

Prácticas Concurrencia y Distribución (16/17)

Arno Formella, Francisco Rodríguez Martínez,
Javier Rodeiro Iglesias, David Olivieri

semana 17–21 abril

Seguimos con la actividad de la metáfora de la lectura concurrente de un libro.

14. Seguramente has notado que realizar la tarea de tal manera que un lector despierta con `notify()` exactamente al siguiente lector que debe leer a continuación y que debe estar en su `wait()` provoca una posible condición de carrera, es decir, que se realiza tal `notify()` antes de que el otro hilo haya entrado en su `wait()`.

¿Cómo puedes resolver este problema?

Inténtalo de las siguientes dos maneras:

- Usa el estado del monitor de Java (`getState()`) como condición para estar seguro que el hilo ya haya entrado en espera antes de efectuar el `notify()` correspondiente. (Este método funcionará pero el manual de Java dice que no se debe usar este método para el fin de sincronizar hilos.)
 - Implementa tu propia condición que puedes chequear, es decir, el hilo que está a punto de ejecutar su `wait()` pone la condición de forma adecuada, el hilo que ejecuta el `notify()` espera de forma adecuada hasta que se haya cumplido la condición. Observa que la condición debe estar implementada en el mismo objeto que usas para tu `notify/wait`-pareja para que esté protegida por los bloques sincronizados correspondientes. (Observerás que necesitas un bucle de espera activa del hilo que debe notificar.)
15. ¿Funcionan las versiones de tu programa también para el caso de un solo hilo? (Observa que en los requisitos este caso nunca fue explícitamente excluido). ¿Qué solución propones?