

Práctico de Concurrencia

Ejemplos de Productor/Consumidor

1. Sean las siguientes clases:
 1. Libro (parte del recurso compartido). Contiene los siguientes atributos:
 1. idLibro: número entero. (clave)
 2. Título: String.
 2. Biblioteca (recurso compartido). Implementada en base a una tabla de hashing que contiene muchos Libros. Es responsable de la sincronización.
 3. Socio (Thread o Runnable). Cada socio retira y luego devuelve el libro retirado. Debe simular un comportamiento usando números aleatorios para elegir idLibro y el tiempo de espera para devolver y el tiempo de espera para volver a retirar.
- Poner todo esto en funcionamiento con una biblioteca de 5 libros y 10 usuarios que compiten por el recurso compartido y esperan cuando no está el libro buscado en la biblioteca.
2. Complicar un poco la cosa y agregar la siguiente complejidad:
 1. Crear una clase Donador (Thread o Runnable) que agregar nuevos libros a la biblioteca. La dificultad está en que la biblioteca puede almacenar hasta 20 libros (puede tener más que 20, pero solamente 20 en sus estantes). O sea que cuando el donador quiera agregar un nuevo libro, deberá esperar hasta que alguien retire alguno, si es que no queda lugar en la biblioteca.
 2. Aumentar el número de Socios para generar un escenario estable.

Sincronización más granularizada.

3. La clase Biblioteca tiene dos tipos de objetos a prestar, libros y revistas. Crear dos clases de socios, SocioLibro y SocioRevista. Bloquear en forma selectiva según esta clasificación. O sea, que un socioLibro no interfiera con un socioRevista y viceversa.
4. Repetir el primer ejercicio, pero delegando la responsabilidad de la sincronización a los threads, usando la sentencia synchronized.