



Teoría:

- 1) Falso. Si el de mediano plazo ejecuta relativamente poco es porque hay poco swapping. Entonces, no debería haber thrashing, sino procesos que empiezan y terminan sin problemas.
- 2) Falso. Si se los comunica usando paso de mensajes, ambos envían un mensaje y ambos quedan bloqueados esperando otro mensaje (por ej un OK del otro), es posible que queden en deadlock (de hecho es un error de algunos tps, cuando recién arrancan). También se da mucho en DBs distribuidas, aunque excede los ejemplos que vemos habitualmente.
- 3) Verdadero. Cuando los niveles se incrementan, también lo hacen los accesos a tablas de páginas. Por ende, tener una TLB con un buen "hit ratio" evita cada vez más accesos.
- 4) Falso. Implica chequear que haya instancias de ese recurso, que no se pase del máximo que puede pedir ese proceso, luego simular otorgarle el recurso y chequear estado seguro y recién ahí otorgarlo.
- 5) Falso. La primera afirmación es correcta ya que teniendo la FAT en RAM, sólo se acceden a bloques de datos, mientras que en EXT se podría requerir acceder a algún bloque de PTR. Sin embargo, la segunda afirmación es incorrecta ya que FAT utiliza una asignación de bloques encadenada/enlazada (no hay un índice por cada archivo)

Practica:

1. SO: el algoritmo el Feedback con una cola de mayor prioridad con RR Q = 3 y una de menor prioridad con FIFO.
 - Se puede ver que en inst 13 KB es desalojado luego de haber ejecutado 3 unidades de CPU por más que aún no se había bloqueado.
 - En inst 18 KB que estaba ejecutando en modo FIFO es desalojado por KA que se encuentra en la cola de mayor prioridad RR:

Biblioteca de hilos de usuarios: SJF.

- En inst 0 se elige UA1 que tiene la ráfaga más corta (1 frente a 2 y 3)
- En inst 15 se elige UA3 en lugar de UA2 (2 frente a 3)

SO: el algoritmo el Feedback con una cola de mayor prioridad con RR Q = 3 y una de menor prioridad con FIFO.

Se puede ver que en inst 13 KB es desalojado luego de haber ejecutado 3 unidades de CPU por más que aún no se había bloqueado.

En inst 18 KB que estaba ejecutando en modo FIFO es desalojado por KA que se encuentra en la cola de mayor prioridad RR:

Biblioteca de hilos de usuarios: SJF.

En inst 0 se elige UA1 que tiene la ráfaga más corta (1 frente a 2 y 3)

En inst 15 se elige UA3 en lugar de UA2 (2 frente a 3)

2. a)

- Identificar: root.conf es un SL de peter.conf. / guest.conf y peter.conf son HL
- Renombrar: peter.conf pasa a ser peter.old. Los HL siguen funcionando. El SL se rompe.
- Borrar peter.conf: el SL se rompe. Todavía se puede acceder al archivo desde guest.conf
- Ejecutar main.sh: No tiene permisos de ejecución

2. b)

50 -> Pista 0 /// 180 -> Pista 1 /// 950 -> Pista 9 /// 20 -> Pista 0

Arranco en la pista 6

6 -> 9 -> 1 -> 0 = 12 pistas

Dentro de los algoritmos sin inanición tenemos FIFO, FSCAN y NSTEP SCAN. Los dos de la familia SCAN van hasta al tope, por lo que para este caso en particular, el más rápido sería FIFO:

Con FIFO: 6 -> 0 -> 1 -> 9 -> 0 = 24 pistas (Los otros, irían de 6-> 9 -> TOPE (1023))

El tiempo de duración del examen final será de 90' a contar desde el momento de comienzo del mismo. Si el alumno por algún motivo comenzara más tarde sólo podrá utilizar el tiempo remanente. Utilice hojas separadas para la teoría / ejercicios.