

Grupo: <input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> PCEO	DNI:
Apellidos:	Nombre:

Instrucciones para realizar el test:

- La puntuación de este test es de 7 puntos.
- Marca con «X» y a bolígrafo en la tabla de respuestas la opción que creas correcta.
- Una respuesta incorrecta resta una correcta. Una pregunta sin contestar ni suma ni resta.
- Debes entregar el enunciado del examen al acabar.**
- Supondremos que 1 KiB = 2¹⁰ bytes, 1 MiB = 2²⁰ bytes y 1 GiB = 2³⁰ bytes. También supondremos que **no existe caché de disco** en aquellos casos en los que haya lecturas y/o escrituras de disco.

Nº	a	b	Respuestas del test	Nº	a	b
1				26		
2				27		
3				28		
4				29		
5				30		
6				31		
7				32		
8				33		
9				34		
10				35		
11				36		
12				37		
13				38		
14				39		
15				40		
16				41		
17				42		
18				43		
19				44		
20				45		
21				46		
22				47		
23				48		
24				49		
25				50		

Preguntas:

- El sistema operativo toma el control de la CPU:
 - Cuando se produce una llamada al sistema, una excepción o una interrupción.
 - Cuando se produce una llamada al sistema o una interrupción, pero no en caso de excepción, dado que en esta situación el hardware aborta directamente el programa.
- La característica del hardware en la que se apoya directamente el sistema operativo para impedir que un proceso se quede con la CPU por tiempo indefinido es:
 - La disponibilidad de dos modos de funcionamiento (modo núcleo y modo usuario) en la CPU.
 - Las interrupciones periódicas generadas por alguno de los relojes hardware con que debe contar el sistema.
- El mecanismo de los buffers que permite el solapamiento de la E/S de un programa con el cómputo del mismo programa requiere de la utilización de:
 - Un dispositivo de acceso aleatorio, como un disco, para ir depositando en una zona concreta del mismo los datos que se van calculando.
 - Una zona concreta de memoria desde donde ir obteniendo los datos que se van necesitando.
- Dados dos sistemas A y B, cuya única diferencia es que el sistema operativo de A implementa multi-programación y el de B multitarea, entonces el tiempo que necesitamos para completar la ejecución de un lote de 10 trabajos:
 - Va a ser mayor o igual en A que en B.
 - Va a ser mayor o igual en B que en A.
- Con respecto a la ejecución del intérprete de órdenes **bash** empleado en las prácticas:
 - Se realiza en modo usuario del procesador puesto que para el núcleo del sistema operativo su ejecución no difiere de la de cualquier otra aplicación de usuario.
 - Se realiza en modo núcleo del procesador puesto que forma parte del núcleo del sistema operativo.
- En un sistema operativo basado en micronúcleo (organización cliente-servidor), es cierto que:
 - Tanto los procesos de tipo cliente como los de tipo servidor se ejecutan con el procesador en modo usuario.
 - Tanto el micronúcleo como los procesos de tipo servidor se ejecutan con el procesador en modo núcleo.
- En un sistema operativo UNIX, la ejecución de la llamada al sistema **fork()** por parte de un proceso:
 - Crea un proceso hijo que es idéntico al proceso que ejecuta la llamada al sistema, salvo que se le pase como argumento un binario, en cuyo caso lo usará para establecer el código y los datos del hijo en el momento de la creación.
 - Crea siempre un proceso hijo que es idéntico al proceso que ejecuta la llamada al sistema, si bien el PID del padre y del hijo son diferentes.
- En un sistema operativo UNIX, cuando un proceso P realiza una llamada al sistema y hasta que comienza a ejecutarse el código correspondiente del sistema operativo se producen:
 - Un cambio de modo y un cambio de contexto.
 - Un cambio de proceso, un cambio de modo y un cambio de contexto.
- Un ejemplo de terminación voluntaria de un proceso como consecuencia de un error sería:
 - Que el proceso compruebe que el número de argumentos pasados no es el requerido y no pueda continuar con la ejecución.
 - Que el proceso trate de realizar un acceso a memoria a una dirección para la cual no tenga permiso.

