UNIVERSIDAD DE MÁLAGA
Dpto. Lenguajes y
Ciencias de la Computación

Programación de Sistemas y Concurrencia
Tema 6 – Monitores

Ejercicio 1.- Supón que un Centro Comercial dispone de unos aseos suficientemente grandes para dar servicio a sus **clientes**. Además de los clientes, el Centro Comercial tiene un **Equipo de Limpieza** que periódicamente limpia los aseos. El sistema debe satisfacer las siguientes condiciones de sincronización:

a) Cualquier número de clientes puede estar simultáneamente utilizando los aseos. Se supone que son tan grandes que, cuando un cliente quiere entrar, siempre hay sitio disponible.

b) Mientras el equipo de limpieza trabaja en los aseos, no puede haber ningún cliente dentro.

Realiza dos implementaciones de este sistema: una injusta para el Equipo de Limpieza, que podría no acceder nunca al aseo si siempre hay clientes; otra justa para el Equipo de Limpieza, que hace que, cuando el Equipo de Limpieza está esperando, cualquier cliente que llegue espere también.

Ejercicio 2. Supón que hay un río cerca de la Escuela de Informática que la separa de un centro de ocio para los estudiantes. Hay estudiantes de dos tipos, los que utilizan móviles Android, y los que utilizan IPhones. Para cruzar el río, existe una barca que tiene 4 asientos, y que no se mueve hasta que no está completa (hay exactamente 4 estudiantes subidos en ella). Para garantizar la seguridad de los pasajeros, no se permite que haya un estudiante Android con tres estudiantes IPhones, ni un estudiante IPhone con tres Android. Cualquier otra configuración de estudiantes en la barca es segura. Implementa una solución a este problema utilizando métodos sincronizados para gestionar la sincronización, teniendo en cuenta que deben satisfacerse las dos siguientes condiciones de sincronización:

-CS1: Un estudiante no puede subirse en la barca hasta que no hay algún asiento libre en la barca y la configuración para él es segura.
-CS2: Un estudiante no puede bajarse de la barca hasta que no se ha terminado el viaje.

El esqueleto de la solución está en el campus virtual. Los estudiantes Android/IPhone llaman al método public void android(int id)/public void iphone(int id)del objeto barca cuando quieren cruzar el río. Para simplificar el problema, se supone que el último estudiante que sube a la barca (el que ocupa el cuarto asiendo) es el que maneja la barca y la lleva al otro extremo del río, es decir, este estudiante es el que decide cuando se ha terminado el viaje, y avisa al resto de los ocupantes de la barca para que se bajen. NOTA: No hay que preocuparse por la orilla desde la que los estudiantes se suben/bajan de la barca.

Ejercicio 3. Supón que átomos de hidrógeno y oxígeno están dando vueltas en el espacio, intentando agruparse para formar moléculas de agua. Para ello es necesario que dos átomos de hidrógeno y uno de oxígeno se sincronicen. Supongamos que cada átomo de hidrógeno y oxígeno está simulado por un proceso. La gestión de la sincronización de los átomos tiene lugar en un objeto gestor de la clase GestorAgua. Cada átomo de hidrógeno llama al método hListo cuando quiere formar parte de una molécula. Del mismo modo los átomos de oxígeno llaman a oListo cuando quieren combinarse con otros dos hidrógenos para formar agua. Los procesos deben esperar en estos métodos hasta que sea posible formar la molécula. Implementa este sistema utilizando métodos sincronizados. El esqueleto de la solución está en el cv.

Ejercicio 4. Barrera. En este ejercicio se trata de sincronizar las iteraciones de una serie de procesos workers. Para ello, cada vez que un worker termina una iteración llama al método finIteracion del objeto Gestor. En este método debe esperar a que el resto de workers también terminen sus iteraciones para comenzar la siguiente. De esta forma, el comportamiento del sistema para N workers debería ser:

N workers terminan la iteración 0-> N workers terminan la iteración 1->...

Implementa el sistema utilizando métodos para sincronizar las hebras. El esqueleto del ejercicio está en el cv.