

Nombre y Apellido:..... Curso:

TEORÍA					PRÁCTICA			NOTA
1	2	3	4	5	1	2	3	

TEORÍA: Responda brevemente las siguientes preguntas. Justifique.

1. Compare los algoritmos RR, VRR y HRRN en términos de Overhead, Monopolización de la CPU y Starvation.
2. ¿Quiénes pueden crear o finalizar Procesos en un sistema? ¿Cómo lo hacen? ¿Qué le sucede al proceso hijo si su padre finaliza inesperadamente?
3. Describa las condiciones necesarias y suficientes para la existencia de un Deadlock. Indique las desventajas que tiene la estrategia de detección y recupero de Deadlock.
4. V o F
 - a. El hecho de utilizar un semáforo implica un cambio de modo de ejecución.
 - b. El planificador de corto plazo según sus decisiones modifica el grado de multiprogramación del sistema.
5. Describa *brevemente* qué ocurre cuando se invoca una *syscall* desde un proceso en modo usuario y cómo se atiende la misma.

PRÁCTICA: Resuelva los siguientes ejercicios justificando las conclusiones obtenidas.

Ejercicio 1

Un sistema corre dos instancias del mismo proceso, utilizando el algoritmo Round Robin, con Q=2. Cuando es necesario, se utiliza una biblioteca que planifica hilos utilizando SJF (sin desalojo), y la misma permite replanificar luego de una I/O. El proceso mencionado se compone de dos hilos que ejecutan como lo indica la tabla:

	Arribo	CPU	I/O	CPU
Hilo A	0	3	1	1
Hilo B	0	2	2	3

- a) Grafique la ejecución mediante un diagrama de Gantt si: los hilos son ULTs, la primera instancia del proceso se crea en el instante 0 y la segunda en el instante 1.
- b) Indique a partir de qué instante cambiaría el diagrama del punto A si:
 - b1) La biblioteca planifica los hilos utilizando FIFO.
 - b2) Si se utilizara la técnica Jacketing

Ejercicio 2

Un conjunto de científicos decide modelar la melodía de una canción perteneciente a una popular banda de rock argentino, con el objetivo de entender la magnitud de su calidad musical. Para ello, desarrollan un proceso que consta de cuatro hilos, donde cada uno toca una nota de dicha melodía (a veces repetidas veces) simulando el sonido de un saxo. La melodía que se pretende modelar es: RE#, RE#, RE#, RE#, DO#, DO#, SOL, SOL#.

Saxo Tenor	Saxo Alto	Saxo Soprano	Saxo Sopranino
<pre>while (true) { tocar (DO#) tocar (DO#) }</pre>	<pre>while (true) { tocar (RE#) }</pre>	<pre>while (true) { tocar (SOL) }</pre>	<pre>while (true) { tocar (SOL#) }</pre>

- a) Sincronice los hilos usando exclusivamente semáforos, para lograr que la melodía deseada se reproduzca infinitamente, sin que se produzca deadlock ni starvation.
- b) Si se desea poder tener un número indefinido de procesos (cada uno con sus cuatro hilos) que toquen concurrentemente la melodía, pero limitando cuántos de ellos pueden realizarla al mismo tiempo ¿Cuáles cambios realizaría al punto anterior?
Asuma que inicialmente solo se desea que hasta 3 procesos toquen la melodía, pero se pretende que ese número pueda incrementarse/decrementarse mediante un proceso extra.

Ejercicio 3

Se dispone de un sistema operativo en el cual la solicitud de recursos constituye una operación bloqueante cuando el recurso no está disponible, y donde la mutua exclusión sobre los recursos es un requisito que no se puede modificar. Dado el siguiente pseudo-código:

Proceso 1	Proceso 2	Proceso 3
<pre>pedir_recurso (B) pedir_recurso (A) pedir_recurso (C) usar_recursos (A, B, C) liberar_recursos ()</pre>	<pre>pedir_recurso (C) pedir_recurso (B) pedir_recurso (A) usar_recursos (A, B, C) liberar_recursos ()</pre>	<pre>pedir_recurso (A) pedir_recurso (C) pedir_recurso (B) usar_recursos (A, B, C) liberar_recursos ()</pre>

- a) Indique una secuencia de ejecución bajo la cual se produzca un deadlock que involucre a todos los procesos, y realice un grafo de asignación de recursos.
- b) Modifique el código para garantizar que no ocurra deadlock, proveyendo al menos dos soluciones distintas basadas en estrategias de Prevención.

Condiciones de aprobación: 3 preguntas correctamente respondidas y 1.5 ejercicios correctamente resueltos.