Sistemas y Aplicaciones Informáticas

Tema 19. Sistemas Operativos: Gestión de Archivos y Dispositivos.

1. Ámbito de docencia.

- Sistemas informáticos monousuario y multiusuario (ASI 1).
- Sistemas informáticos multiusuario y en red (DAI 1).
- Sistemas operativos en entornos monousuario y multiusuario (ESI 1).

2. Gestión de archivos.

2.1. Archivos y directorios.

2.1.1. Archivos. Características. Atributos y descriptores de archivos.

- El sistema operativo necesita almacenar a largo plazo y recuperar posteriormente la información que manejan los procesos, de manera que los datos permanezcan cuando éstos terminan. Para ello utiliza los dispositivos de almacenamiento secundario, realizando una abstracción de sus propiedades físicas a través del concepto de archivo como unidad lógica de almacenamiento.
- Un archivo es un conjunto de datos organizado que reside en un soporte de almacenamiento secundario. Constan de un nombre y unos atributos, y presentan las siguientes características:
  - Son dinámicos, puesto que su tamaño puede variar durante la ejecución del proceso.
  - Proporcionan independencia de los datos respecto de los procesos. Un mismo archivo puede ser utilizado por diferentes procesos y no está limitado al procesos que lo creó.
  - Su capacidad es muy superior a la de la memoria, aunque depende del soporte físico.
- Los atributos son un conjunto de propiedades como contraseña, creador, propietario, longitud, tipo, protección, tamaño, información de creación y modificación, e información de control (oculto, normal, directorio, etc). El sistema operativo almacena los atributos de cada archivo en una estructura de datos denominada descriptor de archivo, localizada:
  - *En disco*. Persiste mientras exista el archivo. Puede residir en la misma ubicación lógica que el archivo o en una zona especial del disco dedicada a ello.
  - *En memoria principal*. Cuando un proceso abre un archivo, el descriptor de éste se asocia con dicho proceso y se carga en memoria, añadiendo algunos campos adicionales.

2.1.2. Estructuras internas de archivo. Modos de acceso. Tipos de archivo.

- Las estructuras internas de archivo más comunes son las siguientes:
  - *Secuencia de bytes*. Consiste en una serie no estructurada de bytes, a los que puede accederse si se conoce su desplazamiento desde el origen. El sistema operativo desconoce el contenido de los archivos, pero debe reconocer la estructura de sus archivos ejecutables.
  - *Secuencia de registros*. Formada por una serie de registros de longitud fija, cada uno con su propia estructura interna. El acceso al archivo se realiza a través de los registros.
  - *Árbol de registros*. Consta de un conjunto de registros de longitud variable estructurados en forma de árbol, y ordenados por un campo clave que se encuentra en una posición fija dentro del registro, y mediante la cual se puede acceder a uno en concreto.
- Existen dos modos de acceso principales a la información de los archivos:
  - *Secuencial*. Para acceder a un registro es necesario recorrer uno a uno por orden todos los registros partiendo del primero. Es necesario controlar cuándo se llega al final del archivo.
  - *Directo*. Se puede acceder a sus registros o bytes en cualquier orden, en escritura y lectura.

- La mayor parte de los sistemas operativos reconocen varios tipos de archivo:
  - *Archivos regulares*. Contienen información del usuario. Pueden ser:
    - **De texto**. Consisten en sucesiones de caracteres producto de la conversión del formato interno de los datos a un código alfanumérico. Son de acceso secuencial y están organizados en registros de longitud indefinida separados por un carácter especial de fin de línea. No hay una relación uno a uno entre los datos existentes en la memoria y los que aparecen en el soporte de almacenamiento.
    - **Binarios**. Son una imagen del formato interno de los datos en la memoria. Son de acceso directo, y se organizan en registros de longitud fija o en secuencias de bytes.
  - *Directorios*. Utilizados para organizar y estructurar el conjunto de archivos del sistema.
  - *Archivos especiales*. Sirven para representar dispositivos de E/S. Pueden ser:
    - **De bloque**. Para dispositivos de E/S que funcionan en modo bloque (discos, cintas).
    - **De carácter**. Para dispositivos de E/S en modo carácter (impresoras, terminales, redes).

