



---

## PROGRAMACIÓN CONCURRENTE

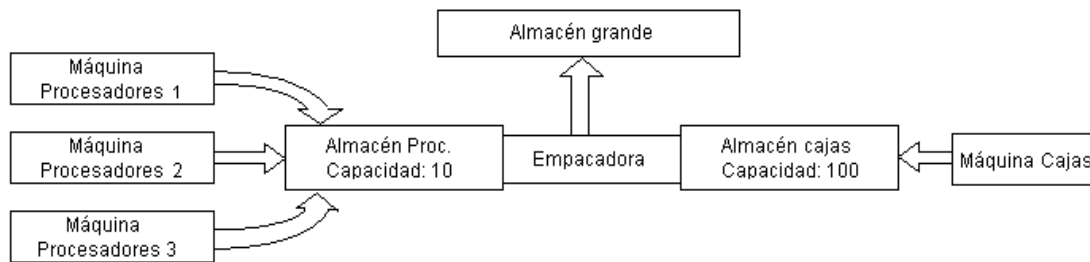
Para todos los problemas descritos a continuación, haga uso de los conceptos vistos en clase referentes a Hilos, Sincronización y Recursos Críticos. Todas las soluciones planteadas deben evitar cualquier tipo de espera activa:

- 1) Considere un sistema de cajeros automáticos que brindan a sus usuarios las siguientes facilidades:
  - Consultar el saldo de una cuenta.
  - Retirar dinero de una cuenta.
  - Depositar dinero en una cuenta.
  - a.- Identifique las secciones críticas.
  - b.- Plantee una solución en Java como Pascal-FC, considerando una sola cuenta bancaria.

- 2) Un estacionamiento de la ciudad, cuya capacidad máxima es de  $P$  puestos, consta de dos taquillas, una de las cuales se encarga de entregar un ticket a los carros que lleguen y la otra se ocupa de recibir el pago cuando un vehículo sale del estacionamiento. Existe la posibilidad de conocer la cantidad de puestos ocupados en cualquier momento, debido a que esta información es registrada cada vez que entra o sale un vehículo. Por lo tanto, sólo una de las taquillas, la que entrega el ticket o la que recibe el pago, puede realizar su trabajo en un mismo instante, ya que el registro de la cantidad de puestos ocupados debe ser correcto en todo momento.

Se quiere que usted:

- a.- Analice el sistema con el fin de determinar los recursos críticos, procesos que intervienen y las políticas a seguir para la sincronización.
  - b.- Diseñe una solución en Java que proporcione información acerca de la cantidad de puestos ocupados en el estacionamiento haciendo uso de monitores.
- 3) Transmeta es una compañía que desarrolla microprocesadores para PCs. Estos microprocesadores requieren de muy poco voltaje para funcionar y su éxito radica en la capacidad de ejecutar parte de las instrucciones en software. La fábrica de Transmeta cuenta con tres máquinas para desarrollar los procesadores, una para crear las cajas donde se empacará el producto y otra que se encarga de empacar los procesadores en las cajas. Una vez que se termina de construir un procesador es enviado a la máquina empacadora, la cual tiene la capacidad de almacenar hasta 10 procesadores. Igualmente sucede con la máquina que construye las cajas. Estas son colocadas en la unidad de almacenamiento de la máquina empacadora, la cual tiene la capacidad de mantener hasta 100 cajas. Una vez que un producto es completado (construido y empacado), se coloca en un almacén tan grande que se puede considerar como infinito para modelar el problema. La situación en general puede ser observada en la figura.



Adicionalmente se requiere que en todo momento se pueda consultar la cantidad de procesadores que han sido construidos y que no se han empacado, así como también la cantidad de cajas que no han sido utilizadas.

Se desea que usted realice lo siguiente:

- a.- Determinar el (los) recurso(s) crítico(s).
- b.- Identifique los procesos involucrados en la situación.
- c.- Implemente una solución en el lenguaje Java que modele la situación planteada (usando monitores y BlockingQueue).

- 4) Existe una barbería en el cual hay un solo barbero, el cual tiene afición por dormir. La barbería tiene una sola silla para el cliente de turno y N sillas para la espera. Si el barbero no tiene clientes este se acuesta en su silla a dormir, hasta que llegue un cliente que lo despierte y lo atienda. Si otros clientes llegan mientras el barbero está trabajando estos se sientan si existen sillas disponibles, si no hay sillas estos salen de la barbería. Se quiere que Ud. Identifique recursos críticos, procesos involucrados, condiciones de sincronización e implemente una solución en lenguaje en Java usando hilos y monitores.
- 5) Cinco filósofos se sientan alrededor de una mesa y pasan su vida cenando y pensando. Cada filósofo tiene un plato de fideos y un tenedor a la izquierda de su plato. Pero para comer los fideos son necesarios dos tenedores y cada filósofo puede tomar el tenedor que esté a su izquierda o derecha, uno por vez (o sea, no puede tomar los dos al mismo tiempo, pero puede tomar uno y después el otro). Si cualquier filósofo coge un tenedor y el otro está ocupado, se quedará esperando, con el tenedor en la mano, hasta que pueda coger el otro tenedor, para luego empezar a comer. Si dos filósofos adyacentes intentan tomar el mismo tenedor a una vez, se produce una condición de carrera: ambos compiten por tomar el mismo tenedor, y uno de ellos se queda sin comer. Si todos los filósofos cogen el tenedor que está a su derecha al mismo tiempo, entonces todos se quedarán esperando eternamente, porque alguien debe liberar el tenedor que les falta. Nadie lo hará porque todos se encuentran en la misma situación (esperando que alguno deje sus tenedores). Entonces los filósofos se morirán de hambre. Este bloqueo mutuo se denomina *deadlock*. Se quiere que Ud. Identifique recursos críticos, procesos involucrados, condiciones de sincronización e implemente una solución tanto en Java como en Pascal-FC.



