1) Una vez iniciado un proceso, todos los elementos de su imagen pueden ser llevados al área de swap (total o parcialmente) si el algoritmo de reemplazo así lo determina

Falso. El PCB nunca puede llevarse al área de swap dado que es la estructura de administración del proceso y nunca se divide en páginas

2) En un programa con altos requisitos de velocidad, reducir la cantidad de llamadas al sistema (reemplazando la lógica necesaria por código usuario cuando sea posible) es una buena elección

Verdadero, reduciria la cantidad de mode switches y cambios de contexto que implica pasar a modo kernel.

3) Copiar un conjunto de archivos comprimidos de un volumen a otro siempre será más lento que moverlos a una carpeta, luego

comprimir la misma, y finalmente copiar dicha carpeta al otro volumen

Verdadero. Si bien la cantidad de bytes transferidos de un volumen a otro sean aproximadamente los mismos, leer un solo archivo

en el volumen origen y escribir un solo archivo en el volumen destino, las operaciones administrativas (crear entradas de directorio,

alocar bloques, etc) se reducen al mínimo si se compara con realizar la misma tarea con muchos archivos.

4) Un sistema en el cual los procesos deben solicitar todos los recursos que van a utilizar al iniciarse, es un sistema en el cual nunca ocurrirá un interbloqueo

Verdadero. Se describe una forma de impedir que ocurra el interbloqueo en la estrategia de Prevención, donde se impide una de las

cuatro condiciones (retención y espera)

5) La corrupción total de la tabla de páginas de un proceso no sería un impedimento para que el mismo siga ejecutando, siempre y cuando el sistema operativo utilice memoria virtual, el proceso no haya escrito páginas y se disponga de espacio libre en memoria ram

Verdadero. El sistema operativo podría frenar la ejecución del proceso, crear una nueva tabla de páginas y cargar algunas paginas

en nuevos frames. Es necesario sobrante de memoria ram porque las páginas cargadas previamente serían imposibles de detectar y

por ende imposibles de liberar (a menos que se realice un barrido completo de todas las tablas de páginas de los procesos)

6) En un sistema con memoria virtual en estado ocioso, incrementar el nivel de multiprogramación siempre permite subir el porcentaje

de uso de la CPU.

Falso. Cuando hay thrashing, aumentar el grado de multiprogramación empeora la situación.

7) Si se copian archivos de un fs de tipo unix a otro de tipo FAT, inevitablemente se perderá cierto tipo de meta-información.

Verdadero. Los filesystem de tipo unix almacenan gran cantidad de metadata en lo que se denomina inodos (cada archivo se corresponde con un inodo). Por otro lado, FAT no esta diseñado para albergar metadata.

8) Podría ocurrir starvation entre distintos hilos que utilizan un mismo semáforo.

Verdadero. Por una mala programación podría ocurrir que un hilo se bloquee haciendo wait de un semaforo del cual nunca se hace signal.

9) Un programa compilado para correr como un proceso estándar con instrucciones privilegiadas, nunca podría ejecutarlas exitosamente.

Verdadero. Nunca podría ejecutarlas sin utilizar los servicios que brinda el so (syscalls)

10) Un sistema con segmentación paginada no sufre de fragmentación externa.

Verdadero, los segmentos se hallan divididos en particiones fijas (paginas) eliminando de esta forma la fragmentacion externa y la necesidad de compactacion

12) Una llamada al sistema bloqueante, realizada por un KLT, podría hacer que el resto de los KLTs del mismo proceso no puedan continuar su ejecución

Verdadero, si todos los klt en cuestion comparten un recurso, uno de ellos hace una syscall bloqueante para acceder al recurso y nunca lo libera por una mala programacion, si los otros aun necesitan el recurso no van a poder continuar su ejecucion.

13) Además de semáforos, es posible garantizar el orden de los procesos utilizando otras herramientas del Sistema Operativo.

