

Grupo601–A-CursodeSistemasOperativos1

EjerciciospropuestosparaProgramaciónConcurrente conPASCAL-FC

Tema:Semáforos

1.- Un conjunto de procesos tienen relaciones de *precedencia* si hay restricciones sobre el orden en el que pueden ejecutarse. Considerar seis procesos. P1 y P2 deben, ambos, ejecutarse antes que P3; P3 debe ejecutarse antes que P4 y P5; y P6 debe ejecutarse después que los demás procesos. Utilizar semáforos para forzar estas relaciones de precedencia. Además, implementar el sistema de manera cíclica para que, por ejemplo, la primera ejecución de P6 pueda realizarse concurrentemente con la segunda de P1 y P2.

2.- Una barbería tiene un solo barbero, una silla para cortar y N sillas para que los clientes esperen cuando el barbero está ocupado. Si no hay clientes, el barbero siempre se encuentra sentado en la silla de barbero durmiendo. Cuando llega un cliente despierta al barbero si se encuentra dormido y se sienta en la silla para hacerse un corte. Si el barbero está atendiendo a otro cliente, se sienta en una de las sillas de espera, pero si todas están llenas, abandona la peluquería.

- Diseñar una solución basada en semáforos generales.
- Diseñar una solución que sólo utilice semáforos binarios.

Nota: este problema es una representación del problema del productor consumidor con buffer acotado. El barbero es el proceso consumidor, mientras que los clientes son los datos consumidos.

3.- Suponer un puente en el que los carros sólo pueden circular en un sentido al mismo tiempo y que a dicho puente pueden llegar carros en dirección Norte o Sur. Los carros que lleguen al puente y que vayan en la misma dirección pueden cruzar al mismo tiempo. En caso de que el puente esté ocupado, los carros que vienen en dirección contraria a los que están cruzando el puente deben esperar.

Este problema es una generalización del problema de los lectores/escritores, pero en este caso se permite que los escritores accedan simultáneamente a la base de datos. Como en el caso del problema de los lectores escritores cada carro puede pasar el tiempo que desee en el puente, es decir, un carro puede adelantar a otros que van en su misma dirección y salir primero.

Implementar una solución basada en semáforos que no se preocupe de la posición indefinida.

Posteriormente implementar una solución que se ajuste a cualquier número de lectores y escritores para que evite la posición indefinida.

4.- Resolver mediante el uso de semáforos el problema de los fumadores: hay tres fumadores y un agente que pone los cigarrillos encima de la mesa. Cada fumador dispone de un surtido suficiente para el resto de su vida de uno de los tres ingredientes (café, azúcar y tabaco). El agente pondrá encima de la mesa un cigarrillo cuando los fumadores lo toman, arma un cigarrillo con los ingredientes y vuelve a repetir el ciclo.

5.- La tribu de los caníbales Ngoro-Ngoro está de excursión de exploradores de una conocida revista de divulgación geográfica. Para celebrar la expedición, 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 hasta que se acaban las reservas de comida. Para comer, cada uno se sirve en su propio plato. Eso sí, de uno en uno. Cuando un caníbal va a comer y encuentra la olla vacía, llama al

cocinero para que vuelva a preparar otros sabrosos platos. Diseñar una solución para el problema de los filósofos utilizando monitores de forma que se evite el bloqueo.

radory loeche ala olla. Pero preparar ue, mientras la olla está llena, el cocinero asada en semáforos que simule la espertar al cocinero cuando no haga falta oqueo, para que no se mueran de ción de los procesos cuando se acaben

6.- Montaña rusa con un carro: Suponer que hay n pasajeros esperando repetidamente para darse una vuelta de $C < n$ pasajeros. Sin embargo, el carro sólo da una vuelta cada C segundos (por ejemplo, 3) y cada pasajero da un paseo por el parque aleatorio entre 0 y 5 segundos antes de volver a la montaña rusa. Diseñar un programa en Pascal - FC que implemente el sistema utilizando sólo semáforos.

ospasajeros y un proceso carro. Los ta en el carro, que tiene una capacidad cuando está lleno. El carro tarda T que está lleno. Después de dar una vuelta de atracciones durante un tiempo montaña rusa para darse otra vuelta. te sistema utilizando sólo semáforos

7.- Montaña rusa con muchos carros : Generalizar el programa anterior para que haya m carros $m > 1$. Como sólo hay un riel, por motivos de seguridad no pueden estar simultáneamente en la montaña rusa, es decir, un carro no debe entrar en la montaña rusa hasta que no esté vacía. Usar únicamente semáforos binarios.

ara los pasajeros, dos carros no decir, un carro no debe entrar en la semáforos binarios.

