



UNIVERSIDAD
DE MÁLAGA

Dpto. Lenguajes y
Ciencias de la Computación

Programación de Sistemas y Concurrencia

Práctica adicional Semáforos y Monitores

Semáforos

1.- La tribu de los caníbales **Ngoro-Ngoro** está de enhorabuena. Han capturado una nutrida expedición de exploradores de una conocida revista de divulgación científica. Para celebrarlo, los Ngoro-Ngoro, que no piensan en el futuro, organizan una fiesta en la que **toda la tribu** baila y come de una **olla central** en la que el **cocinero** va cocinando a los exploradores capturados. Para comer, cada caníbal se sirve en su propio plato de la olla central. Cuando un caníbal va a comer y encuentra la olla vacía, llama al cocinero para que vuelva a preparar otro explorador y lo eche a la olla. Pero preparar un explorador es una tarea bastante cansada, por lo que, mientras la olla no está vacía, el cocinero se retira a su choza a descansar. Diseñar una solución basada en **semáforos binarios** que simule la gran fiesta de los Ngoro-Ngoro en la que se evite despertar al cocinero cuando no haga falta. Suponer que la comida no se acaba nunca.

2.- **El problema de la montaña rusa.** Supón que hay n procesos **pasajeros**, y un proceso **coche**. Los pasajeros esperan repetidamente para darse una vuelta en el coche, que tiene una capacidad $C < n$ de pasajeros. Sin embargo, **el coche sólo da una vuelta cuando está lleno**. El coche tarda T segundos en dar una vuelta, una vez que está lleno. Después de dar una vuelta, cada pasajero da un paseo por el parque de atracciones durante un tiempo aleatorio, antes de volver a la montaña rusa para darse otra vuelta. Diseña un programa que resuelva este problema utilizando sólo **semáforos binarios**.

3.- Considera un nido con n **pájaros bebés** y **dos** pájaros padres (el papá y la mamá). Todos los pájaros comparten un plato común que puede contener a lo sumo B bichitos.

Cada pájaro padre (papá o mamá) da una vuelta volando, atrapa un bichito, vuelve al nido, espera a que haya sitio en el plato, deposita en él el bichito capturado, y repite todas las acciones de nuevo.

Cada pájaro bebé pía un ratito, espera a que el plato tenga algún bichito, lo coge, se lo come y repite de nuevo todas las acciones.

Implementa este sistema utilizando **semáforos binarios**, suponiendo un comportamiento infinito para cada uno de los procesos.

Métodos sincronizados/Locks

1 -**El problema de la cuenta de ahorros.** Una cuenta de ahorros es compartida por varios clientes de un banco. Cada cliente puede depositar o extraer fondos de la cuenta. Supón que la cantidad de cada depósito o extracción es siempre positiva. El balance de la cuenta es siempre la suma de todos los depósitos realizados hasta la fecha, menos las extracciones. **El balance**

nunca puede ser negativo. Diseña una solución basada en métodos sincronizados para este problema. Cuando hay un nuevo depósito en la cuenta, tu solución debería permitir extraer fondos a cualquier petición (o peticiones) de extracción de fondos que estuviera esperando, y que pueda completarse.

2.- Varios procesos (**N**) compiten por utilizar unos cuantos recursos **Rec** en exclusión mutua. El uso correcto de los recursos es gestionado por una clase Control. Cada proceso pide un número de recursos llamando al método del monitor **qrecursos(id,num)**, donde **id** es el identificador del proceso que hace la petición y **num** es un número entero que especifica el número de recursos que pide el proceso. Después de usar los recursos, el proceso los libera ejecutando el método **librecursos(id,num)**. Para asignar los recursos Control utiliza la técnica FCFS (First Come, First Serve) que significa que los procesos son servidos siempre en el orden en el que han realizado sus peticiones. Así un proceso que pide **num** recursos debe esperarse si hay otros procesos esperando aún cuando existan **num** recursos disponibles en el sistema. Implementa la clase **Control**, y varias hebras usuarios que realicen las peticiones a **Control**.

3.- Supongamos un centro de cálculo que dispone de dos tipos de impresoras de características ligeramente diferentes: A y B. Hay documentos que pueden ser impresos en cualquiera de las impresoras, otros que sólo pueden ser impresos en las del tipo A y otros que sólo pueden ser impresos en las de tipo B. Si un proceso quiere imprimir un fichero tiene que esperar a que esté libre una impresora del tipo que puede usar. Diseñar soluciones usando métodos sincronizados que cumpla las reglas anteriores y que asegure el uso en **exclusión mutua** de las impresoras.