2.1.3. Directorios. Contenido y estructura. Directorio raíz y rutas de acceso.

- Los directorios son un tipo especial de archivos que se utilizan como contenedores de archivos y otros directorios. Almacenan en forma de tabla un conjunto de datos por cada uno de los elementos que contienen. Su principal función es asociar el nombre de los archivos y directorios que contienen con la información necesaria para encontrar los bloques de datos en el disco.
- Una entrada de directorio suele almacenar la siguiente información:
  - El nombre simbólico. Las reglas para construir los nombres varían de sistema a sistema.
  - La dirección de inicio en el disco y un puntero a la posición actual de lectura o escritura.
  - Lista de atributos. Protección de acceso, creador, propietario, longitud de registro, fecha y hora de creación, fecha y hora de modificación, tamaño actual, etc.
- Hay varios tipos de estructura de directorios sobre un disco:
  - *Directorio de un nivel*. Existe un único directorio que contiene todas las entradas de archivos. Es la estructura más simple, fácil de entender e implementar, pero no permite clasificar la información y provoca conflictos de nombres en sistemas multiusuario.
  - *Directorio de dos niveles*. Cada usuario dispone de su propio directorio y todos se incluyen dentro de un directorio maestro. Esta estructura soluciona los conflictos de nombres entre usuarios, pero no permite compartir archivos entre ellos.
  - *Directorio de varios niveles*. Un directorio puede incluir otros directorios, y éstos a su vez pueden contener otros directorios. La estructura puede tener forma de árbol jerárquico o de grafo acíclico (éste formado por archivos, directorios y los enlaces a archivos compartidos).
- Todas las estructuras de directorios parten de un un directorio inicial llamado directorio raíz, que debe encontrarse en una dirección conocida por el volumen desde el que se arranca el sistema. Para acceder a un archivo hay que especificar su ruta de acceso a través de los directorios. Los componentes de la ruta se separan mediante algún carácter especial. Existen dos métodos:
  - *Ruta absoluta*. Toma como origen el directorio raíz hasta llegar al archivo, es constante.
  - *Ruta relativa*. Toma como origen el directorio actual hasta llegar al archivo, puede variar en función del directorio actual del sistema en cada instante.

2.2. Gestor de archivos.

2.2.1. Sistemas de archivos. Formateado y estructura general. Funciones.

- Un sistema de archivos es un modo de organización lógica de la información en un dispositivo de almacenamiento, que debe ser independiente del soporte físico concreto sobre el que se encuentre. Reside permanentemente en el dispositivo de almacenamiento, y es el encargado de proporcionar una interfaz entre el usuario y el almacenamiento físico de la información.
- El proceso de creación de un sistema de archivos en una partición de un dispositivo de almacenamiento se denomina formateado, durante el cual se realiza lo siguiente:
  - El espacio ocupado en el dispositivo por el sistema de archivos se divide en bloques numerados lógicamente del cero al máximo del sistema de archivos. Los archivos se dividen a su vez en bloques de igual tamaño numerados lógicamente del cero al máximo del archivo. A cada bloque lógico de un archivo se le asigna un bloque lógico del sistema de archivos.
  - Se escribe en los bloques físicos de la partición la información necesaria para mantener los archivos y directorios (*metainformación*), que en general es la siguiente:
    - **Información de arranque del sistema operativo (boot)**. Contiene código del sistema operativo para la carga del mismo. Sólo contiene información en los discos de arranque.
    - **Descripción de la propia estructura y gestión del sistema de archivos**. Mantiene información sobre el tamaño de bloque, el número máximo de archivos, el estado de ocupación de los bloques de datos y la localización de los descriptores de archivos.
    - **Descriptores de archivos**. Tienen estructura y tamaño variable. Existe un número fijo de descriptores igual al máximo número de archivos que pueden crearse.
    - **Bloques de datos**. Almacenan la información de los archivos.
- Dos ejemplos de estructuras de sistemas de archivos:
  - *En Unix*. Bloque de arranque | Superbloque | Inodos | bloques de datos.
  - *En MS-DOS*. Bloque de arranque | Tabla de asignación | Directorio raíz | bloques de datos.
- Las funciones que desempeña el sistema de archivos son las siguientes:
  - Permitir la manipulación de archivos y la transferencia de información entre ellos.
  - Relacionar el nombre de un archivo con los datos que éste contiene.
  - Proporcionar el acceso controlado a los archivos compartidos (lectura, escritura, ejecución) y la posibilidad del cifrado de la información que contienen.
  - Estructurar los archivos de la manera más apropiada para cada aplicación.
  - Proporcionar posibilidades de copias de seguridad y recuperación de la información.

