

Un sistema de archivos (file system) es una estructura de directorios con algún tipo de organización el cual nos permite almacenar, crear y borrar archivos en diferentes formatos. En esta sección se revisarán conceptos importantes relacionados a los sistemas de archivos.

Almacenamiento Físico de Datos

En un sistema de cómputo es evidente que existe la necesidad por parte de los usuarios y aplicaciones de almacenar datos en algún medio, a veces por periodos largos y a veces por instantes. Cada aplicación y cada usuario debe tener ciertos derechos con sus datos, como son el poder crearlos y borrarlos, o cambiarlos de lugar; así como tener privacidad contra otros usuarios o aplicaciones. El subsistema de archivos del sistema operativo se debe encargar de estos detalles, además de establecer el formato físico en el cual almacenará los datos en discos duros, cintas o discos flexibles. Debe ser conocido por todos que tradicionalmente la información en los sistemas modernos se almacena en discos duros, flexibles y unidades de disco óptico, y en todos ellos se comparten algunos esquemas básicos para darles formato físico: las superficies de almacenamiento son divididas en círculos concéntricos llamados "pistas" y cada pista se divide en "sectores". A la unión lógica de varias pistas a través de varias superficies "paralelas" de almacenamiento se les llama "cilindros", los cuales son inspeccionados al momento de lectura o escritura de datos por las respectivas unidades físicas llamadas "cabezas". Las superficies de almacenamiento reciben el nombre de "platos" y generalmente están en movimiento rotatorio para que las cabezas accedan a las pistas que los componen. Los datos se escriben a través de los sectores en las pistas y cilindros modificando las superficies por medio de las cabezas.

El tiempo que una cabeza se tarda en ir de una pista a otra se le llama "tiempo de búsqueda" y dependerá de la distancia entre la posición actual y la distancia a la pista buscada. El tiempo que tarda una cabeza en ir del sector actual al sector deseado se le llama tiempo de latencia y depende de la distancia entre sectores y la velocidad de rotación del disco. El impacto que tiene las lecturas y escrituras sobre el sistema está determinado por la tecnología usada en los platos y cabezas y por la forma de resolver las peticiones de lectura y escritura, es decir, los algoritmos de planificación.

Algoritmos de planificación de peticiones

Los algoritmos de planificación de peticiones de lectura y escritura a discos se encargan de registrar dichas peticiones y de responderlas en un tiempo razonable. Los algoritmos más comunes para esta tarea son:

- **Primero en llegar, primero en ser servido (FIFO):** Las peticiones son encoladas de acuerdo al orden en que llegaron y de esa misma forma se van leyendo o escribiendo las mismas. La ventaja de este algoritmo es su simplicidad y no causa sobrecarga, su desventaja principal es que no aprovecha para nada ninguna característica de las peticiones, de manera que es muy factible que el brazo del disco se mueva muy ineficientemente, ya que las peticiones pueden tener direcciones en el disco unas muy alejadas de otras. Por ejemplo, si se están haciendo peticiones a los sectores 6,10,8,21 y 4, las mismas serán resueltas en el mismo orden. _ **Primero el más cercano a la posición actual:** En este algoritmo las peticiones se ordenan de acuerdo a la posición actual de la cabeza lectora, sirviendo primero a aquellas peticiones más cercanas y reduciendo, así, el movimiento del brazo, lo cual constituye la ventaja principal de este algoritmo. Su desventaja consiste en que puede haber solicitudes que se queden esperando para siempre, en el infortunado caso de que existan peticiones muy alejadas y en todo momento estén entrando peticiones que estén más cercanas. Para las peticiones 6,10,8,21 y 4, las mismas serán resueltas en el orden 4,6,8,10 y 21.

- Por exploración (algoritmo del elevador): En este algoritmo el brazo se estará moviendo en todo momento desde el perímetro del disco hacia su centro y viceversa, resolviendo las peticiones que existan en la dirección que tenga en turno. En este caso las peticiones 6,10,8,21 y 4 serán resueltas en el orden 6,10,21,8 y 4; es decir, la posición actual es 6 y como va hacia los sectores de mayor numeración (hacia el centro, por ejemplo), en el camino sigue el sector 10, luego el 21 y ese fue el más central, así que ahora el brazo resolverá las peticiones en su camino hacia afuera y la primera que se encuentra es la del sector 8 y luego la 4. La ventaja de este algoritmo es que el brazo se moverá mucho menos que en FIFO y evita la espera indefinida; su desventaja es que no es justo, ya que no sirve las peticiones en el orden en que llegaron, además de que las peticiones en los extremos interior y exterior tendrán un tiempo de respuesta un poco mayor.
- Por exploración circular: Es una variación del algoritmo anterior, con la única diferencia que al llegar a la parte central, el brazo regresa al exterior sin resolver ninguna petición, lo cual proveerá un tiempo de respuesta más cercana al promedio para todas las peticiones, sin importar si están cercas del centro o del exterior.

