

## Segundo Parcial 2015-2016

1. Para poder tener varios procesos en memoria a la vez es necesario la traducción de direcciones lógicas a físicas en tiempo de carga.
2. La protección de la memoria es realizada en tiempo de ejecución por el sistema operativo.
3. La traducción dinámica de direcciones la lleva a cabo la Unidad de Gestión de Memoria en los procesadores actuales de uso común
4. Un esquema de asignación de memoria virtual puede provocar fallos de página.
5. En un esquema de asignación contigua de memoria con particiones dinámicas, el tamaño de las particiones depende del tamaño de los procesos que sean cargados.
6. 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 23456 bytes, este proceso necesitará una tabla de páginas de al menos 7 entradas.
7. En un sistema de gestión de memoria virtual con paginación, la tabla de páginas de un proceso almacena información que permite obtener la dirección física asociada a una dirección virtual de ese proceso
8. En un esquema de memoria virtual con segmentación, la traducción de direcciones virtuales a físicas es más eficiente que en paginación.
9. 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, el tamaño de página es de 2K y la tabla de páginas de nivel 1 ocupa una página. Dado un proceso de tamaño 1 Gbyte de los que se usan únicamente los 1.000.000 bytes de las posiciones más bajas. Será necesarios 4096 bytes para almacenar las tablas de páginas.
10. En un esquema de gestión de memoria virtual con paginación, el sistema integra el concepto de región (código, datos, pila...) que puede compartir, proteger, etc.
11. Un esquema de asignación de memoria física mejora el grado de multiprogramación con respecto a un esquema de asignación de memoria virtual
12. La segmentación de memoria es apropiada para gestionar regiones.
13. La memoria virtual se basa en el principio de localidad.
14. En un sistema de memoria virtual con bus de 16 bits el espacio de direcciones lógicas puede abarcar 2<sup>16</sup> direcciones.
15. Para traducir una dirección virtual a física, lo primero que se mira es en el TLB. Si eso falla, se consulta la tabla de páginas.
16. En gestión de memoria virtual, la pre-paginación es una política válida de lectura.
17. En gestión de memoria virtual, una política Óptima puede verse afectada por la anomalía de Belady.

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.
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.
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.
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.
22. El controlador hardware del dispositivo recibe instrucciones del software independiente del dispositivo del sistema operativo
23. La comunicación con algunos dispositivos hardware pueden realizarse accediendo a posiciones de memoria determinadas.
24. Los dispositivos de e/s de bloques se caracterizan por leer/escribir flujos de bloques de datos de manera secuencial.
25. La parte del sistema operativo que deja la operación a realizar por el dispositivo en los registros de control del controlador hardware es la rutina independiente del dispositivo
26. En dispositivos que soporten el modo de transferencia con Acceso Directo a Memoria la transferencia de datos entre el dispositivo y la memoria la lleva a cabo el sistema operativo.
27. En una operación de e/s el manejador de interrupciones típicamente se ejecuta al finalizar la operación
28. Cada petición que recibe el gestor de dispositivos genera un hilo de ejecución en el gestor independiente del dispositivo
29. Cada petición que recibe el gestor de dispositivos genera un hilo de ejecución en el gestor dependiente del dispositivo
30. Cuando un dispositivo finaliza una operación genera una interrupción que hará que la rutina dependiente del dispositivo sea despertada y pueda continuar con el tratamiento de la e/s
31. La cola de IORBs depende de la rutina independiente del dispositivo y, por lo tanto, sólo hay una en el sistema.
32. Si un disco duro tiene 256 sectores por cilindro, y 512 bytes por sector, 8 cabezas y una capacidad total de 256 MB podemos concluir que tiene 2048 pistas por cada cara.

33. En gestión de E/S, el software dependiente del dispositivo crea los IORBs.
34. En un disco, todas las caras tienen el mismo número de pistas.
35. El tiempo de búsqueda depende de la velocidad de rotación.
36. En políticas de planificación de disco, SSTF ofrece un rendimiento excelente.
37. En políticas de planificación de disco, SCAN puede causar inanición.
38. En políticas de planificación de disco, C-Scan y C-Look tratan todas las peticiones igualmente (es decir, no favorecen pistas centrales o extremas) .
39. Cuando se solicita un bloque de disco no siempre es necesario ir al disco a leerlo.
40. El tiempo de búsqueda (en un disco) suele ser el más alto de los implicados en un acceso a disco.
41. Un disco usa la política de planificación SSTF. 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 57, llega una petición para 50. El orden de servicio de las peticiones es: 10 24 32 46 57 50 90 14.
42. Un disco usa la política de planificación SCAN. 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 57, llega una petición para 50. El orden de servicio de las peticiones es: 10 24 32 46 57 90 50 14.
43. Los sistemas de ficheros que usan habitualmente Windows y Linux interpretan los formatos de ficheros ejecutables o directorios como una secuencia de bytes
44. Los ficheros de texto son ejemplos de ficheros cuyo formato no es conocido para el sistema operativo
45. El almacenamiento contiguo de ficheros se sigue usando en lápices USB.
46. Sea un sistema de ficheros tipo FAT32 con bloques de 512 bytes, y un fichero que tiene asignados los siguientes bloques de disco: 123, 455, 8947, 264, 332,1643. Para localizar el byte 2048 el sistema deberá acceder 4 veces a 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 debe acceder a la entrada 5 en la que encontrará el número de bloque de disco 2984. En este bloque, se deberá localizar el byte número 440.
48. La técnica del Journaling permite asegurar la consistencia de los datos cuando se producen fallos en el sistema.
49. Ext4 usa tablas de índices para la localización de los bloques de los ficheros
50. En la tabla MFT que usa NTFS cada fichero tiene una entrada que constituye el nodo raíz del árbol equilibrado
51. Linux usa el sistema de ficheros Ext4

52. La ventaja de un sistema de asignación contigua frente a la asignación no contigua es la sencillez en la localización de sus datos
53. El tamaño de un bloque se configura al realizar la partición.
54. En el sistema de ficheros ISO9660, el contenido de un fichero es contiguo.
55. En el sistema de ficheros de Unix System V, los i-nodos contienen información para el acceso al directorio donde se encuentran.
56. En NTFS, los primeros bytes del fichero se guardan en el registro de la tabla de ficheros maestra.
57. En el sistema de gestión de ficheros de un sistema operativo se comunica con el sistema de gestión de e/s cuando necesita cargar un bloque que no está en la caché.
58. Una configuración RAID proporciona mayor integridad, tolerancia a fallos y mayor rapidez de acceso a los datos
59. El journaling es una técnica para hacer los sistemas de ficheros más seguros.
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 18 bloques. Necesitará 4 tablas de índices en total.