10. En cuanto a los estados en los que puede estar un proceso, es cierto que:
- Un proceso P podría pasar directamente del estado «Bloqueado Suspendido» al estado «Saliente».
 - Un proceso P podría pasar directamente del estado «Bloqueado» a «En ejecución» cuando la cola de procesos listos estuviese vacía.
11. Un proceso que se encuentra en estado «Bloqueado Suspendido», cambiará de estado...:
- Únicamente cuando el planificador a medio plazo (PMP) así lo decida.
 - Cuando el planificador a medio plazo (PMP) así lo decida o cuando ocurra el evento de E/S que estaba esperando.
12. En cuanto a los estados en los que puede estar un proceso, es cierto que:
- Para que un proceso en estado «Bloqueado Suspendido» pueda hacer uso de la CPU, tiene que pasar antes por otros dos estados diferentes.
 - Para que un proceso en estado «Listo Suspendido» pueda hacer uso de la CPU, tiene que pasar antes por otros dos estados diferentes.
13. En cuanto a la tabla de procesos:
- Tiene una entrada por cada uno de los procesos en estado «Listo», pero no para los procesos en estado «Listo Suspendido» o «Bloqueado Suspendido».
 - Tiene una entrada por cada uno de los procesos con independencia del estado en el que se encuentren.
14. En cuanto al bloque de control de un proceso, es cierto que:
- Almacena los valores de los registros de la CPU tal y como estaban la última vez que al proceso se le quitó la CPU.
 - Almacena los datos de la pila del proceso tal y como estaban la última vez que al proceso se le quitó la CPU.
15. En UNIX, dados dos procesos, P y Q, si suponemos que ambos realizan una llamada al sistema en el mismo instante de tiempo:
- Primero se completará la llamada al sistema de uno de ellos y después la del otro, pero al haber una única copia del SO en memoria, nunca se podrá simultanear la ejecución de las dos llamadas al sistema.
 - Ambas llamadas al sistema podrían ejecutarse simultáneamente (o concurrentemente)
16. Dado un proceso P con dos hilos H1 y H2, podemos decir que:
- H1 y H2 van a ser exactamente iguales, por lo que ejecutan las mismas instrucciones con los mismos datos.
 - H1 y H2 comparten los datos del proceso pero pueden ejecutar instrucciones diferentes.
17. Dado un proceso, P, con dos hilos, H1 y H2, la CPU podrá pasar de ejecutar H1 a H2 (o viceversa) **sin** la intervención del sistema operativo:
- Cuando los hilos estén implementados a nivel usuario.
 - Cuando los hilos estén implementados en el núcleo del sistema operativo.
18. Dado un proceso P con 4 hilos ejecutándose sobre una CPU con 4 núcleos de procesamiento (*cores*), los 4 hilos podrán ejecutarse simultáneamente (uno en cada núcleo):
- Solamente si se implementa soporte para los hilos a nivel de núcleo en el sistema operativo.
 - Siempre, independientemente de cómo estén implementados los hilos.
19. Para un planificador de procesos cuya meta fundamental es la de maximizar la eficiencia en el uso de la CPU:
- Es más conveniente usar un algoritmo de planificación apropiativo.
 - Es más conveniente usar un algoritmo de planificación no apropiativo.
20. En relación con la planificación de procesos, el tiempo de espera de un proceso es:
- El tiempo que hay que esperar desde que empieza el proceso hasta que se obtienen sus resultados.
 - El tiempo que pasa el proceso en la cola de procesos listos.
21. En relación a la planificación de procesos, la planificación SJF (*Shortest Job First*):
- Es apropiativa, puesto que debe asegurar que el proceso en ejecución es en todo momento el de ráfaga de CPU más corta de entre todos los de la cola de listos y el propio proceso que actualmente tiene en la CPU.
 - Es no apropiativa, pues al proceso que está usando la CPU nunca se le puede quitar su uso por muy larga que sea su ráfaga actual y por muy cortas que sean las ráfagas de CPU previstas para los procesos que están en la cola de listos.
22. La planificación *Round Robin*:
- Es no apropiativa, pues hasta que el proceso que está en ejecución no consuma su quantum, el sistema operativo no se puede apropiar de la CPU para dársela a otro proceso.
 - Es apropiativa, pues el proceso que está en ejecución puede ser desalojado de la CPU sin haberse bloqueado ni haber finalizado.
23. De entre los algoritmos de planificación de la CPU vistos en clase, la situación de inanición (*starvation*) por la que un proceso podría no llegar a usar nunca la CPU, podría darse con:
- SRTF y SJF.
 - FCFS y planificación con prioridades estáticas.
24. Dado un algoritmo de planificación *Round Robin* con un valor de quantum de **q** unidades de tiempo (u.t.) y un sistema en el que hay en todo momento **N** procesos (no necesariamente los mismos) en la cola de listos, el tiempo que transcurre desde que un proceso agota su quantum y se le quita la CPU, hasta que la vuelve a tomar es siempre:
- Exactamente igual a $q \times (N-1)$ u.t.
 - Menor o igual a $q \times (N-1)$ u.t.
25. Tenemos una planificación de procesos SJF con estimación de tiempo de ráfaga de CPU. Hay 2 procesos en la cola de listos: P1 y P2. En la última ráfaga de CPU, las estimaciones para P1 y P2 fueron E1 y E2, respectivamente, y los tiempos reales T1 y T2, respectivamente. Entonces:
- Si $E1=E2$ y $T1 < T2$ pasará ejecutarse P1.
 - Si $E1 > E2$ y $T1=T2$ pasará a ejecutarse P1.
26. Dados dos procesos P1 y P2, si en el instante «t» P2 comienza la ejecución del segundo de los tics de una ráfaga de CPU de duración 5 y se desbloquea P1 cuya siguiente ráfaga de CPU tiene una duración de 2 tics, en el instante «t+1» se ejecutará:
- P2 si el planificador es SRTF «oráculo».
 - P1 si el planificador es *Round Robin* con $q=2$ tics.
27. Considera dos procesos P1 y P2 en estado «Bloqueado» y el esquema de planificación *Round Robin*: si en el instante «t» la CPU está ocupada y ambos se desbloquean para ejecutar una ráfaga de CPU de 2 tics (P1) y 5 tics (P2), en el instante «t+1», suponiendo que la CPU queda libre y que estos son los únicos procesos en la cola de listos, se ejecutará:
- P1 o P2 dependiendo del orden en el que se inserten en la cola de listos en el instante «t».
 - P1 siempre, por tener una duración menor de ráfaga de CPU.

