

Nombre y Apellido:..... Curso:

TEORÍA					PRÁCTICA			NOTA
1	2	3	4	5	1	2	3	

TEORÍA: Responda brevemente las siguientes preguntas. Justifique.

1. Explique el ciclo de ejecución, indique en cuál etapa son atendidas las interrupciones, y proporcione dos ejemplos de interrupciones, donde al menos una de ellas provenga del procesador.
2. ¿Sería posible que un ULT, perteneciente a un KLT, cree otro ULT, un nuevo KLT, y un nuevo Proceso? Responda justificando y realice un diagrama con todas las entidades que terminarían existiendo en el sistema, indicando las relaciones entre ellos (imagine también que existía una variable global en el proceso original).
3. Mencione cuáles planificadores inciden sobre el grado de multiprogramación y de qué manera lo hacen. ¿En qué planificador es más importante minimizar el overhead? ¿Por qué?
4. Responda Verdadero o Falso. Justifique:

a. Toda variable que es compartida por más de un proceso o hilo y en la que al menos un proceso realice modificaciones, debe ser considerada sección crítica.

b. Usar hilos de kernel previene la condición de carrera porque son provistos por el sistema operativo.
5. Explique las cuatro condiciones para que exista Deadlock. Elija dos de ellas para mencionar de qué manera se puede prevenir que ocurra Deadlock. Dé un ejemplo para cada caso.

PRÁCTICA: Resuelva los siguientes ejercicios justificando las conclusiones obtenidas.

Ejercicio 1

En un sistema que utiliza Round Robin con Q=3 se ejecutan 2 procesos. El proceso A tiene 2 hilos de usuario cuya biblioteca utiliza SJF sin desalojo. Se sabe además que en el instante 19, el proceso que esté ejecutando, realiza la llamada al sistema YIELD.

- a) Realice el diagrama GANTT según la traza de ejecución que muestra la tabla.
- b) Indique en qué instantes y por qué motivo ocurrieron interrupciones.
- c) En el caso en que la biblioteca ULT utilice jacketing ¿Cuándo comenzaría la ejecución de KLTB1?

Proceso	Hilo	Arribo	CPU	E/S	CPU	E/S	CPU	E/S	CPU
A	ULTA1	0	2	1	1	2	1	1	4
	ULTB1	0	3	2	1	-	-	-	-
B	KLTB1	3	4	2	2	-	-	-	-
	KLTB2	16	5	-	-	-	-	-	-

Ejercicio 2

Ante las continuas fallas de un sistema de preguntas y respuestas usado en clase, se decide programar el propio, modelando los problemas de concurrencia aprendidos en clase. Se dispone de varios procesos Alumno, el cual recibe una pregunta y la responde, dejando el resultado en una LISTA compartida. Los procesos Validador validan dicha respuesta, para que luego alguno de los dos procesos Ranking armen un ranking. Los rankings son de 20 respuestas, y los mismos son armados alternadamente entre los procesos encargados de ello. La lista de preguntas a validar no puede tener más de 30 elementos,

Alumno (N instancias)	Validador (N instancias)	Ranking A (1 instancia)	Ranking B (1 instancia)
P = recibir_pregunta() responder(P, LISTA)	while(true){ R = retirar(LISTA) validar(R) enviar_a_ranking() }	while(true){ armar_ranking() }	while(true){ armar_ranking() }

Sincronice los procesos utilizando semáforos sin que se produzca deadlock ni starvation.

Ejercicio 3

En un sistema que utiliza Evasión como técnica de tratamiento de Deadlocks se tienen las siguientes matrices:

Peticiónes máximas					Recursos asignados				
	R1	R2	R3	R4		R1	R2	R3	R4
P1	2	2	2	1	P1	2	1	1	1
P2	3	3	2	2	P2	1	1	1	1
P3	2	1	2	2	P3	2	0	1	2
P4	3	2	2	4	P4	0	0	2	1
P5	3	1	1	2	P5	1	1	1	0

Recursos totales			
R1	R2	R3	R4
7	5	8	6

En dicho sistema, se realizan los siguientes 3 pedidos de asignación de recursos:

1. P2: (0, 1, 1, 0)
2. P1: (0, 1, 1, 0)
3. P5: (1, 0, 0, 0)
- a) Conteste, justificando, qué decisión tomará el sistema operativo en cada pedido.
- b) Si algún pedido no pudo ser satisfecho inmediatamente, proponga un ejemplo de algún evento que podría ocurrir en el sistema para que sí lo sea. De lo contrario, proponga un nuevo pedido que no sea satisfecho inmediatamente.

Nota: Los 3 pedidos se realizan de forma sucesiva, no deben ser tratados como casos independientes.