**Repaso 1er. Parcial Teórico**

|  |
| --- |
| 1. |

¿El SO necesita tiempo de CPU? **V**

|  |
| --- |
| 2. |

¿Pueden convivir en un mismo SO procesos batch y procesos interactivos? **V**

|  |
| --- |
| 3. |

¿Puede un sistema monousuario ser multitarea? **V**

|  |
| --- |
| 4. |

¿Puede un sistema multiusuario ser monotarea? **F**

|  |
| --- |
| 5. |

¿Puede un programa ejecutarse desde el disco? **F**

|  |
| --- |
| 6. |

¿Puedo planificar el uso de la CPU si no cuento con memoria secundaria? **V**

|  |
| --- |
| 7. |

La interrupción por clock impide que un proceso se apropie del procesador. **V**

|  |
| --- |
| 8. |

Las interrupciones son externas a los procesos. **F**

|  |
| --- |
| 9. |

Un intento de acceder a una dirección ilegal, se trata como un trap. **V**

|  |
| --- |
| 10. |

Un proceso puede acceder al espacio de direcciones de otro proceso si está en modo usuario. F

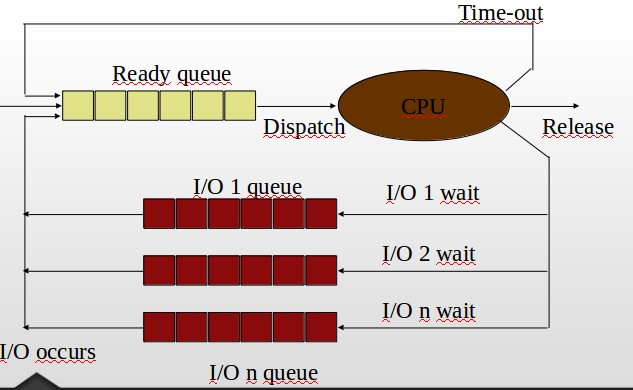
|  |
| --- |
| 11. |

Una llamada al sistema (system call) genera la creación de un proceso del sistema operativo para atender la llamada. **F**

|  |
| --- |
| 12. |

Las llamadas al sistema son la forma que tienen las aplicaciones de comunicarse con el sistema operativo. **V**

13.

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

|  |
| --- |
| 14. |

¿EI sistema operativo permite al usuario abstraerse del hardware y su manejo? **V**

|  |
| --- |
| 15. |

¿Es lo mismo el kernel que el sistema operativo? **F**

|  |
| --- |
| 16. |

La memoria principal es un recurso del tipo multiplexada en el espacio. **V**

|  |
| --- |
| 17. |

El procesador en un sistema monoprocesador es un recurso del tipo multiplexado en el tiempo a cada proceso. **V**

|  |
| --- |
| 18. |

Open → Archivo ¿Se implementa con una system call? **V**

|  |
| --- |
| 19. |

¿Date se implementa con una system call? **V**

|  |
| --- |
| 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. **F**

|  |
| --- |
| 21. |

El estado del proceso está en la PCB. **V**

|  |
| --- |
| 22. |

Un proceso crea a otro mediante un system call. **V**

|  |
| --- |
| 23. |

La cola de procesos está en el disco. **F**

|  |
| --- |
| 24. |

Cuando un proceso se crea, está en disco. **V**

|  |
| --- |
| 25. |

El proceso padre crea al hijo en su propio espacio de direcciones. **F**

|  |
| --- |
| 26. |

Las tablas de archivos correspondientes a los archivos abiertos que está usando un proceso, forman parte de su contexto. **F**

|  |
| --- |
| 27. |

La PCB se crea a partir que el proceso se carga en memoria. **F**

|  |
| --- |
| 28. |

Luego de la system call fork, el proceso padre y el proceso hijo comparten la

PCB. **F**

|  |
| --- |
| 29. |

Si no fuera por la E/S, los procesos no necesitarían system calls. **F**

|  |
| --- |
| 30. |

En modo supervisor, es posible acceder al espacio de direcciones de cualquier proceso. **V**

|  |
| --- |
| 31. |

El contexto de un proceso es lo mismo que su espacio de direcciones. **F**

|  |
| --- |
| 32. |

Para implementar prioridad dinámica o aging por inanición, se tiene en cuenta: **B**

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. **F**

|  |
| --- |
| 34. |

Un cambio de contexto involucra un cambio de modo. **V**

|  |
| --- |
| 35. |

¿Es lo mismo cambio de contexto que cambio de proceso? **F**

|  |
| --- |
| 36. |

¿Es lo mismo cambio de contexto que cambio de modo? **F**

|  |
| --- |
| 37. |

Un fork exitoso produce cambios en la PCB del padre pues se almacena **pid** del hijo.

|  |
| --- |
| 38. |

En el mecanismo de manejo de memoria con particiones, el espacio de direcciones de un proceso está delimitado por los registros **base** y **limite**

|  |
| --- |
| 39. |

El fork devuelve dos valores: **pid padre** al proceso hijo y **pid hijo** al proceso padre.

|  |
| --- |
| 40. |

Un acceso no autorizado a memoria es detectado por: **B**

a) El S.O. b) El Hardware c) No puede detectarse

|  |
| --- |
| 41. |

Las System Calls se ejecutan en “Modo Privilegiado”. **V**

|  |
| --- |
| 42. |

Ante un cambio de contexto, indique cuáles de estos elementos se guarda en la PCB: **A-B-C-D**

