



Nota:

Apellido y Nombre				Profesor		Tomé conocimiento de la nota: (Sólo aplazos)	

Preguntas teóricas					Ejercicios	
1	2	3	4	5	1	2

A) **Teoría:** Explícitamente defina como **VERDADERA** o **FALSA** cada una de estas afirmaciones justificando brevemente.

- 1) Las instrucciones privilegiadas sólo pueden ser ejecutadas por el sistema operativo.
- 2) Los permisos que indican qué puede hacer un Hilo (en términos de operaciones sobre archivos) son iguales para todos los Hilos de ese mismo Proceso.
- 3) Cuando no se utiliza Memoria Virtual, no es necesario el acceso a la TLB debido que no se producirán Page Fault.
- 4) De los tres mecanismos de asignación de datos a archivos (continua, enlazada e indexada), el mecanismo indexado es el que más espacio ocupa en el FCB
- 5) Un proceso cuyos hilos sufren condición de carrera es menos propenso a que los mismos sufran deadlock entre sí.

B) **Práctica:** Resuelva los ejercicios justificando las respuestas

- 1)
Si se tiene un disco rígido de 4 TiB, y se desea formatear con FAT32:

a) ¿Cuál sería el tamaño mínimo de cluster para poder direccionar el disco? (descartando el espacio ocupado por la información administrativa del filesystem)

b) Si en este esquema se almacena un archivo de 491521 bytes, ¿Qué espacio en disco ocuparía?

c) ¿Qué tamaño tendría la FAT si se usara el mismo tamaño de cluster para formatear un pendrive de 1 TiB?

d) Grafique las entradas de directorio, un extracto de la fat y de los bloques del volumen para explicitar la existencia de dos archivos, uno de 10 KiB y otro de 20 KiB.
- 2)
En un sistema que soporta semáforos con bloqueo ejecuta actualmente un proceso con 2 hilos, teniendo en cuenta la siguiente porción de pseudocódigo:

Hilo 1	Hilo 2
<code>uint32_t valor = cálculo() wait(sem1) wait(mutex) acumulador += valor signal(mutex) signal(sem2)</code>	<code>uint32_t valor = cálculo() wait(mutex) wait(sem2) acumulador += valor signal(mutex) signal(sem1)</code>
<code>sem1=1 ; sem2=0 ; mutex=1</code>	

Nota: “acumulador” es una variable global del Proceso

- a) Encuentre un orden de ejecución de los mismos donde se produzca un Deadlock. ¿Qué algoritmo de planificación podría estar siendo usado para que ello ocurra?
- b) Sabiendo que los semáforos utilizados son propios del Proceso, de generarse un Deadlock ¿Tendría algún impacto sobre otros Procesos del sistema? ¿Cambiaría su respuesta si los semáforos estuviesen implementados sin bloqueo?
- c) Proponga un cambio en el pseudocódigo para que ya no pueda ocurrir Deadlock.



Resolución

Teoría

- 1) Verdadero: Las mismas suelen operar sobre el hardware o ser disruptivas si pudieran ser usadas libremente por los procesos de usuario, los mismos solamente acceden a ellas a través de las syscalls dispuestas por el Sistema Operativo.
- 2) Verdadero: Los permisos sobre archivos abiertos son a nivel Proceso ya que el FS normalmente maneja una tabla de archivos abiertos por Proceso.
- 3) Falso: El uso de la TLB es útil en sistemas con y sin memoria virtual para reducir el tiempo en la traducción de direcciones.
- 4) Falso: Al igual que los otros mecanismos, el FCB solamente necesita tener un puntero a bloque, en este caso al bloque índice.
- 5) Verdadero: Al no haber mutua exclusión entre los hilos de ese proceso, no se cumpliría una de las condiciones necesarias para que ocurra deadlock.

Práctica

- 1)
 - a) $2^{42} = 2^{28} * 2^x \Rightarrow x = 14 \Rightarrow$ Tamaño de cluster = 16 KiB
 - b) $491521 \text{ B} / 16 \text{ KiB} = 30 \text{ clusters} + 1 \text{ Byte} \Rightarrow$ Necesito 31 clusters $\Rightarrow 16 \text{ KiB} * 31 = 507.904 \text{ Bytes}$
 - c) Solamente necesito 26 bits para direccionar 1 TiB $\Rightarrow 2^{26} * 4 \text{ Bytes} = 256 \text{ MiB}$
 - d)
archivo1.bin 10240 0
archivo2.bin 20480 1
FAT: [EOF, 2, EOF,.....]
Bloques: [dataArchivo1, dataArchivo2, dataArchivo2,]

- 2)
 - a) Si ejecuta primero el Hilo 2 con un algoritmo FIFO, el mismo tomaría el mutex y luego se bloquearía por sem2, esto haría que luego el Hilo 1 no pueda avanzar al estar el mutex tomado. Lo mismo podría ocurrir en un algoritmo RR si ejecutan alternadamente de a una instrucción.
 - b) No debería impactar en otros Procesos del sistema ya que los semáforos retenidos son usados solamente por ese Proceso. Si los mismos estuviesen implementados con espera activa, esto sería un Livelock y afectaría al resto de los Procesos ya que estaría ocupando tiempo de ejecución en CPU.
 - c) **wait(mutex)** y **wait(sem2)** deberían estar al revés para que el mutex encierre solamente la sección crítica. Otra opción podría ser remover el semáforo mutex por completo, ya que en este caso, el orden de ejecución ya estaría garantizado por los semáforos binarios sem1 y sem2.