Tema - Monitores

1.- Implementar el problema de los filósofos utilizando monitores de forma que se evite el bloqueo.

ndo monitores de forma que se evite el

2.- Suponer 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 la impresora del tipo A y otros que sólo pueden ser impresos en la impresora del tipo B. Si un proceso quiere imprimir un fichero tiene que esperar a que esté libre una impresora del tipo que puede usar. Diseñar una solución usando monitores que cumpla las reglas anteriores y que asegure el uso exclusivo de las impresoras.

s de impresoras de características en ser impresos en cualquiera de las as de tipo A y otros que sólo pueden ser imir un fichero tiene que esperar a que esté r una solución usando monitores que exclusión mutua de las impresoras.

3.- Diseñar una solución basada en monitores que simule la gran fiesta de los Ngoro-Ngoro (problema #5 del tema de semáforos) en la que se evita despertar al cocinero cuando no haga falta (tener en cuenta su mal genio y su gran fuerza) y el bloqueo, para que no se mueran de hambre estos amables caníbales. Controlar la terminación de los procesos.

ule la gran fiesta de los Ngoro-Ngoro vitedespertar al cocinero cuando no haga y el bloqueo, para que no se mueran de ción de los procesos.

5.- Suponer que átomos de hidrógeno y oxígeno están agrupados 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 (H) y uno de oxígeno (O) está simulado por un proceso. Cada átomo H llama al procedimiento Hlisto cuando quiere combinarse en una molécula de agua. De igual forma, cada átomo O llama a Olisto cuando quiere combinarse.

andovuelltas en el espacio intentando necesario que dos átomos de que cada átomo de hidrógeno (H) y como H llama al procedimiento Hlisto igual forma, cada átomo O llama a

El trabajo consiste en escribir un monitor que implemente los procedimientos Hlisto, Olisto y Hacer Agua. Un átomo H tiene que esperarse en Hlisto hasta que otro átomo H haya llamado también a Hlisto y un átomo O haya llamado a Olisto. Del mismo modo, un átomo O debe esperar en Olisto a que dos átomos H hayan llamado a Hlisto. Luego el átomo O, desde Olisto, debe llamar al procedimiento Hacer Agua. Cuando el procedimiento Hacer Agua termine, los procesos deberían terminar sus llamadas a Hlisto y Olisto. Evitar la espera activa, el bloqueo y la posición indefinida. Suponer que tienen k átomos O_2 y k átomos H en el programa.

ente los procedimientos Hlisto, Olisto y hasta que otro átomo H haya llamado Del mismo modo, un átomo O debe Hlisto. Luego el átomo O, desde Olisto, cedimiento Hacer Agua termine, los tres isto. Evitar la espera activa, el bloqueo y os O_2 y k átomos H en el programa.

6.- Suponer que un proceso productor y n procesos consumidores comparten un buffer acotado con b componentes. El productor deposita mensajes en el buffer, y los consumidores los reconocen. Cada mensaje depositado por el productor debe ser recibido por todos los consumidores. Además, cada consumidor debe recibir los mensajes en el orden en el que se depositaron. Sin embargo, los consumidores pueden recibir los mensajes a diferentes

idores comparten un buffer acotado l buffer, y los consumidores los e ser recibido por todos los mensajes en el orden en el que se cibir los mensajes a diferentes

velocidades. Por ejemplo, un consumidor podría recibir b mensajes antes de que un segundo reciba ninguno, si éste último es lento. Implementar un monitor que represente este tipo de comunicación.