28. Supón un esquema de planificación «apropiativo con prioridades dinámicas» (a mayor número, mayor prioridad) en el que, para evitar la inanición, la prioridad de un proceso se aumenta en 2 cada tic que el proceso está en la cola de listos. Si al comienzo del tic $\langle t \rangle$ se empieza a ejecutar un proceso P1 con prioridad 4 con una ráfaga de CPU de duración 5 tics, y en ese mismo momento se desbloquea y se inserta en la cola de listos un proceso P2 con prioridad 1 y ráfaga de CPU de 4 tics, P2 pasará a usar la CPU:
- En el instante $\langle t+2 \rangle$.
 - En el instante $\langle t+3 \rangle$.
29. Considera una planificación de procesos apropiativa con 2 colas (cola 0 de mayor prioridad, cola 1 de menor prioridad) sin posibilidad de que los procesos cambien de cola. Ambas colas se planifican internamente con sendos algoritmos *Round Robin* con quantum de 2 tics. Si en este instante a la cola 0 llega el proceso P0 y a la cola 1 llega el proceso P1, queriendo ambos procesos ejecutar sendas ráfagas de CPU de 4 tics. ¿Qué ocurrirá en los próximos 4 tics?:
- Se ejecutará P0 durante 4 tics y, entonces, se cederá la CPU a P1.
 - Se ejecutará P0 durante 2 tics y, entonces, se cederá la CPU a P1 durante otros 2 tics.
30. Considera una planificación de procesos apropiativa con 2 colas (cola 0 de mayor prioridad, cola 1 de menor prioridad). Ambas colas se planifican internamente con sendos algoritmos *Round Robin*, el de la cola 0 usa un quantum de 2 tics, mientras que el de la cola 1 usa un quantum de 4 tics. Si un proceso de la cola 0 consume su quantum sin bloquearse se pasa a la cola 1. Si en este instante llega a la cola 1 el proceso P1 y dos tics después llega a la cola 0 el proceso P0, queriendo ambos procesos ejecutar sendas ráfagas de CPU de 4 tics, ¿cómo será la ejecución de dichos procesos hasta que ambos ejecuten sus ráfagas de CPU por completo?:
- Dos tics de P1, dos tics de P0, dos tics de P1 y dos tics de P0.
 - Dos tics de P1, cuatro tics de P0 y dos tics de P1.
31. Una invención de la información es:
- Una amenaza a la integridad de la información.
 - Una amenaza a la confidencialidad de la información.
32. El concepto de dominio de protección se asocia a:
- Los procesos y define los objetos a los que puede acceder cada uno de ellos y con qué permisos.
 - Los objetos y define los procesos que pueden acceder a cada uno de ellos y con qué permisos.
33. En cuanto a las listas de control de acceso (ACL):
- Cada una está asociada con un dominio de protección y establece los objetos que son accesibles desde dicho dominio y las operaciones realizables sobre cada uno de ellos.
 - Cada una está asociada con un objeto, y establece los dominios de protección desde los que se puede acceder a dicho objeto y las operaciones realizables sobre el mismo.
34. En UNIX, cada vez que un proceso quiere leer o escribir un fichero que ya tiene abierto:
- Se chequea la ACL del fichero para ver si el proceso dispone de los permisos necesarios para poder realizar la operación.
 - Se comprueba la lista de posibilidades del proceso para ver si puede realizar la operación sobre el fichero.
35. En UNIX, durante la realización del proceso de *login* por parte de un usuario:
- Se accede únicamente al fichero `/etc/passwd` para la validación de la contraseña introducida por el usuario.
