

Nombre y Apellido:..... Curso:

TEORÍA					PRÁCTICA			NOTA
1	2	3	4	5	1	2	3	

TEORÍA: Responda brevemente las siguientes preguntas. Justifique.

- ¿Cuántos cambios de modo implica una función “wrapper”, como mínimo? ¿La cantidad total de cambios de modo de qué depende?
- Indique si es verdadera o falsa la siguiente afirmación, justificando en ambos casos: Atender una interrupción anidada implica un cambio de contexto y un cambio de modo.
- Mencione al menos tres algoritmos de corto plazo que prioricen de alguna forma procesos con ráfagas cortas. ¿Qué forma tiene el sistema operativo de conocer la próxima ráfaga de un proceso para aplicar alguno de estos algoritmos?
- Describa brevemente las estrategias o soluciones para lograr la mutua exclusión, indicando cuáles generan espera activa.
- En un sistema que controla un respirador artificial, ¿cuáles podrían ser estrategias válidas y cuáles no para el tratamiento de deadlocks? ¿Influiría en algo si el sistema no pudiera tolerar mucho overhead?

PRÁCTICA: Resuelva los siguientes ejercicios justificando las conclusiones obtenidas.

Ejercicio 1

En un SO que utiliza Round Robin con Q=4 se ejecutan 2 procesos. Uno de ellos implementa una biblioteca de ULTs que utiliza SJF con desalojo como algoritmo de planificación y maneja sus I/O. Dada la siguiente traza:

Procesos	Hilos	Arribo	CPU	I/O	CPU
PA	ULTA1	0	3	3	1
PA	ULTA2	2	1	2	2
PB	KLTB1	3	3	2	1
PB	KLTB2	6	2	4	2

- Realice el diagrama Gantt.
- Indique en qué instante cambiaría el diagrama del punto a) si la biblioteca de los ULTs utilizara Jacketing para sus I/O.
- Indique en qué instante cambiaría el diagrama del punto a) si ambos procesos usaran ULTs.

Ejercicio 2

Luciano trabaja en el área de sistemas de una empresa de envíos de mercadería y necesita automatizar el proceso de despacho de cajas desde el almacén hasta los camiones autónomos mediante robots, representados en pseudocódigo.

El proceso consiste en que los robots del almacén tomen las cajas a despachar y las dejen en una plataforma compartida con los robots distribuidores. Cada uno de estos deberá tomar estas cajas de la plataforma y las deberá llevar al camión correspondiente al destino de esa caja. Cada camión saldrá una vez que todas las cajas con ese destino estén cargadas.

Notas:

- La plataforma que comparten almacén y distribución tiene capacidad para 5 cajas.
- Cada caja tiene un atributo “destino” que contiene el id del camión correspondiente al mismo.
- Camión tienen una función “get_id()” que devuelve su propio id.
- Hay un total de 100 cajas a despachar, exactamente 25 para cada destino.

Robot de almacén (N instancias)	Robot distribuidor (M instancias)	Camión (4 instancias)
while(1) { caja= tomar_caja() depositar(caja, plataforma) }	while(1) { caja = retirar(plataforma) despachar(caja, caja.destino) }	salir()

Sincronice los procesos utilizando semáforos para que todos los productos sean enviados correctamente.

Ejercicio 3

Un sistema utiliza la técnica de evasión de deadlocks. Se sabe que el estado del sistema en un momento es el siguiente:

Peticiones máximas					Asignaciones				
	R1	R2	R3	R4		R1	R2	R3	R4
P1	2	2	0	1	P1	1	0	0	1
P2	1	0	3	4	P2	0	0	2	0
P3	1	2	4	1	P3	0	0	0	1
P4	0	0	2	0	P4	0	0	1	0

- ¿Cuál debería ser el vector mínimo de recursos totales, considerando que se debe poder satisfacer la nueva petición de P1 (1,1,0,0)? **Nota:** se sabe que el vector no puede ser menor a (2,2,4,4)
- Utilizando el vector definido en 1) y las matrices iniciales del estado del sistema proponga una petición del proceso P3 que, respetando sus peticiones máximas y aunque tenga los recursos disponibles para asignarlos, el SO no los otorgaría en dicho momento.

