

1. En el contexto de un sistema operativo, un directorio/carpeta es un tipo de:
 - a) Recurso físico.
 - b) Recurso lógico.
 - c) Recurso híbrido.
2. Tenemos un sistema multiprogramado con un solo procesador. ¿Cuál de estas características jamás podría afirmarse sobre este sistema?
 - a) Es un sistema virtualizado.
 - b) Es un sistema distribuido.
 - c) Es un sistema de procesamiento por lotes.
3. De los algoritmos que aquí se citan, ¿cuál es el más propenso a sufrir el llamado «efecto convoy»?
 - a) FCFS.
 - b) Round Robin.
 - c) SJF.
4. ¿Cuál de estos tipos de sistemas tiene como requisito principal dar una experiencia interactiva a los usuarios?
 - a) Sistema de procesamiento por lotes.
 - b) Sistema de tiempo real.
 - c) Sistema de tiempo compartido.
5. ¿Qué tipo de sistema es el que planifica los procesos en función de sus plazos de finalización?
 - a) Sistema de tiempo real.
 - b) Sistema de tiempo compartido.
 - c) Sistema de procesamiento por lotes.
6. ¿Qué son las llamadas al sistema (*system calls*)?
 - a) Son procesos de usuario invocados por el núcleo del sistema operativo.
 - b) Son mecanismos del sistema operativo para interceptar las interrupciones del hardware.
 - c) Son servicios que el núcleo del sistema operativo ofrece a los procesos de usuario.
7. ¿Qué objetivo primario persigue la multiprogramación?
 - a) Mejorar la seguridad del sistema.
 - b) Mejorar la usabilidad del sistema.
 - c) Mejorar el rendimiento del sistema.
8. El modelo de diseño que consiste en aliviar el tamaño de un núcleo monolítico haciendo que algunos componentes se puedan incorporar dinámicamente se llama:
 - a) Micronúcleo.
 - b) Sistema operativo en capas.
 - c) Módulos cargables.
9. En la cola de preparados hay dos procesos, A y B. A va a ejecutar una ráfaga de CPU de 100 milisegundos. B va a ejecutar 200 milisegundos. El sistema utiliza un Round Robin con Q=50 milisegundos. Ahora mismo la CPU está libre. ¿Cuántos cambios de contexto se producirán en el sistema para poder planificar estas dos ráfagas?
 - a) 3.
 - b) 6.
 - c) 1.
10. Los métodos multicolos de planificación de CPU:
 - a) Gestionan una cola de preparados y varias colas de espera por CPU.
 - b) Resultan más apropiados para multiprocesadores que los métodos de una sola cola.
 - c) Manejan varias clases de procesos que se planifican según políticas diferentes.

11. ¿Cuál de estos algoritmos de planificación de CPU es siempre expulsivo (*preemptive*)?
 - a) FCFS.
 - b) Round Robin.
 - c) SJF.
12. ¿Qué característica peculiar tiene la instrucción «test-and-set»?
 - a) Se ejecuta de forma atómica.
 - b) Ejecuta dos acciones de forma simultánea (paralela).
 - c) Sólo funciona en multiprocesadores.
13. ¿Cuál de estos componentes del sistema operativo debe ofrecer necesariamente una API?
 - a) El *shell*.
 - b) El núcleo.
 - c) Los programas del sistema.
14. ¿Cómo se puede resolver el riesgo de inanición en los algoritmos basados en prioridades?
 - a) Aplicando prioridades definidas externamente, por el administrador del sistema.
 - b) Estimando la prioridad a partir de una fórmula que tenga en cuenta las duraciones de las anteriores ráfagas de CPU.
 - c) Aumentando la prioridad de los procesos de la cola a medida que esperan.
15. La llamada **pthread_join()**, ¿cómo se podría implementar si usáramos semáforos?
 - a) Con una operación WAIT() sobre un semáforo inicializado a uno. Cualquier hilo que finaliza debe ejecutar un SIGNAL() sobre el semáforo.
 - b) Con una operación WAIT() sobre un semáforo inicializado a cero. Otro hilo debe ejecutar un SIGNAL() en el momento de finalizar.
 - c) Con una operación SIGNAL() sobre un semáforo inicializado a cero. Otro hilo debe ejecutar un WAIT() en el momento de finalizar.
16. Inhibir las interrupciones no es una técnica universal para gestionar secciones críticas. ¿Cuál de estos argumentos es válido para apoyar tal afirmación?
 - a) Esta técnica es inviable en un multiprocesador.
 - b) Esta técnica sólo sirve cuando compiten exclusivamente dos procesos por la sección crítica.
 - c) No todos los sistemas operativos dan soporte a la inhibición de interrupciones.
17. ¿Qué realiza la llamada al sistema **fork()** de Unix?
 - a) Crea un nuevo proceso pesado que ejecuta una copia del proceso padre.
 - b) Crea un nuevo proceso pesado que ejecuta un programa pasado como argumento.
 - c) Crea un nuevo proceso ligero que ejecuta una función pasada como argumento.
18. Habitualmente, el núcleo del sistema operativo:
 - a) Es un hilo (*thread*) de alta prioridad.
 - b) Contiene el código del intérprete de órdenes (*shell*).
 - c) Se ejecuta en modo supervisor.
19. Estamos aplicando Round Robin para planificar procesos. Cuando a un proceso se le expira su cuanto de tiempo, ¿de qué manera cambia su estado?
 - a) Se mantiene en el mismo estado.
 - b) Pasa de estado «en ejecución» a estado «bloqueado».
 - c) Pasa de estado «en ejecución» a estado «preparado».
20. El algoritmo de Peterson para resolver secciones críticas, ¿para cuántos procesos funciona?
 - a) Para dos procesos.
 - b) Para dos o más procesos.
 - c) Para un proceso.
21. Un sistema de interrupciones:
 - a) Exige utilizar un vector de interrupciones.
 - b) Protege la memoria en la que reside el núcleo.
 - c) Ayuda a evitar la espera activa por los dispositivos de E/S.

22. ¿Por qué las operaciones de un semáforo deben ser atómicas?
- a) Para conseguir una implementación más eficiente, que no incurra en esperas activas.
 - b) Para evitar los problemas del acceso concurrente al semáforo.
 - c) Para que se cumpla la condición de progreso.
23. En general, ¿a quién pertenece el código de una rutina de servicio de interrupción?
- a) Es código del *firmware* o de la unidad de control (UC) de la CPU.
 - b) Es código del programa de usuario.
 - c) Es código del núcleo del sistema operativo.
24. Para implementar de forma eficaz un esquema de protección de memoria basado en la pareja de registros base y límite, ¿cuál de estas características debe estar presente en el hardware?
- a) Una jerarquía de memorias de al menos dos niveles.
 - b) Un procesador con dos modos de operación (usuario/sistema).
 - c) Un sistema de interrupciones.