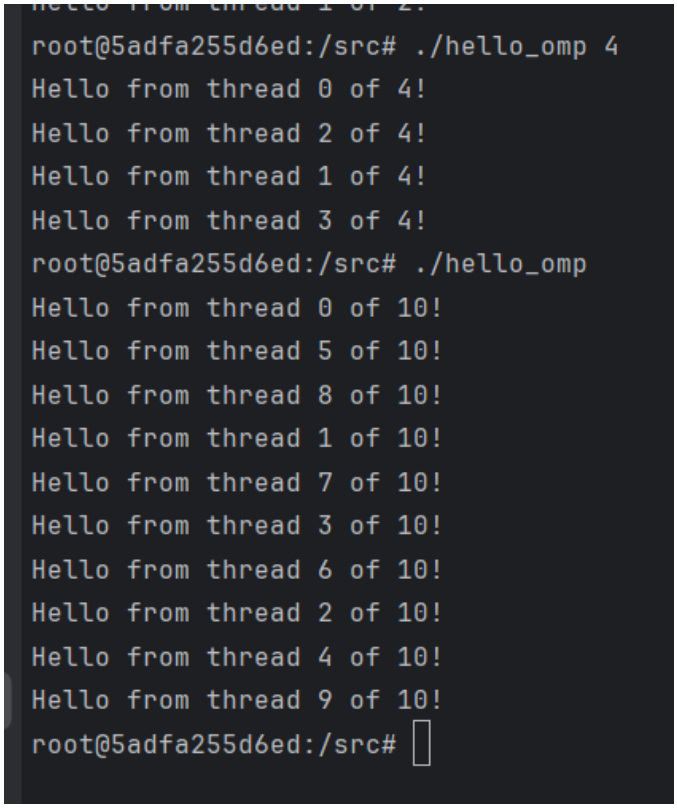
HOJA DE TRABAJO 01

Francisco Castillo 2562

REPOSITORIO

Los archivos del código se encuentran en el siguiente repositorio.

