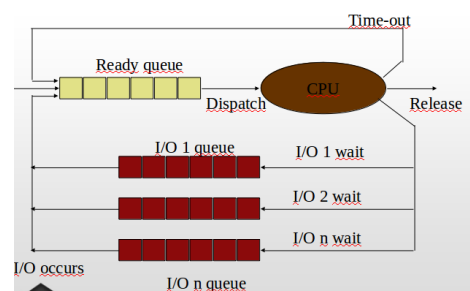


Repaso 1er. Parcial Teórico

1. ¿El SO necesita tiempo de CPU?
2. ¿Pueden convivir en un mismo SO procesos batch y procesos interactivos?
3. ¿Puede un sistema monousuario ser multitarea?
4. ¿Puede un sistema multiusuario ser monotarea?
5. ¿Puede un programa ejecutarse desde el disco?
6. ¿Puedo planificar el uso de la CPU si no cuento con memoria secundaria?
7. La interrupción por clock impide que un proceso se apropie del procesador.
8. Las interrupciones son externas a los procesos.
9. Un intento de acceder a una dirección ilegal, se trata como un trap.
10. Un proceso puede acceder al espacio de direcciones de otro proceso si esta en modo usuario.
11. Una llamada al sistema (system call) genera la creación de un proceso del sistema operativo para atender la llamada.
12. Las llamadas al sistema son la forma que tienen las aplicaciones de comunicarse con el sistema operativo.

13.

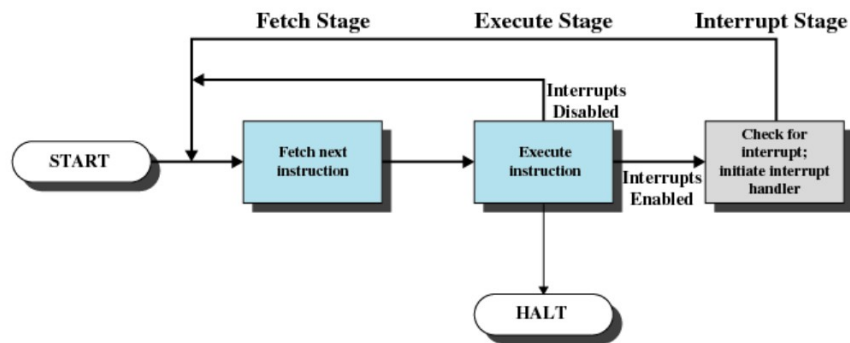
Si tengo muchos procesos orientados a entrada/salida, las colas de solicitudes a los dispositivos de E/S estarán vacías.



14. ¿El sistema operativo permite al usuario abstraerse del hardware y su manejo.
15. ¿Es lo mismo el kernel que el sistema operativo?
16. La memoria principal es un recurso del tipo multiplexada en el espacio.
17. El procesador en un sistema monoprocesador es un recurso del tipo multiplexado en el tiempo **a cada proceso.**
18. Open → Archivo. Se implementa con una system call?
19. Date se implementa con una system call?
20. 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)
21. El estado del proceso está en la PCB. (V o F)
22. Un proceso crea a otro mediante un system call. (V o F)
23. La cola de procesos está en el disco. (V o F)
24. Cuando un proceso se crea, está en disco. (V o F)
25. El proceso padre crea al hijo en su propio espacio de direcciones. (V o F)
26. Las tablas de archivos correspondientes a los archivos abiertos que está usando un proceso, forman parte de su contexto. (V o F)
27. La PCB se crea a partir que el proceso se carga en memoria. (V o F)
28. Luego de la systema call fork, el proceso padre y el proceso hijo comparten la PCB. (V o F)
29. Si no fuera por la E/S, los procesos no necesitarían system calls. (V o F)

