

Programación Concurrente y de Tiempo Real

Grado en Ingeniería Informática

Asignación de Prácticas Número 12

Se le plantean a continuación algunos ejercicios sencillos de programación concurrente con el lenguaje C++23. Debe desarrollar los programas que se piden. En esta ocasión no es obligatorio documentar el código, pero merece la pena aprender a hacerlo utilizando un enfoque parecido al ya conocido para documentar código Java, y Doxygen.

1. Ejercicios

1. Se desea conocer el valor de

$$\int_0^1 \frac{x^3}{(x^4 + 2)} dx$$

utilizando el conocido método estocástico de Monte-Carlo, mediante un programa escrito en C++ que utilice paralelismo mediante división de la carga de trabajo entre varias hebras paralelas. Para ello:

- escriba el programa en un fichero llamado `parIntegral.cpp`.
- se prohíbe el uso de funciones λ ; el motor de ejecución de las hebras residirá en una función con la siguiente signatura:

```
void monteCarloIntegration(int numPoints)
```

- Cada hebra dispondrá de un acumulador independiente de puntos válidos en una variable `hits`, que se sumará a un acumulador global en una variable `totalHits`; esta última estará bajo control de exclusión mutua utilizando un cerrojo de clase `mutex`.
- A continuación, escriba una función principal (`main`) que lea desde teclado el número de puntos a lanzar y el número de hebras paralelas; tras ello, creará las hebras, lanzará su ejecución y esperará a que terminen. Finalmente, efectuará los cálculos necesarios para obtener la integral buscada y ofrecerá en pantalla ese valor. La interacción input-output del programa con el usuario se describe a continuación, donde `xxxxxxx` indica el número de puntos a lanzar, `yy` el número de hebras paralelas y `zzzzzz` el valor de la integral.

Ingrese el numero de puntos a lanzar: xxxxxxxx
Ingrese el numero de tareas paralelas: yy
Aproximacion: zzzzzz

2. Aunque el lenguaje C++ dispone de barreras desde la iteración de 2020, en esta ocasión deseamos simularlas mediante un monitor escrito específicamente para ello con este lenguaje. Desarrolle el código necesario para lograrlo según las siguientes especificaciones:

- se prohíbe el uso de funciones λ .
- el código residirá en un fichero `simBarrier.cpp`, que ofrecerá una clase `simBarrier` compuesta por los métodos siguientes:

```
void toWaitOnBarrier();  
void resetBarrier()
```

- El primer método bloquea a una hebra en la barrera hasta que todas las citadas en ella llegan; el segundo método permite reiniciar la barrera, y el número de hebras que permiten abrir la barrera se fija a través del constructor de clase.
- A continuación, escriba una función principal (`main`) que instancie seis hebras como objetos de la clase `thread` cuyo motor de ejecución contendrá **única y exclusivamente**, el siguiente código:

```
printf("%d llegando a barrera...\n", this_thread::get_id());  
bar.toWaitOnBarrier();  
printf("%d saliendo de barrera...\n", this_thread::get_id());
```

- las salidas por consola asociadas a la función `printf` del motor de ejecución anterior deberán estar protegidas bajo control de exclusión mutua; es libre de utilizar la técnica de control que considere oportuna para ello.
- El programa principal lanzará un grupo de tres hebras, que se sincronizarán en el objeto `barrera` de clase `simBarrier`. A continuación, el programa principal reiniciará la barrera, y lanzará el segundo grupo de tres hebras. El output impreso deberá ser (el orden de las impresiones de las tareas de un grupo puede variar, pero el orden de los grupos no) el siguiente, donde los números que identifican a cada hebra pueden variar en su máquina.

```
main creando barrera para tres hebras...  
1878738688 llegando a barrera...  
1870345984 llegando a barrera...  
1727735552 llegando a barrera...  
1727735552 saliendo de barrera...  
1878738688 saliendo de barrera...  
1870345984 saliendo de barrera...  
main reseteando barrera para tres nuevas hebras...  
1727735552 llegando a barrera...  
1870345984 llegando a barrera...
```

1878738688 llegando a barrera...
1878738688 saliendo de barrera...
1727735552 saliendo de barrera...
1870345984 saliendo de barrera...