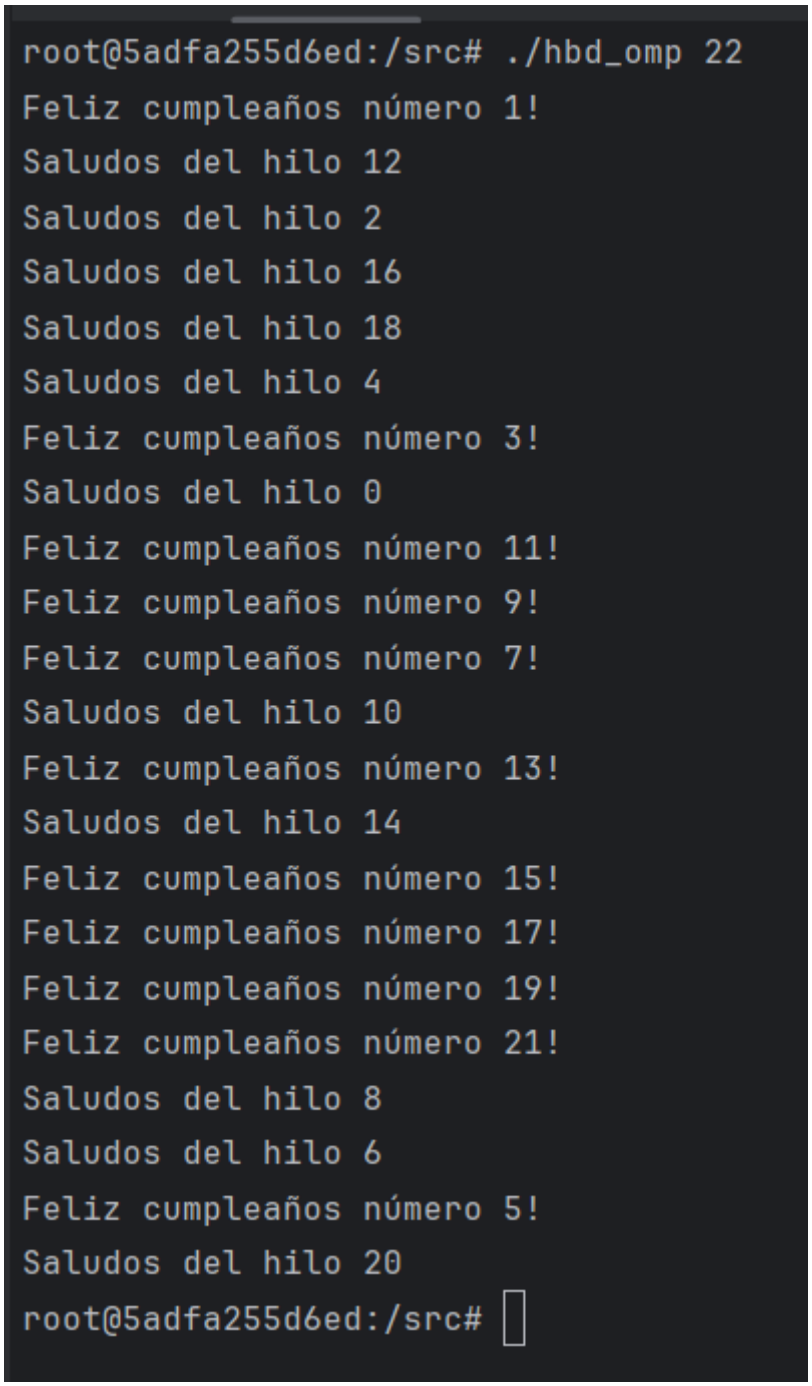
EJERCICIO 1



¿POR QUÉ, AL EJECUTAR SU CÓDIGO, LOS MENSAJES NO ESTÁN DESPLEGADOS EN ORDEN?

Esto es debido a que su creación y su proceso de ejecución se hace de manera paralela, por lo que es imposible predecir el orden.

EJERCICIO 2


root@5adfa255d6ed:/src# ./hbd_omp 22
Feliz cumpleaños número 1!
Saludos del hilo 12
Saludos del hilo 2
Saludos del hilo 16
Saludos del hilo 18
Saludos del hilo 4
Feliz cumpleaños número 3!
Saludos del hilo 0
Feliz cumpleaños número 11!
Feliz cumpleaños número 9!
Feliz cumpleaños número 7!
Saludos del hilo 10
Feliz cumpleaños número 13!
Saludos del hilo 14
Feliz cumpleaños número 15!
Feliz cumpleaños número 17!
Feliz cumpleaños número 19!
Feliz cumpleaños número 21!
Saludos del hilo 8
Saludos del hilo 6
Feliz cumpleaños número 5!
Saludos del hilo 20
root@5adfa255d6ed:/src#

EJERCICIO 3

```
root@2eda6f324d3d:/src# ./riemann_x2 2 10
n = 1.000000e+07
f(x) = x^2
[a, b] = [2, 10]
Approximation = 330.666628
```

```
root@2eda6f324d3d:/src# ./riemann_2x3 3 7
n = 1.000000e+07
f(x) = 2x^3
[a, b] = [3, 7]
Approximation = 1159.999874
root@2eda6f324d3d:/src#
```

```
root@2eda6f324d3d:/src# ./riemann_sin 0 1
n = 1.000000e+07
f(x) = sin(x)
[a, b] = [0, 1]
Approximation = 0.459698
root@2eda6f324d3d:/src#
```

EJERCICIO 4

```
root@2eda6f324d3d:/src# ./riemann_omp2_x2 2 10 2
Thread 0: [a, b] = [2.000000, 6.000000]
Thread 1: [a, b] = [6.000000, 10.000000]
n = 1.000000e+07
f(x) = x^2
[a, b] = [2, 10]
Approximation = 330.666628
root@2eda6f324d3d:/src#
```