Asignación del espacio de almacenamiento

El subsistema de archivos se debe encargar de localizar espacio libre en los medios de almacenamiento para guardar archivos y para después borrarlos, renombrarlos o agrandarlos. Para ello se vale de localidades especiales que contienen la lista de archivos creados y por cada archivo una serie de direcciones que contienen los datos de los mismos. Esas localidades especiales se llaman directorios. Para asignarle espacio a los archivos existen tres criterios generales que se describen enseguida.

- Asignación contigua: Cada directorio contiene la los nombres de archivos y la dirección del bloque inicial de cada archivo, así como el tamaño total de los mismos. Por ejemplo, si un archivo comienza en el sector 17 y mide 10 bloques, cuando el archivo sea accesado, el brazo se moverá inicialmente al bloque 17 y de ahí hasta el 27. Si el archivo es borrado y luego creado otro más pequeño, quedarán huecos inútiles entre archivos útiles, lo cual se llama fragmentación externa.
- Asignación encadenada: Con este criterio los directorios contienen los nombres de archivos y por cada uno de ellos la dirección del bloque inicial que compone al archivo. Cuando un archivo es leído, el brazo va a esa dirección inicial y encuentra los datos iniciales junto con la dirección del siguiente bloque y así sucesivamente. Con este criterio no es necesario que los bloques estén contiguos y no existe la fragmentación externa, pero en cada "eslabón" de la cadena se desperdicia espacio con las direcciones mismas. En otras palabras, lo que se crea en el disco es una lista ligada.
- Asignación con índices (indexada): En este esquema se guarda en el directorio un bloque de índices para cada archivo, con apuntadores hacia todos sus bloques constituyentes, de manera que el acceso directo se agiliza notablemente, a cambio de sacrificar varios bloques para almacenar dichos apuntadores. Cuando se quiere leer un archivo o cualquiera de sus partes, se hacen dos accesos: uno al bloque de índices y otro a la dirección deseada. Este es un esquema excelente para archivos grandes pero no para pequeños, porque la relación entre bloques destinados para índices respecto a los asignados para datos es incosteable..

Métodos de acceso en los sistemas de archivos

Los métodos de acceso se refiere a las capacidades que el subsistema de archivos provee para acceder datos dentro de los directorios y medios de almacenamiento en general. Se ubican tres formas generales: acceso secuencial, acceso directo y acceso directo indexado.

- Acceso secuencial: Es el método más lento y consiste en recorrer los componentes de un archivo uno en uno hasta llegar al registro deseado. Se necesita que el orden lógico de los registros sea igual al orden físico en el medio de almacenamiento. Este tipo de acceso se usa comunmente en cintas y cartuchos.

- **Acceso directo:** Permite acceder cualquier sector o registro inmediatamente, por medio de llamadas al sistema como la de seek. Este tipo de acceso es rápido y se usa comúnmente en discos duros y discos o archivos manejados en memoria de acceso aleatorio. _ **Acceso directo indexado:** Este tipo de acceso es útil para grandes volúmenes de información o datos. Consiste en que cada archivo tiene una tabla de apuntadores, donde cada apuntador va a la dirección de un bloque de índices, lo cual permite que el archivo se expanda a través de un espacio enorme. Consume una cantidad importante de recursos en las tablas de índices pero es muy rápido.

Operaciones soportadas por el subsistema de archivos

Independientemente de los algoritmos de asignación de espacio, de los métodos de acceso y de la forma de resolver las peticiones de lectura y escritura, el subsistema de archivos debe proveer un conjunto de llamadas al sistema para operar con los datos y de proveer mecanismos de protección y seguridad. Las operaciones básicas que la mayoría de los sistemas de archivos soportan son:

- **Crear (create) :** Permite crear un archivo sin datos, con el propósito de indicar que ese nombre ya está usado y se deben crear las estructuras básicas para soportarlo.
- **Borrar (delete):** Eliminar el archivo y liberar los bloques para su uso posterior.
- **Abrir (open):** Antes de usar un archivo se debe abrir para que el sistema conozca sus atributos, tales como el dueño, la fecha de modificación, etc. _ **Cerrar (close):** Después de realizar todas las operaciones deseadas, el archivo debe cerrarse para asegurar su integridad y para liberar recursos de su control en la memoria.
- **Leer o Escribir (read, write):** Añadir información al archivo o leer el carácter o una cadena de caracteres a partir de la posición actual. _ **Concatenar (append):** Es una forma restringida de la llamada 'write', en la cual sólo se permite añadir información al final del archivo. _ **Localizar (seek):** Para los archivos de acceso directo se permite posicionar el apuntador de lectura o escritura en un registro aleatorio, a veces a partir del inicio o final del archivo.
- **Leer atributos:** Permite obtener una estructura con todos los atributos del archivo especificado, tales como permisos de escritura, de borrado, ejecución, etc.
- **Poner atributos:** Permite cambiar los atributos de un archivo, por ejemplo en UNIX, donde todos los dispositivos se manejan como si fueran archivos, es posible cambiar el comportamiento de una terminal con una de estas llamadas.
- **Renombrar (rename):** Permite cambiarle el nombre e incluso a veces la posición en la organización de directorios del archivo especificado. Los subsistemas de archivos también proveen un conjunto de llamadas para operar sobre directorios, las más comunes son crear, borrar, abrir, cerrar, renombrar y leer. Sus funcionalidades son obvias, pero existen también otras dos operaciones no tan comunes que son la de 'crear una liga' y la de 'destruir la liga'. La operación de crear una liga sirve para que desde diferentes puntos de la organización de directorios se pueda acceder un mismo directorio sin necesidad de copiarlo o duplicarlo. La llamada a 'destruir la liga' lo que hace es eliminar esas referencias, siendo su efecto la de eliminar las ligas y no el directorio real. El directorio real es eliminado hasta que la llamada a 'destruir liga' se realiza sobre él.

