

Fundamentos de los Sistemas Operativos · EII / ULPGC
curso 2018/2019, convocatoria ordinaria · test de la primera parte

1. Si dispusiéramos de una cantidad de memoria RAM infinita (o tan grande como queramos), ¿qué componente del sistema operativo dejaría de tener utilidad?
 - a) El sistema de protección de memoria.
 - b) El sistema de memoria virtual.
 - c) El sistema de caché de disco.
2. Tenemos un servidor empresarial que se va a dedicar exclusivamente a imprimir nóminas y cargar recibos a clientes. Estas acciones deben ejecutarse con periodicidad mensual para cada cliente. ¿Cuál de estas modalidades de procesamiento le viene mejor a este servidor?
 - a) Procesamiento por lotes.
 - b) Tiempo compartido.
 - c) Tiempo real.
3. La característica llamada «independencia del dispositivo» implica lo siguiente:
 - a) El sistema operativo opera de forma independiente a los dispositivos de E/S.
 - b) El sistema operativo transfiere los datos desde o hacia la E/S de manera independiente a las interrupciones de los dispositivos.
 - c) El sistema operativo ofrece una interfaz uniforme para acceder a los dispositivos de E/S.
4. ¿Cuál de estas funciones en C es una llamada al sistema de UNIX/Linux?
 - a) read()
 - b) fprintf()
 - c) main()
5. ¿Cuál de estos sistemas no se puede dar?
 - a) Un sistema de tiempo compartido no multitarea.
 - b) Un sistema multiusuario no multitarea.
 - c) Un sistema por lotes no multitarea.
6. Un programa de usuario ejecuta una instrucción ilegal. Tras ello, ¿cuál de estas acciones ocurre primero?
 - a) Se lanza una rutina de servicio de interrupción.
 - b) Se conmuta a modo núcleo o modo sistema.
 - c) Se aborta el programa.
7. ¿Una aplicación de usuario puede ejecutar código del núcleo?
 - a) Sí, de forma controlada, por ejemplo cuando se ejecuta una llamada al sistema.
 - b) Sí, el código del núcleo debe estar disponible para que las aplicaciones de usuario lo puedan utilizar.
 - c) No, el núcleo debe mantenerse protegido de cualquier acceso desde las aplicaciones de usuario.
8. En las arquitecturas de núcleos, ¿en qué consiste el sistema de módulos cargables?
 - a) En los núcleos monolíticos, es un sistema que permite la carga dinámica de módulos al núcleo.
 - b) En los micronúcleos, es un sistema que permite agregar al núcleo módulos con servicios adicionales ejecutables en modo usuario.
 - c) En los diseños por capas, es un sistema que permite cargar o inyectar módulos del sistema dentro de aplicaciones de usuario.
9. El sistema operativo:
 - a) Actúa en respuesta a eventos externos.
 - b) Es una interfaz entre el procesador y los dispositivos de E/S.
 - c) Nunca consume tiempo de CPU a costa de los programas de usuario.
10. ¿Cuál es la forma típica de solicitar operaciones a un *shell*?
 - a) Mediante órdenes o comandos.
 - b) Mediante interrupciones software.
 - c) Mediante llamadas una API.

Fundamentos de los Sistemas Operativos · EII / ULPGC
curso 2018/2019, convocatoria ordinaria · test de la primera parte

