



Tema: Sincronización de Procesos

Este práctico deberá ser realizado utilizando diversas herramientas como ser esquemas, mapas conceptuales, cronologías, etc... todos acompañados por videos explicativos (pudiendo utilizar aplicaciones web como Zoom, Meet) con la participación de todo el grupo, los cuales deberán ser compartidos en el Drive.

El video no deberá superar los 20 minutos y no más de media carilla por ejercicio.

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

- 1) Ejemplifique los conceptos de Concurrencia y sus tipos, proceso cooperativo y procesos independientes.
- 2) Explique con un ejemplo Sección Crítica y los tres requisitos que debe cumplir cualquier solución para resolver un problema se Sección Crítica.
- 3) Sean dos procesos A y B, sincronizarlos para que ejecuten de manera alternada (B,A,B,A,...).
- 4) Sean los procesos A, B y C, sincronizarlos para que ejecuten de la siguiente manera: C,A,B,A,C,A,B,A,....
- 5) Sincronizar A y B de tal manera que siempre el resultado de la ejecución sea 50 y 200. Determinar si necesita uno o más semáforos y el o los valores de inicialización. Dado el siguiente segmento de código:

A	B
X = 199	X = 500
X = X + 1	X = X/10
Print (X)	Print(X)

- 6) Dadas la siguiente secuencia lógica y los valores iniciales de los semáforos, verificar si esta es la secuencia normal de ejecución y si está bien realizada la sincronización.

X	Y	Z
P(S) P(C)	P(R) P(B)	P(R)
V(R) V(S)	V(B) V(S)	V(C)

Con los valores iniciales de S=1, R=0, B=0, C=1 y cuya secuencia normal de ejecución sería XYXZXXYZ

- 7) Sean los procesos A, B y C, sincronizarlos para que ejecuten de manera alternada (A,B,C,A,B,C...).

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

- 8) Explique cómo se ejecutan los siguientes procesos, P1, P2 y P3 tienen prioridades de 1, 5 y 10, respectivamente (10 es prioridad más alta que 1). Los procesos ejecutan el siguiente código:

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

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 de *prioridades expulsivo*.

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_critica_X y Y si es sección_critica_Y) se está ejecutando del proceso correspondiente y calcule el tiempo de retorno.

ms	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
P1																																						
P2																																						
P3																																						

- 9) Sean 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.

<u>Proceso 1</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);