

# Concurrencia y Paralelismo

Grado en Informática 2024

## Práctica 1 – Contadores

Vamos a implementar un sistema donde varios thread comparten dos contadores: `increment`, que empieza en 0, y `decrement`, que empieza en un `total`. Cada thread itera un número de veces restando 1 a `decrement` y sumando 1 a `increment`. En cada paso la suma de las dos variables debería ser `total`.

El programa 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 para el ejercicio 3

El código proporcionado parsea las opciones e inicia los threads. Al probarlo se puede comprobar que no protege las variables correctamente, por lo que al final de la ejecución la suma de `increment` y `decrement` no es igual al total.

Partiendo de este código se pide:

### Ejercicio 1 (Proteger los contadores)

Proteja los contadores usando un `mútex`. No use variables globales.

**Ejercicio 2 (Iteraciones)** Cambie el comportamiento de las iteraciones para `iterations` especifique el número total de iteraciones que se hacen entre todos los threads. Es decir, si el número de iteraciones es 100, deberían hacerse 100 incrementos en `increment` y 100 decrementos en `decrements`

Pista: Use un contador compartido para las iteraciones.

### Ejercicio 3 (Cambie los contadores a arrays)

Cambie los contadores de un valor único a un array. Todos los contadores del array de decrementos deberían inicializarse a `total`. Cada thread selecciona aleatoriamente en cada iteración una posición para decrementar y una para incrementar.

Para incrementar la concurrencia, bloquee cada posición en los arrays de forma independiente. Esto es, debería poderse acceder a `increment[0]` al mismo tiempo que a `increment[1]`. Si se ha implementado correctamente, la suma de los valores de los dos arrays debería ser `total * tamaño array`.

**Ejercicio 4 (Desplazamiento del incremento)** Cree un nuevo tipo de thread que escoja dos posiciones aleatoriamente en el array de incrementos, decrementa una posición e incrementa la otra, dejando el total del array sin cambios. Asegúrese de hacerlo de forma que cualquier otro thread que observe el array de incrementos y decrementos vea siempre la suma como total.

Utilice el número de threads especificado en las opciones como el número de threads de este nuevo tipo a crear. Las estructuras con los argumentos para estos nuevos threads deberían estar en el stack del thread inicial.

**Ejercicio 5 (Desplazamiento del decremento)** Haga lo mismo que en el ejercicio anterior, pero para el array de decremento. En este ejercicio las estructuras con los argumentos para los nuevos threads deben estar en el heap.

## Entrega

La fecha límite de entrega es el 18 de febrero. El código inicial está disponible a través de github classroom en <https://classroom.github.com/a/W8HddegP>. 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 18 de febrero.