

Introducción a los Sistemas Operativos

Repaso 1er. Parcial Teórico

1. ¿El SO necesita tiempo de CPU?
2. ¿Pueden convivir en un mismo SO la modalidad Batch y la Interactiva?
3. ¿Puede un sistema monousuario ser multitarea?
4. ¿Puede un sistema multiusuario ser monotarea?
5. ¿Los sistemas time sharing son una consecuencia de la multiprogramación?
6. ¿Puede un programa ejecutarse desde el disco?
7. ¿Puedo planificar el uso de la CPU si no cuento con memoria secundaria?
8. La interrupción por clock impide que un proceso se apropie del procesador.
9. La interrupción es externa al proceso.
10. Un intento de acceder a una dirección ilegal, se trata como una excepción o trap.
11. Un proceso puede acceder al espacio de direcciones de otro proceso si esta en modo usuario.
12. Una llamada al sistema (system call) genera un proceso del sistema operativo.
13. La llamada al sistema es la forma que tiene la aplicación de comunicarse con el sistema operativo.
14. Si tengo varios trabajos orientados a entrada/salida, las colas de solicitudes a los dispositivos estarán vacías.
15. Buffer, es espacio en memoria, y spool, espacio en disco.
16. Si varios usuarios mandan a imprimir a la misma impresora, los archivos se almacenan temporariamente en el buffer.
17. El sistema operativo permite al usuario abstraerse del hardware y su manejo.
18. Es lo mismo el kernel que el sistema operativo?
19. Las bibliotecas proveen módulos de sw para ser invocados en tiempo de compilación.
20. La memoria principal es un recurso del tipo multiplexada en el espacio.
21. El procesador en un sistema monoprocesador es un recurso del tipo multiplexada en el tiempo a cada proceso.
22. Open (archivo) se implementa como una system call?
23. Date se implementa como una system call?
24. Un proceso tiene un stack en modo usuario y un stack en modo supervisor. Como no se usan a la vez, ocupan la misma dirección de memoria. (V o F)
25. El estado del proceso está en la PCB. (V o F)
26. Un proceso crea a otro mediante un system call. (V o F)
27. La cola de procesos está en el disco. (V o F)
28. Cuando un proceso se crea, está en disco. (V o F)
29. El proceso padre crea al hijo en su propio espacio de direcciones. (V o F)
30. Las tablas de archivos que usa el proceso forma parte de su contexto. (V o F)
31. La PCB se crea a partir que el proceso se carga en memoria. (V o F)
32. El proceso padre y el hijo comparten la PCB. (V o F)
33. Si no fuera por la E/S, los procesos no necesitarían system calls. (V o F)
34. Para que un proceso pueda acceder al espacio de direcciones de otro proceso, debe estar en modo supervisor. (V o F)
35. El contexto de un proceso es lo mismo que su espacio de direcciones. (V o F)
36. Para impedir que un proceso se apropie del procesador, existen las interrupciones por clock.
37. Para implementar prioridad dinámica o aging, se tiene en cuenta: a) cuanto tiempo de CPU usó el proceso recientemente; b) cuanto tiempo de espera tiene acumulado

