

Apellidos:	Nombre:
DNI:	Grupo:

Instrucciones para realizar el test:

- La puntuación de este test es de 5 puntos.
- Tiempo: **30 minutos**.
- No olvides poner tus apellidos, nombre, DNI y grupo al que perteneces al principio de la hoja.
- Marca con una «X», en la tabla que hay a continuación y en bolígrafo, una única respuesta para cada cuestión.
- Una respuesta incorrecta resta una respuesta correcta. Una pregunta sin contestar ni suma ni resta.
- Debes entregar la hoja del examen al acabar.** El test no puede sacarse del aula.
- Supondremos que $1\text{ KiB} = 2^{10}$ bytes, $1\text{ MiB} = 2^{20}$ bytes y $1\text{ GiB} = 2^{30}$ bytes. También supondremos que **no existe caché de disco** en aquellos casos en los que haya lecturas y/o escrituras de disco.

Nº	a	b	Respuestas del test	Nº	a	b
1				16		
2				17		
3				18		
4				19		
5				20		
6				21		
7				22		
8				23		
9				24		
10				25		
11				26		
12				27		
13				28		
14				29		
15				30		

Preguntas:

- De manera general, un sistema operativo lo podemos ver como controlador de recursos,
 - pues oculta el funcionamiento del hardware al usuario, mostrándole una interfaz más sencilla y amigable de los recursos del sistema.
 - pues proporciona una asignación controlada de los procesadores, memoria, dispositivos de E/S, etc., a los distintos procesos que compiten por ellos.

- En relación a la protección del sistema operativo, el procesador debe disponer de dos modos de ejecución (modo núcleo y modo usuario):
 - Para evitar que el sistema se bloquee porque un programa se haga con el control de la CPU (por ejemplo, porque entre en un bucle infinito).
 - Para evitar que un programa pueda ejecutar ciertas instrucciones de la CPU, como, por ejemplo, las de E/S.
- Las llamadas al sistema
 - son programas que acompañan al sistema operativo para ofrecer un entorno más cómodo para el desarrollo y ejecución de programas.
 - definen la interfaz entre el sistema operativo y un programa en ejecución.
- Un mismo proceso puede comprender varios programas,
 - y un mismo programa puede dar lugar a varios procesos.
 - pero un programa solamente puede dar lugar a un único proceso.
- El pseudoparalelismo:
 - Consiste en ejecutar varios procesos simultáneamente gracias a tener varias CPUs y/o una CPU con varios núcleos.
 - Se da cuando cada núcleo de la CPU es utilizado por varios procesos, pasando la ejecución de uno a otro rápidamente.
- Cuando un proceso P_i hace una llamada al sistema `fork()`:
 - Se crea un nuevo proceso P_j que es una copia idéntica del P_i , con lo que ambos procesos, P_i y P_j , podrán modificar los datos que tienen compartidos en memoria RAM.
 - Se crea un nuevo proceso P_j que continúa su ejecución de manera independiente de P_i .
- Cuando un proceso P_i hace una llamada al sistema `exec()`:
 - Se crea un nuevo proceso P_j con el código y datos del programa que se indica como argumento a la llamada.
 - El proceso P_i sustituye su código y datos por los del programa cuyo nombre se indica como argumento a la llamada. A continuación, P_i comienza la ejecución del nuevo código desde el principio.
- En relación con el estado de un proceso:
 - Un proceso pasa directamente de estado “en ejecución” a “listo” cuando lo decide el planificador, independientemente del código que esté ejecutando dicho proceso.
 - Un proceso pasa directamente de estado “bloqueado” a estado “en ejecución” cuando el evento externo que está esperando ya ha finalizado exitosamente, con lo que dicho proceso puede continuar su ejecución.
- El Planificador a Medio Plazo (PMP) se encarga de decidir:
 - Si un proceso en estado “bloqueado suspendido” pasa directamente a estado “listo suspendido”.
 - Si un proceso en estado “bloqueado suspendido” pasa directamente a estado “bloqueado”.
- El Bloque de Control de Procesos (BCP) de un proceso de usuario:
 - Contiene todo aquello que está relacionado con dicho proceso: su código, sus datos, su pila,...
 - No puede ser modificado directamente por dicho proceso, sino únicamente por el sistema operativo.
- En Unix, cuando un proceso de usuario realiza una llamada al sistema:
 - Dicho proceso pasa de modo usuario a modo núcleo y ejecuta el código oportuno.
 - Se produce un cambio de proceso en CPU: el proceso de usuario pasa a estado bloqueado mientras que el proceso del sistema operativo toma el control en modo núcleo para atender dicha llamada.