Verdadero, se puden utilizar tecnicas de IPC (como envio de mensajes)

14) Todos los algoritmos de planificación, que no se basan en el orden de llegada a "listos" para elegir el siguiente proceso, presentan inanición.

Verdadero dado que pueden haber procesos en la cola de listos que nunca vayan a ser atendidos por la llegada de nuevos procesos que son elegidos basados en el sistema de eleccion del algoritmo.

15) Es posible que un proceso entre en thrashing, siendo el único en ejecución en un determinado momento

Verdadero si está dividido en paginas muy chicas va a estar mas tiempo swapeando que ejecutando el proceso.

16) Una entrada-salida bloqueante puede ser realizada como no bloqueante, aunque esto carece de utilidad

F. Tiene mucha utildiad Por ejemplo, cuando se usan ULTs con jacketing , la biblioteca de ULTs puede elegir otro hilo de usuario para ejecutar que ya estaba listo.

17) Al poner a correr una gran cantidad de procesos IO Bound concurrentemente, descubre que la performance de los procesos CPU bound que ya estaban en ejecución baja notablemente, pese a que sus Entrada-Salida se realizan utilizando DMA. Esto podría ocurrir porque la técnica de DMA roba ciclos.

V. Roba ciclos de bus

18) La tabla de páginas invertidas no permite compartir memoria a los procesos

V. Porque en la definición original de la estructura de la tabla, solo se tiene # página, pid, bits de control y la direccion del proximo chain (si es que tiene); por lo tanto no hay forma de asociar dos o más procesos a un mismo frame.

19) Los ULTs de un mismo proceso pueden quedar en deadlock utilizando semáforos

F: si se bloquea un hilo, se bloquean los otros y por lo tanto no llega a formarse la espera circular.

20) Para una partición de swap es fundamental que su File System garantice la integridad de los datos, usando algún mecanismo.

Falso una particion de swap puede inclusive no tener file system, en una particion sin formato.

F, ya que ante algún evento(por ej se apaga la pc) esos datos se pierden.

21) El mecanismo de DMA, configurado bajo un bus de datos, permite que el intercambio de datos entre el módulo de DMA y el de E/S se realice en el bus de sistema.

Aca vale preguntar si cuando dice “configurado bajo un bus de datos” significa:

1. que se le otorga un bus aparte a la dma para que se comunique con el modulo de E/s

Falso, si se le otorga un bus de datos al dma, el intercambio entre el modulo de dma y de I/O se hacen a traves de ese bus de datos, y no el de sistema (stallings 483 figure 11.3)

b) que hay un Solo bus de datos (el cual seria el bus del sistema)

Verdadero, el dma se comunica a traves del bus de sistema con el modulo de E/S y la memoria (fue la primer implementacion de dma bastante ineficiente)

22) En un sistema en el que ejecutan muchos procesos y cada uno de ellos posee un conjunto residente muy pequeño, esto implica un porcentaje muy bajo de fallo de páginas.

Falso implica un porcentaje alto de fallos de pagina.

23) En un File System tipo Unix, el nombre no es un atributo del archivo, ya que este no se encuentra en el I‑nodo. Esto permite que los usuarios conozcan a los archivos por sus alias.

Falso alias no es un concepto de Unix, el nombre del archivo se encuentra en el directorio

24) En un sistema operativo multitarea UNIX, con paginación por demanda se

encuentra el siguiente grafo de asignación de la figura, que posee un ciclo con

recursos con un solo punto de entrada. R1 es una impresora, R2 un frame de

memoria y R3 un registro con permiso de escritura para los procesos P3 y P1. El

grafo demuestra que están en deadlock.



Verdadero, hay espera circular, condición suficiente para que haya deadlock.

25) La sincronización mediante Semáforos Contadores, funciona gracias a que la variable del semáforo no puede ser accedida directamente por los procesos de usuarios.

Falso.” These mechanisms can just as readily be used by user processes if the processor and OS make the mechanisms available.” stallings 8th 214

26) La memoria se divide en marcos o frames de igual tamaño que los segmentos.

