

**Respuestas Orientadoras e incompletas (en algunos casos) a las preguntas de revisión. Favor tomarlas como orientadoras.**

### **Revisión – Unidad VII – Sistema de Archivos**

1. ¿Por qué se dice que los archivos son una abstracción asociada al almacenamiento secundario?

La abstracción es un recurso valioso para manejar la complejidad. Detrás de una abstracción puede ocultarse cierto grado de complejidad. La mayoría de los usuarios manejan el concepto abstracto "archivo o fichero", en la mayoría de las aplicaciones el concepto central es el "archivo" como una colección de datos, persistente, compartible entre procesos y con una estructura propia. Esta es la abstracción presentada al usuario que oculta la complejidad de implementación de esta colección de datos grabada en n bloques de un dispositivo de almacenamiento secundario.

2. Responda V o F: Todo SO debe organizar sus archivos bajo una estructura jerárquica.

Falso. Existen muchos SO's (CP/M, VMS, MVS, etc.) que presentan una abstracción de su sistema de archivos -que, si bien esta organizada bajo la abstracción "archivo"- en forma no jerárquica.

3. Responda V o F: El concepto de archivo esta asociado con la persistencia de datos.

Verdadero.

4. ¿Cuál es la función principal de un Sistema Gestor de Ficheros?

Proporcionar servicios a usuarios y aplicaciones para el uso de ficheros.

5. ¿Cuáles son las operaciones típicas que se aplican sobre ficheros?

crear() ..., borrar() ..., abrir() ..., cerrar() ..., leer() ..., escribir() ....

6. Defina los siguientes conceptos: Campo, Registro, Fichero, Base de Datos.

Campo: elemento básico, valor único, de longitud fija o variable, de un tipo de dato determinado.

Registro: conjunto de campos relacionados que se pueden tratar como la representación de una unidad, de longitud fija o variable.

Fichero: Colección de registros, de campos similares, con restricciones en cuanto al control de acceso (permisos), entidad única desde el punto de vista del usuario y las aplicaciones, referenciada por nombre.

Base de Datos: Colección de datos relacionados, formada por uno o más ficheros, diseñada para ser usada por una o más aplicaciones, se requiere de un software adicional para su manipulación: SGBD (sistema gestor de base de datos) separado del SO.

7. ¿Qué ventajas le provee al desarrollador de aplicaciones el trabajar con un SO que posea un Sistema de Gestión de Ficheros?

Le permite concentrarse en el problema a resolver en vez de ponerse a escribir código para lidiar con los dispositivos, el espacio ocupado y libre en disco, la planificación de los dispositivos, etc. El SGF provee una interfase uniforme (E-S lógica) que le permite al usuario y a las aplicaciones acceder a los registros de los archivos en forma lógica, ocultando los detalles de implementación.

8. Responda V o F: El Sistema Gestor de Ficheros no se ocupa de la administración de permisos sobre los ficheros.

Falso. Dentro del soporte para la compartición de archivos (entre distintos usuarios) que debe proporcionar el SGF se encuentran los derechos de acceso.

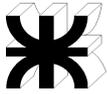
9. Responda V o F: Los drivers tienen un nivel de abstracción mayor que la I/O lógica.

Falso. Los drivers son los de menor nivel de abstracción, puesto que interactúan con el hardware (en términos de señales, registros de cpu, canales, etc.); mientras que la E-S lógica interactúa con los usuarios y las aplicaciones (en términos de archivos, registros lógicos).

10. Responda V o F: Los programas del usuario interactúan directamente con los drivers.

Falso. Los programas interactúan con la E-S lógica.

11. Responda V o F: El sistema básico de I/O trabaja a nivel de bloques.



Verdadero. Realiza la E-S física, manejo de buffers, etc.

12. Responda V o F: El supervisor de I/O básico trabaja con las políticas de planificación de I/O.

Verdadero. Inicia y termina las peticiones de E-S, planificación de E-S, colas, estados de ficheros, asignación de buffers.

13. Responda V o F: Los datos básicos de un fichero son mantenidos por el sistema de I/O lógica.

Verdadero. El sistema de E-S lógica trabaja con los usuarios y aplicaciones permitiendo el acceso a archivos y registros lógicos; así como también mantiene los datos básicos de los ficheros. Pag.552.

