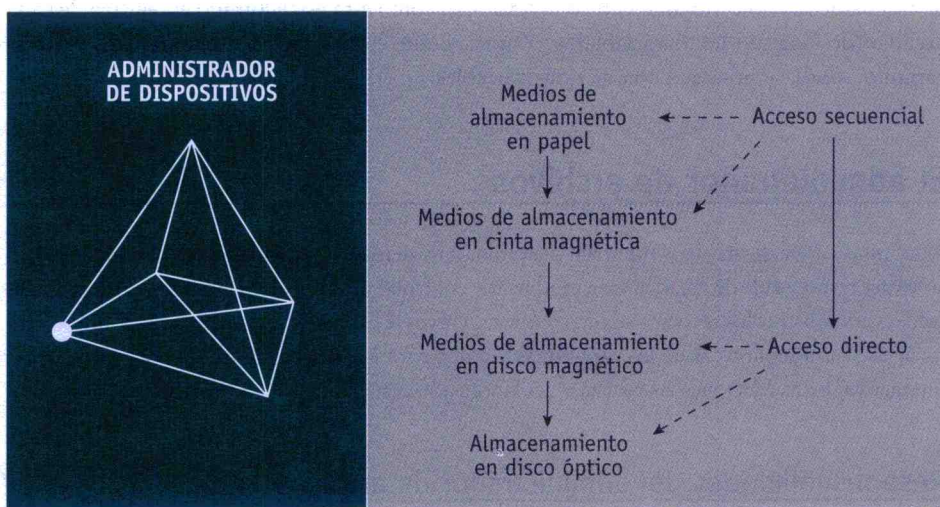


Capítulo 8 | Administración de archivos



“El conocimiento es de dos tipos. Sabemos sobre un tema o sabemos dónde encontrar información al respecto.”

Samuel Johnson (1709-1784)

Objetivos de aprendizaje

Después de completar este capítulo, el lector debe ser capaz de describir:

- Los fundamentos de la administración de archivos y la estructura del sistema de administración de archivos.
- Los convencionalismos para nombrar archivos, incluyendo el rol de las extensiones.
- La diferencia entre formato de registros de longitud fija y variable.
- Las ventajas y desventajas de las técnicas de almacenamiento contiguo, no contiguo e indexado de archivos.
- Comparaciones entre acceso secuencial y directo a archivos.
- Técnicas de control de acceso y cómo se comparan.
- El rol de la compresión de datos en el almacenamiento de archivos.

El administrador de archivos controla todos los archivos en el sistema. En este capítulo aprenderemos la forma en que los archivos están organizados lógicamente, cómo están almacenados físicamente, cómo se tiene acceso a ellos y a quién se permite tener acceso. También estudiaremos la interacción entre el administrador de archivos y el administrador de dispositivos.

La eficiencia del administrador de archivos depende de cómo están organizados los archivos del sistema (secuenciales, directos o indexados secuenciales); cómo están almacenados (contiguamente, no contiguamente, o indexados); cómo está estructurado el registro de cada archivo (de longitud fija o variable), y cómo se controla el acceso a estos archivos. En este capítulo abordaremos cada una de estas variables.

El administrador de archivos

El administrador de archivos (también denominado sistema de administración de archivos) es el software responsable de crear, borrar, modificar y controlar el acceso a los archivos, así como también de administrar los recursos usados por los archivos. El administrador de archivos proporciona apoyo para bibliotecas de los programas y datos para usuarios en línea, y para computación interactiva. Estas funciones las realiza en colaboración con el administrador de dispositivos.

Responsabilidades del administrador de archivos

El administrador de archivos tiene un trabajo complicado. Está a cargo de los componentes físicos del sistema, sus recursos de información y las políticas usadas para almacenar y distribuir los archivos. Para llevar a cabo sus responsabilidades, debe realizar estas cuatro funciones:

1. Seguir la pista del lugar en que está almacenado cada archivo.
2. Usar una política que determine dónde y cómo han de almacenarse los archivos, asegurándose de usar eficientemente el espacio de almacenamiento disponible y proporcionar acceso eficiente a los archivos.
3. Asignar cada archivo cuando a un usuario se le ha permitido el acceso al archivo y luego registrar su uso.
4. Liberar el archivo cuando debe devolverse a almacenamiento y comunicar su disponibilidad a otros que pueden estar esperándolo.

