Prácticas Concurrencia y Distribución (16/17)

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

semana 20–24 marzo

4. Exclusión mutua sobre un objeto totalmente sincronizado

En esta actividad queremos implementar sincronización entre hilos. Usamos la metáfora de una biblioteca (el programa principal) con un libro (el objeto sincronizado) y varios lectores (los hilos participantes).

Considera los siguientes requisitos:

- Se pasa al programa 3 parámetros: número de lectores, número de páginas del libro, tiempo de lectura por página.
- El programa principal crea el libro con p páginas y un tiempo t (milisegundos ya que usaremos `sleep()`) de lectura por página.
- Los hilos lectores tienen un número de identificación, un tiempo de espera entre lecturas, y una referencia al libro por leer en común.
- El programa principal crea todos los lectores, lanza su actividad, y sincroniza con todos los hilos creados (bucle de `joins`).
- El objeto libro es totalmente sincronizado e internamente cuenta las páginas ya leídas. El libro tiene un método que indica, si quedan páginas por leer.
- El método de lectura de una página del libro debe imprimir en pantalla qué hilo está leyendo qué página (por ejemplo: implementa un `Read(int idHilo)`).

Una vez elaborado la estructura principal del programa, considera las tareas siguientes:

1. Implementa en el método `run()` de los hilos lectores un bucle que lee el libro página por página (llamando al método `Read()` de la clase `Book`) y aposteriori espera un tiempo de $t/(2n)$ (donde t es el tiempo de lectura usado en el método `Read()` y n es el número de hilos lectores participantes).

   El bucle termina cuando el libro indica que ya se han leído todas sus páginas.

   Lanza el programa por ejemplo con 10 hilos, 200 páginas y 20 milisegundos de tiempo de lectura por página.

   Una persona novata en el tema argumenta lo siguiente:

   Dado que el primer hilo que se lanza está leyendo la primera página del libro durante bastante tiempo, todos los demás hilos se ponen a la espera (`Read()` sincronizada) de leer la suya.

Cuando le toca al segundo lector, lee otra vez tan lento que el primero ya está de nuevo a la espera de leer su siguiente página. Y así seguidamente con todos los lectores.

Entonces el progama tiene obviamente el siguiente comportamiento: en el orden como el hilo principal finalmente lanza los hilos, éstos leen siempre en el mismo orden cada uno una página. Por eso al final del progama cada hilo habrá leído el mismo número de páginas y se han leído exactamente todas las páginas del libro.

¿Qué opinas sobre este ingenuo? Lo confirmas? Haz las pruebas! Abajo unas pistas como aumentar el programa.

2. Aumenta tu método `run()` de los lectores para que cada uno cuente el número de páginas que ha leído y que imprima este número con el mensaje de terminación.

3. Implementa la espera con el `sleep()` de tal manera que se llama a la función solamente si el tiempo de espera es más grande que zero. Lanza entonces el programa también con un valor zero de tiempo de lectura ($t = 0$).

4. Intercambia en el bucle del método `run()` las líneas tal que primero se espera y luego se lee la página del libro. Observa que pasa al final del programa!