2.2.2. Sistema de gestión básica de archivos. Funciones. Enlaces físicos y simbólicos.

- El sistema de gestión básica de archivos es la parte del sistema operativo que se encarga de la estructuración y la gestión del almacenamiento físico de los archivos. Se divide en:
  - *Módulo de organización de archivos*. Recibe la información que el sistema de archivos le envía a partir de un nombre de archivo simbólico. Se encarga de traducir las direcciones de bloques lógicos en direcciones de bloques físicos.
  - *Sistema básico de archivos*. Recoge las direcciones de bloques físicos y envía una serie de órdenes al manejador de dispositivo apropiado para leer y escribir bloques de datos.

- Las funciones que desempeña el sistema de gestión básica de archivos son las siguientes:
  - Recibir las llamadas al sistema sobre gestión de archivos.
  - Localizar los bloques del disco que contienen los datos de un archivo.
  - Gestionar el espacio de almacenamiento de los discos.
  - Gestionar la caché de disco.
  - Comunicarse con los manejadores de dispositivo para indicarles la operación a realizar.
- La compartición de archivos permite que un archivo pueda aparecer simultáneamente en varios directorios sin necesidad de realizar varias copias del mismo. Para ello se utiliza una conexión especial entre un directorio y un archivo de otro directorio denominada enlace. Existen dos tipos:
  - *Enlaces físicos*. Son entradas de directorio con el mismo número de descriptor de archivo que el archivo enlazado. Por tanto sólo pueden crearse enlaces físicos con archivos del mismo sistema de archivos. A todos los efectos funcionan como copias del archivo original, por lo que sólo se puede eliminar un archivo si se eliminan todos sus enlaces físicos.
  - *Enlaces simbólicos*. Son archivos con descriptor de archivo propio que contienen la ruta de acceso al archivo enlazado. Por tanto permiten enlazar archivos entre diferentes sistemas de archivos. Si se elimina un archivo, todos sus enlaces simbólicos quedan inservibles.

2.3. Gestión del espacio en disco.

2.3.1. Asignación contigua.

- Cada archivo ocupa un conjunto de bloques consecutivos en el disco. La situación de un archivo queda perfectamente determinada por medio de la dirección de su primer bloque y su longitud. Los dispositivos no direccionables como las cintas sólo permiten este tipo de asignación.
- Presenta los siguientes inconvenientes:
  - Se parte del hecho de conocer el espacio máximo que ocupará un archivo en el instante de su creación, algo que con frecuencia es desconocido. Si el archivo crece y no hay suficientes bloques libres contiguos, es necesario cambiarlo de posición dentro del disco.
  - Si se manejan archivos grandes, el tamaño de los bloques libres contiguos tiende a ser cada vez más pequeño resultando inutilizables, por lo que se produce fragmentación externa.

2.3.2. Asignación no contigua.

- El contenido del archivo se almacena en disco en bloques no consecutivos. De esta manera los archivos pueden crecer o disminuir sin restricción, se realiza un mejor aprovechamiento del espacio de disco y desaparece la fragmentación externa. Sin embargo, se complica la gestión ya que es necesario llevar el control de la secuencia de bloques que forman un archivo.
- Existen varias posibilidades de gestión:
  - *Lista enlazada de bloques*. Cada archivo está formado por una lista enlazada de bloques de datos, en la que cada bloque contiene un puntero al siguiente bloque. La entrada de directorio guarda la dirección del primer bloque del archivo, y el último bloque contiene una marca de fin de archivo. Presenta los siguientes inconvenientes:
    - El acceso a un determinado bloque supone pasar por todos los bloques anteriores.
    - Si se daña cualquier puntero al siguiente bloque, el resto del fichero queda ilocalizable.