38. Es lo mismo cambio de contexto que cambio de proceso?
39. Es lo mismo cambio de contexto que cambio de modo?
40. Un cambio de contexto involucra un cambio de modo.
41. Un cambio de modo involucra un cambio de contexto
42. Un fork exitoso produce cambios en la PCB del padre pues se almacena del hijo.
43. El espacio de direcciones de un proceso está delimitado por los registros y
44. EL fork devuelve dos valores: ... al proceso hijo y al proceso padre.
45. El acceso no autorizados por parte de un proceso a una posición de memoria es detectado por:
a) El S.O. b) El Hardware c) No es detectado nunca
46. Las Systems Calls se ejecutan en "Modo Privilegiado". V o F
47. Ante un cambio de contexto, indique cuáles de estos elementos se guarda en la PCB:
a) tabla de páginas; b) pila de usuario; c) tabla de archivos abiertos; d) estado del proceso
48. El chequeo de la existencia de una interrupción se realiza entre los pasos de "Fetch" y "Execute" de cada ciclo de instrucción
49. El vector de interrupciones siempre debe estar en memoria
50. Un system call fork, provocará cambio de contexto
51. Un proceso swapeado en estado listo (ready to run) no compete por CPU.
52. El scheduler de short term se ejecuta con menos frecuencia que el de long term.
53. El cambio de contexto lo hace el scheduler de long term.
54. Cuando un proceso se crea, se pone en la cola de procesos, hasta que es elegido por el scheduler de long term.
55. Cuando a un proceso se le termina su quantum, pasa a estado de espera.
56. El scheduler de medium term es quien decide el cambio entre nuevo y ready.
57. El scheduler de short term es quien hace pasar al proceso de estado ready a running.
58. En la planificación de CPU se trata de maximizar la productividad, minimizar el tiempo de respuesta.
59. El tiempo de retorno, es el tiempo desde que se inicia hasta que termina, sumando cpu, espera en colas.
60. Supongamos que un proceso está en espera swapeado y se cumple el evento por el que estaba esperando. El proceso queda en estado de listo en memoria secundaria.
61. Según el diagrama visto: puede un proceso pasar del estado de nuevo (creado) a listo swapeado? SI - NO
62. Un proceso puede pasar de esperar en memoria secundaria a esperar en memoria real.
63. El scheduler de medium term maneja el grado de multiprogramación.
64. El disco permitió implementar la planificación de procesos.
65. Cuando a un proceso se le termina su quantum, pasa a estado de espera.
66. En un sistema monoprocesador, cuando se ejecuta la rutina de interrupciones, el resto de los procesos quedan en espera.
67. En un ambiente multiprocesador; a) cada procesador tiene su propia cola de procesos. B) hay una cola de procesos para todos los procesos
68. Priorizar los procesos cortos a los largos elimina el efecto convoy.
69. Conviene que un proceso en espera swapeado retorne a memoria principal?
70. En un ambiente interactivo y batch, que se maneja con colas múltiples con retroalimentación... conviene usar algoritmos preemptivos?
71. Indique cuál es la combinación que representa la sucesión de actividades que realiza el dispatcher:

- a) Cambio de contexto;
b) Cambio de Modo;
c) Salto a primer instrucción a ejecutar;
d) Carga en memoria del proceso elegido

72. Indique que puede ocurrir cuando tengo demasiados procesos orientados a I/O:
73. a) Se incrementa el uso de CPU; b) se saturan las colas de dispositivo;
74. Cuando se carga un proceso en memoria, se hace en modo usuario.
75. En la administración particionada de memoria, no puede haber swapping.
76. En la administración particionada de memoria hay fragmentación interna.
77. En la estrategia best fit, se elige de entre las particiones disponibles, la que produzca menor fragmentación interna.
78. En la estrategia worst fit, se elige de entre las particiones disponibles, la que produzca mayor fragmentación interna.
79. La más rápida de las estrategias de asignación de particiones de memoria, es la first fit.
80. Resolver la dirección en el momento de la carga, exigirá que el proceso se ejecute siempre en el mismo lugar de la memoria.
81. Si se resuelven las direcciones en el momento de la compilación y quieren cambiarse, se debe recompilar.
82. En la Administración particionada fija, el grado de multiprogramación lo dá la cantidad de particiones.
83. La resolución de direcciones en el momento de la carga facilita la administración paginada.
84. El espacio de direcciones de un proceso está delimitado por los registros y
85. El scheduler de short term se ejecuta con menos frecuencia que el de long term. (V o F)
86. Un proceso tiene un stack en modo usuario y un stack en modo supervisor. Como no se usan a la vez, ocupan la misma dirección de memoria. (V o F)
87. El cambio de contexto lo hace el scheduler de long term. (V o F)
88. El estado del proceso está en la PCB. (V o F)
89. Un proceso crea a otro mediante un system call. (V o F)
90. Cuando un proceso se crea, se pone en la cola de procesos, hasta que es elegido por el scheduler de long term. (V o F)
91. La cola de procesos está en el disco. (V o F)
92. Cuando a un proceso se le termina su quantum, pasa a estado de espera. (V o F)
93. Cuando un proceso se crea, está en disco. (V o F)
94. El scheduler de medium term es quien decide el cambio entre nuevo y ready. (V o F)
95. El scheduler de short term es quien hace pasar al proceso de estado ready a running. (V o F) Distinguir entre “pasar” y “elegir”