UNIVERSIDAD DE CARABOBO
FACULTAD EXPERIMENTAL DE CIENCIAS Y TECNOLOGÍA
DEPARTAMENTO DE COMPUTACIÓN
SISTEMAS OPERATIVOS

Semestre 1-2017
Prof. José Canache
Prep. Viviana Rodríguez

PRÁCTICA #2

Sincronización de Procesos

1. Realice un algoritmo utilizando semáforos de modo que tres procesos que compartan un recurso lo hagan de manera exclusiva.

2. Se tienen los procesos A, B, y C iniciados en forma concurrente. Utilizando semáforos, sincronizar los procesos de manera que la ejecución de las funciones cumpla con el siguiente orden: B1( ) -> C1( ) -> C2( ) -> A1( ) -> B2( ) -> A2( )

```
Proceso A        Proceso B        Proceso C
A1();            B1();            C1();
A2();            B2();            C2();
```

3. Se tienen las siguientes secuencias de ejecución:

a) La secuencia permitida es: ABCDEABCDEABCDEABCDEABCDEABCDEABCDEABCDE…
b) La secuencia permitida es: ACDEBCDEACDEBCDEACDEBCDEACDEBCDE…
c) La secuencia permitida es: (AoB)CDE(AoB)CDE(AoB)CDE(AoB)CDE(AoB)CDE…
d) La secuencia permitida es: (AoB)CE(AoB)(AoB)DE(AoB)CE(AoB)(AoB)DE…

Realizar la sincronización de los procesos utilizando semáforos para cada uno de los casos especificados.

4. Suponga que inician concurrentemente los procesos P1, P2, P3, P4, P5 y P6.

```
Proceso Pi(){
     //Código del proceso Pi
}
```

Se desea definir, inicializar y utilizar los semáforos necesarios en el cuerpo de cada proceso para que se verifiquen las siguientes condiciones:

- P1 debe terminar antes que P2 y P3 empiecen.
- P2 debe terminar antes que P4 y P5 empiecen.
- P3 debe terminar antes que P5 empiece.
- P6 debe empezar después que P3 y P4 termine.

5. En una tienda de mascotas se quiere mantener a todos sus perros felices. Los perros comparten un cojín y una pelota para jugar. Todos los perros quieren inicialmente acostarse en el cojín y después jugar con la pelota. Sin embargo, se encuentran con el inconveniente de que solo uno de ellos puede jugar con la pelota al mismo tiempo y solo tres de ellos pueden usar el cojín. Definir un proceso que ejecuten los perros de forma concurrente para que se sincronicen estas actividades usando semáforos.

6. Considere un sistema con tres procesos fumadores y un proceso agente. Cada fumador está continuamente armando y fumando cigarrillos. Sin embargo, para armar un cigarrillo, el fumador necesita tres ingredientes: tabaco, papel y fósforos. Uno de los procesos fumadores tiene papel, otro tiene el tabaco y el tercero los fósforos. El agente tiene una cantidad infinita de los tres materiales. El agente coloca dos de los ingredientes sobre la mesa. El fumador que tiene el ingrediente restante armaría un cigarrillo y se lo fuma, avisando al agente cuando termina. Entonces, el agente coloca dos de los tres ingredientes y se repite el ciclo. Escriba un programa para sincronizar al agente y los fumadores utilizando semáforos.

7. Un grupo de estudiantes está estudiando para un examen de Sistemas Operativos. Los estudiantes solo pueden estudiar cuando están comiendo pizza. Cada estudiante ejecuta el siguiente ciclo: mientras (verdad) {agarra un pedazo de pizza; estudia mientras come pizza}. Si un estudiante se da cuenta que se acabó la pizza, ese estudiante se duerme hasta que llegue otra pizza. El primer estudiante en descubrir que el grupo se quedó sin pizza se encarga de llamar y ordenar una nueva pizza antes de irse a dormir. Cada pizza tiene **S** pedazos. Escribir una solución que sincronice los procesos estudiantes y pizzería.

8. Se tiene una única lavadora que puede lavar 10 prendas y para aprovechar al máximo el jabón en estos tiempos de escasez, no se enciende hasta estar totalmente llena. Suponer que se tiene un proceso **L** para simular la lavadora y un conjunto de procesos para representar a cada prenda. Escriba el código que resuelva este problema de sincronización, indicando los semáforos utilizados y sus respectivos valores iniciales teniendo en cuenta los siguientes requisitos:

    - El proceso **L** invoca EstoyLista() para indicar que la ropa puede empezar a ser cargada. Un procoeso **P** invoca EntroALavadora() una vez que la lavadora está lista. No pueden ingresar dos prendas a la lavadora al mismo tiempo. Al terminar el lavado la lavadora invoca a PuedenDescargarme().
    - Cada prenda invoca SaquenmeDeAqui() una vez que la lavadora indicó que puede ser descargada y termina su proceso. Una vez vacía la lavadora espera nuevas prendas mediante EstoyLista().