UNIVERSIDAD DE MURCIA

Introducción a los Sistemas Operativos

Examen final de teoría (temas 1-4). Test A. 24/11/2023

Grupo: ☐ 1 ☐ 2 ☐ 3 ☐ PCEO	DNI:
Apellidos:	Nombre:

Instrucciones para realizar el test:

1. La puntuación de este test es de 7 puntos. **La parte de problemas de este examen solo se calificará cuando en este test haya 16 o más respuestas correctas tras descontar las penalizaciones.**
2. Marca con ≪X≫ y a bolígrafo en la tabla de respuestas la opción que creas correcta.
3. Una respuesta incorrecta resta una correcta. Una pregunta sin contestar ni suma ni resta.
4. **Debes entregar el enunciado del examen al acabar**.
5. Supondremos que 1 KiB = 2^{10} bytes, 1 MiB = 2^{20} bytes y 1 GiB = 2^{30} bytes. También supondremos que **no existe caché de disco** en aquellos casos en los que haya lecturas y/o escrituras de disco.

Respuestas del test

Nº	a	b
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		
21		
22		
23		
24		
25		

Nº	a	b
26		
27		
28		
29		
30		
31		
32		
33		
34		
35		
36		
37		
38		
39		
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		
50		

Preguntas:

1. Las llamadas al sistema:
   a) Permiten a los programas de usuario tener acceso a una serie de funcionalidades y servicios proporcionados por el sistema operativo que simplifican el uso de la máquina.
   b) Las proporciona el conjunto de bibliotecas que acompaña al compilador, sin tener necesariamente que estar implementadas en el sistema operativo, sino como una forma de evitar que tengamos que reescribir código que necesitamos frecuentemente.

2. En el mecanismo de multitarea:
   a) Es necesario que el planificador de procesos sea apropiativo de cara a poder ir pasando el uso de la CPU de un proceso a otro.
   b) Un proceso puede utilizar la CPU todo el tiempo que precise de manera consecutiva, sin cederla, independientemente de la cantidad y de la prioridad de los procesos que estén esperando en la cola de listos.

3. En los sistemas operativos distribuidos:
   a) Se suele proporcionar tolerancia a fallos de una máquina, asumiendo otra máquina del sistema su labor de forma transparente al usuario.
   b) El usuario es el encargado de decidir en qué máquina del sistiema distribuido se ejecutarán sus procesos.

4. El mecanismo de interrupciones periódicas:
   a) Permite evitar que un proceso de usuario que no hace llamadas al sistema monopolice la CPU.
   b) Se encarga de atender las interrupciones que se producen periódicamente en los dispositivos de E/S, como puede ser el disco duro cuando se le solicita que lea una secuencia de bloques.

5. Un sistema operativo basado en el modelo cliente-servidor (microkernel):
   a) Implementa los servicios que acceden directamente al hardware como procesos en modo usuario, de manera que si un servicio falla, no corrompe el núcleo.
   b) Permite la posibilidad de que un proceso cliente pueda solicitar servicios a un proceso servidor sin que ambos tengan que estar ejecutándose en la misma máquina física.

6. En una CPU con más de un núcleo:
   a) La planificación de la CPU puede provocar que un proceso vaya ejecutándose en diferentes núcleos durante su vida.
   b) Si asignamos dos hilos de un mismo proceso a dos núcleos diferentes, conseguimos que se ejecuten de forma aislada sin compartir recursos entre ellos.

7. En relación con la ejecución de procesos, el pseudoparalelismo:
   a) Sigue estando presente aunque la CPU tenga varios núcleos que permiten paralelismo real.
   b) Es otra forma de llamar al paralelismo que usa hilos en modo usuario.

8. Durante la codificación de un programa:
   a) Lo habitual es que en nuestro código tengamos en cuenta que hay muchos procesos ejecutándose en paralelo junto con el nuestro.
   b) No podemos presuponer con exactitud el tiempo que tardará en ejecutarse cualquier parte del código.