Preguntas capsciosas si las hay, no indica en que esquema estamos hablando, asi que lo mas seguro es que este hablando de la disposicion logica independientemente del esquema que se utiliza.

Falso, Almost invariably, main memory in a computer system is organized as a linear,

or one-dimensional, address space, consisting of a sequence of bytes or words.

27) En entornos que no permitan E/S asíncronas, estas pueden ser simuladas utilizando hilos

Verdadero, se pueden utilizar I/O no bloqueantes mediante jacketing

28) Si la TLB posee una buena tasa de aciertos, es posible que un fallo de página no genere accesos a disco

Falso, si bien es posible que un fallo de pagina no genere acceso a disco (si es una pagina invalida), no tiene relacion con la tasa de aciertos de la tlb.

29) Dos procesos que comparten ciertos recursos pero no utilizan semáforos podrían quedar en Deadlock

Verdadero, podrian utilizar otra tecnica que garantice la exclusion mutua, y quedar en deadlock, por ej monitores.

30) El algoritmo Virtual Round Robin puede presentar inanición si se ejecutan concurrentemente demasiados procesos CPU Bound.

Falso, el algoritmo Virtual round robin no sufre de inanicion.

31) Al ejecutar por consola “./benchmark”, desde el directorio raíz del FS Unix, se corre correctamente el Benchmark. Esto no sería posible si el programa estuviese en la partición FAT32.

Falso. Si “./benchmark” fuera un symbolic link apuntando al path absoluto del ejecutable en la otra particion, entonces sería posible ejecutar el programa

32) El proceso “Benchmark” podría generar el mismo gantt si los hilos fuesen ULTs, utilizando una biblioteca de hilos que tenga algún algoritmo apropiado.

La validez de esta pregunta varia si uno objeta que se puede utilizar jacketing o no,

33) Con una TLB de 4 entradas se generarían menos fallos de página

Falso. La TLB no disminuye la cantidad de fallos de página. Disminuye los accesos a memoria a la hora de traducir páginas.

34) Dado que esta PC tiene un único procesador, no es recomendable que los semáforos del Sistema Operativo realicen espera activa

Verdadero. La espera activa generaría demasiado overhead (a diferencia de un sistema multiprocesador donde otros procesos

podrían ejecutar mientras en una cpu se ocupa una región crítica). Para un caso así, se podrían deshabilitar las interrupciones, dado

que suele ser menos costoso porque solo hay un procesador

35) Cada operación de E/S asociada a un fallo de página se realiza utilizando una entradasalida bloqueante

Verdadero. Es necesario que el KLT en cuestión no continúe hasta que la página esté cargada en memoria principal, dado que en

su ejecución da por sentado que todas las páginas que necesita estarán en memoria

36) Si un volumen formateado con FAT16 es formateado nuevamente con FAT32 manteniendo el tamaño de los clusters, el máximo espacio direccionable sigue siendo el mismo porque es el mismo disco.

Falso. El máximo espacio de direccionamiento esta dado por la capacidad de direccionamiento de cada FS en fat 16 es 2^16\* tamaño de cluster y en fat 32 es 2^28 \*tamaño de cluster (claramente fat 32 es mas grande si usan el mismo tamaño de cluster)

37) En Buddy System la fragmentaciòn externa es mayor que en la paginación con memoria virtual.

Falso La paginacion no sufre fragmentación externa

Verdadero? paginacion no sufre fragmentacion externa, asi que buddy al sufrirla es mayor.

38) Una página involucrada en una E/S podría ser elegida como víctima de un algoritmo de reemplazo. Para evitar entorpecer dicha operación, la única opción es “lockear” dicha página para que sea ignorada por el algoritmo de reemplazo.

