

Tema: Sincronización de Procesos

Este práctico deberá ser realizado utilizando diversas herramientas como ser esquemas, mapas conceptuales, cronologías, etc... La autoevaluación de pares (integrantes del grupo) deberá ser subida antes de finalizar la clase de práctica

**El desarrollo del práctico deberá cumplir con los estándares de las normas APA.**

**Recuerden que deberán utilizar la cuenta institucional que les fueron asignadas al momento de la inscripción.**

1) Realice un resumen con los siguientes conceptos: procesos independientes, concurrencia, paralelismo, pseudoparalelismo, condición de carrera, sincronización, exclusión mutua, las condiciones que garantizan la sección crítica, solución por hardware, software, construcciones especiales, espera activa, comunicación entre procesos.

2) Sean los siguientes programas concurrentes:

```

VAR
a, b: semáforo;
m: integer;
PROCEDURE P1;
BEGIN
  REPEAT
    wait(a);
    wait(b);
    m:=m-1;
    write(m);
    signal(b);
    signal(a);
  UNTIL false
END;

PROCEDURE P2;
BEGIN
  REPEAT
    wait(a);
    wait(b);
    m:=m+1;
    write(m);
    signal(b);
    signal(a);
  UNTIL false
END;

(* Prog. Principal *)
BEGIN
  .....
  .....
  COBEGIN
    P1;
    P2;
  COEND
END.

```

¿Con qué valor deben inicializarse los semáforos 'a' y 'b' para garantizar que no se produzcan interbloqueos, preservando el acceso en exclusión mutua sobre la variable m?, realiza prueba de escritorio de todas las opciones. Explique los conceptos empleados para justificar la elección de la respuesta correcta.

- 0 y 1.
- 1 y 0.
- 1 y 1.
- Ninguna opción es correcta.

3) Tres procesos, P1, P2 y P3. Los procesos ejecutan el siguiente código:

Process P1	Process P2	Process P3
<b>begin</b>	<b>begin</b>	<b>begin</b>
<codigo_A>	<codigo_A>	<codigo_A>
<b>wait(X);</b>	<b>wait(Y);</b>	<b>wait(X);</b>
<seccioncritica_X>	<seccioncritica_Y>	<seccioncritica_X>
<b>signal(X);</b>	<b>signal(Y);</b>	<b>signal(X);</b>
<codigo_B>	<codigo_B>	<codigo_B>
<b>end;</b>	<b>end;</b>	<b>end;</b>

Los semáforos X e Y están inicializados a 1. El código A necesita 2 ms de tiempo para ejecutarse, el código B 4 ms y las secciones críticas 6 ms. Las operaciones **wait** y **signal** son instantáneas y no consumen tiempo. P1 comienza a ejecutarse a los 0 ms, P2 a los 4 ms y P3 a los 8 ms. Hay una única CPU y el algoritmo de planificación utilizado para determinar qué proceso se ejecuta en cada instante es el RR con Q= 2.

a) **Marcar** en el diagrama siguiente en cada instante de tiempo que parte de código (A si es código, B si es código, X si es sección crítica \_X y Y si es sección crítica \_Y) se está ejecutando del proceso correspondiente.

m	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37				
s																																										
P1																																										
P2																																										
P3																																										

4) Dados los siguientes procesos y sus respectivas secuencias de código, indica si existiría o no situación de interbloqueo y explica por qué. En cualquier caso, indica también la salida por pantalla y el valor final de los semáforos. Supón que inicialmente todos los semáforos tienen valor cero, y luego que todos los semáforos tienen valor 1, explique qué situaciones pueden presentarse. Realice las pruebas de escritorios.

<u>Proceso1</u>	<u>Proceso 2</u>	<u>Proceso 3</u>
Printf ("3");	Wait(s1);	Wait (s2);
Signal (s3);	Printf ("1");	Wait (s4);
Printf ("4");	Wait (s3);	Printf ("2");
Signal (s2);	Signal (s4);	Printf ("5");
Signal (s1);	Wait(s3);	Signal (s3);

5) Se quiere asegurar Exclusión mutua, agregue el o los semáforos que considere correctos considerando que solo se les solicita "Exclusión Mutua" y luego analice que pasa si quiero aplicar sincronización y quiero que la salida sea 90 y luego 200. Explique los conceptos aplicados.

a) Dado el siguiente código:

.....	.....
X=199	X=900
X=x+1	X=x/10
Print(X)	Print(X)
.....	.....

b) Indique las salidas.

6) Colocar los semáforos de tal manera que los procesos A, B, C se ejecuten siempre en la secuencia BACABACA.

A	B	
P(s)	P(k)	P(r)
----	----	----
----	----	----
----	----	----
V(k)	V(r)	V(s)

7) Se muestran 2 procesos concurrentes que comparte una variable global x (las restantes variables son locales a los procesos).

Sincronizar los procesos para que P1 utilice todos los valores de x suministrados por el proceso P2.

Declaración, Inicialización de Variables y Semáforos var x: entero	
P1	P2
<pre>While (TRUE) {   m = 2 * x - n;   imprimir (m); }</pre>	<pre>while (TRUE) {   leer_teclado (d);   x := d - c * 5; }</pre>