9. En un sistema Unix, tenemos un proceso P ejecutándose en modo usuario que realiza una llamada al sistema. Durante la ejecución del código de dicha llamada, el proceso P se bloquea, pasándose a ejecutar el proceso Q en modo núcleo que, tras un cierto tiempo, pasa a modo usuario. En todo este transcurso:
   a) Se han producido 2 cambios de modo, 1 cambio de proceso y 3 cambios de contexto.
   b) Se han producido 3 cambios de modo, 2 cambios de proceso y 1 cambio de contexto.

10. En un sistema Unix, la realización de una llamada al sistema `fork()` por parte de un proceso P:
    a) Provoca la creación de un nuevo proceso Q independiente que se ejecuta siempre y cuando P esté vivo. Si P muere, también muere Q.
    b) Crea un nuevo proceso Q que contendrá esencialmente el mismo código y datos que P, por lo que habrá que tener presente en el código la forma de hacer que cada proceso se encargue de la labor que le corresponde.

11. En relación con los cambios de estado de un proceso:
    a) Un proceso en el estado Listo solo puede pasar al estado Bloqueado si pasa previamente por el estado de Ejecución.
    b) El número de procesos en estado Ejecución depende del tipo de planificador, no del número de núcleos que tenga nuestra CPU.

12. Tenemos únicamente 3 procesos en nuestra máquina: P, Q y R. En el momento t, P está Bloqueado, Q en Ejecución y R en estado Listo. Entonces, los estados de estos procesos en el instante $t+1$ podrían ser:
    a) P:Listo, Q:Bloqueado, R:Listo.
    b) P:Listo, Q:Bloqueado, R:Ejecución.

13. En Unix, cuando un proceso crea un nuevo proceso hijo:
    a) El BCP del proceso hijo es una copia idéntica del BCP del proceso padre.
    b) Ambos procesos compartirán el mismo segmento de código.

14. Tenemos una CPU con 2 núcleos y un sistema operativo sin soporte para hilos. Si un proceso con 2 hilos quiere utilizar ambos núcleos de la CPU:
    a) No será posible, y ambos hilos tendrán que ejecutarse necesariamente en un mismo núcleo.
    b) No tendrá que hacer nada especial porque el SO se encarga de asignar cada hilo a un núcleo diferente.

15. Si el sistema operativo tiene implementados el soporte para hilos dentro del núcleo:
    a) Cada uno de los hilos se podrá bloquear de forma independiente.
    b) Cada proceso puede tener un planificador que se adapte a los requerimientos de sus hilos.

16. En relación con la planificación de procesos, teniendo en cuenta la meta de equidad:
    a) El planificador intenta conseguir que el tiempo de CPU que reciben todos los procesos sea aproximadamente el mismo.
    b) El planificador asignará a cada proceso más o menos tiempo de CPU en función de las características del propio proceso.

17. En relación a la planificación de procesos, el tiempo de espera de un proceso es:
    a) El tiempo que pasa un proceso en el estado Bloqueado.
    b) El tiempo que pasa un proceso en el estado Listo.

18. En relación con la planificación de procesos:
    a) Un proceso se dice que está limitado por CPU cuando necesita más núcleos de CPU para ejecutarse de los que ofrece la CPU.
    b) Un proceso limitado por CPU tendrá normalmente muchas ráfagas largas de CPU.

19. Utilizando el algoritmo de planificación de procesos FCFS:
    a) Se puede producir un efecto convoy que perjudica a los procesos limitados por E/S.
    b) Minimizamos el tiempo de respuesta de los procesos.

20. En un planificador Round-Robin:
    a) Tenemos que definir un mecanismo adicional para evitar que haya procesos que jamás cojan la CPU.
    b) El tiempo que tarda un proceso en coger la CPU está acotado.

21. En un planificador de múltiples colas con realimentación, se cumple que:
    a) Un proceso de una cierta cola, tras ejecutarse y bloquearse, puede desbloquearse en una cola diferente.
    b) Todos los procesos se ejecutan una cantidad de tiempo proporcional a la prioridad de su cola.

22. Usando un planificador de procesos SRTF:
    a) El proceso en ejecución no puede perder la CPU mientras no se bloquee y no entre ningún proceso en la cola de listos que no estuviera cuando se le asignó la CPU.
    b) Se consigue minimizar el tiempo que un proceso se encuentra bloqueado.