Verdadero, cuando no se usa io buffering

If paging is being used, at least the page containing the target locations must be locked into main memory

39) Antes de que un proceso intente ejecutar una instrucción privilegiada, el sistema operativo valida el modo de ejecución y finaliza el proceso si el mismo no tiene el modo adecuado

Falso, a nivel hardware se detecta en la psw el bit de modo, y si esta en modo usuario queriendo ejecutar una instruccion privilegiada, se genera un trap que le avisa al so, y se termina el proceso.

40) Comparando con los UTL, los KLT son más flexibles con respecto a las operaciones bloqueantes aunque menos portables.

Verdadero, los klt son menos portables porque no todos los sistemas operativos soportan el uso de hilos klt

41) El mapeo de archivos a memoria permite a los procesos manejar archivos muy grandes de manera eficiente y sencilla.

Verdadero, es mas eficiente llevarlo a memoria y manipularlo como si fuera un puntero.

42) Usando semáforos implementados con espera activa no hay posibilidad de deadlocks, ya que los procesos no se bloquean esperando para ingresar a una región crítica.

Falso los semaforos no se pueden implementar con espera activa .

43) Si una system call se ejecuta de forma no bloqueante, igualmente realizará al menos un cambio de modo.

Verdadero, en algun momento el SO tiene que atender la system call y para eso debe haber un cambio de modo.

44) Lecturas secuenciales de archivos muy grandes requieren mayor cantidad de accesos a bloques en EXT que en FAT (asumiendo archivos y bloques de igual tamaño).

Verdadero, por los accesos a los bloques de punteros.

45) El algoritmo SJF con estimadores no puede aplicarse si el comportamiento de los procesos es muy errático

Falso. puede aplicarse con un alfa adecuado a favorecer a las ejecuciones más recientes

46) En un sistema que utiliza un planificador de corto plazo sin desalojo, los únicos cambios de contexto que ocurren son los generados por el abandono voluntario de la cpu por parte del proceso en ejecución, ya sea por finalización o llamada a syscall

Falso, puede haber un cambio de contexto por una interrupcion.

47) Los ULTs que son creados por una biblioteca que implementa las entradas salidas mediante la técnica de jacketing se comportan igual que los KLTs.

Falso, son igual de flexibles porque las I/O no bloquean a los hilos, pero los hilos de usuario no se comportan igual en ambientes con multiprocesadores

48) Es posible realizar symbolic links sobre archivos de otros file systems, aún si los mismos no son mantenidos. Sin embargo, no se puede hacer lo mismo con los hardlinks, al menos que ambos file systems sean de tipo UFS (Unix FS).

Falso. no se pueden utilizar hard links que apunten a inodos de volumenes distintos

49) Si bien la técnica de Buddy System permite particionar la memoria en huecos muy pequeños, puede dejar huecos imposibles de asignar.

Verdadero, no siempre se puede compactar los huecos en buddy system

50) Si en un sistema no se utiliza ninguna técnica de detección/evasión/prevención de deadlocks, si ocurre un deadlock, no se podrá realizar ningún trabajo útil.

Falso, solo no se podra realizar trabajo util en los procesos que sufrieron deadlock, el resto del sistema si trabaja de forma independiente a esos procesos no se va a ver afectado

51) La diferencia sustancial entre RAID 5 y RAID 6 es que se necesita un disco adicional para conformar las paridades.

Verdadero, necesita un disco mas ya que cada disco tiene dos bloques de paridad por franja

52) Al utilizar tabla de páginas invertidas disminuye el tiempo de exploración en la tabla.

Verdadero, por el rapido acceso que produce la utilizacionde hashing

53) Si los bloques asignados al administrador de memoria son variables se genera fragmentación externa.

Verdadero; la asignacion variable sufre de fragmentacion externa dejando huecos los cuales luego se deben compactar para un manejo optimo de la capacidad de la memoria

54) Es posible que una E/S realizada desde un KLT no bloquee al resto de los KLT’s del mismo padre.

Verdadero, los klt son vistos por el so como procesos, por lo que en un contexto en el que no haya mala sincronizacion entre sus recursos compartidos, no tendrian que bloquearse.

