



Puntuación: para cada pregunta <ul style="list-style-type: none">• 100% si está bien contestada• -50% si no está bien contestada• 0 si se deja en blanco	UO: Nombre:	Modelo 1
---	------------------------------	-----------------

Cuestiones Verdadero / Falso

IMPORTANTE: Debes contestar a cada pregunta independientemente, con VERDADERO o FALSO.

- Los sistemas actuales realizan la traducción de direcciones lógicas a físicas en tiempo de carga.
Falso. Los sistemas actuales realizan la traducción en tiempo de ejecución a través de la Unidad de Gestión de Memoria
- La protección de la memoria impide el acceso a direcciones de memoria fuera del espacio de direcciones del proceso que solicita el acceso
Cierto. De ello se encarga la Unidad de Gestión de Memoria antes de traducir la dirección y localizar su contenido
- La protección, al igual que la traducción, la lleva a cabo la Unidad de Gestión de Memoria en los procesadores actuales de uso común
Cierto. Son las dos funciones que realiza el hardware
- Un esquema de asignación de memoria real puede provocar fallos de página.
Falso. En ese caso estaríamos hablando de gestión de memoria virtual
- En un esquema de asignación contigua de memoria con particiones fijas, el tamaño de las particiones depende del tamaño de los procesos que sean cargados.
Falso. En este esquema el tamaño de las particiones está prefijado y los procesos deben cargarse en particiones de igual o mayor tamaño, desperdiciando parte del espacio de dicha partición que no puede asignarse a otro proceso
- Dado un sistema de gestión de memoria virtual con paginación, en el que se emplea tabla de páginas de un único nivel, con páginas de 4K. Dado un proceso cuyo espacio de direcciones ocupa 47982 bytes, este proceso necesitará una tabla de páginas de 12 entradas.
Cierto. 47982Bytes dividido entre el tamaño de la página son 11,71. Esto implica que se distribuye en 12 páginas, y por tanto la tabla tendrá 12 entradas.
- En un sistema de gestión de memoria virtual con paginación, la tabla de páginas de un proceso almacena información que permite detectar los fallos de páginas de ese proceso
Cierto, se trata del bit de presencia que indica si la página está o no cargada
- En un esquema de memoria virtual con paginación que disponga de TLB, algunas de las páginas del proceso estarán cargadas en la TLB
Falso, en la TLB no se guardan parte de las páginas del proceso sino parte de la tabla de páginas del proceso
- Dado un sistema de gestión de memoria virtual con paginación, en el que se emplea tabla de páginas de dos niveles, en los que cada entrada de la tabla ocupa 2 bytes, y tamaño de página de 2K. Dado un proceso de tamaño 1Gbyte de los que se usan únicamente los 4.700.000 bytes de las posiciones más bajas (del byte 0 al 4.699.999). Será necesarios 1026 Kbytes para almacenar la tabla de páginas.



Falso. En una tabla de páginas de 2 niveles sólo se crean las páginas que sean necesarias en el segundo nivel. Si solo se usan los primeros 4.700.000 bytes, solo se usan $(4700000/2048 = 2294,92)$ 2295 páginas. Para localizar estas páginas dividimos 2295/1024 entradas que tiene cada tabla. Necesito, por tanto 3 tablas de páginas de segundo nivel y una de primer nivel. Esto significa $4 * 2K = 8K$ bytes para almacenar la tabla de páginas. Si usara todo el espacio de direcciones entonces si necesitaría 26K puesto que $2MB / 2K = 512K$ páginas. Cada tabla tiene 1024 entradas, necesito 512 tablas de segundo nivel + 1 de primero. $513 * 2k = 1026K$ para la tabla de páginas.

10. En un esquema de gestión de memoria virtual con segmentación, el sistema integra el concepto de región (código, datos, pila...) que puede compartir, proteger, etc. En un sistema de paginación no hay este concepto.

Cierto.

11. Un esquema de asignación de memoria virtual mejora el grado de multiprogramación con respecto a un esquema de asignación de memoria física

Cierto. Aumenta el número de procesos que pueden estar simultáneamente en memoria ejecutándose.

