



***Programación Concurrente 2017***  
***Trabajo Práctico N° 6***

1. **La cena de los filósofos:** 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. Para comer los fideos son necesarios dos tenedores y cada filósofo sólo puede tomar los que están a su izquierda y derecha. Si cualquier filósofo toma un tenedor y el otro está ocupado, se quedará esperando, con el tenedor en la mano, hasta que pueda tomar 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 toman 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 interbloqueo o *deadlock*. El problema consiste en encontrar un algoritmo que permita que los filósofos nunca se mueran de hambre.
  - a) Realizar un programa concurrente de forma que utilizando algún mecanismo de sincronización coordine la cena de los filósofos evitando el interbloqueo.
2. **El barbero dormilón:** El problema consiste en una barbería en la que trabaja un barbero que tiene un único sillón de barbero y varias sillas para esperar. Cuando no hay clientes, el barbero se sienta en una silla y se duerme. Cuando llega un nuevo cliente, éste o bien despierta al barbero para que lo atienda o —si el barbero está afeitando a otro cliente— se sienta en una silla. Si todas las sillas están ocupadas se va.
  - a) Realizar un programa concurrente de forma que utilizando algún mecanismo de sincronización coordine el funcionamiento de la barbería, teniendo en cuenta que el máximo de sillas para espera es de 4.
3. **Transbordador:** Un transbordador permite pasar coches de un lado de un río al otro. Los coches viajan por el lado este del río, cruzan el río y continúan su viaje por el lado oeste (nunca vuelven). El transbordador tiene espacio para 10 coches y espera a estar lleno para cruzar el río. Cuando ha cruzado y descargado los coches, vuelve vacío.

Considere en el transbordador operaciones **ir** y **volver**

- **ir** hace que el transbordador cruce con los coches
- **volver** lo hace volver vacío.

Se trata de implementar este problema con hilos, resolviendo la concurrencia con el empleo de algún mecanismo de sincronización (monitor, semáforos, locks).



**Nota: Resolver el problema teniendo en cuenta que, en el caso en que se quiera desbloquear a varios procesos debe hacerse en cadena, esto es, cada proceso debe desbloquear al siguiente.**

4. **El problema del taxista:** Considere los siguientes dos hilos: un taxista y usted. Usted necesita un taxi para llegar a su destino y el taxista necesita que un pasajero tome el taxi para cobrar por su trabajo. Por lo tanto cada uno tiene una tarea. Usted espera tomar el taxi y viajar hasta que el taxista le notifique que ha llegado a su destino. Resultaría molesto para usted y para el preguntar cada dos segundos “¿ya llegamos a destino?”. Entre distintos viajes, el taxista quiere dormir en el taxi hasta que otro pasajero necesite ser conducido hasta algún lugar no desea despertarse de su siesta cada cinco minutos para ver si arribo un pasajero. Por lo tanto, ambos hilos preferirían realizar su trabajo de la manera más relajada posible. Usted y el taxi necesitan alguna forma de comunicar sus necesidades al otro. Mientras usted esta ocupado caminando por la calle buscando un taxi, el taxista esta durmiendo plácidamente en la cabina. Cuando usted le avisa que quiere tomar el taxi, el se despierta y comienza a conducir. Cuando usted arriba a su destino, el taxista le notifica que ha llegado y continúa con su trabajo. El taxista debe ahora esperar y dormir nuevamente una siesta hasta que el próximo pasajero llegue.
- a) Diseñe una solución de comunicación de los hilos que modele la situación antes planteada.