

Fundamentos de los Sistemas Operativos · curso 2018/2019 · examen parcial · TEST

1. En comparación con los sistemas monolíticos, ¿qué se puede afirmar sobre los micronúcleos?
 - a) Son más lentos.
 - b) Ofrecen menos servicios.**
 - c) Son más flexibles.
2. ¿Por qué un fichero ejecutable binario de Windows no se puede ejecutar de forma directa en Linux?
 - a) Porque el formato del ejecutable de Windows es diferente al de Linux.**
 - b) Porque la arquitectura del código máquina de Windows es diferente al de Linux.
 - c) Porque el hardware de manejo de memoria soportado por Windows es diferente al de Linux.
3. ¿Cuál de estos tres componentes es imprescindible en un sistema operativo?
 - a) Cargador de programas.**
 - b) Gestor de memoria virtual.
 - c) Planificador de procesos.
4. ¿Cuál de estas características es esencial para un núcleo de sistema operativo?
 - a) Está instalado en el disco de arranque.
 - b) Se ejecuta en modo privilegiado.**
 - c) Ofrece servicios de manejo de directorios (carpetas).
5. Si un sistema tuviera memoria RAM infinita (o tan grande como necesitemos), ¿cuál de estos servicios del sistema operativo seguiría teniendo sentido que existiera?
 - a) Memoria virtual.
 - b) Intercambio (*swapping*).
 - c) Protección de memoria.**
6. En el diseño de los sistemas operativos, ¿qué conseguimos gracias a la independencia del dispositivo?
 - a) Que la CPU y la E/S puedan operar de forma independiente.
 - b) Más rapidez en las transferencias con los dispositivos.
 - c) Una interfaz de programación uniforme con los dispositivos.**
7. En un programa en C bajo Linux aparece esta llamada al sistema: **write(1,B,1200)**. ¿Qué es lo que hace?
 - a) Lee 1200 bytes desde B y los transfiere a la salida estándar.**
 - b) Escribe 1200 caracteres en la zona de memoria apuntada por B.
 - c) Escribe 1200 bytes a partir de la posición 1 del fichero B.
8. La instrucción **syscall** que implementan los procesadores x86 y MIPS, ¿es una instrucción privilegiada?
 - a) Sí, porque es una instrucción que genera una interrupción software.
 - b) No, porque si fuera privilegiada los procesos de usuario no podrían realizar llamadas al sistema.**
 - c) No es necesario que lo sea, pero podría definirse como privilegiada para evitar usos inadecuados.

9. Mientras el núcleo del sistema operativo está ejecutando código que atiende una llamada al sistema, llega al sistema una interrupción de un dispositivo hardware. El núcleo atiende la interrupción. ¿En qué modo de ejecución se atiende la interrupción?
- a) En modo núcleo, ya que se va a atender una interrupción.**
 - b) En modo usuario, ya que se estaba atendiendo una llamada al sistema.
 - c) En el modo que designa el dispositivo que interrumpió.
10. ¿Qué es una llamada al sistema?
- a) Un mecanismo para atender interrupciones del hardware.
 - b) Cada una de las funciones de la biblioteca estándar del C.
 - c) La solicitud de un servicio al núcleo del sistema operativo.**
11. ¿Cuál de estos componentes del sistema operativo debe ofrecer necesariamente una API?
- a) El núcleo.**
 - b) El shell.
 - c) La GUI.
12. ¿Qué objetivo primario persigue la multiprogramación?
- a) Mejorar el rendimiento del sistema.**
 - b) Mejorar la seguridad del sistema.
 - c) Mejorar la estabilidad del sistema.
13. Tenemos un servidor empresarial de una empresa de telefonía móvil que se va a dedicar exclusivamente a calcular las facturas mensuales a los clientes y hacer los cargos en sus cuentas bancarias. ¿Cuál de estas modalidades de procesamiento le viene mejor a este servidor?
- a) Procesamiento por lotes.**
 - b) Procesamiento por tiempo compartido.
 - c) Procesamiento de tiempo real.
14. ¿Cómo se llama la arquitectura de núcleos en la cual se pueden añadir dinámicamente componentes a un núcleo básico, ejecutándose estos componentes en modo sistema/núcleo?
- a) Arquitectura de micronúcleo.
 - b) Arquitectura por capas virtualizadas.
 - c) Arquitectura de módulos cargables.**
15. Alguien propone un algoritmo de planificación de procesos mediante el cual, cuando un procesador queda libre, entra a ejecución el proceso de la cola de preparados que en ese momento está ocupando menos memoria RAM. ¿Qué se puede afirmar de ese algoritmo?
- a) No es implementable.
 - b) Tiene riesgo de inanición.**
 - c) Es expulsivo.
16. En un ordenador tipo PC se utiliza un algoritmo de planificación de procesos basado en prioridades, en el cual la prioridad de un proceso es inversamente proporcional al tiempo que ha disfrutado de la CPU en los últimos 10 minutos. ¿Qué clase de procesos resulta favorecida por este algoritmo?
- a) Las aplicaciones interactivas.**
 - b) Los procesos intensivos en CPU.
 - c) Los procesos del sistema operativo.
17. En un algoritmo Round Robin, si el cuanto de tiempo Q es muy pequeño:
- a) El algoritmo degenera en un FCFS.
 - b) El reparto del tiempo de procesador entre los procesos se vuelve injusto.
 - c) Los cambios de contexto penalizan el rendimiento del sistema.**