 - Además de `/etc/passwd`, es necesario también acceder al fichero `/etc/shadow`.
36. En UNIX, si desde el intérprete de órdenes se ejecuta el programa `passwd`, cuyo fichero ejecutable pertenece al usuario `root` y tiene el bit SETUID activado, entonces, el proceso resultante:
- Ejecutará el programa `passwd` con el procesador en modo usuario.
 - Ejecutará el programa `passwd` con el procesador en modo núcleo.
37. Un fichero especial de caracteres:
- Está asociado con un dispositivo de entrada/salida.
 - Es un fichero regular de tipo texto.
38. La creación de la entrada de directorio para un nuevo fichero regular, se realiza:
- A través de la llamada al sistema `Create` usada para crear el fichero.
 - Mediante la llamada al sistema `Writedir` aplicada al directorio en el que quedará el fichero.
39. Dado un fichero regular con tamaño T bytes, si el tamaño de bloque lógico es B bytes y se cumple que $T > B$, entonces el número de bloques lógicos que se utilizarán para almacenar su contenido:
- Será mayor para la estrategia de implementación de ficheros de lista ligada con índice que para la basada en nodos-i.
 - Será igual para las estrategias de implementación de ficheros de asignación contigua y lista ligada con índice.
40. En cuanto a la fragmentación interna:
- Puede aparecer en la estrategia de implementación de ficheros de asignación contigua pero no en la basada en nodos-i.
 - Puede aparecer tanto en la estrategia de implementación de ficheros basada en nodos-i como en las basadas en lista ligada.
41. En una implementación de ficheros mediante lista ligada sin índice, el valor del byte Z del bloque X ($X \neq 0$) de un fichero cuyo primer bloque (el bloque 0) se encuentra en la dirección D de disco puede obtenerse:
- Leyendo X+1 bloques del disco.
 - Leyendo D+X bloques del disco.
42. Dado un sistema de ficheros tipo UNIX, suponiendo la estructura de nodos-i vista en clase con un total de 13 punteros, que todas las entradas de directorio tienen una longitud de exactamente 256 bytes y que el tamaño de bloque lógico es de 1 KiB, un directorio con 50 entradas (sin incluir las entradas $\langle . \rangle$ y $\langle .. \rangle$) ocupa un total de:
- 13 bloques lógicos de la zona de datos.
 - 14 bloques lógicos de la zona de datos.
43. Supón dos sistemas de ficheros, SF1 de tipo MS-DOS y SF2 de tipo UNIX, ambos usando el mismo tamaño de bloque lógico de B bytes. Dado un directorio D cuyas entradas ocupan exactamente 1 bloque de la zona de datos tanto en SF1 como en SF2 y que tanto el índice en SF1 como el nodo-i de D en SF2 están ya cargados en memoria (lo indicado es lo único que hay en memoria en ese momento), el número de accesos a disco para obtener un listado de los ficheros regulares de D junto con su tamaño:
- Será igual en SF1 y SF2.
 - Será mayor en SF2 que en SF1.
44. Si suponemos que cualquier directorio ocupa un solo bloque de la zona de datos, que tanto el nodo-i del directorio raíz como su bloque de datos se encuentran ya en memoria, y que cada nodo-i se encuentra en un bloque diferente, entonces para poder leer el byte X del fichero `/home/alumno/ExamenISO/ejercicio1.sh` habrá que leer:
- 9 bloques de disco como *mínimo*.
 - 11 bloques de disco como *máximo*.