Por ejemplo, el sistema de archivos es como una biblioteca, donde el administrador de archivos desempeña el papel del bibliotecario que lleva a cabo las mismas cuatro tareas:

1. Un bibliotecario usa el catálogo para seguir la pista de cada artículo en la colección; cada entrada enumera el número de llamada y los detalles que ayudan a los lectores a encontrar el libro que buscan.
2. El bibliotecario depende de una política para almacenar todos los artículos en la colección, incluyendo los libros de gran tamaño, revistas, libros en cinta, DVD, mapas y videos. Y es necesario ordenarlos físicamente de modo que sea posible encontrarlos cuando se requieran.

3. Cuando se solicita, el libro se extrae de su librero y en los registros de préstamo se escribe el nombre del prestatario.
4. Cuando se devuelve el libro, el bibliotecario hace la anotación correspondiente en los registros de préstamo y vuelve a colocarlo en su sitio.

En un sistema de cómputo, el administrador de archivos sigue la pista de sus archivos con directorios que contienen el nombre del archivo, su ubicación física en almacenamiento secundario e información importante sobre cada archivo.

La política del administrador de archivos determina dónde se almacena cada archivo, cómo el sistema y sus usuarios; puede tener fácil acceso a los archivos: mediante comandos que son independientes de los detalles del dispositivo. Además, la política debe determinar quién tendrá acceso a qué material, lo cual implica dos factores: flexibilidad de acceso a la información y protección subsecuente. El administrador de archivos hace lo anterior al permitir acceso a archivos compartidos, proporcionando acceso distribuido y permitiendo que los usuarios naveguen a través de directorios públicos. Mientras tanto, el sistema operativo debe proteger sus archivos contra malfuncionamientos del sistema y proporcionar verificaciones de seguridad mediante números de cuenta, contraseñas y palabras de bloqueo para preservar la integridad de los datos y salvaguardarlos contra manipulaciones. Estas técnicas de protección se explicarán más tarde en este capítulo.

El sistema de cómputo *asigna* un archivo al activar el dispositivo idóneo de almacenamiento secundario y cargar el archivo en la memoria a la vez que actualiza sus registros de quién está usando qué archivo.

Finalmente, el administrador de archivos *desasigna* o libera un archivo al actualizar las tablas del archivo y volviendo a escribir el archivo (si se ha revisado) en el dispositivo de almacenamiento secundario. Cualquier proceso que esté esperando tener acceso a ese archivo es notificado entonces de su disponibilidad.

Definiciones

Antes de continuar, tomaremos un minuto para definir algunos elementos básicos de los archivos, ilustrados en la figura 8.1, que se relacionan con nuestro análisis del administrador de archivos.

(Figura 8.1)

Los archivos constan de registros. Los registros constan de campos.

Registro 19 →	Campo A	Campo B	Campo C	Campo D
Registro 20 →	Campo A	Campo B	Campo C	Campo D
Registro 21 →	Campo A	Campo B	Campo C	Campo D
Registro 22 →	Campo A	Campo B	Campo C	Campo D

Un **campo** es un grupo de bytes relacionados que el usuario puede identificar mediante un nombre, tipo y tamaño. Un **registro** es un grupo de campos relacionados.

Un **archivo** es un grupo de registros relacionados que contienen información a usar por programas de aplicación específicos para generar reportes. El tipo de archivo contiene datos y algunas veces se denomina archivo plano porque no tiene ninguna conexión con otros archivos; a diferencia de las bases de datos, carece de dimensionamiento.