11. ¿Cómo se llama el componente del sistema operativo que almacena los BCP correspondientes a los procesos que están dispuestos a ejecutar instrucciones en la CPU?
 - a) Cola de preparados (*ready queue*).
 - b) Intercambiador (*swapper*).
 - c) Planificador de medio plazo (*medium-term scheduler*).
12. ¿Qué sucede a medida que el cuanto de tiempo (Q) del Round Robin va disminuyendo?
 - a) Aumenta el tiempo medio de retorno de los procesos.
 - b) Disminuye el tiempo medio de respuesta de los procesos.
 - c) Disminuye el tiempo medio de espera de los procesos.
13. Tenemos en la cola de preparados tres procesos, P1, P2 y P3, colocados en ese orden y con duraciones respectivas 1, 2 y 3 milisegundos. ¿Cuál de estos algoritmos de planificación provocará un mayor tiempo de espera medio?
 - a) FCFS.
 - b) Round Robin con Q=1 milisegundo.
 - c) SJF.
14. En la planificación de CPU, ¿en qué circunstancias es aconsejable recurrir a la técnica de «envejecimiento» (*aging*)?
 - a) Cuando utilizamos un algoritmo de planificación expulsivo.
 - b) Cuando utilizamos un algoritmo de planificación no expulsivo.
 - c) Cuando utilizamos algoritmos de planificación basados en prioridades.
15. ¿Cuándo se hace conveniente realizar equilibrado de carga (*load balancing*) en la planificación de multiprocesadores?
 - a) Cuando algunos procesadores se desequilibran porque hay demasiados procesos cargando o accediendo a una sección crítica.
 - b) Cuando existen procesadores ociosos mientras otros están cargados de trabajo.
 - c) Cuando el tipo de procesos que hay cargados en algunos procesadores no se corresponde con su arquitectura.
16. Tenemos un sistema de un solo procesador, al que han llegado tres procesos de una única ráfaga de CPU, con duraciones respectivas de 1, 2 y 4 milisegundos. Una vez planificados por la CPU, se han medido sus tiempos de espera han resultado ser de cero milisegundos en los tres casos. ¿Qué explicación puede tener este resultado?
 - a) Los procesos han llegado en instantes diferentes, de forma que nunca han entrado en conflicto por la CPU.
 - b) El algoritmo de planificación es un Round Robin con un valor de Q muy pequeño, de forma que se reduce al mínimo el tiempo de espera.
 - c) Hay algún error de medida, ya que el tiempo de espera no puede ser nulo al mismo tiempo para los tres procesos.
17. ¿Cuál de estos algoritmos de planificación de CPU es por su propia naturaleza NO expulsivo?
 - a) FCFS.
 - b) Round Robin.
 - c) SJF.
18. De los algoritmos que aquí se citan, ¿cuál puede provocar inanición de alguna clase de procesos?
 - a) FCFS.
 - b) Round Robin.
 - c) SJF.
19. El problema de la sección crítica no se puede dar:
 - a) En un sistema con un solo procesador.
 - b) En un sistema de procesamiento por lotes.
 - c) En un sistema sin multiprogramación.

Fundamentos de los Sistemas Operativos · EII / ULPGC
curso 2018/2019, convocatoria ordinaria · test de la primera parte

20. ¿Cuál es la finalidad principal de las instrucciones atómicas, como «test-and-set»?
- Facilitar la construcción de soluciones eficientes para el acceso a datos compartidos.
 - Aumentar la velocidad en el acceso a la memoria principal.
 - Aumentar la velocidad de ejecución de los hilos en sistemas multiprocesadores.
21. ¿Cómo se llama la propiedad de un algoritmo concurrente que indica que el resultado de su ejecución puede ser en parte imprevisible?
- Exclusión mutua.
 - Atomicidad.
 - No determinismo.
22. ¿Qué diferencia hay entre concurrencia y paralelismo?
- El paralelismo exige la existencia de más de un procesador, mientras que la concurrencia se puede dar con un único procesador.
 - El paralelismo es una característica del software, mientras que la concurrencia es una característica del hardware.
 - El paralelismo está asociado sistemas con procesos pesados, mientras que la concurrencia está asociada a sistemas con procesos ligeros (hilos).
23. Tenemos una variable entera «i» que actualmente vale 1. En un momento dado, tres hilos ejecutan concurrentemente la instrucción $i = i+1$. ¿Cuál es el valor final de «i» una vez que se han ejecutado las tres instrucciones?
- 4.
 - Puede ser 1, 2, 3 o 4.
 - Puede ser 2, 3 o 4.
24. ¿Con qué finalidad hemos definido las instrucciones DORMIR y DESPERTAR en la construcción de algoritmos concurrentes?
- Para programar alarmas que mantengan dormido a un hilo durante un tiempo.
 - Para evitar la inanición de procesos que no disfrutan del procesador un tiempo suficiente.
 - Para no tener que resolver las esperas mediante bucles de espera activa.