45. En UNIX, si creamos el fichero regular F1 y seguidamente creamos un enlace simbólico a F1 llamado ES_{F1} :
- El valor del contador de enlaces tanto para ES_{F1} como para F1 será 1.
 - El valor del contador de enlaces tanto para ES_{F1} como para F1 será 2.
46. Dado un sistema de ficheros con un tamaño de bloque lógico de B bytes, direcciones de disco de D bytes y una zona de bloques de datos de N bloques, el número de bloques lógicos requeridos para llevar el registro de los bloques libres es:
- $\lceil \frac{N}{8 \times B} \rceil$ si se usa la estrategia del mapa de bits.
 - $\lceil \frac{N \times D}{8 \times B} \rceil$ si se usa la estrategia de la lista ligada de bloques.
47. En cuanto a la caché de disco, es cierto que:
- Es una colección de bloques que pertenecen, desde el punto de vista lógico, al disco, pero que se mantienen temporalmente en la memoria principal.
 - Es una colección de bloques que pertenecen, desde el punto de vista lógico, a la memoria principal, pero que se mantienen temporalmente en disco.
48. El algoritmo de reemplazo discutido en clase usado en una caché de disco:
- Siempre da prioridad al reemplazo de los bloques de metadatos que se han modificado, para evitar inconsistencias en el sistema de ficheros.
 - Entre dos bloques con datos de ficheros, reemplazará siempre antes al que más tiempo hace que no se usa.
49. Dado un disco con un total de N sectores de S bytes cada uno, en general, aumentar el tamaño de bloque lógico de B a $4 \times B$ bytes provoca que:
- El tamaño en bytes de los nodos-i aumente.
 - El número de bits del mapa de bits para bloques de datos disminuya.
50. Dado un sistema de ficheros tipo UNIX, en el que hay un total de I nodos-i, cada uno con un tamaño de S bytes, y en el que el tamaño de bloque lógico es de B bytes, si en un instante determinado se emplean un total de N bloques indirectos (BSI, BDI y BTI), el tamaño total de la tabla de nodos-i en ese instante será de:
- $\lceil \frac{I \times S}{B} \rceil$ bloques lógicos.
 - $\lceil \frac{I \times S}{B} \rceil + N$ bloques lógicos.

Grupo:	DNI:
Apellidos:	Nombre:

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	

Respuestas

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	