Una **base de datos** aparece con el administrador de archivos como un tipo de archivo, pero las bases de datos son más complicadas, porque en realidad son grupos de archivos relacionados que están interconectados a varios niveles a fin de proporcionar a los usuarios flexibilidad de acceso a los datos almacenados. Si la base de datos del usuario requiere una estructura específica, el administrador de archivos debe ser capaz de proporcionarla.

Los **archivos del programa** contienen instrucciones y los archivos de datos contienen datos, pero en cuanto al almacenamiento concierne, el administrador de archivos los trata exactamente de la misma forma.

Los **directorios** son archivos especiales con listados de los nombres de los archivos y sus atributos. Los datos recolectados para supervisar el rendimiento del sistema y proporcionar su contabilidad, se reúnen en archivos. De hecho, todo programa y archivo de datos a los que tiene acceso el sistema de cómputo, así como toda pieza del software de la computadora, se tratan como un archivo.

Interacción con el administrador de archivos

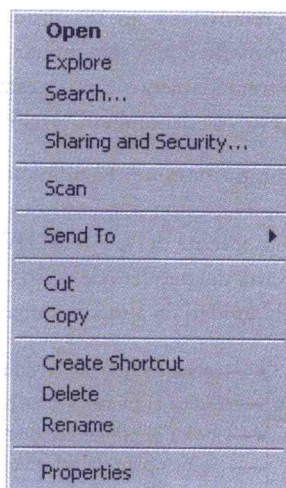
El usuario se comunica con el administrador de archivos, el cual responde a comandos específicos.

Algunos ejemplos de comandos interactivos, mostrados en la figura 8.2, son OPEN, DELETE, RENAME y COPY. En realidad, los archivos pueden crearse con otros términos específicos del sistema; por ejemplo, la primera vez que un usuario da un comando para guardar un archivo, en realidad se trata del comando CREATE. En otros sistemas operativos, el comando OPEN FILE dentro de un programa indica al administrador de archivos que es necesario crear un archivo.

Estos comandos y muchos más fueron diseñados para ser muy fáciles de usar, así que están desprovistos de las instrucciones detalladas para correr el dispositivo (información en el dispositivo) donde el archivo puede ser almacenado. Es decir, éstos son dispositivos independientes. Por tanto, para acceder a un archivo, el usuario no necesita conocer su ubicación física en el paquete de discos (el cilindro, superficie, y sector), el medio en el sean almacenadas (cinta de archivo, disco magnético, disco óptico, o memoria flash), o la red específicos. Eso está bien porque el acceso a archivos es un proceso complicado. Cada comando lógico se descompone en una secuencia de señales de bajo nivel que accionan las acciones paso a paso realizadas por el dispositivo y supervisan el avance de la operación al probar el estado

(Figura 8.2)

Menú típico de opciones de archivos.



del dispositivo. Por ejemplo, cuando el programa de un usuario emite un comando para leer el registro de un disco, la instrucción READ debe descomponerse en los pasos siguientes:

1. Mover las cabezas de lectura/escritura hacia el cilindro o pista donde debe estar el registro.
2. Esperar por la demora de rotación hasta que el sector que contiene el registro deseado pasa bajo la cabeza de lectura/escritura.
3. Activar la cabeza de lectura/escritura idónea y lee el registro.
4. Transferir el registro a la memoria principal.
5. Enviar una bandera para indicar que el dispositivo está libre para satisfacer otra solicitud.

Mientras se lleva a cabo todo lo anterior, el sistema debe verificar condiciones de posibles errores. El administrador de archivos efectúa todo esto, liberando a usuario de incluir en cada programa las instrucciones de nivel bajo para cada dispositivo a usar: la terminal, el teclado, la impresora, el CD, la unidad de disco, etc. Sin el administrador de archivos, cada programa requeriría incluir instrucciones para operar todos los diferentes tipos de dispositivos y todos los diferentes modelos dentro de cada tipo. Al considerar el rápido desarrollo y la sofisticación creciente de los dispositivos de E/S, sería impráctico, y ciertamente no muy amigable para el usuario, solicitar que cada programa incluya estos detalles operacionales mínimos. Esta es la ventaja de la independencia de los dispositivos.

