

Nombre y Apellido:..... Curso: .....

TEORÍA					PRÁCTICA			NOTA
1	2	3	4	5	1	2	3	

TEORÍA: Responda brevemente las siguientes preguntas. Justifique.

1. Describa paso a paso qué ocurriría si un Proceso intentara escribir, por error, una variable perteneciente exclusivamente a otro Proceso. ¿Ocurriría lo mismo si fuera el Sistema Operativo quien escribiera equivocadamente en la variable de un Proceso?
2. Compare las técnicas de Prevención, Evasión y Detección de Deadlock en términos de posibilidad de ocurrencia del deadlock, overhead y limitaciones para la solicitud/asignación de recursos.
3. Considere la existencia de varios Procesos que leen y escriben una porción de memoria compartida y garantizan la mutua exclusión usando un semáforo bien inicializado y bien ubicado ¿Podrían sufrir deadlock? ¿Es posible que alguno de los Procesos sufra inanición sabiendo que el orden de desbloqueo de los Procesos en el semáforo no está garantizado?
4. Responda por V o F justificando en ambos casos.

a. Un ULT podría encontrarse en estado 'Running' mientras que su Proceso asociado está en estado 'Ready'

b. Un microkernel, en comparación con un kernel monolítico, tiene la ventaja de poder agregar nuevas funcionalidades sin la necesidad de recompilar todo el código, pero es menos tolerante a fallos y por ende más inestable.
5. Compare los algoritmos SJF sin desalojo, HRRN y VRR en términos de aging, starvation, prioridad al planificar procesos CPU Bound y prioridad para procesos E/S Bound.

PRÁCTICA: Resuelva los siguientes ejercicios justificando las conclusiones obtenidas.

**Ejercicio 1** 2 klt's de 2 procesos tienen 2 ult's que realizan pequeñas ráfagas de CPU y E/S enviando a un ROUTER un archivo zip a un servidor ftp remoto el cual, primeramente, pasa por un dispositivo "COMPR" que lo comprime. Las planificaciones de las e/s como los de los ult's son FIFO. Cada E/S la maneja el SO aprovechando que además maneja un VRR con un q=2. Si se cuenta con solo 1 procesador, se solicita que realice el Gantt correspondiente en base a las siguientes premisas:

1. Las E/S pasan por la biblioteca de hilos.
2. Las E/S no pasan por la biblioteca de hilos

PRO	HILO	T LLEG	HILO	CPU	COMPR	CPU	ROUTER	CPU
PROCESO_A	KLTA	1	ULT1	3	4	1	3	1
			ULT2	1	3	2	4	2
PROCESO_B	KLTB	0	ULT3	2	3	1	4	1
			ULT4	1	4	3	3	3

Ejercicio 2

Un señor de Corea del Sur decide inventar un sistema de criptomonedas conformado por un par de monedas relacionadas entre sí: UST y LUNA. En dicho ecosistema, los Traders (que son varios) deciden intercambiar UST por LUNA permanentemente, dado que es su estilo de transacción. Existen dos "Arbitradores", encargados de realizar la operación solicitada por el Trader. Para ello, destruyen la moneda vendida por el Trader, y crean la moneda comprada. Se sabe que no pueden existir más de 20 operaciones pendientes de ser ejecutadas, y que los dos Arbitradores se turnan para procesar las operaciones pendientes. Dado el siguiente pseudo código, sincronízelo usando semáforos respetando el enunciado establecido.

Variable compartida: Lista

Trader	Arbitrador A	Arbitrador B
<pre>while(TRUE){   Op.vender = UST   Op.comprar = LUNA   depositar(Op, Lista) }</pre>	<pre>while(TRUE){   Op = retirar(Lista)   destruir(Op.vender)   crear(Op.comprar) }</pre>	<pre>while(TRUE){   Op = retirar(Lista)   destruir(Op.vender)   crear(Op.comprar) }</pre>

Ejercicio 3

Usando la estrategia de detección y recupero:

1. Indique qué procesos están involucrados en el deadlock y cuáles no.
2. Recupere el sistema terminando la menor cantidad de procesos que menos tiempo estuvieron ejecutando en el sistema, teniendo en cuenta que los primeros procesos en comenzar a ejecutar son los de menor ID.
3. Indique algún criterio por el cual el orden de terminación de los procesos en deadlock sea diferente.

Solicitudes Pendientes

	R1	R2	R3	R4
P id=1	2	1	1	0
P id=2	1	0	0	3
P id=3	1	2	0	0
P id=4	1	1	1	1
P id=5	0	1	0	0

Recursos Asignados

	R1	R2	R3	R4
P id=1	1	1	1	1
P id=2	1	0	2	0
P id=3	0	0	0	2
P id=4	0	0	0	0
P id=5	0	1	0	0

Recursos Totales

R1	R2	R3	R4
3	3	3	3