

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

TEORÍA					PRÁCTICA			NOTA
1	2	3	4	5	1	2	3	

**TEORÍA:** Responda brevemente las siguientes preguntas. Justifique.

- ¿Cuál es la diferencia entre una Syscall y una función Wrapper? ¿En qué caso utilizaría cada una y por qué?
- ¿En qué se diferencian las técnicas de prevención y los de detección de deadlock? ¿En qué casos utilizaría cada estrategia?
- Definir el concepto de sección crítica. ¿Qué condiciones debe cumplir? Mostrar un ejemplo del uso de soluciones hardware dentro de las primitivas wait/signal.
- Verdadero o falso:
  - Utilizando semáforos en hilos ULT, no requieren realizar un cambio de modo para ejecutar las operaciones de wait/signal.
  - Para compartir memoria entre procesos o entre KLTs se necesita intervención del SO. Entre ULTs no es necesario, debido a que se gestionan en espacio de usuario.
- Los algoritmos de planificación con desalojo ¿Qué eventos tienen en cuenta para la re-planificación? ¿Por qué los algoritmos sin desalojo no?

**PRÁCTICA:** Resuelva los siguientes ejercicios justificando las conclusiones obtenidas.

**Ejercicio 1**

Peter quiere tomarse unos buenos mates. Para ello, saca yerba de un frasco de la cocina, la coloca en su despolvillador homologado, y luego de agitarlo sirve la yerba limpia en el mate. Otros compañer@s toman, cada uno con su propio mate, pero todos comparten el despolvillador. Dado que Peter usa yerba premium, tiene su propio frasco, mientras que el resto comparte otro. Si los frascos de la cocina se vacían, alguien los llena (siempre ambos a la vez). Mientras los frascos están vacíos nadie debería tomarlos, y cada frasco rinde 20 recargas. Los frascos empiezan llenos.

Peter (1 instancia)	Llenador (1 instancia)	Compañer@ matero (N instancias)
<pre>while (TRUE) {   yerba = tomarDe(frascoPremium);   yerbaLimpia = usar(despol, yerba);   mate = cargarCon(yerbaLimpia);   disfrutarUnosVerdes(mate); }</pre>	<pre>while (TRUE) {   llenar(frascoPremium);   llenar(frasco); }</pre>	<pre>while (TRUE) {   yerba = tomarDe(frasco);   yerbaLimpia = usar(despol, yerba);   mate = cargarCon(yerbaLimpia);   saborearUnosVerdes(mate); }</pre>

Teniendo en cuenta el pseudocódigo presentado, sincronice utilizando únicamente con semáforos.

**Ejercicio 2**

Sabiendo que P1 llega en t=0 y P2 en t=1, y dado el siguiente diagrama de gantt:

P1 – KLT1			IO	IO					F				
P1 – KLT2					IO					F			
P2 – KLT3						IO							F

- ¿Qué algoritmos pueden haber sido utilizados para la planificación?
- Indique en qué instantes de tiempo se produjeron context y/o mode switches.
- Si en t=2 se ingresa un proceso nuevo en el sistema, con sus ráfagas: CPU 1, IO 1, CPU 2. Realice el nuevo diagrama de gantt utilizando alguno de los algoritmos encontrados en la primer parte.

**Ejercicio 3**

En un sistema que utiliza la estrategia de detección y recupero de deadlock, se disponen de recursos cuya disponibilidad actual es: RA = RB = RD = 0; RC = RE = 1; y por otro lado el estado actual es:

	RA	RB	RC	RD	RE
P1	0	0	0	2	0
P2	1	1	0	0	0
P3	0	0	1	0	0
P4	0	0	0	0	4
P5	0	0	0	0	1
P6	0	0	0	0	0

Recursos asignados

	RA	RB	RC	RD	RE
P1	1	0	0	0	0
P2	0	0	2	1	0
P3	0	1	0	0	0
P4	0	0	0	0	0
P5	0	0	0	0	2
P6	0	1	0	0	0

Solicitudes pendientes

- Indique si existe actualmente un deadlock en el sistema, indicando los procesos involucrados en el mismo.
- Realice un grafo de asignación de recursos. En base al mismo y la información obtenida del punto anterior, indique qué procesos están bloqueados (y su motivo) y cuáles no.
- Recupere el sistema eliminando la existencia de deadlocks, sabiendo que la estrategia es eliminar procesos, solo se puede eliminar uno, y el orden de selección de la víctima está determinado por la prioridad (prioridad(P6) > prioridad (P5) > prioridad (P4), etc))

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