12. La paginación de memoria es apropiada para gestionar regiones.

Falso. En la paginación no hay manejo de regiones. En la segmentación sí.

13. La memoria virtual se basa en el principio de localidad.

Cierto.

14. En un sistema de memoria virtual con bus de 16 bits el espacio de direcciones lógicas puede abarcar 2^{16} direcciones.

Cierto.

15. Para traducir una dirección virtual a física, lo primero es mirar en la tabla de páginas. Si eso falla, se consulta el TLB.

Falso. Es al revés.

16. En gestión de memoria virtual, FIFO es una política válida de lectura.

Falso. FIFO es una política de reemplazo.

17. En gestión de memoria virtual, una política LRU puede verse afectada por la anomalía de Belady.

Falso. La anomalía de Belady afecta a la política FIFO.

18. En un sistema de memoria virtual basado en el algoritmo del reloj, con una memoria física asignada de 4 marcos, y cuyos contenidos son: (Página=12|bit de referencia=1) (15|1) (18|0) (45|1), y el "puntero al siguiente marco" apunta al segundo, cuando se solicita la página 49, el segundo marco se usa para cargar esa página 49, y la 15 abandona la memoria física.

Falso. El segundo marco tiene bit de referencia a 1, así que bajará a 0 pero no se usará. Se usará el tercero.

19. En un sistema de memoria virtual basado en el algoritmo del reloj, con una memoria física asignada de 4 marcos, y cuyos contenidos son: (Página=12|bit de referencia=1) (15|1) (18|0) (45|1), y el "puntero al siguiente marco" apunta al segundo, cuando se solicita la página 49, el tercer marco se usa para cargar esa página 49, y la 18 abandona la memoria física.

Cierto.

20. En un sistema de memoria virtual basado en el algoritmo del reloj, con una memoria física asignada de 4 marcos, y cuyos contenidos son: (Página=12|bit de referencia=1) (15|1) (18|0) (45|1), y el "puntero al siguiente marco" apunta al segundo, cuando se solicita la página 49, después de terminar de procesar esa petición el bit de referencia para el segundo marco es 0.

Cierto.



21. En un sistema de memoria virtual basado en el algoritmo del reloj, con una memoria física asignada de 4 marcos, y cuyos contenidos son: (Página=12|bit de referencia=1) (15|1) (18|0) (45|1), y el "puntero al siguiente marco" apunta al segundo, cuando se solicita la página 49, después de terminar de procesar esa petición el "puntero al siguiente marco" apunta al tercero.
Falso. Se ha usado el tercero, y apuntará al cuarto.
22. El controlador hardware del dispositivo recibe instrucciones del controlador software (driver) del sistema operativo
Cierto, son los dos elementos que se comunican entre sí. El sistema operativo a través del controlador software da la orden, la recibe el controlador hardware y la transmite al dispositivo en sí.
23. La comunicación con el controlador hardware a través de puertos utiliza instrucciones máquina privilegiadas
Cierto. Son privilegiadas para que sólo puedan ejecutarse en modo núcleo, y por tanto solo el sistema operativo las pueda ejecutar.
24. Cualquier programa de usuario puede comunicarse directamente con el controlador hardware a través de las operaciones port-in y port-out
Falso. No es posible utilizarlas porque son instrucciones máquina privilegiadas. Sólo puede usarlas el sistema operativo.
25. El sistema operativo deja la operación a realizar por el dispositivo en los registros de control del controlador hardware
Cierto. Los controladores hardware tienen registros de control, de datos y de estado para comunicarse con el sistema operativo
26. El modo de transferencia con Acceso Directo a Memoria permite al controlador hardware el acceso a la memoria principal del ordenador
Cierto, es el propio controlador el que realiza la transferencia de datos del dispositivo a la memoria del ordenador
27. Una transferencia con Acceso Directo a Memoria no es una entrada/salida por interrupciones
Falso. Si que es una e/s por interrupciones. El controlador envía la interrupción cuando ya ha hecho la transferencia.
28. Cada petición que recibe el gestor de dispositivos genera un hilo de ejecución en el gestor independiente del dispositivo
Cierto. Se genera un hilo que crea la petición y se la pasa a la cola del manejador dependiente del dispositivo
29. Cada petición que recibe el gestor de dispositivos genera un hilo de ejecución en el gestor dependiente del dispositivo
Falso. El gestor dependiente es un hilo que gestiona una petición y espera su realización antes de seguir con la siguiente
30. Cuando un dispositivo finaliza una operación genera una interrupción que será trasladadas al manejador independiente del dispositivo
Falso. Serán trasladadas al manejador dependiente que es el que está esperando por la finalización de dicha operación
31. La cola de IORBs es utilizada tanto por el manejador dependiente como por el independiente del dispositivo.
Cierto. El manejador independiente la usa para añadir la petición y el dependiente la usa para servir la petición
32. Si un disco duro tiene 256 sectores por cilindro, y 512 bytes por sector, 4 cabezas y una capacidad total de 256 MB podemos concluir que tiene 2048 pistas por cada cara.



