

# Prácticas Concurrencia y Distribución (16/17)

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

semana 3–7 abril

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

Con los experimentos (y el gráfico) de la semana anterior se ha observado que la lectura de un libro con cierto número de páginas realizada con un número creciente de lectores cada vez consume más tiempo si se usa tanto el método `notify()` como el método `notifyAll()`.

Obviamente este comportamiento se observa porque se despiertan muchos lectores a los cuales todavía no les toca su turno. Debe ser posible modificar el programa de tal manera que se despierta con un solo `notify()` exactamente el lector correspondiente a la página por leer. De esta manera el tiempo de ejecución del programa debe ser más o menos constante independientemente del número de lectores participantes.

Entonces:

10. Modifica tu programa para que ya no contenga ningún `sleep` y ninguna salida a la consola (menos las dos de comienzo y terminación del hilo principal), mejor dependiendo de una variable booleana para que se pueda activar/desactivar este comportamiento mediante parámetro por línea de comando.
11. Genera de nuevo el gráfico del tiempo de ejecución manteniendo el número de páginas muy alto (por ejemplo 500.000) y aumentado el número de lectores hasta 1000 (por ejemplo usando la secuencia 3 5 10 30 50 100 300 500 1000).
12. Modifica el programa para conseguir el objetivo que un `notify()` despierta exactamente el siguiente hilo por actuar. Activa las salidas del programa y verifica que sigue la lectura correctamente según los requisitos establecidos.
13. Suprime de nuevo las salidas y esperas y genera el gráfico del tiempo de ejecución manteniendo el número de páginas muy alto (por ejemplo 500.000) y aumentado el número de lectores hasta 1000 (por ejemplo usando la secuencia 3 5 10 30 50 100 300 500 1000).

¿Consigues la mejora en tiempo de ejecución que se puede esperar?