- 6) Se tiene un túnel entre un punto A y un punto B. El túnel solo se puede usar en una sola dirección (esto es de A hacia B o de B hacia A) en cualquier momento dado. Debido a la afluencia de automóviles en el punto A, se debe permitir partir tres automóviles de A hacia B por cada automóvil de B hacia A.

Se desea que Ud. implante una solución usando Pascal-FC

- 7) En Computación Gráfica el CPU construye el Display Buffer y el GPU representa el DB en el monitor. Obviamente mientras el DB esta siendo construido no puede ser representado, tampoco es viable que el GPU espere mientras el DB es construido por el CPU. Para evitar este problema se usa la técnica de Doble Buffer, esto es: mientras el Buffer1 es construido por el CPU, el Buffer2 es representado por el GPU. Una vez finalizados ambos procesos, los buffers son intercambiados y el proceso comienza nuevamente. Se desea que Ud. diseñe una solución usando `java.util.concurrent`, `exchangers` y `Rendezvous` en Pascal-FC.

- 8) El problema de las 8 torres consiste en colocar 8 torres en un tablero de ajedrez sin que hayan dos en la misma fila o columna. Para resolver este problema concurrentemente, se colocan inicialmente las torres aleatoriamente en el tablero. Si no se ha obtenido una solución, cada torre debe de moverse a una fila o columna contigua. Este proceso continúa hasta obtener una solución. Suponga que cuenta con los siguientes métodos:

```
boolean posición_valida(int i, int j)
```

```
//indica se la posición ji,j del tablero es válida, es decir,
```

```
//si se encuentra en los límites del tablero.
```

```
int proximo_movimiento()
```

```
// Indica el próximo movimiento a realizar.
```

```
//Retorna 1 si el movimiento es hacia arriba, 2 hacia la derecha, 3 hacia
```

```
//abajo, y 4 hacia la izquierda.
```

En función de lo descrito, se desea que Ud.:

a.- Indique recursos críticos, proceso y problemas de sincronización que podrían surgir.

b.- Elabore el programa con el lenguaje de programación Java. (Use Monitores).

- 9) En una empresa hay un director, unos jefes de sección y un grupo de técnicos bajo la responsabilidad de cada jefe. Debido a que las secciones no tienen trabajo especializado, la forma de repartir tareas en esta empresa es la siguiente: el director tiene un tablón en la puerta en el que va escribiendo las tareas a realizar. Los jefes de sección visitan regularmente el tablón, seleccionan una tarea y la borran. A renglón seguido estos jefes utilizan el mismo sistema con los técnicos de tal forma que cuando llegan a su despacho escriben en el tablón la tarea que han borrado del tablón del director. Los técnicos de cada sección cada vez que acaban con una tarea necesitan encontrar otra a realizar. Para ello visitan el tablón del jefe, eligen una tarea y la borran del tablón. Además, esta empresa tiene un sistema antiestrés que hace que cada jefe o director sepa el máximo número de tareas que puede poner en el tablón sin que sus subordinados sufran estrés. El director sabe que puede haber M tareas para los J jefes y cada jefe sabe que puede tener en su tablón  $N_i$  tareas (donde i indica el número identificativo del jefe correspondiente) para sus  $T_i$  técnicos.



- a.- Identifique el(los) recurso(s) crítico(s) y procesos que intervienen
- b.- Realice el ejercicio usando monitores

- 10) En un sistema OpenVMS se desea sincronizar el acceso a una base de datos entre una serie de procesos que efectúan lecturas y escrituras en la misma. Existen múltiples procesos lectores, pero sólo un proceso escritor. Para evitar que la información se corrompa o que se acceda a información inconsistente, el sistema debe cumplir los siguientes requisitos:
- a) En un momento dado, a la base de datos pueden acceder o bien un número indeterminado de procesos lectores, o bien el proceso escritor, pero no los dos tipos de procesos a la vez.
  - b) Cuando el proceso escritor desee acceder a la base de datos, ningún proceso lector podrá comenzar una nueva lectura. Por supuesto, el escritor esperará que terminen los posibles procesos que ya estuvieran usando la base de datos.
  - c) Cuando un proceso lector desee acceder a la base de datos únicamente lo hará si el proceso escritor no la está usando en esos momentos ni tampoco está esperando para acceder. En caso contrario, esperará.
  - d) El sistema no tendrán esperas activas y los procesos podrán usar la base de datos tan pronto como sea posible, siempre que se cumplan las condiciones mencionadas anteriormente.