14. ¿Qué es un método de acceso? ¿Para qué sirve?

Es el nivel de la arquitectura del sistema de ficheros más cercano al usuario. Sirve para proporcionar una interfaz estándar entre las aplicaciones, sistemas de ficheros y dispositivos de almacenamiento. Es la forma de manipular la estructura física del archivo, cómo leer, cómo grabar un registro lógico; existen distintas formas de hacerlo según el método de acceso utilizado. Esta vinculado con las distintas estructuras de los archivos.

15. ¿A qué nos referimos con la "organización de un fichero"?

Idem anterior. Estructura lógica de los registros determinados por su forma de acceso.

16. ¿Qué relación hay entre la "organización de un fichero" y los "métodos de acceso"?

La organización de un fichero se refiere a su estructura física mientras que el método de acceso se refiere al algoritmo que debe aplicarse -acorde con la estructura física del archivo- para manipular los registros del archivo. Por ejemplo, si un archivo tiene una organización de tipo Hash, el método de acceso debe ser a través de un valor clave el cual será traducido a una dirección de registro lógico utilizando para ello una función de dispersión. ...

17. Responda V o F: En cuanto a la administración de ficheros, el SO trabaja a nivel de bloques físicos (registros físicos) mientras que las aplicaciones de usuario trabajan a nivel de registros lógicos.

Verdadero.

18. ¿Qué diferencia hay entre registro físico y registro lógico?

Registro físico: son la unidad de transferencia entre el dispositivo de almacenamiento y la memoria principal. Un dispositivo de almacenamiento secundario lee y graba "de a registros físicos" o bloques físicos.

Registro lógico: es el registro de un archivo desde el punto de vista del programador, formado por n campos que describen a una entidad determinada (artículo, proveedor, cliente, etc.). Un registro físico -según su tamaño- podrá contener 0, 1 o n registros lógicos más algunos bytes de desperdicio (fragmentación interna).

19. Describa la organización de archivo de tipo pila (stack). ¿Qué características tiene?

Es la organización más sencilla, el orden de los registros equivale al orden de llegada, los registros pueden ser de longitud variable, con igual cantidad de campos o distinta cantidad, lo cual implica el uso de delimitadores de registros/campos y/o campos autodescriptivos. Acceder a un registro determinado implica una búsqueda exhaustiva, organización inadecuada para la mayoría de las aplicaciones, generalmente se usa como una forma de almacenamiento previo a su proceso.

20. Responda V o F: En una organización de tipo pila puedo procesar registros en forma directa.

Falso. El acceso requiere de una búsqueda exhaustiva, secuencial.

21. Describa la organización de archivo de tipo secuencial. ¿Qué características tiene?

Registros de longitud fija, igual cantidad de campos, no autodescriptivos, posicionales, conjunto de 1 o más campos que forman clave primaria, almacenamiento según clave primaria, procesamiento batch, ideal para almacenamiento en cinta o disco, el orden físico de los registros coincide con el orden lógico. Generalmente se actualizan mezclándose con otro fichero de tipo pila, obteniéndose un nuevo archivo secuencial actualizado.



22. Responda V o F: En una organización de tipo secuencial el o los campos clave no tienen ninguna importancia.

Falso. Ya que los registros se guardan físicamente según la clave.

23. Responda V o F: Un archivo secuencial indexado, al igual que un archivo secuencial se encuentra ordenado por uno o más campos clave.

Verdadero. Ambas organizaciones mantienen la misma estructura, pero el secuencial indexado agrega dos estructuras adicionales: un archivo índice que permite acceso aleatorio por clave y un archivo de desbordamiento.

24. ¿Qué ventajas adicionales provee la organización secuencial indexada por sobre la organización secuencial?

Permite acceso aleatorio por clave.

25. Responda V o F: Un archivo secuencial indexado ocupa menos espacio en disco que un archivo secuencial.

Falso. Puesto que requiere dos archivos adicionales.

26. ¿Qué es un índice?