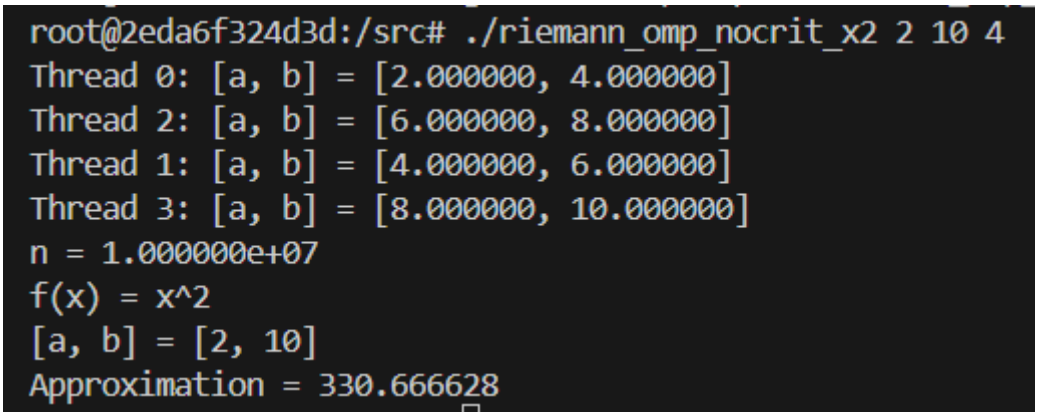
```
root@2eda6f324d3d:/src# ./riemann_omp2_2x3 3 7 4
Thread 0: [a, b] = [3.000000, 4.000000]
Thread 1: [a, b] = [4.000000, 5.000000]
Thread 2: [a, b] = [5.000000, 6.000000]
Thread 3: [a, b] = [6.000000, 7.000000]
n = 1.000000e+07
f(x) = 2x^3
[a, b] = [3, 7]
Approximation = 1159.999874
```

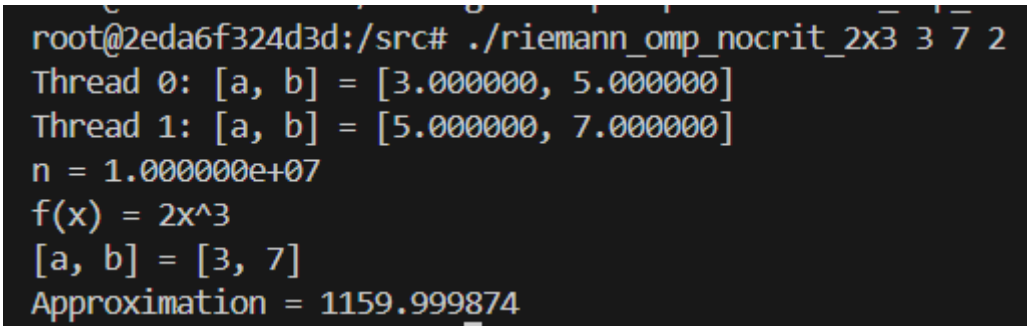
```
root@2eda6f324d3d:/src# ./riemann_omp2_sin 0 1 1
Thread 0: [a, b] = [0.000000, 1.000000]
n = 1.000000e+07
f(x) = sin(x)
[a, b] = [0, 1]
Approximation = 0.459698
```

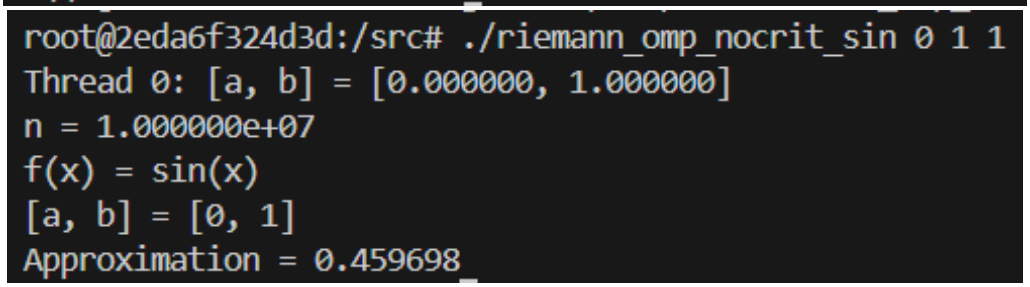
¿POR QUÉ ES NECESARIO EL USO DE LA DIRECTIVA #PRAGMA OMP CRITICAL?

Esto se debe que, para realizar la suma es necesario leer el valor de la variable previo a la operación. Paralelizar el proceso de adición crea la situación de *rrace conditions*, por lo que, si no se coloca, provocaría valores erróneos.

EJERCICIO 5







¿QUÉ DIFERENCIA HAY ENTRE USAR UNA VARIABLE GLOBAL PARA AÑADIR LOS RESULTADOS A UN ARREGLO?

En ambos casos se evitan las *race* conditions. Pero, el utilizar un arreglo permite que cada hilo escriba su resultado en una posición única del arreglo, lo que elimina la necesidad de sincronización explícita. En cuanto a la sección crítica, todos los hilos acceden al mismo espacio de memoria, este método de sincronización asegura que solo un hilo pueda ejecutar el código a la vez; esto introduce un posible *bottleneck*.