23. Si un usuario no autorizado modifica maliciosamente el software de acceso a una base de datos provocando que ningún usuario pueda acceder a una parte de la información almacenada:
    a) Se vulnera el requisito de integridad del software y el de disponibilidad de los datos.
    b) Se vulnera el requisito de integridad de los datos y el de disponibilidad del software.

24. La política de seguridad de un sistema:
    a) Es la encargada de implementar los mecanismos de seguridad del sistema.
    b) Es independiente de los mecanismos de seguridad del sistema, siempre que dichos mecanismos sean suficientemente generales.

25. En un dominio de protección:
    a) Hay una entrada por cada usuario del sistema en la que se indica qué operaciones puede realizar dicho usuario sobre los diferentes objetos del sistema.
    b) Hay un conjunto de entradas que asocian objetos del sistema con las operaciones que se pueden realizar sobre dicho objeto.

26. Un problema que aparece al implementar listas de posibilidades es:
    a) que solamente existe la lista de posibilidades de un proceso mientras se está ejecutando, por lo que se necesita una forma de preservar los permisos cuando el ordenador se apaga.
    b) que no es factible que un proceso ceda una posibilidad a otro proceso, cuando esto es una práctica común entre un proceso y sus hijos.

27. Cuando un usuario U intenta conectarse a un sistema Unix y proporciona una contraseña P:
    a) El sistema operativo tiene un fichero encriptado donde se almacenan las contraseñas de manera que no sean accesibles por extraños.
    b) El sistema operativo debe buscar en el fichero apropiado el tipo de encriptación y la base que tiene almacenados para U.

28. En relación con la gestión de contraseñas en Unix, si dos usuarios cambian sus contraseñas y casualmente deciden utilizar la misma:
    a) Sus contraseñas cifradas serán distintas porque lo que se encripta es la concatenación de la contraseña y el UID, que es diferente para cada usuario del sistema.
    b) Sus contraseñas cifradas serán distintas porque lo que se encripta es la concatenación de la contraseña y un valor aleatorio denominado base.

29. En un sistema Unix tenemos un fichero llamado `anotaciones.txt`, con permisos `rw- r-- ---`, cuyo usuario propietario es `pepe` y cuyo grupo propietario es `empleados`. Tenemos también otro usuario, llamado U (el UID de U es distinto del UID de `pepe`), que pertenece a los grupos `pepe`, X y Z. El usuario U lanza un proceso P, por lo que:
    a) Si el GID de alguno de los grupos `pepe`, X o Z es igual al GID de `empleados` entonces P podrá leer `anotaciones.txt`.
    b) Si el GID de `pepe` es igual al GID de `empleados` entonces P podrá modificar `anotaciones.txt`.

30. En un sistema Unix tenemos el fichero `anotaciones.txt`, con permisos `rws --- --x`, cuyo usuario propietario es `pepe` y cuyo grupo propietario es `empleados`. El usuario U lanza un proceso P. Entonces:
    a) P podrá leer `anotaciones.txt` solamente si se cumple que el UID de U es igual al UID de `pepe`.
    b) P podrá leer `anotaciones.txt` gracias al bit SETUID de este fichero, que hará que temporalmente tome el papel de `root` que puede acceder a cualquier fichero.

31. Los ficheros especiales de caracteres:
    a) Constan de líneas de texto que se pueden ver y modificar tal cual en un editor de texto común y solo pueden accederse secuencialmente.
    b) Están asociados a ciertos dispositivos de E/S.

32. En relación con los ficheros en Unix:
    a) Las tuberías son ficheros especiales que por un extremo se escriben byte a byte, secuencialmente, pero por el otro se puede acceder a cualquier byte almacenado en dicha tubería.
    b) El tamaño de un fichero en Unix se almacena en su nodo-i.

33. En relación con los directorios en Unix:
    a) La llamada al sistema `writedir` permite modificar entradas de un directorio.
    b) La llamada al sistema `unlink` puede, bajo ciertas circunstancias, modificar el contenido del nodo-i correspondiente al fichero que recibe como parámetro sin eliminarlo.