Cierto. $256 * 512 = 131.072$ Bytes por cilindro. $256MB / 131.072Bytes = 2048$ cilindros existentes, por tanto 2048 pistas por cada cara.

33. En gestión de E/S, el software dependiente del dispositivo recibe IORBs.

Cierto.

34. En un disco, el número de caras es igual que el número de cabezas de lectura/escritura.

Cierto.

35. El tiempo de latencia depende de la velocidad de rotación.

Cierto.

36. En políticas de planificación de disco, FCFS es muy fácil de implementar, y por tanto ofrece un rendimiento excelente.

Falso. Es fácil de implementar, pero no ofrece un buen rendimiento, porque no intenta evitar movimientos de las cabezas.

37. En políticas de planificación de disco, las que pueden causar inanición son FCFS y SSTF.

Falso. FCFS no causa nunca inanición.

38. En políticas de planificación de disco, las únicas que tratan todas las peticiones igualmente (es decir, no favorecen pistas centrales o extremas) son Scan y Look.

Falso. Scan y Look favorecen las pistas centrales. Las que no favorecen ninguna en particular son FCFS, C-Scan y C-Look.

39. Cuando se solicita un bloque, si está en la caché de bloques, esa petición no llega a procesarse en los manejadores de dispositivo (ni dependientes ni independientes).

Cierto.

40. El tiempo de búsqueda (en un disco) depende de la velocidad de rotación.

Falso. El movimiento de la cabeza no depende de cuán rápido esté rotando el disco bajo ella.

41. Un disco usa la política de planificación Look. La cabeza está inicialmente en el cilindro 0. Las peticiones pendientes son 10, 46, 90, 24, 57, 32. Cuando está procesando la 32, llega una petición para 14. Cuando está procesando 46, llega una petición para 59. El orden de servicio de las peticiones es: 10 24 32 46 57 59 90 14.

Cierto.

42. Un disco usa la política de planificación Look. La cabeza está inicialmente en el cilindro 0. Las peticiones pendientes son 10, 46, 90, 24, 57, 32. Cuando está procesando la 32, llega una petición para 14. Cuando está procesando 46, llega una petición para 59. El orden de servicio de las peticiones es: 10 24 32 46 57 90 59 14.

Falso. La 59 no se procesa "a la vuelta", sino "a la ida".

43. Los Sistemas de ficheros que usan habitualmente Windows y Linux interpretan los ficheros del usuario como una secuencia de bytes

Cierto, no hace otra interpretación, salvo en los formatos de algunos específicos como los ejecutables o los directorios.

44. Los ficheros ejecutables, o los ficheros de texto son ejemplos de ficheros cuyo formato es conocido para el sistema operativo

Falso, los ficheros ejecutables si son reconocidos por el sistema pero los de texto simplemente son para él una secuencia de bytes.

45. Actualmente no se usa el almacenamiento contiguo de ficheros



Falso. Se sigue usando en CDs

46. Sea un sistema de ficheros tipo FAT32 con bloques de 512 bytes, y un fichero que tiene asignados los siguientes bloques de disco: 1018, 456, 8358, 143, 294,2984. Para localizar el byte 3000 el sistema deberá acceder 5 veces a la FAT

Cierto. $3000/512 = 5,86$. Esto significa que hay que acceder al sexto bloque del fichero (2984) y para ello hay que recorrer 5 entradas de la FAT.

