



UNIVERSIDAD
DE MÁLAGA

Dpto. Lenguajes y
Ciencias de la Computación

Programación de Sistemas y Concurrencia Práctica nº5 Tema 6 - Semáforos

1. En un sistema industrial existen tres sensores que realizan mediciones del nivel de temperatura, humedad y luz respectivamente. Cuando se han recogido mediciones de los tres sensores, existe un dispositivo “trabajador” encargado de realizar ciertas tareas según las mediciones realizadas.

El dispositivo **no puede** comenzar a realizar sus tareas hasta que se han recogido mediciones de los tres sensores, y los sensores **no pueden** volver a realizar mediciones hasta que el dispositivo finaliza sus tareas. El proceso se repite de forma indefinida de manera que cuando el dispositivo finaliza sus tareas, volverá a esperar a que haya mediciones de los tres sensores.

Realizar utilizando semáforos el modelado de dicho sistema. Modelar el dispositivo trabajador y cada sensor como una hebra (con lo cual habrá un total de 4 hebras). Modelar el proceso de realizar mediciones y las tareas del dispositivo con retrasos aleatorios y valores de tipo entero. Inicialmente puede suponerse que los sensores pueden comenzar haciendo peticiones.

2. En una cadena de montaje existe un robot encargado de colocar productos de 3 tipos diferentes (1, 2 o 3) en la cadena de montaje. Otros robots, retiran los productos de la cadena de montaje para realizar su empaquetado, teniendo en cuenta que están especializados en un solo tipo de producto (1, 2 o 3), ignorando los que no son de su tipo. Finalmente, se quiere llevar un control del total de productos empaquetados (independientemente de su tipo).

Modelar utilizando semáforos el sistema descrito con las siguientes indicaciones:

- Modelar cada robot como una hebra (1 colocador y 3 empaquetadores, uno para cada tipo de producto).

- Los productos son colocados de uno en uno en la cadena, y solamente en posiciones libres (se puede considerar que en la cadena de montaje caben un máximo N de elementos). Si no hay posiciones libres el robot colocador tendrá que esperar hasta que algún producto sea retirado de la cadena.

- Los robots empaquetadores se especializan en un tipo de producto (1, 2 o 3) en tiempo de inicialización.

- Los robots empaquetadores comprueban si hay algún elemento de su tipo en la cadena ignorando los productos que no sean de su tipo. Si hay algún producto de su tipo lo retiran de la cadena (sólo 1 producto cada vez) y la posición queda libre para colocar nuevos productos, en caso contrario se quedan a la espera de que haya nuevos productos.

- Los robots empaquetadores de distinto tipo pueden funcionar a la vez.
- Tanto el colocador como los empaquetadores nunca acaban.
- Cada vez que un robot empaquetador procesa un producto, la cuenta total de productos empaquetados debe aumentar y mostrarse un mensaje por pantalla.

3. Se quiere realizar la simulación simplificada del tráfico aéreo de un aeropuerto teniendo en cuenta las siguientes características:

- Se dispone de una única pista de aterrizaje compartida por todos los aviones.
- Los aviones pueden aproximarse por el norte o por el sur, por ello, para controlar el tráfico aéreo existe una hebra controladora para el norte y otra para el sur.
- Existen N aviones representados cada uno con una hebra que periódicamente intentan aterrizar en el aeropuerto (por el lado que se desee). El procedimiento para aterrizar es el siguiente:
 - o El avión “solicita permiso” para aterrizar al controlador del lado por el que se aproxima y queda a la espera (utilizando algún semáforo) de que se le dé permiso.
 - o Los procesos controladores atienden peticiones de aterrizaje (de una en una). Cuando les llega una petición, tienen que esperar hasta que la pista esté disponible y entonces avisan al avión correspondiente de que ya puede aterrizar. Si no hay peticiones, las hebras controladoras deben dormirse. Cuando el avión haya terminado de aterrizar se indicará a la hebra controladora para que pueda liberar la pista.
 - o Cuando un avión tenga permiso para aterrizar, se simulará el aterrizaje con un tiempo aleatorio durante el que la pista está ocupada.

Se deben mostrar mensajes indicando cuándo un avión comienza a aterrizar y cuándo finaliza el aterrizaje.

4. En el problema del “puente estrecho” se quiere simular el paso de coches por un puente estrecho de manera que simultáneamente sólo pueden circular coches en un único sentido. Implementar en Java y con semáforos este problema con las siguientes características:

- Existen N coches (representando cada coche con una hebra) intentando pasar de izquierda a derecha o de derecha a izquierda.
- En cada momento el puente puede estar o vacío u ocupado por coches que circulen únicamente en un sentido (izquierda-derecha o derecha-izquierda). Por simplificar, se puede suponer que pueden circular al mismo tiempo un número indefinido de coches en el mismo sentido.
- Los coches periódicamente realizan el mismo proceso: esperar hasta que puedan entrar al puente, pasar el puente y salir del puente y, por simplificar, siempre en el mismo sentido. Después de salir del puente, esperan un tiempo aleatorio hasta volver a intentar entrar. Tampoco es necesario controlar que los coches que entren antes, salgan antes del puente que otros que llegan después.

Realizar 2 versiones de este problema:

- Versión “injusta” donde no se controla que continuamente puedan estar pasando coches de un mismo sentido sin dejar pasar nunca a los del otro sentido.

- Versión “justa” donde se controla que no se acumulen coches indefinidamente en uno de los 2 lados, bien controlando que ya han pasado un número “suficiente” de coches de un sentido, bien controlando que la cola de coches en el otro sentido supera cierto límite.

El programa deberá funcionar correctamente en todas las situaciones. No serán válidas soluciones que utilicen espera activa ni hebras adicionales.

5. Implementar el problema del productor-consumidor utilizando varios productores y varios consumidores con la particularidad de que el buffer en esta ocasión es NO destructivo. Esto es: cada vez que un productor coloca un elemento en el buffer, el elemento tiene que ser consumido por todos los consumidores (una sola vez por cada consumidor) antes de poder liberar la posición asociada al buffer. El resto del funcionamiento es similar al productor-consumidor normal.

Por ejemplo, si en el buffer tenemos los elementos [1,2,3,4] y 3 consumidores, se podría tener que cada consumidor ha consumido los siguientes elementos:

C1: 1, 2

C2: 1

C3: 1, 2, 3

Hasta que un elemento no ha sido consumido por todos los consumidores, la posición no se libera. En el ejemplo anterior, el elemento “1” ha sido consumido por todos los consumidores y su posición en el buffer podría ser utilizada para colocar otro elemento. El elemento “2”, por el contrario, sólo ha sido consumido por C1 y C3 por lo que su posición todavía no puede ser liberada en el buffer.