12. Tenemos un proceso P_i que tiene 2 hilos H_{i1} y H_{i2} . Entonces:
- H_{i1} y H_{i2} son exactamente iguales, por lo que ejecutan las mismas instrucciones con los mismos datos.
 - H_{i1} y H_{i2} comparten los datos del proceso pero pueden ejecutar instrucciones diferentes.
13. En una implementación de hilos a nivel de usuario, si el hilo de un proceso realiza una llamada al sistema bloqueante:
- Se bloqueará completamente el proceso y todos sus hilos.
 - Se bloquearán solamente el hilo que hizo la llamada y aquellos otros hilos del proceso que dependan de la resolución de dicha llamada.
14. En relación a 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.
15. La necesidad de planificación de trabajos en CPU:
- surge desde el momento en que se tienen varios trabajos en CPU que se encuentren bloqueados a la espera de poder utilizar la E/S.
 - surge desde el momento en que se tienen varios trabajos en memoria que pueden ejecutarse.
16. La planificación *round robin*:
- Es no apropiativa, pues hasta que el proceso que está en ejecución no consume 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.
17. Una invención de la información es:
- Una amenaza a la integridad de la información.
 - Una amenaza a la confidencialidad de la información.
18. El dominio de protección...
- donde se ejecuta un proceso define los objetos a los que puede acceder dicho proceso y con qué permisos.
 - de un objeto define los procesos que pueden acceder a dicho objeto y con qué permisos.
19. En Unix, por defecto, la ACL de cada fichero es restringida porque:
- se limita a tres permisos: lectura, escritura y ejecución.
 - se limita a tres conjuntos de usuarios: propietario, grupo y otros.
20. En Unix, si un fichero ejecutable tiene el bit SETUID activo, entonces si un usuario lo ejecuta:
- el proceso que se crea no tendrá el EUID de dicho usuario sino del propietario del fichero.
 - el proceso que se crea no tendrá el EUID del propietario del fichero sino del usuario que lo ejecuta.
21. Existen 2 tipos de ficheros regulares:
- Ficheros especiales de bloques y ficheros especiales de caracteres.
 - Ficheros de texto y ficheros binarios.
22. Para un fichero (o un directorio):
- Existen innumerables rutas relativas.
 - Existen dos rutas absolutas: una partiendo del directorio raíz y otra partiendo desde el directorio actual.
23. Para realizar una lectura de un directorio:
- Se utiliza la operación **read** (como en los ficheros) pues los directorios son simplemente un tipo de ficheros.
 - Se utiliza la operación **readdir** debido a que el sistema operativo es el que determina el formato de los directorios.
24. En una implementación de ficheros mediante lista ligada sin índice, para la lectura del byte Z, perteneciente al bloque X de un fichero F cuyo primer bloque (el bloque 0) se encuentra en la dirección D de disco,
- se tendrá que leer únicamente el bloque D+X del disco.
 - se tendrán que leer X+1 bloques del disco.
25. Si suponemos que cualquier directorio tiene sólo un bloque 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 primer byte del fichero `/home/alumno/apuntes/tema1.pdf`:
- Habrà que leer 8 bloques de disco.
 - Habrà que leer 7 bloques de disco.
26. Una caché de disco:
- 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.
27. En una implementación de ficheros mediante lista ligada e índice, el tamaño de la tabla o índice:
- Depende del tamaño del dispositivo de almacenamiento y del tamaño de los bloques.
 - Depende del tamaño de los ficheros y del tamaño de los bloques.
28. En una implementación de ficheros mediante nodos-i,
- Existe una única tabla para todo el sistema de ficheros, llamada nodo-i, que contiene un conjunto de punteros para acceder a todos los bloques del dispositivo de almacenamiento, bien directamente o través de bloques indirectos (BSI, BDI y BTI).
 - La cantidad de memoria RAM utilizada para nodos-i depende del número de ficheros abiertos en un momento dado.
29. En Unix, si tenemos el fichero `fic1` y creamos un enlace físico llamado `lin1` a dicho fichero y, tras ello, borramos `fic1`:
- El enlace `lin1` se quedará inconsistente, pues apuntará a un fichero que ya no existe.
 - Se sigue pudiendo acceder mediante el enlace `lin1` al contenido del fichero eliminado `fic1`.
30. Tenemos un disco donde el control de bloques libres de la zona de datos se gestiona mediante un mapa de bits. Si en dicha zona se tiene que L bloques están libres y M bloques están ocupados:
- El mapa de bits estará formado al menos por $(L + M)/8$ bytes.
 - El mapa de bits estará formado por L bits.

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

Respuestas del test

Nº	a	b
16		X
17	X	
18	X	
19		X
20	X	
21		X
22	X	
23		X
24		X
25	X	
26	X	
27	X	
28		X
29		X
30	X	