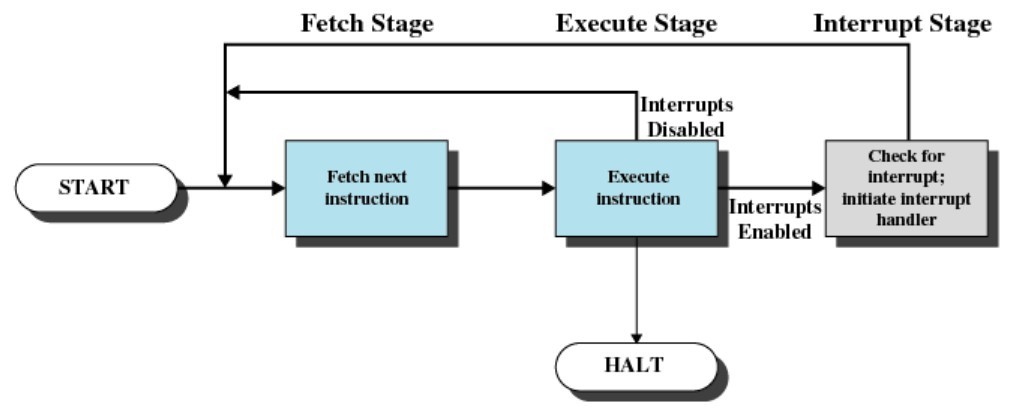
a)tabla de páginas b)pila de usuario

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 **F**



|  |
| --- |
| 44. |

El vector de interrupciones siempre debe estar en memoria **V**

|  |
| --- |
| 45. |

Un system call fork, provocará cambio de contexto **F**

|  |
| --- |
| 46. |

Un proceso swappeado en estado listo (ready to run) no compite por CPU. **V**

47. El scheduler de short term se ejecuta con menos frecuencia que el de long

term. **F**

|  |
| --- |
| 48. |

El cambio de contexto lo hace el scheduler de long term. **F**

|  |
| --- |
| 49. |

Cuando a un proceso se le termina su quantum, pasa a estado de espera. **F**

|  |
| --- |
| 50. |

El scheduler de medium term es quien decide el cambio entre nuevo y ready. **F**

|  |
| --- |
| 51. |

El scheduler de short term es quien hace pasar al proceso de estado ready a running. **F**

|  |
| --- |
| 52. |

En la planificación de CPU se trata de maximizar la productividad, minimizar el tiempo de respuesta. **V**

|  |
| --- |
| 53. |

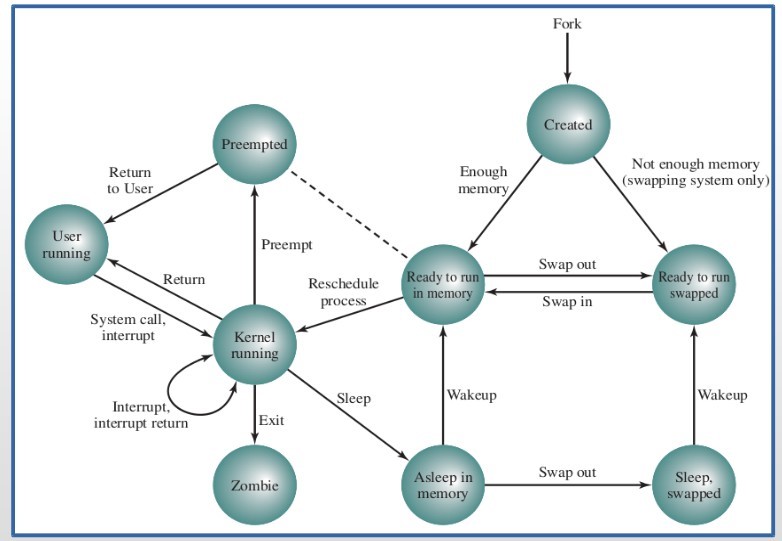
El tiempo de retorno, es el tiempo desde que se inicia hasta que termina, sumando cpu, espera en colas. **V**

|  |
| --- |
| 54. |

Supongamos que un proceso está en espera swappeado y se cumple el evento por el que estaba esperando. El proceso queda en estado de listo en memoria secundaria. **V**

|  |
| --- |
| 55. |

Según el diagrama visto: ¿puede un proceso pasar del estado de nuevo (creado) a listo swappeado? **V**



|  |
| --- |
| 56. |

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

|  |
| --- |
| 57. |

El scheduler de medium term maneja el grado de multiprogramación. **V**

|  |
| --- |
| 58. |

El disco permitió implementar la planificación de procesos. **F**

|  |
| --- |
| 59. |

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

|  |
| --- |
| 60. |

En un ambiente con procesos interactivos y batch, que maneja colas multinivel.

¿Conviene usar algoritmos apropiativos? **V**

|  |
| --- |
| 61. |

Indique cuál es la combinación que representa la sucesión de actividades que realiza el dispatcher: **A-B-C-D**

1. Cambio de contexto c) Salto a primer/próxima instrucción a ejecutar

|  |
| --- |
| 62. |

1. Cambio de Modo d) Carga en memoria del proceso elegido
2. Indique que puede ocurrir cuando solamente se tienen muchos procesos orientados a I/O **B**
   1. Se incrementa el uso de CPU
   2. Se saturan las colas de dispositivo

|  |
| --- |
| 63. |

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

|  |
| --- |
| 64. |

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

|  |
| --- |
| 65. |

Seleccione la(s) opción(es) correcta(s). Cada entrada en la tabla de segmentos está compuesta por: **D-E-F**

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