Es un archivo adicional al archivo de datos. Tiene la estructura de un archivo secuencial, con -al menos- dos campos: el campo clave y un puntero a una dirección de registro dentro del archivo de datos (también este puntero puede utilizarse como indirección a otro nivel dentro del mismo índice).

27. ¿Qué es un índice exhaustivo?

Es un índice que tiene una entrada por cada valor de clave.

28. ¿Qué es un índice no exhaustivo?

Es un índice que tiene una entrada por cada grupo de valores de clave (permite apuntar hacia la "vecindad" de la clave buscada). Pueden incluirse sólo algunas claves en el índice (para reducir su tamaño) y luego apuntar hacia otro nivel o bien hacia la vecindad de la clave buscada en el archivo de datos y continuar con una búsqueda secuencial a partir de allí.

29. Responda V o F: Para mejorar el rendimiento de un archivo secuencial indexado voy a crear una clave por cada atributo.

Falso. Porque a medida que se agregan índices se penalizan las grabaciones, puesto que, además de actualizar los datos del registro también se deben actualizar los índices.

30. Responda V o F: Un archivo secuencial indexado puede contener registros de longitud variable.

Falso. Todos los registros deben ser de longitud fija.

31. ¿Por qué hay necesidad de un "archivo de desborde" (overflow file) en un archivo secuencial indexado?

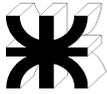
Para enlazar (y mantener en forma ordenada por el campo clave) los nuevos registros que se agregan en el archivo.

32. ¿Cómo funciona un archivo de acceso directo?

Tiene una estructura similar a un archivo secuencial, pero sin mantener un orden por clave primaria. Este tipo de archivos es accesible en forma aleatoria, a través del número de registro lógico (0 .. N-1). Generalmente se utiliza combinado con una técnica de hashing, utilizando una función de dispersión para lograr un acceso a través de clave primaria.

33. ¿Cómo funciona un archivo hash (hashed file)?

Idem anterior. Stallings asume que un archivo de acceso directo es sinónimo de un archivo hash. En este caso se implementa una función hash (o función de dispersión) que transforma una clave primaria en dirección de registro lógico. Son archivos muy rápidos para acceder por clave primaria, no requieren de ninguna estructura de datos adicional al archivo de datos. El problema habitual es que debe implementarse una forma de dar soporte a las claves sinónimas (dadas dos claves primarias distintas resulta que ambas, luego de aplicarles la función hash, apuntan al mismo registro lógico).



34. ¿Cuáles son las posibles implementaciones de un archivo hash?

Siguiendo con el punto anterior. Existen dos implementaciones posibles: hashing lineal y desbordamiento encadenado. En el primer caso, ante una colisión, se continúa dentro del archivo de datos secuencialmente hasta encontrar un lugar libre donde almacenar el registro (esto tiene el efecto adverso de poder generar más colisiones en forma indirecta). En el segundo caso, se agrega al archivo de datos un campo adicional que sera un puntero hacia el primer registro colisionado. En todos los registros que no tengan colisiones, este campo estará vacío. Entonces, cuando ocurre una colisión, los registros colisionados se mantienen -a partir de un area de desborde (en un archivo separado o a partir del registro  $n + 1$ )- como una lista simplemente enlazada.

35. ¿Qué es un directorio? ¿Qué contiene?

Es una proyección entre los usuarios, las aplicaciones y los ficheros. Mantiene información sobre cada archivo del sistema, incluye información básica (nombre,tipo,organización), información de direccionamiento (volumen/disco,direccion física inicial,tamaño,tamaño asignado), información de control de acceso (dueño,claves de acceso,permisos), información de uso (fecha creación,usuario creador,fecha ultima lectura,usuario ultimo lector,fecha ultima actualizacion,usuario ultimo actualizador,fecha ultima copia de seguridad,uso actual (info.sobre el uso actual, que proceso tiene este archivo abierto, si esta bloqueado, si esta actualizado en memoria y aun no en disco,etc.)).

36. ¿Qué operaciones pueden aplicarse sobre un directorio?

buscar(), crear fichero(), borrar fichero(), listar directorio(),actualizar directorio().

37. ¿Cuál es la información de control asociada a un directorio?

Información de control de acceso (dueño,claves de acceso,permisos).

