UNIVERSIDAD DE MÁLAGA
Dpto. Lenguajes y Ciencias de la Computación

Programación de Sistemas y Concurrencia
Práctica adicional
Monitores

Monitores

1.- Supongamos que N niños asisten a una fiesta de cumpleaños, en la que se les ofrece de forma repetida raciones de tarta de chocolate que se encuentran en una bandeja. Como las raciones se acaban con frecuencia, existe un pastelero que, cuando lo avisan, pone una nueva tarta en la bandeja (suponemos que cada tarta da lugar a un número fijo R de raciones). Cuando un niño quiere una ración de tarta, va a la bandeja y la coge. Si ve que la bandeja se ha quedado vacía, avisa al pastelero para que traiga una nueva tarta. Implementa este sistema utilizando métodos sincronizados, teniendo en cuenta que deben satisfacerse las dos condiciones de sincronización siguientes:

-CS1: Un niño que quiere una ración de tarta debe esperar si la bandeja está vacía.

-CS2: El pastelero no pone una nueva tarta en la bandeja, hasta que la bandeja está vacía.

Supón que el pastelero es perezoso, por lo que mientras que no lo avisan se queda durmiendo un ratito. El esqueleto del sistema se encuentra en el campus virtual. Cuando un niño quiere una ración llama al método public void quieroRacion(int id) del objeto bandeja.

Por otro lado, el pastelero llama al método public void tarta() para poner una nueva tarta sobre la bandeja. Supón que la bandeja está inicialmente vacía.

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.