

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) Explique con ejemplos el concepto de Concurrencia y Procesos Concurrente.
- 2) Explique con ejemplos cuando un proceso cooperativo y cuando un proceso es independiente.
- 3) Explique con un ejemplo Sección Crítica y los tres requisitos que debe cumplir cualquier solución para resolver un problema se Sección Crítica.
- 4) 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)

- 5) Dados los siguientes procesos con variables compartidas, sincronizarlos para garantizar la mutua exclusión sobre ellas.

variables_compartidas a = b = 1;	
Proceso 0	Proceso 1
<pre>variable_local d = 1; While (TRUE){   a = a + d;   d = d * d;   b = b - d; }</pre>	<pre>variable_local e = 2; While (TRUE){   b = b * e;   e = e ^ e;   a++; }</pre>

- 6) Dada la siguiente secuencia lógica y los valores iniciales:

X	Y	Z	Q
W(B)	W(S)	W(C)	W(D)
W(T)	****	****	****
*****	****	****	****
S(C)	S(T)	S(D)	S(B)
			S(S)

Valores iniciales S =1, T=0, B=1, C=0, D=0. Explique cuál es la secuencia normal de ejecución.

- 7) Dado un sistema con N procesos del mismo programa, sincronice su código mediante semáforos para respetar el límite de tres instancias del recurso usado.

<b>Programa</b>
<pre>while (TRUE){   id_recurso = pedir_recurso();   usar_recurso(id_recurso); }</pre>

8) Dado un sistema con los siguientes tipos de procesos, sincronice su código mediante semáforos sabiendo que hay tres impresoras, dos escáners y una variable compartida.

Proceso A (n instancias)	Proceso B (n instancias)	Proceso C (n instancias)
<pre>While (TRUE){   usar_impresora();   variable_compartida++; }</pre>	<pre>While (TRUE){   variable_compartida++;   usar_escanner(); }</pre>	<pre>While (TRUE){   usar_escanner();   usar_impresora(); }</pre>

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

A	B	C
P(l)	P(m)	P(n)
_____	_____	_____
V( )	V( )	V( )

10) 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(R)	P(R)
P(C)	P(B)	_____
_____	_____	_____
_____	_____	V(C)
V(R)	V(B)	
V(S)	V(S)	

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

11) 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
<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 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 crítica \_X y Y si es sección crítica \_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																																								

12) La barbería está regentada por un peluquero, tiene una silla para pelar a los clientes y N sillas para que esperen los clientes. Si no hay clientes, el peluquero espera durmiendo a que entren en la peluquería. Cuando llega un cliente, tiene que despertar al peluquero. Si llegan más clientes mientras el peluquero está pelando a un cliente se sientan si hay sillas libres o se van si no pueden sentarse.

**Explique la solución planteada indicando los procesos involucrados, los tipos de semáforos utilizados y la función que cumplen cada uno.**

<pre> Cliente (i)  wait(em); if (clientesEsperando &lt; N ) {     clientesEsperando++;     signal(clientes);     signal(em);     wait(peluquero);     //Cortándose el pelo } else //No hay sitio para esperar     signal(em);         </pre>	<pre> Peluquero  while(1) {     wait(clientes);     wait(em);     clientesEsperando-- ;     signal (peluquero);     signal(em);     //Cortando el pelo }         </pre>
--	---