

Práctica 3: Semáforos (2)

- 1- Dar una implementación Java para el problema del productor/consumidor con un buffer acotado y múltiples productores y consumidores.
- 2- Dar una implementación de la clase `bufferAcotado` que provea las operaciones de escritura y lectura de forma tal que las mismas sean seguras cuándo se utilizan en un contexto de múltiples productores/consumidores.
- 3- Considere el siguiente problema en donde existen lectores y escritores de una estructura de datos compartida. Los lectores sólo desean leer el recurso compartido mientras que los escritores desean modificarlos. Claramente no hay problemas si dos o más lectores acceden a la estructura simultáneamente. Sin embargo, si un escritor accede de manera simultánea con otro lector o escritor, entonces el resultado es indeterminado (los escritores deben tener acceso exclusivo al recurso).
 - a) Dar una solución al problema utilizando semáforos en la cual se privilegien a los lectores (si un lector está accediendo al recurso, entonces los nuevos lectores tendrán prioridad sobre los escritores, los que deberán esperar hasta que no haya más lectores).
 - b) Dar una implementación *java* de una clase `Buffer` que provea las operaciones para leer y escribir que sean seguras para su uso concurrente.
 - c) Notar que la solución dada en el punto a) introduce el problema de inanición de los escritores. Proponer una solución alternativa (basada en semáforos) que evite este problema.
 - d) Modificar la implementación dada en b) según la solución encontrada en c).
- 4- En una oficina hay un baño unisex con n toilets. El baño puede ser utilizado tanto por hombres como por mujeres, pero no puede ser utilizado por hombres y mujeres al mismo tiempo. Dar una solución usando semáforos. La solución debe ser libre de deadlocks pero no necesariamente libre de inanición.
- 5- En un centro de hemoterapia se reciben donaciones de sangre. Para realizar las extracciones, que toman un tiempo aleatorio, se cuenta con 4 camillas. Las personas que llegan a donar sangre son atendidas en orden de llegada, pero mientras no haya camillas libres esperan en la sala de espera. En la sala de espera hay 10 revistas desactualizadas que las personas leen mientras esperan su turno y un televisor sin audio sintonizado en el canal crónica-tv. Los donantes siempre prefieren leer una revista, pero si no hay ninguna disponible miran el noticiero. Cuando se libera una camilla se llama a la siguiente persona en la sala de espera, quien deja la revista si tenía una. Ni bien una revista es liberada las personas mirando el televisor compiten por tomarla.
 - a. Identificar los "procesos concurrentes" y los recursos compartidos en el escenario presentado.
 - b. Dar una solución utilizando semáforos que modele el comportamiento explicado.
 - c. Implementar la solución dada en Java.

- 6- Un edificio de dos pisos cuenta con un ascensor kosher que puede transportar al máximo N personas. El ascensor continuamente alterna entre planta baja y primer piso. Este espera en un piso como máximo un minuto o hasta llenarse, luego de ello (lo que ocurra primero) cambia de piso. Dar una solución que simule a este problema, teniendo en cuenta que cada persona debe ser un thread (que puede llegar a cualquiera de los pisos) y el mecanismo de sincronización a utilizar debe ser semáforos. El ascensor puede ser modelado con uno o más threads.
- 7- Se debe modelar el comportamiento de un ferry que conecta dos ciudades y que puede transportar al máximo 100 coches. El ferry alterna entre las dos ciudades. El mismo espera en una ciudad hasta cargar al menos 50 coches, luego de lo cual se dirige hacia la ciudad en la costa opuesta. Implementar en Java una clase que simule el comportamiento del ferry, teniendo en cuenta que cada coche, que es modelado como un thread independiente, puede llegar a una de las dos ciudades y deberá esperar hasta tanto pueda abordar el coche. Utilizar semáforos como mecanismo de sincronización.