34. Con respecto a los enlaces y las rutas en Unix:
    a) Un enlace simbólico siempre contiene la ruta absoluta al fichero enlazado, de esa manera aunque movamos el enlace, sigue funcionando.
    b) Un enlace simbólico contiene la ruta exacta usada para crear el enlace, de esa manera, si movemos el enlace y lo hemos creado con una ruta relativa, puede dejar de funcionar.

35. En la implementación de un Sistema de Ficheros por parte del Sistema Operativo:
    a) El dispositivo de almacenamiento secundario se gestiona como un array lineal de bloques.
    b) Por cada fichero F, el Sistema Operativo tiene almacenada tantas direcciones de disco como bytes tenga F con el objetivo de poder acceder a cualquiera de estos bytes de manera directa (acceso aleatorio).

36. Dado un sistema de ficheros implementado mediante asignación adyacente, si el bloque D (contando desde 0) de disco almacena el bloque X (contando desde 0) del fichero F, entonces:
    a) El bloque $(D - X)$ de disco seguro que almacena un bloque de F.
    b) El bloque $(D + 1)$ de disco seguro que almacena un bloque de F.

37. En una implementación de ficheros mediante asignación contigua en un disco con bloques de 4 KiB, para modificar el último byte de un fichero de 40 960 bytes:
    a) Se tendrán que realizar dos operaciones con bloques de disco: 1 lectura y una escritura del bloque 9 del fichero.
    b) Se tendrá que realizar una única operación de escritura: un nuevo bloque al final del fichero (bloque 10) conteniendo el byte.

38. En una implementación de ficheros mediante lista ligada sin índice, con bloques de B bytes y direcciones de disco de d bytes, tenemos un fichero con T bytes. El número de lecturas de bloque que tenemos que realizar para leer el último bloque del fichero es:
    a) $\lceil T/(B-d) \rceil$
    b) $\lceil (T-d)/B) \rceil$

39. Sean dos discos duros D1 y D2 idénticos formateados con un sistema de ficheros con asignación mediante lista ligada sin índice. D1 usa un tamaño de bloque B y D2 usa un tamaño de bloque $2 \times B$. Copiamos un mismo fichero F a ambos discos, por lo que en ambos está solamente ese fichero. Entonces:
    a) La fragmentacion interna en ambos sistemas seguro que es la misma.
    b) La fragmentación externa en ambos sistemas seguro que es la misma.

40. Sean dos discos duros D1 y D2 idénticos. Formateamos D1 con un sistema de ficheros con asignación mediante lista ligada sin índice y D2 con un sistema de ficheros con asignación mediante lista ligada con índice. D1 usa un tamaño de bloque B y D2 usa un tamaño de bloque $2 \times B$. Si suponemos que en caso de D2 el índice se encuentra ya en memoria, entonces:
    a) El número de accesos a disco que se necesitan en D1 para llegar al último bloque del fichero será la mitad de los que se necesitan en D2.
    b) El número de accesos a disco que se necesitan para llegar al último bloque del fichero en D2 es independiente del tamaño del bloque.

41. En relación con los sistemas de ficheros, un nodo-i en un sistema de ficheros con estructura Unix tradicional:
    a) Es un bloque de disco que contiene la lista de bloques de disco que componen un fichero.
    b) Tendrá un tamaño fijo, independiente del tamaño de los ficheros y del número total de ficheros existentes en el sistema de ficheros.

42. Tenemos un sistema de ficheros con bloques de 2 KiB, direcciones de bloque de 4 bytes y el nodo-i tiene una estructura Unix tradicional (10 punteros directos, 1 puntero a BSI, 1 puntero a BDI y 1 puntero a BTI). Entonces, para el fichero más grande que se podría gestionar, se necesitarían:
    a) 263 171 bloques de metadatos.
    b) 263 181 bloques de metadatos.

43. Tenemos un sistema de ficheros Unix con bloques de 1 KiB, direcciones de bloque de 4 bytes y el nodo-i tiene una estructura tradicional (10 punteros directos, 1 puntero a BSI, 1 puntero a BDI y 1 puntero a BTI). Entonces, ¿cuántos bloques en total se necesitarían traer de disco a memoria principal para poder leer el byte X de un fichero, si suponemos que el nodo-i ya está en memoria principal?
    a) Mínimo: 1 bloque. Máximo: 4 bloques.
    b) Mínimo: 1 bloques. Máximo: 66 052 bloques.