Configuración de volumen típica

Normalmente, los archivos activos de una computadora residen en unidades de almacenamiento secundario. Algunos dispositivos acomodan unidades de almacenamiento removibles –como CD, DVD, disquetes, dispositivos USB y otros medios removibles– de modo que los archivos que no se usan a menudo pueden almacenarse fuera de línea y montarse sólo cuando el usuario los solicita específicamente. Otros dispositivos presentan unidades de almacenamiento integradas, como discos duros y paquetes de discos no removibles.



Los recursos de red están dados por lo general por una etiqueta volumen lógica para su fácil identificación, tal como z, x, w, etc.

Cada unidad de almacenamiento, removible o no, se considera un **volumen**, y cada volumen puede contener varios archivos, de modo que se denominan “volúmenes multiarchivos”. Sin embargo, algunos archivos son extremadamente largos y están contenidos en varios volúmenes; de manera no sorprendente, éstos se denominan “archivos multivolúmenes”.

Cada volumen en el sistema tiene un nombre. El administrador de archivos escribe este nombre y otra información descriptiva, como se muestra en la figura 8.3, en un sitio de fácil acceso en cada unidad; la parte más interna del CD o del DVD, el principio de una cinta, o el primer sector de la pista más externa del paquete de discos. Una vez identificado, el sistema operativo puede interactuar con la unidad de almacenamiento.

Fecha de creación	← Fecha en que fue creado el volumen
Apuntador hacia el área del directorio	← Indica el primer sector en que está almacenado el volumen
Apuntador hacia el área de archivos	← Indica el primer sector en que está almacenado el archivo
Código del sistema de archivos	← Se usa para detectar volúmenes con formatos incorrectos
Nombre del volumen	← Nombre asignado por el usuario

(Figura 8.3)

El descriptor del volumen, almacenado al principio de cada volumen, incluye el nombre del volumen y otra información vital sobre la unidad de almacenamiento.

El **directorio maestro de archivos (MFD)** se almacena inmediatamente después del descriptor de volumen y enumera los nombres y las características de todos los archivos contenidos en ese volumen. Los nombres de los archivos en el MFD pueden referirse a archivos del programa, archivos de datos y/o archivos del sistema. Y si el administrador de archivos atiende subdirectorios, también aparecen enumerados en el MFD. El resto del volumen se usa para almacenamiento de archivos.

Los primeros sistemas operativos atendían un solo directorio por volumen. Este directorio era creado por el administrador de archivos y contenía los nombres de los archivos, usualmente organizados en orden alfabético, espacial o cronológico. Aunque resultaba fácil de implementar y mantener, este esquema presentaba algunas desventajas importantes.

- Se requería mucho tiempo para buscar un archivo individual, específicamente si el MFD estaba organizado en orden arbitrario.
- Si el usuario tenía más de 256 pequeños archivos almacenados en el volumen, el espacio del directorio (con límite de 256 archivos), podía llenarse antes que el espacio de almacenamiento del disco. Entonces, el usuario recibía un mensaje de “disco lleno” cuando sólo estaba lleno el directorio.
- Los usuarios sólo podían crear subdirectorios para agrupar los archivos que estaban relacionados.
- No era posible que usuarios múltiples resguardaran sus propios archivos de otros usuarios, porque todo el directorio estaba libremente a disposición de cualquier usuario en el grupo solicitado.
- Cada programa en todo el directorio requería un nombre único, incluso los directorios que daban servicio a muchos usuarios, de modo que sólo una persona que estuviera usando ese directorio podía tener un programa denominado PROG1.

Esto podía producir estragos en un curso introductorio de informática. Por ejemplo, ¿qué ocurría si la primera persona denominaba PROG1 a la primera asignación de programación? Entonces, el resto de la clase podía tener opciones interesantes: escribir un nuevo

programa y nombrarlo de otra forma; escribir un programa nuevo y llamarlo PROG1 (con lo cual se borraría la versión original); o simplemente abrir PROG1, modificarlo y luego guardarlo con los cambios realizados. Con la última opción, toda la clase podía terminar con un programa único, tal vez terrorífico.

