

Nombre y Apellido: Curso:

TEORÍA					PRÁCTICA			NOTA
1	2	3	4	5	1	2	3	

TEORÍA: Responda brevemente las siguientes preguntas. Justifique.

1. ¿En qué consiste el ciclo de ejecución de una instrucción? ¿Podría una interrupción surgir como consecuencia de dicho ciclo? (en caso afirmativo brinde un ejemplo)
2. ¿Qué comparten un proceso padre y un proceso hijo? ¿Y en el caso de dos hilos del mismo proceso?
3. Compare los algoritmos FIFO, Round Robin y SJF en términos de Equidad, Overhead y Starvation.
4. Mencione los requisitos mínimos y deseables para la correcta aplicación de la mutua exclusión. Indique al menos dos ventajas del uso de semáforos por sobre soluciones de software
5. Explique las condiciones necesarias y suficientes para que ocurra interbloqueo.

PRÁCTICA: Resuelva los siguientes ejercicios justificando las conclusiones obtenidas.

Ejercicio 1

Se dispone de un sistema operativo con una biblioteca de ULTs que planifican round robin con q = 2, y un planificador a nivel S.O. algoritmo SJF con desalojo. Dada la siguiente traza de ejecución:

	Llegada	Cpu	i/o	Cpu
Proceso 1 – ULTA	0	6	-	-
Proceso 2 – ULTB	2	3	4	3
Proceso 3 – ULTC	3	1	2	2
Proceso 4 – ULTD	6	2	-	-

- a) Realice el diagrama de GANTT
- b) Indique los instantes en los cuales intervino el planificador de corto plazo y/o el de largo plazo
- c) Considerando el diagrama de 7 estados posibles de un proceso, indique todos los estados que necesariamente fueron utilizados, y cuáles no (justifique cada caso)

Ejercicio 2

Desarrolle en pseudo-código las siguientes funciones de semáforos:

Prototipo función	Descripción
<code>void wait(sem_t semaforo, int qty);</code>	Una variante de <i>wait</i> donde se puede especificar una cantidad al modificar el semáforo
<code>void signal(sem_t semaforo, int qty);</code>	Una variante de <i>signal</i> donde se puede especificar una cantidad al modificar el semáforo
<code>int try_wait(sem_t semaforo, int qty);</code>	Una variante no bloqueante de <i>wait</i> donde se puede especificar una cantidad al modificar el semáforo

Nota: recuerde utilizar algún mecanismo que evite condiciones de carrera en el uso del semáforo en sí mismo. Asimismo, la variante no bloqueante debe usar el valor de retorno para indicar cualquier problema ocurrido.

Ejercicio 3

Recursos Asignados

	R1	R2	R3	R4
P1	1	1	2	0
P2	2	1	0	1
P3	1	0	0	1
P4	1	0	1	1
P5	0	0	0	0

Peticiones Actuales

	R1	R2	R3	R4
P1	3	2	2	2
P2	3	1	0	0
P3	3	3	3	2
P4	1	1	1	0
P5	1	1	1	1

Recursos Totales

R1	R2	R3	R4
5	3	4	3

- a) Dadas las matrices, detecte que procesos se encuentran en Deadlock. Justifique.
- b) Siendo P3 un proceso con alta prioridad y que P1, P2, P4 y P5 tienen baja prioridad pero la misma entre ellos. Encuentre el mejor y peor orden de finalización de procesos para recuperarse del Deadlock, teniendo en cuenta que el proceso 3 debe finalizar correctamente.

Condiciones de aprobación: 3 preguntas correctamente respondidas y 1,5 ejercicios correctamente resueltos.