

2. PROGRAMACIÓN CON OPENMP

Ejercicio 1:

En el primer ejercicio, hemos implementado un programa que imprime “hola mundos” desde múltiples hilos en paralelo, indicando el número de cada hilo. A partir de la salida, podemos concluir que se han utilizado 16 hilos, y que cada uno se ejecuta en paralelo sin seguir un orden específico. Esto podría parecer un error si estuviéramos trabajando con programación secuencial, donde se espera un orden predefinido. Sin embargo, en programación paralela, los hilos no siguen un orden fijo; la ejecución se distribuye entre los hilos y el más rápido en completar su tarea imprime su mensaje primero. Por lo tanto, el comportamiento observado es normal y esperado en un entorno de programación paralela.

```
root@566c8d7af7c1:/workdir# ./ejec_ejercicio1
Hola desde el hilo 1 de 16
Hola desde el hilo 2 de 16
Hola desde el hilo 8 de 16
Hola desde el hilo 6 de 16
Hola desde el hilo 9 de 16
Hola desde el hilo 15 de 16
Hola desde el hilo 5 de 16
Hola desde el hilo 3 de 16
Hola desde el hilo 14 de 16
Hola desde el hilo 4 de 16
Hola desde el hilo 11 de 16
Hola desde el hilo 0 de 16
Hola desde el hilo 12 de 16
Hola desde el hilo 10 de 16
Hola desde el hilo 13 de 16
Hola desde el hilo 7 de 16
```

Ejercicio 2:

En este ejercicio, se usa OpenMP para hacer más rápida la suma de los elementos de una lista. Primero, definimos un número $N=1000$ y un arreglo A (lista) de tipo double (números decimales con 2 veces más decimales que los float normales), inicializado con valores de 0.00 a N

Luego, usamos OpenMP con la cláusula `reduction(+:suma)`, que permite paralelizar el cálculo, consiguiendo que varios hilos sumen partes del arreglo al mismo tiempo y luego combinen los resultados sin errores. Así, el programa aprovecha mejor los recursos del procesador.

Al final, el programa muestra en pantalla la suma total. Aunque el beneficio en rendimiento es limitado para un tamaño pequeño de datos, el uso de OpenMP es crucial en tareas con mayor carga computacional, donde la paralelización mejora significativamente la eficiencia.

```
root@566c8d7af7c1:/workdir# ./ejec_ejercicio2  
Suma total: 499500.000000
```