

## Trabajo Práctico Nº 2

**Tema:** Introducción a Concurrencia

**Fecha Inicio:** 22/08/2024    **Fecha de Entrega:** 05/09/2024

### Actividades:

- 1) Suponga que dos tareas deben realizar las siguientes acciones:

|        |   |
|--------|---|
| Tarea1 | <i>Begin</i><br>$A = A + 100$ (I1 T1)<br>$B = B - 100$ (I2 T1)<br><i>End</i>  |
| Tarea2 | <i>Begin</i><br>$A = 1.21 * A$ (I1 T2)<br>$B = 0.5 * B$ (I2 T2)<br><i>End</i> |

Si A inicia en 50 y B inicia en 150, indique mediante una tabla el resultado de la ejecución secuencial, primero Tarea1 y luego Tarea2, y viceversa. A continuación, suponga que ambas tareas se ejecutan de manera simultánea, indique el orden de ejecución de cada instrucción teniendo en cuenta todas las combinaciones posibles de las mismas, e indique el resultado final de cada variable (A y B) al finalizar cada combinación.

- 2) Considerando el siguiente programa concurrente, calcular para él, el conjunto de los posibles valores finales para la variable x.

```

Vars x: Ent := 0;
      s1: sem := 1;
      s2: sem := 0

P1::      P2::      P3::

wait(s2)  wait(s1)  wait(s1)
wait(s1)  x := x*x    x := x+3
x := 2*x  send(s1)    send(s2)
send(s1)  send(s1)    send(s1)

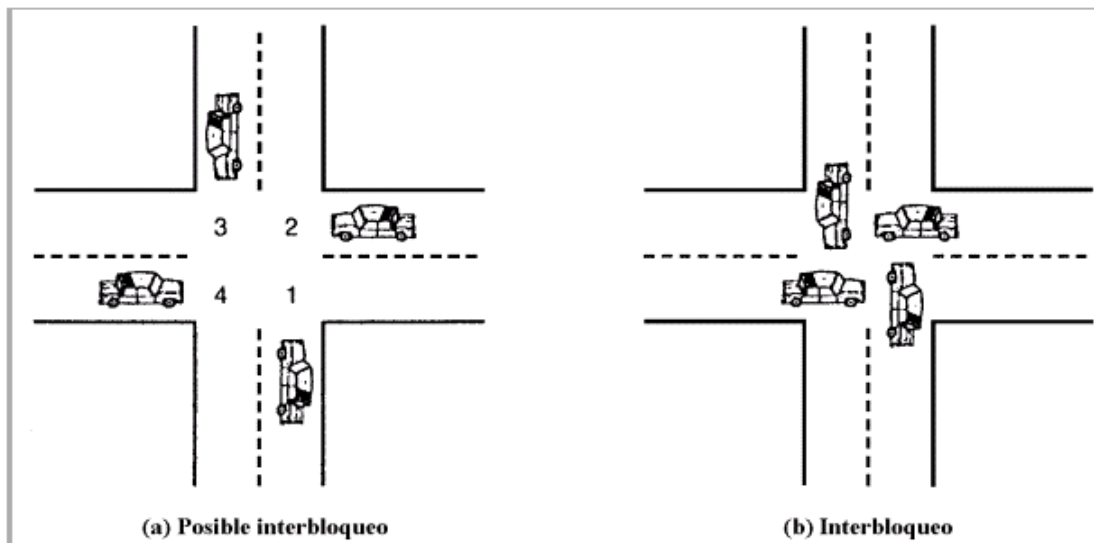
```

- 3) En la siguiente imagen, extraída de un cruce de calles, cada auto desea seguir su ruta (a), y de no tener algún mecanismo de orden o control se producirá un choque, traducido, un interbloqueo (b).

Los cuatro cuadrantes de la intersección son los recursos compartidos sobre los que se demanda control, si los cuatro coches desean atravesar el cruce, las necesidades de recursos son las siguientes:

- El coche que va hacia el norte necesita los cuadrantes 1 y 2
- El coche que va hacia el oeste necesita los cuadrantes 2 y 3
- El coche que va hacia el sur necesita los cuadrantes 3 y 4

- El coche que va hacia el este necesita los cuadrantes 4 y 1



Implemente mediante **pseudocódigo** una rutina que permita la circulación de los autos y evite el interbloqueo, para ello debe emplear semáforos como mecanismo de sincronización, y debe identificar cada auto por el sentido hacia donde desea ir (N, S, E y O).

- Investigue como y mediante qué mecanismos se implementa concurrencia en los siguientes lenguajes de programación: Java, C# y Python. Indique además los nombres de las librerías para cada lenguaje.

- Cuáles son los posibles valores de  $x$ .

```

global int y = 0 , z = 0;
thread {
  int x ;
  x = y + z ;
}

thread {
  y = 1;
  z = 2;
}
  
```

- Existen 2 procesos concurrentes donde **S** no puede ejecutar **S<sub>2</sub>** hasta que **R** haya ejecutado **R<sub>1</sub>**. ¿Cuántos semáforos deberías emplear y en que valores los deberías inicializar? ¿Dónde pondrías las operaciones **wait()** y **signal()**?

| R              | S              |
|----------------|----------------|
| R <sub>1</sub> | S <sub>1</sub> |
| R <sub>2</sub> | S <sub>2</sub> |

- Considere los siguiente hilos

```
thread {  
    print ( " R " );  
    print ( " OK " );  
}  
  
thread {  
    print ( " I " );  
    print ( " OK " );  
}  
  
thread {  
    print ( " O " );  
    print ( " OK " );  
}
```

Empleando semáforos, se debe asegurar que el resultado final de la impresión sea “R I O OK OK OK”.

- 8) Mediante semáforos y **pseudocódigo**, resuelva el siguiente problema: Una barbería posee una sala de espera con n sillas y una sala para el barbero. Si no hay clientes esperando, el barbero duerme. Si un cliente entra a la barbería y todas las sillas están ocupadas, el cliente se retira. Si el barbero está ocupado, pero hay sillas, el cliente se sienta en una de las sillas libres. Si el barbero está dormido, el cliente lo despierta. Escriba un programa para coordinar al barbero y los clientes.

### **NOTA**

Los puntos obligatorios de este TP y que serán presentados por los grupos son: Punto 3, Punto 5, Punto 6 y Punto 7.