

Nombre y Apellido:..... Curso:

TEORÍA					PRÁCTICA			NOTA
1	2	3	4	5	1	2	3	

TEORÍA: Responda brevemente las siguientes preguntas. Justifique.

1. ¿Cómo se relacionan las interrupciones con los cambios de contexto y cambios de modo? ¿Podría ocurrir un cambio de contexto sin cambio de modo? Ejemplifique.
2. Luciano posee un servidor para que su alumnado de SisOp pueda cargar consultas. El mismo cuenta con 8 CPUs y se desea que atienda peticiones de forma concurrente priorizando que ante una eventual falla no deje de atender peticiones y pueda recuperarse. ¿Crearía procesos, hilos (y de qué tipo) o una combinación de estos para atender las peticiones? Justifique. Mencione alguna desventaja que pueda identificar de su propuesta.
3. Compare los algoritmos de “FIFO”, “Prioridades (con desalojo)” y “Round Robin” en términos de: overhead, starvation, posibilidad de que algún proceso monopolice la CPU.
4. Responda Verdadero o Falso, justificando en ambos casos:

a. Si tenemos 2 o más procesos accediendo a un recurso en común en forma concurrente se considera una región crítica y debemos sincronizarlo.

b. Puede ser más performante utilizar semáforos implementados con espera activa que con bloqueo si la región crítica es muy corta.
5. Proponga un pseudocódigo en que pueda generarse un deadlock y muestre un orden de ejecución en que se genere deadlock y otro en el que no.

PRÁCTICA: Resuelva los siguientes ejercicios justificando las conclusiones obtenidas.

Ejercicio 1

En un sistema que utiliza Virtual Round Robin con Q=3 se ejecutan 2 KLT que utilizan 2 dispositivos de entrada/salida, el KLT A tiene dos hilos ULT que utilizan las funciones brindadas por su biblioteca para usar los dispositivos de entrada/salida, mientras que el KLT B tiene otros 2 hilos ULT que lo hacen llamando syscalls de forma directa, ambas bibliotecas de hilos planifican con FIFO.

- a) Realice el diagrama GANTT según la traza de ejecución que muestra la tabla.
- b) ¿En qué instantes interviene el SO y por qué motivos lo hace? Identifíquelos.
- c) Si las bibliotecas de ULT implementaran jacketing, ¿En qué instante cambiaría la planificación?

Proceso	Hilo	Arribo	CPU	Router (I/O)	CPU	Disco (I/O)	CPU
PA/KLT A	ULT A1	1	4	4	1	1	1
	ULT A2	0	2	1	3	3	1
PB/KLT B	ULT B1	3	2	1	2	2	1
	ULT B2	4	3	1	4	2	1

Ejercicio 2

Luciano pone en práctica sus conocimientos de hilos y desarrolla un pseudocódigo en el cual se desea que los hilos se creen en orden ascendente (del 1 al 4), que la variable “contador” llegue (y no supere) a 100, luego se imprima el mensaje “Contador llegó a 100” y por último finalice el programa. Sincronice el mismo utilizando solamente semáforos, eliminando condiciones de carrera y sin generar deadlocks ni starvation.

hilo_main:
crear_hilo(hilo_1);
crear_hilo(hilo_3);
finalizar_programa();

hilo_2:
while(1){
 contador++;
}

hilo_3:
printf("Contador llegó a 100");
//termina el hilo

hilo_1:
crear_hilo(hilo_2);
crear_hilo(hilo_4);
//termina el hilo

hilo_4:
while(1){
 contador++;
}

Notas:

- “contador” es una variable entera global inicializada en 0.
- Se considera que un hilo es creado en el momento que se ejecuta la función crear_hilo.

Ejercicio 3

Un sistema capaz de detectar y recuperarse de Deadlock se encuentra en la siguiente situación:

RECURSOS ASIGNADOS					SOLICITUDES ACTUALES					RECURSOS TOTALES			
	R1	R2	R3	R4		R1	R2	R3	R4				
P1	2	1	2	1	P1	1	3	3	2	R1	R2	R3	R4
P2	0	2	2	1	P2	0	1	0	1				
P3	0	0	0	0	P3	0	1	0	3				
P4	3	0	2	1	P4	1	3	5	2				
P5	1	2	0	2	P5	3	0	3	0				
										6	6	6	6

- a) Determine si existe un deadlock y qué procesos están involucrados. ¿Cómo lo solucionaría? Sabiendo que en este sistema se decide finalizar el proceso con mayor cantidad de recursos asignados para evitar finalizar varios procesos.
- b) Si el sistema tuviese una instancia más de R1 y R3, ¿Se presentaría la misma situación que el punto anterior?