- *Tabla de localización*. Consiste en mantener una tabla en memoria en la que a cada bloque de disco le corresponde una entrada. El contenido de cada entrada puede ser el número del siguiente bloque de datos del archivo, una marca de fin de archivo o una marca de bloque libre. La entrada en el directorio guarda el número del primer bloque de datos de archivo. Presenta el inconveniente de que el acceso a un determinado bloque supone recorrer la tabla de asignación comenzando por la entrada correspondiente al primer bloque.
- *Tabla de bloques de índices*. Existen en el disco bloques de datos y bloques de índices a bloques de datos. Para cada archivo existe una tabla que relaciona el descriptor del archivo con sus correspondientes bloques de datos y bloques de índices. Los bloques de índices pueden apuntar a su vez a otros bloques de índices, posibilitando la existencia de archivos de grandes tamaños. Presenta el inconveniente de que los bloques de índices ocupan espacio en disco que deja de estar disponible para almacenar datos.

2.3.3. Gestión de bloques libres.

- Debido a que el espacio de disco es limitado, se hace necesario reutilizar el espacio ocupado por archivos que han sido borrados para almacenar nuevos archivos. Por tanto el sistema operativo debe mantener una lista de bloques libres.
- Al crear un archivo, el sistema operativo examina esta lista en busca de bloques libres, los asigna y elimina dichos bloques de la lista. Cuando se borra un archivo de forma efectiva, los bloques que éste ocupaba se añaden a la lista de bloques libres.
- Hay tres métodos para llevar el control de los bloques libres:
  - *Lista enlazada en memoria*. El sistema sólo mantiene un puntero al primer bloque libre, el cual apunta al siguiente, y así sucesivamente hasta llegar al último bloque libre.
  - *Indexación*. Trata el espacio libre como si fuera un archivo y utiliza una tabla índice en la que existe una entrada para cada bloque libre del disco.
  - *Mapa de bits*. Cada bloque del disco se representa por medio de un bit, que estará puesto a un valor lógico concreto si el bloque está asignado a algún archivo y a su complementario cuando el bloque esté libre.

2.3.4. Tamaño óptimo de los bloques.

- Es necesario determinar el tamaño de bloque, puesto que es lo mínimo que ocupará un archivo:
  - Los bloques grandes suponen un desperdicio de espacio en el caso de ficheros pequeños.
  - Los bloques pequeños implican que cada archivo ocupará muchos bloques. Por tanto el acceso a un archivo supone que la cabeza de lectura-escritura tiene que realizar muchos saltos por la superficie del disco para acceder a todos los bloques.
- El compromiso que se toma en la actualidad consiste en elegir un tamaño de bloque múltiplo en potencia de dos de 512 bytes (tamaño de un sector), que oscila entre 1 y 64 KB.

2.4. Protección e integridad de archivos.

2.4.1. Protección de archivos. Dominios y matriz de acceso.

- La protección de archivos es un conjunto de mecanismos utilizados para controlar el acceso de los procesos a su contenido, puesto que un proceso sólo debe poder acceder a aquellos recursos que necesita en ese momento para completar su tarea y para los cuales está autorizado.

