

Programación Concurrente

Práctica 2: Semáforos

1. Dados los siguientes threads

P_1 :	P_2 :
Print(A);	Print(E);
Print(B);	Print(F);
Print(C);	Print(G);

utilizar semáforos para garantizar que:

- a) A se muestra antes que F
- b) F se muestra antes que C

2. Dados

P_1 :	P_2 :
Print(C);	Print(A);
Print(E);	Print(R);
	Print(O);

utilizar semáforos para garantizar que las únicas salidas posibles sean ACERO y ACREO.

3. Considerar los siguientes tres procesos:

P_1 :	P_2 :	P_3 :
Print(R)	Print(I)	Print(O)
Print(OK)	Print(OK)	Print(OK)

Utilizar semáforos para garantizar que el único resultado impreso será R I O OK OK OK (asumimos que Print es atómico)

4. Dados

P_1 :	P_2 :	P_3 :
repeat	repeat	repeat
Print(A);	Print(E);	Print(H);
Print(B);	Print(F);	Print(I);
Print(C);	Print(G);	while true;
Print(D);	while true;	
while true;		

Agregar semáforos para garantizar que:

- a) cantidad de F \leq cantidad de A
- b) cantidad de H \leq cantidad de E
- c) cantidad de C \leq cantidad de G

5. Considere los siguientes dos procesos

T_1 = while true do print(A)

T_2 = while true do print(B)

- a) Utilizar semáforos para garantizar que en todo momento la cantidad de A y B difiera al máximo en 1.
- b) Modificar la solución para que la única salida posible sea ABABABAB...

6. Los siguientes procesos cooperan para calcular el valor N2 correspondiente a la suma de los primeros N números impares. Los procesos comparten las variables N y N2 inicializadas de la siguiente manera: N = 50 y N2 = 0.

$P_1 :$ <pre>while (N > 0) do N = N - 1 end while mostrar (N2)</pre>	$P_2 :$ <pre>while (true) do N2 = N2 + 2 * N + 1 end while</pre>
--	---

- a) Dar una solución utilizando semáforos que garantice el cálculo correcto de N2.
 - b) Dar una implementación en Java de la solución.
 - c) Cómo cambia la solución si N2 se calcula como $N2 = N2 + 2 * N - 1$.
7. Modelar con semáforos un transbordador entre dos costas. El transbordador tiene capacidad para N personas y funciona de la siguiente manera. Espera en una costa hasta llenarse y automáticamente cambia de costa. Cuando llega a una costa se bajan todas las personas y suben las que están esperando. Cada persona debe ser modelada como un thread independiente del transbordador. Implementar su solución en Java.
 8. En un gimnasio hay cuatro aparatos, cada uno permite trabajar un grupo muscular distinto. Los aparatos se cargan con discos (todos del mismo peso). Cada cliente del gimnasio posee una rutina que le indica qué aparatos usar, en qué orden y cuánto peso utilizar en cada caso. Notar que la rutina podría incluir usar repetidamente el mismo aparato. Como norma, el gimnasio exige que cada vez que un cliente termina de utilizar un aparato descargue todos los discos y los coloque en el lugar destinado a su almacenamiento (lo que incluye usos consecutivos del mismo aparato).
 - a) Modele este escenario utilizando semáforos. Indique cuáles son los recursos compartidos y objetos activos.
 - b) Provea un código que simule el funcionamiento del gimnasio, garantizando exclusión mutua en el acceso a los recursos compartidos y que esté libre de deadlocks y livelocks.
 - c) Indique si su solución está libre de inanición. En caso contrario explique cómo podría resolverlo.
 9. Considere el siguiente juego. Existen dos tipos de participantes: los *generadores* y los *consumidores* de bolitas. Los generadores crean bolitas de a una a la vez y las colocan en una bolsa (suman un punto por cada bolita que logran colocar en la bolsa). Los consumidores toman pares de bolitas y suman un punto por cada par (pero no pueden tomar mas de un par por vez). No se conoce a priori la cantidad de participantes que están jugando, pero se debe garantizar en todo momento que la bolsa no contenga mas bolitas que la cantidad de generadores que se encuentran jugando.
 Proponer una solución basada en semáforos que sea libre de deadlocks para todo escenario en donde participen 2 o mas generadores.
 10. Se desea modelar una planta de refinamiento donde vehículos autónomos trasladan productos en distintos estados de procesamiento. La planta consta de una plataforma donde se recibe la materia prima, otra donde se depositan los productos terminados, 8 máquinas procesadoras y 4 vehículos.
 Cada máquina es capaz de descargar el contenido de un vehículo, realizarle algún procesamiento y cargar el producto refinado en un vehículo. No necesariamente se carga el producto procesado en el mismo vehículo del cual se descargó la materia prima.
 Cada vehículo posee una lista ordenada de cargas y descargas que debe efectuar en distintas máquinas o plataformas. Las acciones de carga, descarga, procesamiento y desplazamiento de los vehículos consumen un tiempo no despreciable. Por lo que es importante garantizar que los vehículos permanecen junto a las máquinas mientras se realizan las acciones de carga y descarga. Además, las acciones de desplazamiento y procesamiento deben poder realizarse simultáneamente.
 Modele este escenario utilizando semáforos. Asumiendo que:
 - Siempre es posible realizar una acción de carga en la plataforma de recepción y siempre es posible descargar en la plataforma de entrega.
 - Ninguna ruta requiere descargar en la plataforma de recepción ni cargar en la de entrega.
 - Un vehículo esperando a ser cargado no obstruye a un vehículo siendo descargado por la misma máquina.