María Camila Caicedo – Informe programación paralela

1. Productor–Consumidor:

Se desarrolló un buffer circular compartido en el que el productor inserta datos y el consumidor los retira. Se empleó mutex y variables de condición para evitar condiciones de carrera. Permite comprender los mecanismos de sincronización en memoria compartida.

Imagen que contiene Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

1. Multiplicación Matriz–Vector:

Cada hilo procesa un bloque de filas de la matriz, multiplicándolas por el vector x. Las filas se dividen equitativamente y cada hilo actualiza las posiciones correspondientes en el vector y. Asegura un reparto equilibrado de la carga.

Texto

El contenido generado por IA puede ser incorrecto.

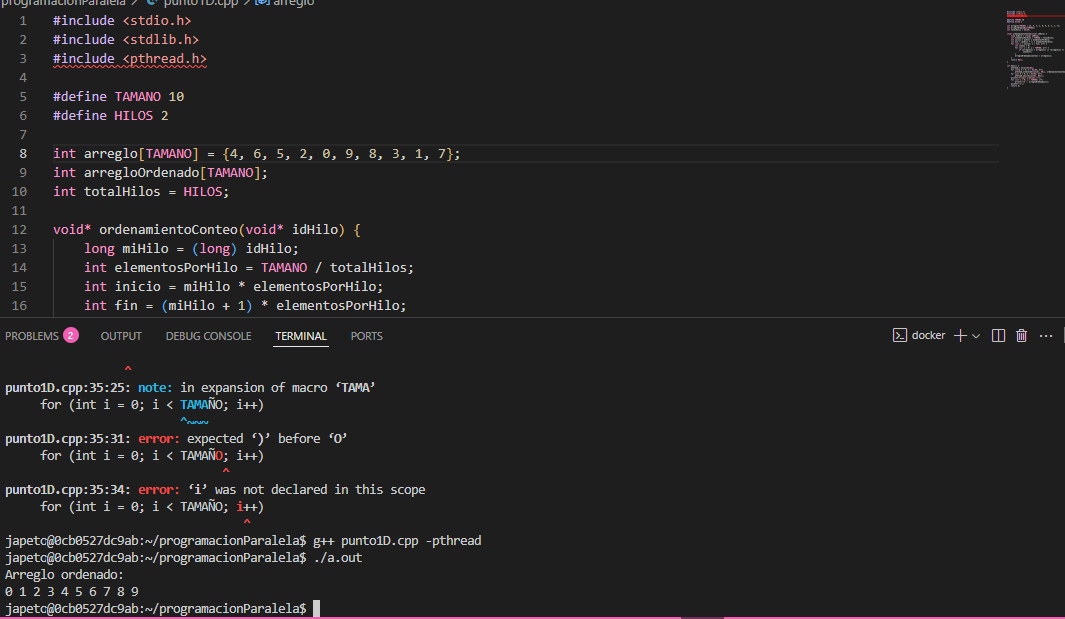
1. Regla del Trapecio:

Se aproximó la integral definida en [a, b] con n trapecios. Cada hilo calcula su intervalo local y se utilizó un mutex para acumular la suma parcial en la variable integral. Ejemplifica un proceso de reducción con exclusión mutua.



1. Count Sort Paralelo:

Se implementó una variante paralela del algoritmo Count Sort, en la que cada hilo calcula posiciones de un subarreglo y escribe directamente en el arreglo resultado. Muestra cómo algoritmos simples pueden paralelizarse sin conflictos.



Ejercicio 2: Suma de un Arreglo Grande

a: Se utilizó Pthreads con exclusión mutua. Cada hilo calculó una suma parcial y se sincronizó con pthread\_mutex\_t.

b: Se aplicó OpenMP con la directiva #pragma omp parallel for reduction(+:sum), lo que permitió paralelizar el cálculo de manera sencilla. Los tiempos obtenidos evidencian que OpenMP fue más eficiente que Pthreads, gracias a la optimización implícita de la reducción.

Texto

El contenido generado por IA puede ser incorrecto.

Ejercicio 3: Multiplicación de Matrices

Se implementó la operación **C = A × B** utilizando **OpenMP**, donde cada hilo procesó un bloque de filas de la matriz **A**.

* Se generaron matrices de prueba con valores aleatorios.
* Gracias a la paralelización con directivas de OpenMP, cada hilo escribió únicamente en la sección de la matriz **C** que le correspondía, evitando cualquier conflicto de concurrencia.

Texto

El contenido generado por IA puede ser incorrecto.