44. Tenemos la ruta `/home/alumno`, donde `alumno` es un enlace que acabamos de crear al directorio `/externo/alumno` donde hemos movido nuestro HOME para tener más espacio. Se sabe que:
    - Solo están en memoria el nodo-i del directorio raíz y el bloque de datos del directorio raíz.
    - Los ficheros y directorios tienen un solo bloque cada uno.
    - Cada nodo-i se encuentra en un bloque diferente.

    Entonces, ¿cuántos bloques necesitaremos leer de disco para traer a memoria el primer bloque de datos del fichero `/home/alumno/doc.txt`?
    a) 10 si el enlace es simbólico.
    b) 6 si el enlace es físico.

45. En cuanto a los tipos de enlaces en Unix, es cierto que ...
    a) Es posible crear un enlace simbólico a un fichero regular que se encuentra en un sistema de ficheros de un disco distinto de donde se creará el enlace.
    b) Es posible crear un enlace físico a un fichero regular que se encuentra en un sistema de ficheros de un disco distinto de donde se creará el enlace.

46. En Unix, si creamos un fichero F y, a continuación, creamos 6 enlaces a dicho fichero F, de los que 4 son simbólicos y 2 son físicos, ¿cuántos nodos-i se habrán creado en total?
    a) 3.
    b) 5.

47. Con respecto a las diferencias entre los mapas de bits y las listas de bloques para llevar el control de los bloques libres de un disco, es cierto que:
    a) El número de bloques que se necesitan para mantener un mapa de bits *siempre* es menor que el número de bloques que se necesitan para llevar la lista de bloques libres, debido a que almacenan mucho más eficientemente la información (solo necesitan un bit por bloque, mientras que la lista necesita un entero).
    b) Por mucho que ocupe una lista de bloques libres, en realidad no nos resta espacio de la zona de datos del disco, puesto que, conforme vamos consumiendo bloques, la lista también va desapareciendo.

48. En relación con el control de bloques libres en un sistema de ficheros con asignación mediante lista ligada con índice (FAT):
    a) Cada entrada de la FAT podría usarse para indicar que el bloque correspondiente está libre si almacenamos en ella un valor especial, que no sea un número de bloque válido, como forma de expresarlo.
    b) Necesitamos una estructura adicional, bien sea un mapa de bits o una lista de bloques para poder gestionar los bloques libres, como en cualquier otro sistema de ficheros.

49. La caché de disco:
    a) Se utiliza para mantener el mayor tiempo posible en memoria principal algunos bloques de metadatos como los bloques doblemente indirectos, ya que estos bloques suelen tener muchas referencias en intervalos cortos de tiempo.
    b) Conlleva un control extra para forzar la escritura frecuente de bloques modificados que sean importantes para la consistencia del sistema de ficheros.

50. En un sistema de ficheros Unix tradicional:
    a) El número de bloques de disco que se dedican al mapa de bits de nodos-i es habitualmente inferior al que se dedica al mapa de bits de bloques.
    b) La zona dedicada a nodos-i contiene, además de los propios nodos-i, los bloques BSI, BDI y BTI.

UNIVERSIDAD DE MURCIA

Introducción a los Sistemas Operativos

Examen de teoría: test (A). 24 de noviembre de 2023

Grupo:	DNI:
Apellidos:	Nombre:

Respuestas del PROFESOR

Nº	a	b
1	X	
2	X	
3	X	
4	X	
5		X
6	X	
7	X	
8		X
9	X	
10		X
11	X	
12		X
13		X
14	X	
15	X	
16		X
17		X
18		X
19	X	
20		X
21	X	
22	X	
23	X	
24		X
25		X

Nº	a	b
26	X	
27		X
28		X
29	X	
30	X	
31		X
32		X
33		X
34		X
35	X	
36	X	
37	X	
38	X	
39		X
40		X
41		X
42	X	
43	X	
44	X	
45	X	
46		X
47		X
48	X	
49		X
50	X	