

UO:

Nombre:

*Puntuación por pregunta*

- 100% si acierto
- -50% si fallo
- 0 si en blanco

**Cuestiones Verdadero / Falso****En la hoja de respuestas MARCA UNA A cuando sea VERDADERO y UNA B cuando sea FALSO.**

1. \*La gestión de la interrupción se realiza en modo núcleo (o supervisor)
2. Los programas del usuario llaman al Sistema Operativo a través del Shell
3. \*Cuando se produce una interrupción activa el hardware guarda algunos registros del procesador
4. El primer chequeo del hardware lo realiza el Sistema Operativo.
5. \*Cada vez que se un proceso finaliza, se activa el sistema operativo
6. \*Cada vez que se produce una interrupción de reloj se ejecuta el Sistema Operativo
7. Las operaciones de e/s se ejecutan siempre en el procesador
8. Cuando se produce un error de fallo de programa se produce una interrupción hardware.
9. \*Después de una instrucción TRAP se ejecuta código del Sistema Operativo
10. \*La comunicación entre procesos es tarea del gestor de procesos
11. La localización de espacio en memoria para un proceso es tarea del gestor de procesos
12. El Bloque de Control de un proceso guarda, entre otras cosas, los valores de los registros del procesador en todo momento.
13. \*El PID, la prioridad y el estado del proceso en memoria se guardan en el BCP de un proceso
14. Multitarea y multiproceso son conceptos equivalentes
15. \*La alternancia de los programas entre instrucciones de procesamiento y operaciones de e/s abren una forma de implementar de la multitarea
16. Siempre que hay una interrupción se guarda el estado del procesador en el BCP del proceso interrumpido
17. \*Los procesos en estado listo sólo necesitan al procesador para ser ejecutados.
18. Un proceso puede pasar de estado listo a bloqueado.
19. \*Cuando hay una interrupción los registros del procesador se guardan en la pila del sistema
20. \*La pila del sistema almacena la palabra de estado del procesador del proceso interrumpido en el momento de la interrupción.
21. Toda interrupción de reloj implica un cambio de proceso
22. \*Cuando el gestor de procesos decide hacer un cambio de proceso, los datos de algunos registros del procesador que había al producirse la interrupción se guardan en el BCP del proceso que abandona la ejecución
23. \*El cambio entre hilos de la misma tarea tiene menos coste que el cambio entre procesos
24. \*Hilos de la misma tarea comparten espacio de direcciones
25. \*Hilos de la misma tarea comparten BCP
26. El planificador a largo plazo es el responsable de rebajar el grado de multiprogramación del sistema.

27. Hilos de la misma tarea comparten pila de ejecución
28. \*En una política de planificación de turno rotatorio o cíclica, si un proceso en estado ejecutando agota su cuanto es expulsado al final de la cola de listos para ejecución.
29. \*En una política de planificación de turno rotatorio o cíclica un cuanto demasiado pequeño puede introducir mucha sobrecarga al sistema.
30. En una política de planificación con prioridades expulsivas, si llega un proceso con mayor prioridad que el que se está ejecutando, el sistema lo expulsa y lo pasa al estado bloqueado hasta que el de mayor prioridad finalice.
31. La decisión de qué proceso de entre los procesos listos pasa a ejecución constituye la planificación a medio plazo
32. \*Las prioridades dinámicas con envejecimiento sirven para evitar que los procesos sufran inanición

Proceso	Instante llegada	Prioridad <i>Prioridad</i> $2 > 1$	Tiempo de CPU
D	9	2	3
C	6	2	6
B	3	1	4
A	0	1	7

Escenario para preguntas 33, 34, 35 y 36

33. \*En el escenario anterior, utilizando prioridades expulsivas, en el instante 6 se pasará a ejecutar el proceso C.
34. En el escenario anterior, utilizando round-robin (sin prioridades) con cuanto de 2 unidades de tiempo, en el instante 3 se pasará a ejecutar el proceso B.
35. \* En el escenario anterior, utilizando prioridades expulsivas, el tiempo de respuesta del proceso A será 0.
36. \*En el escenario anterior, utilizando FCFS el tiempo de retorno del proceso B será 11.
37. \*Escribir en una tubería vacía puede hacer que un proceso dormido se despierte.
38. \*Las señales *kill/pause* permiten la sincronización de procesos.
39. \*Siempre que se modifica el valor de una variable compartida es necesario un mecanismo de sincronización
40. \*Los procesos que hacen *wait* sobre un semáforo con valor cero pasan al estado bloqueado y entran en una cola
41. \*Los *Mutex (lock/unlock)* permiten controlar el acceso a un recurso en exclusión mutua
42. Para que dos procesos sean concurrentes deben ejecutarse en la misma máquina
43. \*En un esquema de memoria real con paginación, cada entrada de la tabla de páginas guarda el número del marco donde se encuentra cargada.
44. Traducir direcciones en tiempo de ejecución es tarea del Gestor de Memoria
45. El esquema de memoria real y asignación contigua con particiones variables se produce fragmentación interna.
46. El problema del productor – consumidor no necesita sincronización para evitar condiciones de carrera
47. En un esquema de memoria real y asignación contigua con particiones fijas, cada partición puede albergar varios programas simultáneamente
48. \*Ubicar a un proceso en memoria es tarea del Gestor de Memoria
49. \*Compartir memoria entre procesos es tarea del Gestor de Memoria
50. \*La protección del espacio de memoria de cada proceso es realizada por la MMU.