Algunas facilidades extras de los sistemas de archivos

Algunos sistemas de archivos proveen herramientas al administrador del sistema para facilitarle la vida. Las más notables es la facilidad de compartir archivos y los sistemas de 'cotas'.

La facilidad de compartir archivos se refiere a la posibilidad de que los permisos de los archivos o directorios dejen que un grupo de usuarios puedan accederlos para diferentes operaciones" leer, escribir, borrar, crear, etc. El dueño verdadero es quien decide qué permisos se aplicarán al grupo e, incluso, a otros usuarios que no formen parte de su grupo. La facilidad de 'cotas' se refiere a que el sistema de archivos es capaz de llevar un control para que cada usuario pueda usar un máximo de espacio en disco duro. Cuando el usuario excede ese

límite, el sistema le envía un mensaje y le niega el permiso de seguir escribiendo, obligándolo a borrar algunos archivos si es que quiere almacenar otros o que crezcan. La versión de UNIX SunOS contiene esa facilidad.

Sistemas de Archivos Aislados

Los sistemas de archivos aislados son aquellos que residen en una sola computadora y no existe la posibilidad de que, aún estando en una red, otros sistemas puedan usar sus directorios y archivos. Por ejemplo, los archivos en discos duros en el sistema MS-DOS clásico se puede ver en esta categoría.

Sistemas de Archivos Compartidos o de Red

Estos sistemas de archivos es factible accederlos y usarlos desde otros nodos en una red. Generalmente existe un 'servidor' que es la computadora en donde reside el sistema de archivos físicamente, y por otro lado están los 'clientes', que se valen del servidor para ver sus archivos y directorios de manera como si estuvieran localmente en el cliente. Algunos autores les llaman a estos sistemas de archivos 'sistemas de archivos distribuidos' lo cual no se va a discutir en este trabajo.

Los sistemas de archivos compartidos en red más populares son los provistos por Netware, el Remote File Sharing (RFS en UNIX), Network File System (NFS de Sun Microsystems) y el Andrew File System (AFS). En general, lo que proveen los servidores es un medio de que los clientes, localmente, realicen peticiones de operaciones sobre archivos los cuales con 'atrapadas' por un 'driver' o un 'módulo' en el núcleo del sistema operativo, el cual se comunica con el servidor a través de la red y la operación se ejecuta en el servidor. Existen servidores de tipo "stateless y no-stateless". Un servidor "stateless" no registra el estado de las operaciones sobre los archivos, de manera que el cliente se encarga de todo ese trabajo. La ventaja de este esquema es que si el servidor falla, el cliente no perderá información ya que ésta se guarda en memoria localmente, de manera que cuando el servidor reanude su servicio el cliente proseguirá como si nada hubiese sucedido. Con un servidor "no-stateless", esto no es posible.

La protección sobre las operaciones se lleva a cabo tanto en los clientes como en el servidor: si el usuario quiere ejecutar una operación indebida sobre un archivo, recibirá un mensaje de error y posiblemente se envíe un registro al subsistema de 'seguridad' para informar al administrador del sistema de dicho intento de violación.

En la práctica, el conjunto de permisos que cada usuario tiene sobre el total de archivos se almacena en estructuras llamadas 'listas de acceso' (access lists).

Tendencias actuales

Con el gran auge de las redes de comunicaciones y su incremento en el ancho de banda, la proliferación de paquetes que ofrecen la compartición de archivos es común. Los esquemas más solicitados en la industria es el poder acceder los grandes volúmenes de información que residen en grandes servidores desde las computadoras personales y desde otros servidores también. Es una realidad que la solución más socorrida en las empresas pequeñas es usar Novell Netware en un servidor 486 o superior y acceder los archivos desde máquinas similares.

A veces se requieren soluciones más complejas con ambientes heterogéneos:

diferentes sistemas operativos y diferentes arquitecturas. Uno de los sistemas de archivos más expandidos en estaciones de trabajo es el NFS, y prácticamente todas las versiones de UNIX traen instalado un cliente y hasta un servidor de este servicio. Es posible así que una gran cantidad de computadoras personales (de 10 a 80) accedan grandes volúmenes de información o paquetería (desde 1 a 8 Gygabytes) desde una sola estación

de trabajo, e incluso tener la flexibilidad de usar al mismo tiempo servidores de Novell y NFS. Soluciones similares se dan con algunos otros paquetes comerciales, pero basta ya de `goles'. Lo importante aquí es observar que el mundo se va moviendo poco a poco hacia soluciones distribuídas, y hacia la estandarización que, muchas veces, es `de facto'.