55) El vector de estados, el cual se implementa como PCB, mantiene el estado de cada uno de los ciclos de ejecución del proceso.

Falso,el pcb mantiene la informacion con respecto al estado del proceso en la informacion de control del proceso.

56) Se detecta que el planificador de largo plazo está ejecutándose con mayor frecuencia que el de mediano plazo. Si el grado de multiprogramación se mantuvo, esto podría significar que está ocurriendo thrashing.

Falso, cuando hay thrashing el grado de multiprogramacion no se mantiene, sino que aumenta.

57) No hay forma de que dos procesos sincronizados, que ejecutan en diferentes computadoras, queden en deadlock entre sí.

Falso, pueden estar mal sincronizados.

58) Cuantos más niveles hay en un esquema de paginación multinivel, más importante es el uso de una TLB.

Verdadero, porque al haber paginacion multinivel hay mas accesos por los distintos niveles en la tabla de pagina, si hay un tlb hit, se puede reducir bastante la cantidad de accesos.

59) Implementar la estrategia de “Evasión” implica sólo verificar que el Sistema esté en Estado Seguro cuando se pide un recurso.

Falso, evasion ademas requiere conocer todos los recursos de antemano que van a utilizar los procesos y simular las peticiones de forma que al menos exista una secuencia que determine que el estado es seguro.

60) En FAT, la lectura de un bloque de datos podría requerir menos accesos a disco que con EXT (con un igual tamaño de bloque). A su vez, FAT permite el acceso directo gracias a su asignación de bloques indexada.

Falso, FAT tiene asignacion entrelazada.

61) La única forma de activar el Sistema operativo para que realice alguna tarea, cuando se está ejecutando un proceso de usuario es por medio de una system call.

Falso, puede ocurrir una interrupcion

62) En un sistema con hilos a nivel kernel, necesito usar PCBs a pesar de que el sistema operativo planifica y encola los TCBs.

Verdadero, el pcb siempre es necesario ya que es una estructura de control de los procesos.

63) Cuando un proceso referencia a una página, el sistema operativo primero valida que la misma corresponda al espacio de direcciones del proceso, y si la página es inválida finaliza el proceso.

Falso, primero se fija si esta en la tlb.

64) Un sistema operativo puede detectar el thrashing mediante el análisis de la relación entre los porcentajes de utilización de la CPU y el grado de multiprogramación del sistema.

Falso tambien es necesario conocer el porcentaje de utilizacion de la unidad de paginacion

65) Una condición de carrera en un sistema que soporta hilos podría provocar un deadlock.

Falso en condicion de carrera no puede ocurrir deadlock porque no existe la exclusion mutua.

66) En un programa donde existan condiciones de carrera, seguramente se producirán interbloqueos en el corto o largo plazo

Falso, leer 65)

67) Si el único proceso que iba a realizar un signal(s) muere antes de efectuar la operación, siendo ‘s’ un semáforo de tipo general/contador, otros procesos que esperaban por ese cambio de estado quedarán bloqueados por siempre

Verdadero, si el contador queda en 0 y ese proceso era el que tenia que aumentar el contador, los demas quedaran bloqueados a la espera del signal.

68) Un sistema operativo con un planificador de corto plazo sin desalojo puede desestabilizarse fácilmente, perdiendo el control general del sistema.

Falso, no depende directamente del planificador de corto plazo. si el grado de multiprogramacion es muy elevado, superando las limitaciones de procesamiento del sistema, el sistema se desestabiliza, independientemente del algoritmo.

69) En un sistema con hilos a nivel kernel, el PCB necesita guardar referencias a la pila de ejecución de cada hilo debido a que los mismos podrían querer acceder a variables globales.

70) La tabla de páginas invertida siempre permite realizar un mejor aprovechamiento de la memoria que las tabla de páginas de dos niveles.

Verdadero, la paginacion multinivel puede ocupar mucho espacio en la memoria, ademas de que la paginacion invertida el acceso es mucho mas rapido.