- Se basa en el concepto de dominio como una serie de parejas formadas por un recurso del sistema y el conjunto de operaciones que se pueden llevar a cabo sobre él, denominado derechos de acceso. Un recurso puede pertenecer a varios dominios con distintos derechos de acceso.
- Cada proceso trabaja dentro de un dominio, el cual especifica los recursos a los cuales puede tener acceso. Un proceso puede cambiar de un dominio a otro si el sistema operativo lo permite.
- Las relaciones entre dominios y recursos se pueden representar de forma abstracta mediante una matriz denominada matriz de acceso, en la cual las filas representan dominios y las columnas recursos. Una manera de representar la matriz es mediante una tabla formada por tripletas ordenadas (dominio, recurso, derechos), y para simplificarla se implementa:
  - *Por columnas (listas de acceso para recursos).* A cada recurso se le asocia una lista ordenada con todos los dominios que acceden al recurso y los derechos de cada dominio. Por ejemplo, los bits de protección de lectura, escritura y ejecución de los ficheros UNIX.
  - *Por filas (listas de capacidades para dominios).* A cada dominio se le asocia una lista ordenada con todos los recursos a los cuales puede tener acceso y los derechos sobre cada recurso. Permite llevar a cabo la revocación de acceso a un determinado recurso.
- Las matrices de acceso son dinámicas, se pueden crear y eliminar recursos, dominios y derechos. En cada momento la matriz determina lo que puede hacer un proceso, no lo que no está autorizado. Está integrada en el sistema y es producto de la política de administración.

2.4.2. Integridad de archivos. Copias de seguridad y replicación de discos (RAID).

- La integridad de los archivos está relacionada con la protección frente a la destrucción y alteración accidental o intencionada de la información almacenada. Los problemas causados por la utilización de bloques defectuosos o de fallos del sistema en mitad de una operación de E/S de un bloque pueden resolverse mediante algunas utilidades del sistema operativo (fsck, scandisk).
- Para prevenir la pérdida de la información se recurre a dos técnicas:
  - *Copias de seguridad*. Se tratan de volcados de la información a proteger en un dispositivo de almacenamiento alternativo para permitir una recuperación posterior. Pueden ser completos o incrementales (sólo se copian los datos modificados desde la última copia).
  - *Replicación de discos*. Se trata de almacenar información redundante en un conjunto de discos con el fin de aumentar la tolerancia a fallos y mejorar el rendimiento. Existen siete configuraciones (RAID 0 a RAID 6) con diferentes niveles de redundancia, y pueden ser implementados por hardware o por software.

3. Gestión de discos.

3.1. Estructura física. Tiempos de búsqueda, latencia y transferencia.

- Están formados por una serie de discos magnéticos superpuestos que giran alrededor de un eje común. Se leen o escriben mediante un conjunto de cabezas, una por cada superficie de disco, montadas sobre un brazo que puede hacer mover todo el conjunto a lo largo del radio del disco.
- Cada disco está organizado en pistas concéntricas, que a su vez están divididas en sectores. Todos los sectores tienen una capacidad igual de bytes, aunque a medida que se alejan del centro del disco son más grandes. Para una posición dada de las cabezas de lectura/escritura, la serie de pistas accesibles de todos los discos forman un cilindro.

- El tiempo necesario para ejecutar una operación de entrada/salida es la suma de los tiempos de:
  - *Búsqueda*. Tiempo necesario para que las cabezas se desplacen hasta la pista deseada.
  - *Latencia*. Tiempo de rotación del disco desde el sector actual hasta el sector deseado.
  - *Transferencia*. Tiempo necesario para transferir un sector entre el disco y el buffer de E/S.