38. ¿Quién es el dueño (owner) de un archivo y qué operaciones puede hacer sobre un archivo de su propiedad?

El dueño del fichero es su propietario, su creador y puede hacer cualquier tipo de operación sobre un archivo de su propiedad.

39. ¿Cómo se identifican unívocamente los archivos dentro de un sistema de ficheros jerárquico?

A través del nombre de camino (path, conjunto de directorios hasta el directorio que contiene al archivo) + nombre del fichero. Ejemplo, el archivo myfile se encuentra en el directorio /home/grchere por lo tanto, unívocamente el archivo se identifica como /home/grchere/myfile.

40. ¿Qué es el directorio actual (current directory) o directorio de trabajo (working directory)?

Permite hacer referencia en forma relativa a un archivo o directorio. El directorio actual o directorio de trabajo es un nombre de camino (path) asociado con el usuario interactivo actual o con un proceso actualmente en ejecucion. En forma interactiva, en una consola unix/linux, el usuario puede conocer su directorio actual utilizando el comando "pwd" quien muestra en pantalla el directorio actual.

41. ¿Que implicancias tiene el hecho de que un archivo sea compartido por n procesos ejecutados por n usuarios?

El SO debe aplicar algun tipo de disciplina sobre el fichero utilizando algún tipo bloqueo (a nivel del fichero completo o a nivel de registros individuales) para implementar una exclusión mutua entre distintos procesos que pretenden acceder al mismo fichero.

42. ¿Qué clases de permisos sobre ficheros Ud. conoce?

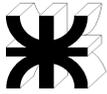
ninguno ..., conocimiento ..., ejecucion ..., lectura ..., adición ..., actualización ..., cambio de protección ..., borrado ... .

43. ¿Qué clases de usuarios Ud. conoce sobre los cuales se pueden aplicar distintos accesos?

Usuario específico, Grupo de usuarios y Todos los usuarios.

44. ¿Qué tipo de bloqueos (locks) pueden aplicarse sobre un archivo para garantizar exclusión mutua?

Bloqueo a nivel de archivo o bloqueo de registros individuales.



45. El Sistema de Ficheros debe gestionar el espacio de almacenamiento secundario. Un bloque físico es una unidad de I/O de almacenamiento secundario, existen 3 enfoques en cuanto al tipo de bloques físicos. Indique cuáles son estos 3 enfoques y cuáles de ellos son incompatibles con las implementaciones tradicionales de memoria virtual.

1. Bloques fijos: ...

2. Bloques expandidos de longitud variable: ...

3. Bloques no expandidos de longitud variable: ...

2 y 3 son incompatibles con las implementaciones tradicionales de memoria virtual.

46. Al asignar bloques a un fichero, hay 2 enfoques: el estático y el dinámico. Describa brevemente cada uno de ellos y qué relación guardan en cuanto a la cantidad de bloques a asignar a un nuevo archivo.

La asignación estática de bloques implica pre-asignación, es decir, que los usuarios, las aplicaciones deberían indicar de antemano cuánto espacio ocupará un fichero, lo cual podría llevar a un malgasto de espacio por sobre-estimación.

La asignación dinámica asigna bloques a medida que lo necesita el fichero.

En ambos casos, el dilema común se encuentra en estimar el tamaño de porción apropiado, ¿qué cantidad de bloques asigno a un fichero?, es una tensión entre la eficiencia del fichero Vs eficiencia del sistema. Tener espacio contiguo incrementa el rendimiento del fichero (pero puede ocasionar que el SO agote los espacios contiguos requeridos para otras asignaciones), muchas porciones pequeñas incrementa el tamaño de las tablas necesarias para gestionar la asignación, porciones de tamaño fijo simplifica la re-asignación de espacio, porciones variables o pequeñas de tamaño fijo minimiza el espacio malgastado.

47. Existen 3 métodos de asignación de ficheros (file allocation): Asignación Contigua, Encadenada o Enlazada e Indexada. Describa brevemente cada uno de ellos.

1. Asignación Contigua: Pre-asignación. Se asigna un conjunto contiguo de bloques y existe una tabla de asignación de ficheros que contiene el nombre del fichero, el bloque inicial y la longitud. Se producirá fragmentación externa. ....