30. En modo supervisor, es posible acceder al espacio de direcciones de cualquier proceso. (V o F)
31. El contexto de un proceso es lo mismo que su espacio de direcciones. (V o F)
32. Para implementar prioridad dinámica o aging por inanición, se tiene en cuenta:
- a) cuanto tiempo de CPU usó el proceso recientemente;
 - b) cuanto tiempo de espera tiene acumulado
33. Un cambio de modo involucra un cambio de contexto.
34. Un cambio de contexto involucra un cambio de modo.
35. Es lo mismo cambio de contexto que cambio de proceso?
36. Es lo mismo cambio de contexto que cambio de modo?
37. Un fork exitoso produce cambios en la PCB del padre pues se almacena del hijo.
38. En el mecanismo de manejo de memoria con particiones, el espacio de direcciones de un proceso está delimitado por los registros y
39. El fork devuelve dos valores: ... al proceso hijo y al proceso padre.
40. Un acceso no autorizado a memoria es detectado por:
- a) El S.O.
 - b) El Hardware
 - c) No puede detectarse
41. Las Systems Calls se ejecutan en "Modo Privilegiado". V o F
42. 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

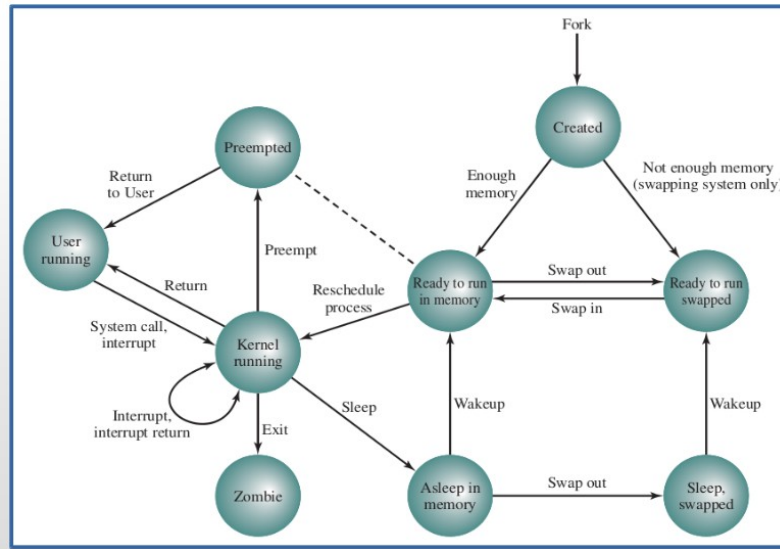
43. El chequeo de la existencia de una interrupción se realiza entre los pasos de "Fetch" y "Execute" de cada ciclo de instrucción



44. El vector de interrupciones siempre debe estar en memoria
45. Un system call fork, provocará cambio de contexto
46. Un proceso swapeado en estado listo (ready to run) no compite por CPU.
47. El scheduler de short term se ejecuta con menos frecuencia que el de long term.
48. El cambio de contexto lo hace el scheduler de long term.
49. Cuando a un proceso se le termina su quantum, pasa a estado de espera.
50. El scheduler de medium term es quien decide el cambio entre nuevo y ready.
51. El scheduler de short term es quien hace pasar al proceso de estado ready a running.
52. En la planificación de CPU se trata de maximizar la productividad, minimizar el tiempo de respuesta.
53. El tiempo de retorno, es el tiempo desde que se inicia hasta que termina, sumando cpu, espera en colas.

54. 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.

55. Según el diagrama visto: puede un proceso pasar del estado de nuevo (creado) a listo swapeado? SI – NO



56. Un proceso puede pasar de esperar en memoria secundaria a esperar en memoria real.

57. El scheduler de medium term maneja el grado de multiprogramación.

58. El disco permitió implementar la planificación de procesos.

59. En un sistema monoprocesador, cuando se atiende una interrupción (se ejecuta una rutina de manejo de interrupciones) todos los procesos quedan en espera.

60. En un ambiente con procesos interactivos y batch, que maneja colas multinivel.
¿Conviene usar algoritmos apropiativos?

61. 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/próxima instrucción a ejecutar;
- d) Carga en memoria del proceso elegido

62. Indique que puede ocurrir cuando solamente se tienen muchos procesos orientados a I/O:

- a) Se incrementa el uso de CPU;
- b) Se saturan las colas de dispositivo;

63. Cuando se carga un proceso en memoria, se hace en modo usuario.

64. En paginación los procesos utilizan direcciones lógicas que son necesarias traducir a direcciones físicas.

65. Seleccione la(s) opción(es) correcta(s). Cada entrada en la tabla de segmentos está compuesta por:

- A) Desplazamiento
- B) Nro de segmento
- C) Nro. Página
- D) Dirección base del segmento
- E) tamaño
- F) Bits de Control