3.2. Planificación del disco. Objetivos. Algoritmos de planificación.

- La planificación del disco consiste en la ordenación de la cola de peticiones de entrada/salida sobre dicho disco en los sistemas de multiprogramación, debido a que la generación de peticiones puede ser mucho más rápida que la atención de las mismas.
- Los objetivos de la planificación de discos son los siguientes:
  - Minimizar el movimiento mecánico de las cabezas del disco.
  - Aumentar productividad del disco (número de peticiones/sg.).
  - Disminuir la varianza y los tiempos medios de respuesta.
  - Evitar la inanición de peticiones.
- Se dispone de distintos algoritmos para elegir la petición siguiente a ser servida:
  - *Primero en llegar, primero en ser servido (FCFS)*. Consiste en atender las solicitudes por orden de llegada. Trata a todas las peticiones por igual, pero perjudica a las peticiones situadas al final de la cola y hace que el número de movimientos del brazo sea elevado.
  - *Tiempo de búsqueda más corto primero (SSTF)*. Se atiende la petición que requiere el menor movimiento de la cabeza de lectura/escritura desde su posición actual. Tiende a favorecer a las pistas del centro del disco y a retrasar las peticiones más antiguas.
  - *Búsqueda en ascensor (SCAN)*. Las cabezas empiezan en un extremo del disco y se desplazan hacia el otro sirviendo las peticiones pendientes que encuentra a su paso hasta alcanzar la última pista. Se cambia entonces la dirección de las cabezas y el rastreo sigue en sentido opuesto. Si durante el recorrido aparecen nuevas solicitudes se atienden primero, produciendo esperas en las anteriores. Las pistas centrales siguen siendo más visitadas que las exteriores, pero menos que en SSTF, y tiene menor varianza del tiempo de respuesta.
  - *SCAN de N pasos*. Es una planificación SCAN en la que sólo se da servicio a las peticiones que se encuentran en espera al comienzo del recorrido. Las nuevas solicitudes que aparecen durante un recorrido de las cabezas se agrupan para atenderlas en el recorrido inverso posterior. Tiene menos varianza en los tiempos de respuesta que SSTF y SCAN.
  - *Búsqueda circular (C-SCAN)*. Es una planificación SCAN en la que cuando las cabezas llegan a un extremo del disco vuelven inmediatamente al otro extremo para comenzar un nuevo recorrido. No discrimina ningún cilindro y tiene una varianza muy baja.
  - *LOOK y C-LOOK*. Son mejoras de las planificaciones SCAN y C-SCAN respectivamente, en las que se cambia la dirección cuando no hay más peticiones en la dirección actual.

4. Ejemplos de sistemas operativos.

4.1. Gestión de archivos en Windows XP.

- Los archivos son secuencias de bytes cuyos descriptores se almacenan en los propios directorios a los que pertenecen. Existe un árbol de directorios por cada dispositivo lógico del sistema.

- Los archivos pueden tener los siguientes atributos: oculto, normal o sin atributos especiales, sólo lectura, sistema, temporal (se mantiene en memoria el mayor tiempo posible) y escritura atómica (sus datos son críticos; esto hace que se aumente la frecuencia de escritura de memoria a disco).
- El gestor de archivos (Executive) usa un mecanismo de escritura diferida de los cambios realizados a un archivo en memoria, de manera que se escribe a disco cuando se cierra el archivo, o cuando el sistema está desocupado, o cuando es necesario hacer interacambio.
- La E/S a archivos puede realizarse de dos modos:
  - *Modo síncrono*. Es el habitual. El hilo que inicia una operación de E/S sobre un archivo es suspendido por el Executive hasta que esa E/S se completa. Cuando un hilo abre un archivo para un acceso síncrono, el sistema operativo asigna a dicho hilo un puntero de lectura/escritura exclusivo sobre el archivo. El puntero se modifica al leer y al escribir.
  - *Modo asíncrono*. El hilo que inicia una operación de E/S puede seguir ejecutándose, y cuando necesite los datos de la E/S se pondrá en espera, de manera que si para ese tiempo la E/S se ha completado, obtendrá los resultados inmediatamente. No existe un puntero de lectura/escritura, por lo que hay que indicar la dirección de inicio de cada acceso.
- Se soportan tres tipos de sistemas de archivos distintos:
  - *FAT16*. Basado en asignación no contigua de bloques por tabla de localización. Permite direccionar como máximo 2^{16} bloques de 32 KB (2 GB). El sector de arranque ocupa un sector (512 Bytes). El directorio raíz tiene un tamaño limitado y se encuentra en una posición fija del disco. Consume pocos recursos del sistema y es el mejor para discos y/o particiones de menos de 200MB. No pueden aplicarse permisos sobre archivos y directorios.
  - *FAT32*. Es una mejora de FAT16. Amplía el espacio de direccionamiento a 32 bits, aunque sólo se usan 28 y las 4 restantes están reservadas para el sistema. En la práctica permite crear particiones de unos 124 GB. El sector de arranque ocupa 32 sectores y el directorio raíz es un archivo más, permitiendo su crecimiento y evitando la limitación de archivos.
  - *NTFS 3.1*. Es el sistema de archivos nativo de Windows XP. Utiliza direccionamiento de 64 bits y presenta características de seguridad (listas de control de acceso y cifrado de ficheros), administración de cuotas de disco, compresión individual y registro de transacciones (journaling). Soporta enlaces físicos (**Fsutil**) y simbólicos, y admite montar particiones FAT16, FAT32 o NTFS. Es el mejor para volúmenes de 400 MB o más.

