la región paralela*/
}

/*código 1*/
#include <stdio.h>
int main() {
    int i;
    printf("Hola Mundo\n");
    for(i=0;i<6;i++)
        printf("Iter:%d\n",i);
    printf("Adios \n");
}
```

Una vez que se ejecute el programa. ¿Qué salida se obtiene? y ¿ Que mensajes de advertencia se mostraron? \_\_\_\_\_

\_\_\_\_\_

Para que se agreguen las directivas del compilador, se debe utilizar una bandera que lo indique, en este caso es `-openmp` o `/Qopenmp` para Linux y Windows.

**Actividad 2:** Agregar la bandera para poder hacer uso de las directivas, compilar y ejecutar nuevamente. ¿Qué se observa a la salida? ¿Qué sucede al agregar la región paralela?

\_\_\_\_\_

\_\_\_\_\_

**Actividad 3:** Una forma de cambiar el número de hilos de la región paralela es modificando la variable de ambiente `OMP_NUM_THREADS`. Cambiar la variable de ambiente, compilar y ejecutar el programa nuevamente. ¿Qué sucede?

\_\_\_\_\_

\_\_\_\_\_

Proyecto PAPIME PE104911

Pertinencia de la enseñanza del cómputo paralelo en el currículo de las ingenierías

Práctica - **Introducción a Open Multiprocessing (Open MP)**

Facultad de Ingeniería – UNAM

Ahora que se sabe cómo crear una región paralela y que pasa al tener una en el código de un programa, se requiere conocer otros constructores y cláusulas que permitan paralelizar ese código. Un constructor que permite dividir el trabajo entre el número de hilos es la directiva **for** que permite dividir las iteraciones de una estructura de repetición **for** entre el número de hilos de la región paralela, un ejemplo se muestra en el código 2

*/\*código 2\*/*

**#pragma omp parallel**

**#pragma omp for**

```
for(i=0;i<100;i++) {  
    Realizar Trabajo();  
}
```

**Actividad 4:** Al código que se compiló en la actividad 2, agregar el constructor **for** antes del ciclo **for**. Compilar y ejecutar el programa. ¿Qué se observa?

---

## Referencias

**Multicore Programming for Academia, Intel Software College, 2007**