

Práctica: Sincronización

José Luis Quiroz Fabián

Marzo 2019

1 Introducción

Como hemos estudiado, la concurrencia en muchos casos puede mejorar la funcionalidad de nuestras aplicaciones ya que nos permite realizar varias tareas de forma simultánea. Ejemplos de lo anterior es cuando utilizamos varias pestañas en nuestro navegador web, o bien cuando nuestras aplicaciones de oficina revisan al momento el texto que estamos escribiendo. Sin embargo, el precio a pagar al desarrollar aplicaciones concurrentes es sin duda la complejidad subyacente en las mismas a fin de garantizar un resultado esperado o determinista. No obstante, se han implementado diferentes mecanismos de sincronización a fin de garantizar un resultado "correcto" en la ejecución de nuestras aplicaciones. En esta práctica estudiamos los candados y los semáforos.

2 Ejemplos

2.1 Candados `ReentrantLock`: `lock-unlock`

1. REVISAR EJEMPLO PAQUETE `uam.candados.ejemplo1`
2. REVISAR EJEMPLO PAQUETE `uam.candados.ejemplo2`
3. REVISAR EJEMPLO PAQUETE `uam.candados.ejemplo3`

2.2 Semáforos

1. REVISAR EJEMPLO PAQUETE `uam.semaforos.ejemplo1`

3 Ejercicios

- Considerando el ejemplo que soluciona de forma secuencial el problema de las N-Reinas, realizar una solución concurrente donde cada hilo encuentre un conjunto de soluciones (número de soluciones parciales) y las almacene en una variable compartida de tipo *int* (no un arreglo). Al final el hilo principal muestra el valor de la variable compartida.

- Modificar el problema del productor-consumidor para aceptar N productores y M consumidores y un almacén de tamaño $K > 2$. Encerrar a cada productor y consumidor en un ciclo para que produzca muchas veces o consuma muchas veces según sea su rol.