4.2. Gestión de archivos en GNU/Linux.

- Linux utiliza de manera nativa el sistema de archivos Ext2, basado en el sistema UNIX System V. Un archivo es una secuencia de bytes representado por un descriptor denominado inodo, que se identifica con un número, cuyo contenido es el siguiente:
  - Número de identificación de usuario y grupo del propietario del archivo.
  - Tipo de archivo y permisos de acceso para el propietario, su grupo y el resto de usuarios.
  - Fecha de última modificación y número de enlaces físicos al archivo.
  - Doce punteros a bloques de datos y tres punteros a bloques de índices con uno, dos y tres niveles de indirección respectivamente.
- Ext2 divide la partición en grupos de bloques, cada uno de los cuales contiene lo siguiente:

    - *Un superbloque*. Es una estructura de datos que contiene información global sobre el sistema de archivos, como el tamaño de los bloques y el número de bloques e inodos libres.
    - *Un descriptor de grupo*. Almacena información sobre la localización de los mapas de bits, el número de bloques e inodos libres en el grupo y el número de directorios que contiene.
    - *Dos mapas de bits*. Guardan información sobre los bloques libres y los inodos libres respectivamente. Un mapa de bit es una estructura de datos en la que cada bit representa si un bloque o un inodo se encuentra libre u ocupado.
    - *Una tabla de inodos*. Contiene una lista de todos los inodos libres y ocupados del grupo
    - *Un área de datos*. Contiene los bloques de datos y de índices del grupo. La división del sistema de archivos en grupos mantiene los inodos y los bloques de datos más cerca.
- Cada entrada de directorio contiene el nombre simbólico de los archivos que contiene y su número de inodo. Existe un único árbol de directorios sobre el que puede montar en cualquier rama otro sistema de archivos distinto. El usuario no percibe en qué sistema de archivos residen realmente sus datos. Cada sistema de archivos puede ser de tipos diferentes y tiene su propia metainformación. El sistema operativo en todo caso utiliza la metainformación del sistema de archivos actual, llevando el control de los sistemas de archivos montados.
- El sistema de archivos utiliza tres estructuras de datos en memoria principal:
    - *Tabla de inodos*. Cuando se abre un archivo se debe cargar en memoria su inodo en una Tabla de inodos en memoria, la cual funciona de forma semejante a una tabla hash.
    - *Tabla de archivos abiertos*. Tabla única que contiene una entrada por cada archivo abierto. Contiene el tipo de apertura (lectura/escritura) y un puntero al inodo del archivo abierto.
    - *Tabla de descriptores de archivos de proceso*. Existe una por cada proceso. Contiene una entrada por cada archivo abierto por el proceso. Apunta a una entrada en la tabla de archivos abiertos. Cuando se abre un archivo al usuario se le devuelve un puntero a este descriptor.
- Linux permite montar varios tipos de sistemas de archivos distintos. Para ello dispone de un sistema de archivos virtual (VFS) que proporciona una capa de abstracción adicional entre los programas de usuario y las distintas implementaciones de cada sistema de archivos. El VFS es un sistema de archivos genérico que ofrece una estructura jerárquica y conceptual de archivos y directorios, siguiendo el modelo de sistemas de ficheros de UNIX.
- Un sistema de ficheros especial de Linux es el directorio /proc, en el que se crea un subdirectorio de nombre igual a su PID por cada proceso existente en el sistema. Contiene información sobre los procesos, y otra mucha información acerca de la CPU, las particiones de disco, los dispositivos, los vectores de interrupción, los sistemas de ficheros, los módulos cargados, etc.
- El sistema de ficheros Ext3 es una implementación de Ext2 con journaling, que se basa en llevar un registro de todas las transacciones de lectura/escritura que sufre un sistema de archivos para aumentar la tolerancia a errores.