LISTA DE EJERCICIOS DE CONCURRENCIA

Procesos en equilibrio

En un sistema concurrente existen dos tipos de procesos, A y B, que compiten por utilizar cierto recurso. Al recurso se accede mediante la rutina de solicitarlo, esperar hasta que se pueda usar, usarlo y luego liberarlo.

En cualquier momento puede haber un máximo de N procesos de cualquier tipo usando el recurso (N es constante). Por otro lado, para que un proceso de tipo A pueda entrar a emplear el recurso, debe haber al menos el doble de procesos de tipo B que procesos de tipo A dentro del recurso.

Los fumadores

En torno a una mesa hay cuatro personas: tres fumadores y un proveedor. Cada fumador fuma cigarrillos uno detrás de otro. Para poder fumarse un cigarrillo se necesitan tres ingredientes: papel, tabaco y fósforos. Uno de los fumadores tiene papel, el otro tabaco y el otro fósforos. El proveedor tiene una cantidad infinita de los tres ingredientes.

Inicialmente, el agente coloca dos de los ingredientes en la mesa, escogidos al azar. El fumador que tiene el ingrediente que falta toma lo que hay en la mesa, lía un cigarrillo y se lo fuma. Cuando termina, se lo indica al proveedor. El proveedor entonces coloca otros dos ingredientes y el ciclo se repite una y otra vez.

Se pide construir un programa compuesto por cuatro hilos (tres fumadores y el proveedor) que estén correctamente sincronizados.

La anterior sería una modalidad muy básica del problema con la que se obtendría una calificación mínima en la práctica. Como opción adicional, se puede variar el comportamiento de los fumadores para permitir una mayor concurrencia:

- Cuando un fumador toma lo que hay en la mesa y se pone a fumar, el proveedor inmediatamente toma dos ingredientes y los coloca en la mesa. De esta forma, mientras un fumador está fumando, se da la posibilidad de que otro fumador pueda tomar la pareja de ingredientes que hay en la mesa y así lleguemos a tener varios fumadores al mismo tiempo. (Aunque esto no ocurrirá si el proveedor coloca justo el par de ingredientes que no le sirven a los fumadores en espera, pero eso depende del azar).

El taller de costura

Tenemos un taller de costura (sitúenlo en Indonesia, para dar ambiente), dedicado a hacer jerséis. En su interior tenemos a tres personas trabajando de sol a sol por cuatro duros, para que los occidentales podamos comprarlos baratitos (la globalización, ya saben...)

Una persona está continuamente fabricando mangas, que va depositando en un cesto. El cesto tiene una capacidad limitada: cuando se llena, la costurera deja de coser más mangas hasta que hay hueco libre. Otra persona está continuamente fabricando los cuerpos de los jerséis, que también deposita en su correspondiente cesta de capacidad limitada. Una tercera persona se encarga continuamente de ensamblar jerséis,

cogiendo en cada caso dos mangas de la cesta de mangas y un cuerpo de la cesta de cuerpos.

Se trata de escribir el código que sincronice a estas tres personas, de forma que las dos primeras personas no avancen si su cesta está llena, y que la tercera persona no avance si le faltan piezas para hacer un nuevo jersey.

Se supone que las capacidades de las dos cestas son constantes y distintas (supongan, p.ej. que son dos constantes enteras "NumMaxMangas" y "NumMaxCuerpos").

Los filósofos (the dining philosophers)

Otro problema planteado por Dijkstra. Cinco filósofos orientales dedican su existencia a meditar y comer, alternativamente. Los filósofos están sentados en torno a una mesa circular con cinco sillas, cada una de un filósofo. En el centro de la mesa hay un cuenco de arroz. En la mesa hay cinco palillos, uno a cada lado de cada filósofo (uno a su izquierda y otro a su derecha). Para comer, un filósofo necesita usar los dos palillos que tiene a cada lado. Si otro filósofo ha cogido uno de los palillos, deberá esperar a que lo deje en la mesa.

Se trata de implementar este sistema garantizando que no se producen interbloqueos. Cada filósofo será un hilo diferente.

El barbero dormilón (the sleeping barber)

Este problema fue planteado por Edsger Dijkstra. Una barbería tiene una sala de espera con N sillas, más la silla de barbero. Si no hay clientes, el barbero se echa a dormir. Si un cliente entra en la barbería y encuentra todas las sillas ocupadas, se marcha de la barbería. Si el barbero está ocupado, pero hay sillas disponibles, el cliente se sienta en una de ellas. Si el barbero está dormido, el cliente lo despierta. Se trata de escribir un programa que coordine al barbero y los clientes, cada uno de los cuales se corresponde con un hilo.

El puente estrecho

Construya un sistema para gestionar el tráfico sobre un puente estrecho. El puente lo pueden atravesar vehículos desde ambos extremos. Como el puente tiene un solo carril, en un momento dado sólo puede haber vehículos cruzándolo en un único sentido. El puente tiene una capacidad de carga limitada: si hay más de tres vehículos, se hunde. Implemente una solución en la que cada vehículo sea un hilo. El algoritmo que sigue cada vehículo debe ser parecido a esto:

```
void vehículo (sentido) {
   llegar_al_puente(sentido)
   cruzar_puente(sentido)
   salir_del_puente(sentido)
}
```

El parámetro *sentido* indica en qué sentido se cruza el puente (o sea, que puede tener dos valores, por ejemplo *izquierda* y *derecha*).

Garantice que no ocurren colisiones y que el puente no se hunde.

Como añadido opcional, intente que no se produzcan esperas indefinidas.

Problema del baño mixto

Ante la carencia de baños en el Edificio de Informática, el administrador ha decidido que los pocos baños sanos sean de uso mixto: pueden entrar tanto hombres como mujeres, pero con la condición de que simultáneamente sólo pueda haber personas de un único sexo. Implemente la gestión de uno de estos baños (cada persona se simula como un hilo que entra y sale del baño).

- Añadido 1: el baño tiene una capacidad limitada.
- Añadido 2: diseñar una solución libre de inanición.

Caníbales y misioneros

En el África profunda viven junto a un río una tribu de caníbales y un grupo de misioneros. El río se puede cruzar mediante un bote, con capacidad para tres personas. Como las aguas son bravas, el bote siempre tiene que transportar carga completa. Si se sientan dos misioneros y un caníbal, éstos le comen el tarro al caníbal, hasta el punto de que el caníbal se arroja al agua: hay que evitar esta situación. Las demás combinaciones no dan problemas.

El sistema debe lanzar varios hilos, uno por cada persona. Escriba dos procedimientos: LlegaMisionero() y LlegaCaníbal(), para cuando cada persona llega al bote. Estos procedimientos dejan al hilo esperando hasta que el bote se encuentre lleno y seguro. Cuando esto ocurra, el procedimiento retorna y se considera que el barco vuelve a quedar vacío.

Radiado atómico (atomic broadcast)

Tenemos un proceso productor y N procesos consumidores que se comunican a través de un búfer de capacidad C. Cada elemento producido va destinado a todos los consumidores: un elemento en el búfer no desaparece hasta que ha sido leído por todos los consumidores. Cada consumidor va leyendo los elementos producidos de forma secuencial, a su propio ritmo: puede haber consumidores mucho más adelantados que otros.