2. Asignación Encadenada: Con o sin pre-asignación. Se asignan bloques individuales, donde cada uno contiene un puntero al siguiente elemento (fichero=lista simplemente enlazada de bloques), es decir, no están contiguos. Tabla de asignación de ficheros contiene nombre de fichero y puntero al bloque inicial de la lista. No hay fragmentación externa. Asignación sencilla. ....

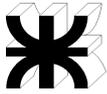
3. Asignación Indexada: La tabla de asignación de ficheros contiene nombre de fichero y puntero al bloque índice, el cual a su vez, contiene los bloques donde se encuentra el fichero. Elimina la fragmentación externa y aprovecha el principio de proximidad. Permite usar porciones de tamaño fijo o variable....

48. ¿Qué es "la tabla de asignación de ficheros" (file allocation table)?

La tabla de asignación de ficheros es una estructura de datos (tabla) que guarda la "traza" de las porciones de memoria secundaria asignadas a cada fichero.

49. ¿Por qué razón se produce "fragmentación externa" al utilizar asignación contigua?

Porque cuando se borra un archivo quedará un "hueco" de bloques libres, si dicho hueco se encuentra entre dos archivos, ese espacio no puede ser reutilizado eficientemente a menos que se vuelva a asignar un nuevo archivo de exactamente la misma cantidad de bloques que el archivo borrado. Esto puede llegar al extremo de que un sistema tenga miles de bloques libres en "huecos" pequeños generados entre archivos y no pueda asignar espacio contiguo de un nuevo archivo aunque éste no supere el espacio libre. En tal caso, el disco necesitará de un proceso de defragmentación para "reacomodar" todos los archivos a partir del comienzo del disco, dejando los bloques libre al final.



50. ¿Por qué razón la asignación encadenada o enlazada no tendría buen rendimiento sobre un archivo de acceso directo?

Porque ir al i-esimo bloque implica recorrer la lista enlazada hacia adelante a partir de un bloque inicial y así sucesivamente para cada petición. Obtener el i-esimo bloque puede implicar la lectura de muchos bloques.

51. ¿Por qué razón se necesita hacer una compactación o una consolidación del disco luego de cierto tiempo de uso?

Para transformar el espacio libre en espacio libre contiguo y al añadir nuevos ficheros a éstos se les podría otorgar bloques contiguos favoreciendo el principio de proximidad.

52. ¿Existe alguna diferencia dentro de la tabla de asignación archivos cuando usamos una asignación indexada utilizando porciones fijas que cuando usamos porciones variables?

No, en ambos casos la FAT solo guarda el puntero al bloque índice. Si cambia la estructura del bloque índice que en el caso de las porciones variables se debe agregar además del número de bloque, su longitud.

53. Gestionar el espacio en disco implica gestionar el espacio libre y ocupado. La tabla de bits es un método posible para gestionar espacio libre. Indique cuál es su principal desventaja.

La principal desventaja es que el tamaño de la tabla de bits crece proporcionalmente al tamaño del disco, ocupando mucho espacio y haciendo las búsquedas muy lentas, incluso aunque la misma se cargue en memoria principal. La mayoría de los SO que usan esta técnica utilizan estructuras de datos auxiliares en memoria (a modo de resumen de la tabla de bits) para acelerar la ubicación de espacio libre.

54. En UNIX/Linux, una estructura de tipo inode esta asociado a un único archivo, entonces ¿Por qué se dice que puede haber muchos nombres de archivos asociados a un mismo inode?

Debido a que en Unix (al igual que en Windows) se puede enlazar una o más entradas del directorio a un archivo físico determinado. Cada i-node representa un único archivo físico, pero sobre un archivo físico puede haber n enlaces. Cada enlace tendrá una entrada el directorio que estará apuntando a un mismo i-nodo. Fig. 12.14 Pag. 579 (podría haber alguna entrada del directorio apuntando al mismo i-nodo).

55. Responda V o F: En UNIX/Linux los directorios son estructuras especiales, distintas a un archivo normal.

Falso. Un directorio es un archivo normal cuyo contenido es una lista de nombres (que representan archivos u otros directorios) más un número que indica un desplazamiento dentro de la tabla de nodos-i.