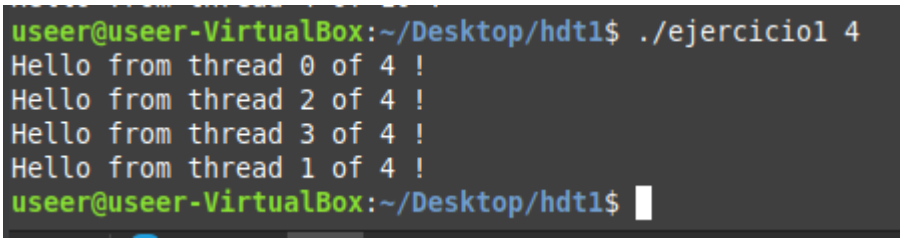
Carol Arevalo
20461

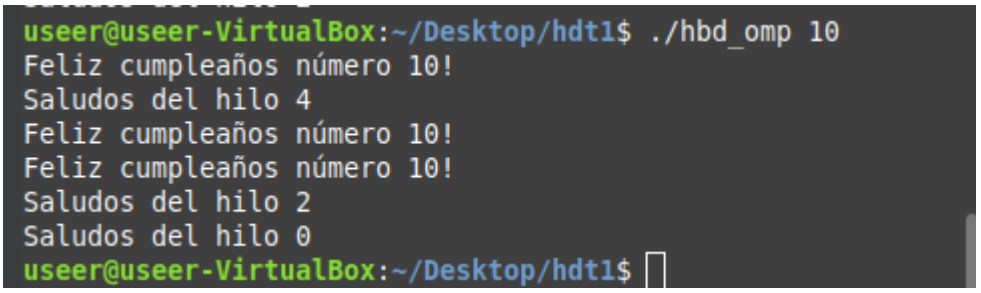
1. Ejercicio 1



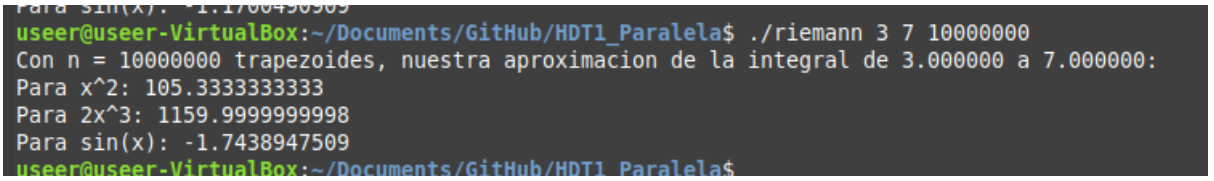
¿Por qué al ejecutar su código los mensajes no están desplegados en orden?

Cuando se ejecutan threads en paralelo, no hay garantía de que se ejecuten en un orden específico. Los threads pueden competir por recursos de la CPU y ejecutarse en momentos diferentes.

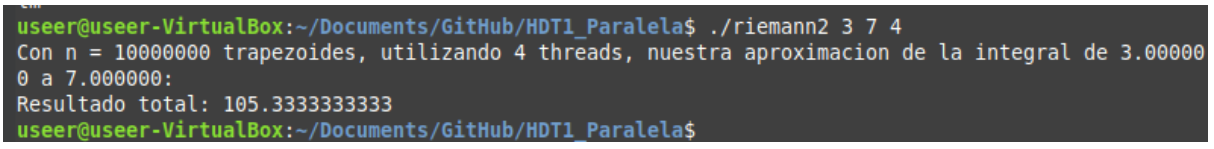
2. Ejercicio 2



3. Ejercicio 3



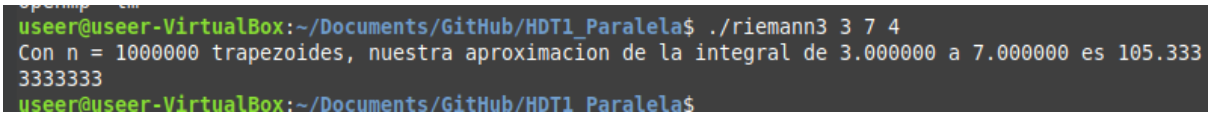
4. Ejercicio 4



¿Por qué es necesario el uso de la directiva #pragma omp critical?

Al usar #pragma omp critical alrededor de la operación de suma en la sección paralela, se asegura que solo un thread pueda ejecutar esa operación a la vez. Esto garantiza que los cálculos locales de cada thread se sumen de manera ordenada y segura a la variable total_sum, evitando posibles race conditions y que asi se produzca el resultado correcto.

5. Ejercicio 5



¿Qué diferencia hay entre usar una variable global para añadir los resultados a un arreglo?

Cuando se utiliza una variable global se necesita sincronizar adecuadamente el acceso de los hilos a esa variable, mientras que al almacenar los resultados locales en un arreglo evita la necesidad de esa sincronización, pues cada hilo tiene su propio espacio en el arreglo para almacenar el resultado local. Esto elimina problemas con race conditions.