- a.- Identifique el(los) recurso(s) crítico(s) y procesos que intervienen.
- b.- Proponga una solución, usando monitores, para sincronizar el acceso a la base de datos.

- 11) El “más veloz” es un juego que permite determinar cuan veloces pueden ser los participantes. El juego consta de 4 jugadores, divididos en dos equipos, y un árbitro. Los jugadores deben comunicarse con el árbitro para que éste pueda conformar los equipos. El árbitro va formando los equipos a medida que llegan los jugadores. Una vez que los equipos están formados, el árbitro da la señal de inicio de juego. Los jugadores deben buscar las pelotas proporcionadas por el árbitro. Éste reparte 30 pelotas. No existe un orden preestablecido para atender a los jugadores. Cuando los jugadores obtienen una pelota deben de depositarla en una caja. Sólo un jugador a la vez debe de colocar su pelota en la caja. Debe tenerse un control sobre los jugadores que colocan las pelotas para determinar a qué equipo pertenece y llevar el conteo de las pelotas por equipo. El árbitro es el que indica cuál es el equipo ganador del juego.

Se desea que realice lo siguiente:

- a.- Determine el (los) recurso (s) crítico (s).
- b.- Identifique los procesos involucrados en la situación.
- c.- Implemente una solución en el lenguaje Java que modele la situación planteada.



- 12) Existe una conjetura que dice que si se tienen 1000 monos y 1000 máquinas de escribir en algún momento, escribiendo al azar, generarán una gran novela. Debido a que probar esta teoría en vivo es problemático, Ud. ha sido seleccionado para desarrollar un sistema que simule a los 1000 monos escribiendo al mismo tiempo. Para desarrollar la simulación Ud. debe considerar lo siguiente:
1. Cada mono escribe un número aleatorio de caracteres y luego descansa un tiempo para pensar lo que continuará escribiendo.
  1. Existe un “depósito de texto” (cuya capacidad es de 10.000 caracteres), donde es colocada toda la información escrita por los monos. Esta información es colocada cada vez que un mono se retira para pensar, siguiendo una política FIFO (*first in first out*).
  2. Existe un “Recoge-letras” que toma una cantidad aleatoria ( $k$ ) de caracteres del “depósito de texto”, respetando la política FIFO. Una vez obtenido el texto, el “Recoge-Letras” tarda  $k$  milisegundos en procesarlo.
  3. Si en algún momento el “depósito de texto” está lleno y algún mono quiere escribir, éste recibirá una descarga de corriente eléctrica que lo mantendrá sin escribir por un tiempo aleatorio.
  4. Cuando un mono ha recibido 3 descargas de corriente eléctrica, la próxima vez que trate de escribir con el “depósito de texto” lleno, se le bloqueará definitivamente la máquina de escribir y su contador de descargas eléctricas volverá a cero.

En función de lo descrito, se desea que Ud.:

- a.- Identifique los procesos y el (los) recurso(s) crítico(s) presentes en el problema.
  - b.- Identifique las condiciones de sincronización presentes en el problema y describa las políticas de sincronización necesarias para asegurar el buen funcionamiento.
  - c.- Muestre un algoritmo de alto nivel para las clases que implementan hilos de ejecución (*threads*) suponiendo que se utiliza Java como lenguaje de programación. Indique la información de control (atributos) necesarios e indique las acciones que se deben realizar en el método `run()`. Indique para cada una de las clases identificadas el número de instancias necesarias en el problema.
  - d.- Muestre con un nivel de detalle superior al utilizado en la sección anterior, la clase que encapsula el recurso crítico. Recuerde utilizar los recursos que provee Java.
  - e.- Suponiendo que el lenguaje de programación utilizado es Pascal-FC, muestre mediante un diagrama los procesos necesarios para resolver el problema y las entradas que debe proveer cada una.
- 13) En la práctica de bateo de la LPBV se dice que un bateador logra conectar un hit cada dos (2) lanzamientos. El lanzador realiza un lanzamiento cada 10 segundos. Se desea que Ud. implemente una solución en Pascal-FC utilizando canales.
- 14) Se requiere que Ud. realice una simulación en Pascal-FC usando canales en donde: una máquina produzca números enteros de forma aleatoria (entre 0 y 100) y los envíe uno a uno a otra máquina que será la encargada de desplegarlos en pantalla, luego de enviar 100 números, esta toma un reposo de 5 segundos para evitar sobrecalentamiento, una vez culminado, el proceso se repite. Si la primera máquina envía un 0 dejará de enviar números, y la otra al recibirlo, lo mostrará y dejará de recibir, finalizando la simulación.