47. Sea un sistema de ficheros tipo ext2, con asignación indexada con tablas de índices de 13 entradas, con bloques de 512 bytes, y un fichero que tiene asignados los siguientes bloques de disco: 1018, 456, 8358, 143, 294,2984. Para localizar el byte 3000 el sistema será más rápido que si fuese FAT32.

Cierto, solo tiene que hacer un acceso a la tabla de índices, a la posición 9.

48. La técnica del Journaling permite acelerar la velocidad de acceso a los datos de los ficheros.

Falso. Permite asegurar la consistencia de los datos cuando se producen fallos en el sistema.

49. NTFS usa tablas de índices para la localización de los bloques de los ficheros

Falso, utiliza árboles equilibrados.

50. En la tabla MFT que usa NTFS cada fichero tiene una entrada que constituye el nodo raíz del árbol equilibrado

Cierto.

51. Las agrupaciones son conjuntos de ficheros del mismo directorio

Falso. Son conjuntos de bloques del disco

52. El tiempo que tarda el sistema en localizar un bloque de un fichero en un sistema de asignación contigua depende de la posición de éste en el fichero (cuanto más lejos del principio más se tarda en localizar)

Falso. El tiempo es el mismo (dirección de comienzo + byte de acceso)

53. Un bloque es lo mismo que un sector.

Falso. No tiene por qué. Un bloque puede constar de varios sectores.

54. En el sistema de ficheros ISO9660, el contenido de un fichero es contiguo.

Cierto.

55. En el sistema de ficheros de Unix System V, el número total de i-nodos disponibles se ajusta dinámicamente dependiendo de las necesidades.

Falso. El número total de i-nodos disponibles es fijo y se establece al crear el sistema de ficheros.

56. En NTFS, un fichero (incluso uno que no esté vacío) podría no necesitar ningún bloque de datos.

Cierto. Si es lo bastante pequeño, puede caber en el espacio de los metadatos.

57. En el sistema de gestión de ficheros de un sistema operativo, el sistema de ficheros virtual pide bloques directamente al manejador independiente del dispositivo.

Falso. Se los pide al módulo de organización de archivos, que a su vez se los pide al servidor de bloques, que a su vez trata con los manejadores.

58. Una configuración RAID 5 distribuye la información de paridad entre todos sus discos

Cierto.

59. El journaling y los mapas de bits son dos técnicas para hacer los sistemas de ficheros más seguros.

Falso. El journaling sí, pero los mapas de bits son para gestionar el espacio disponible.



60. Supongamos un sistema de ficheros con asignación indexada por tablas de índices (estilo System V). Cada tabla, sea del nivel que sea, tiene 13 entradas, y la primera tiene 10 referencias directas, 1 indirecta, 1 con doble indirección y 1 con triple indirección. Un fichero tiene 11 bloques: 54, 22, 45, 34, 35, 36, 129, 86, 470, 212, 46. La onceava entrada de la primera tabla de índices contiene 46
Falso. Contiene una referencia a otra tabla de índices, que sí apunta a 46.
61. Supongamos un sistema de ficheros con asignación indexada por tablas de índices (estilo System V). Cada tabla, sea del nivel que sea, tiene 13 entradas, y la primera tiene 10 referencias directas, 1 indirecta, 1 con doble indirección y 1 con triple indirección. Un fichero tiene 28 bloques. Necesitará 3 tablas de índices en total.
Falso. Referencia 10 bloques en la primera tabla, otros 13 en la segunda, y en la tercera, que es de doble indirección, habrá una referencia a una cuarta tabla que es la que apunta al resto de bloques.
62. Supongamos un sistema de ficheros con asignación indexada por tablas de índices (estilo System V). Cada tabla, sea del nivel que sea, tiene 13 entradas, y la primera tiene 10 referencias directas, 1 indirecta, 1 con doble indirección y 1 con triple indirección. Un fichero tiene 28 bloques. Necesitará 4 tablas de índices en total.
Cierto.