18. Los sistemas operativos actuales utilizan la técnica de afinidad al procesador (*processor affinity*), que consiste en procurar que un hilo se ejecute siempre en el mismo procesador una y otra vez. ¿Qué beneficio tiene esta técnica?
- Ayuda a mantener el equilibrio de carga de trabajo entre procesadores.
 - Aprovecha las porciones de memoria del hilo que están en la caché del procesador afín.**
 - Ayuda a resolver conflictos de acceso simultáneo a la cola de procesos por parte de varios procesadores.
19. ¿Para qué sistemas es más conveniente utilizar Round Robin no expulsivo?
- Para sistemas interactivos.
 - Para sistemas por lotes.
 - Ese algoritmo no existe.**
20. Los métodos multicolos de planificación de CPU:
- Manejan varias clases de procesos que se planifican según políticas diferentes.**
 - Resultan más apropiados para multiprocesadores que los métodos de una sola cola.
 - Gestionan una cola de preparados y varias colas de espera por CPU y dispositivos de E/S.
21. ¿Cuál de las siguientes afirmaciones acerca de los cambios de contexto es FALSA?
- Son improductivos, por lo que deben realizarse lo más rápidamente posible.
 - Los realiza el planificador de medio plazo o medio nivel (*swapper*).**
 - Los lleva a cabo el módulo llamado despachador (*dispatcher*).
22. El Bloque de Control de Proceso (BCP):
- Es la estructura de datos que el sistema operativo utiliza para representar un proceso.**
 - Es el componente del s.o. que se encarga de controlar la entrada y salida de procesos a la CPU.
 - Es la estructura que da soporte a la cola de control de procesos en estado «preparado».
23. ¿Cuál de estas políticas de planificación de procesos logra un mejor tiempo medio de espera?
- FCFS.
 - SJF.**
 - Round Robin.
24. Tenemos un sistema multiprogramado con una única CPU. ¿Cuál de las siguientes afirmaciones NUNCA podría ser cierta para ese sistema?
- Podría ser un sistema concurrente y además de tiempo real.
 - Podría ser un sistema de tiempo compartido y además de procesamiento paralelo.**
 - Podría ser un sistema de procesamiento por lotes y además multiusuario.
25. ¿Qué implica que un algoritmo de sección crítica cumple la propiedad de progreso?
- Que la sección crítica está siempre ocupada.
 - Que si alguien solicita entrar en sección crítica, se le acaba atendiendo en un tiempo finito.
 - Que si la sección crítica está libre y alguien solicita entrar en ella, alguien acabará entrando en ella.**

26. ¿En qué consiste el interbloqueo?
- a) **Es un ciclo de bloqueos mutuos entre procesos con la consecuencia de que todos los procesos se quedan bloqueados irreversiblemente.**
 - b) Es un bloqueo entre uno o más procesos y un recurso del sistema operativo al que se debe acceder en exclusión mutua.
 - c) Es la situación en la que un proceso bloquea indefinidamente el acceso a la sección crítica a uno o más procesos diferentes.
27. ¿Qué diferencia hay entre concurrencia y paralelismo?
- a) **El paralelismo implica un hardware multiprocesador, mientras que la concurrencia se puede dar dentro de un único procesador.**
 - b) Son dos términos que se pueden emplear de forma intercambiable.
 - c) La concurrencia es un caso particular de paralelismo.
28. ¿Qué característica peculiar tiene la instrucción **test-and-set**?
- a) **Se ejecuta de forma atómica.**
 - b) Sólo funciona en multiprocesadores.
 - c) Ejecuta dos acciones (comparar y dar valor) de forma simultánea.
29. Inhibir las interrupciones no sirve como técnica universal para controlar el acceso a una sección crítica. ¿Cuál de estos argumentos es válido para apoyar tal afirmación?
- a) **Esta técnica normalmente es inviable en un multiprocesador.**
 - b) Un proceso de usuario no tiene capacidad de inhibir las interrupciones.
 - c) Esta técnica sirve para dos procesos, pero no es generalizable a $N > 2$ procesos.
30. ¿Para qué sirve la operación **pthread_create**?
- a) **Crea un nuevo proceso ligero.**
 - b) Crea un nuevo proceso pesado.
 - c) Crea un nuevo proceso mediano.