

Concurrencia y Paralelismo

Grado en Informática 2023

Práctica 1 – Array de Enteros

Implemente un sistema en el que varios threads comparten un array que empieza con sus posiciones a 0. Cada thread debe realizar un número de iteraciones, donde en cada iteración incrementa una posición aleatoria del array. Si el programa funciona correctamente, la suma de los valores guardados en el array debería ser el número de threads multiplicado por el número de iteraciones.

El código proporcionado implementa la versión secuencial de este problema, y acepta las siguientes opciones:

- `-i n`, para controlar el número de iteraciones que cada thread va a hacer.
- `-t n`, para especificar el número de threads que vamos a crear.
- `-s n`, para controlar el tamaño del array.
- `-d n`, para cambiar el tiempo de espera entre operaciones.

Partiendo de este código se pide:

Ejercicio 1 (Crear los threads y proteger el array)

El programa debe crear el número de threads indicados en las opciones, y proteger el acceso concurrente al array mediante un mutex. El mutex debe pasarse en los parámetros de cada thread, y no debería ser global.

Cada thread debe estar identificado por un número correlativo desde 0 hasta el número de threads menos 1, que imprimirá en sus mensajes por pantalla.

Ejercicio 2 (Mutex por posición) Cada posición del array se puede incrementar de manera independiente. Modifique la implementación para que haya un mutex distinto para cada posición del array.

Ejercicio 3 (Añadir thread que muevan valores entre posiciones)

Añada otros n threads que seleccionen aleatoriamente dos posiciones del array, y resten uno de la primera posición y sumen uno en la segunda posición. Estos threads también deberían hacer el número especificado de iteraciones.

En ningún momento debería ser posible que otro thread viera el valor en tránsito, es decir, si antes de empezar el movimiento la suma del array es n , en todo momento la suma tiene que ser $\geq n$. Puede ser mayor si otro thread está incrementando otra variable al mismo tiempo.

Ejercicio 4 (Iteraciones) Ahora mismo cada thread realiza un número de operaciones especificado con la opción `-i`. Cambie el comportamiento para que ese número de iteraciones sea el realizado entre todos los threads de cada tipo.

Es decir, si se especifica que hay que hacer 100 iteraciones, deberían hacerse 100 incrementos y 100 movimientos entre todos los threads. Si el programa funciona correctamente, la suma de todo el array debería ser igual al número de iteraciones.

La cuenta de las iteraciones debería hacerse mediante un contador compartido.

Entrega

La fecha límite de entrega es el 19 de febrero. El código inicial está disponible a través de github classroom en <https://classroom.github.com/a/oxRzrn1i>. En el acceso inicial se pedirá la creación del grupo de prácticas. Usad como nombre los logins de las personas que componen el grupo separados por un guión (-). Una vez creado el repositorio, quien lo ha hecho puede añadir a la segunda persona del grupo.

La corrección se hará sobre el contenido del repositorio al final del día 19 de febrero.