Acerca de los subdirectorios

Administradores de archivos más recientes crearon un MFD para cada volumen que contiene entradas para archivos y para subdirectorios. Un **subdirectorio** se crea cuando un usuario abre una cuenta para tener acceso al sistema de cómputo. Aunque este directorio del usuario se trata como un archivo, su entrada en el MFD está señalada para indicar al administrador de archivos que este archivo es en realidad un subdirectorio y que posee propiedades únicas: de hecho, sus registros son nombres de archivos que apuntan a archivos.

Aunque ésta era una mejora con respecto al esquema de directorio único (hoy todos los estudiantes pueden nombrar PROG1 a su primer programa), no resuelve los problemas que encuentran usuarios prolíficos que desean agrupar sus archivos en orden lógico para mejorar la accesibilidad y eficiencia del sistema.

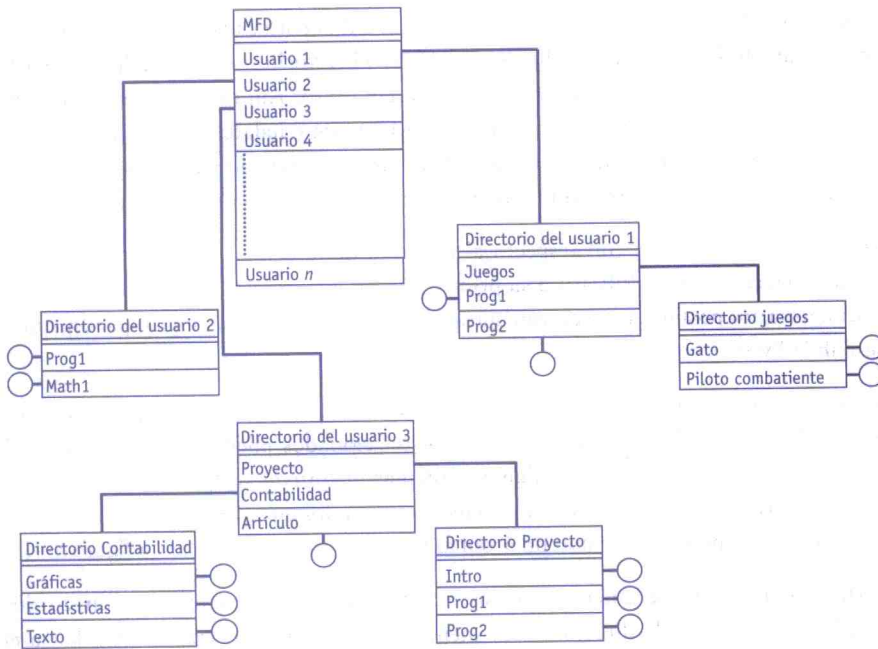
Los sofisticados administradores de archivos actuales permiten que el usuario elabore sus propios subdirectorios, de modo que los archivos relacionados pueden agruparse entre sí. Muchos usuarios de computadoras y algunos sistemas operativos denominan “carpetas” a estos subdirectorios. Esta estructura es una extensión de la organización previa de directorio de dos niveles, y se implementa como un árbol al revés, como se muestra en la figura 8.4.

Las estructuras de árbol permiten que el sistema busque de manera eficiente directorios individuales, ya que en cada directorio hay menos entradas. Sin embargo, la ruta hacia el archivo solicitado puede llevar a través de varios directorios. Para toda solicitud por algún archivo, el MFD es el punto de entrada. En realidad, el MFD suele ser transparente para el usuario; sólo es accesible para el sistema operativo. Cuando el usuario desea tener acceso a un archivo específico, el nombre del archivo se envía al administrador de archivos. Éste busca el directorio del usuario en el MFD y luego en el directorio del usuario y en todos los subdirectorios busca el archivo solicitado y su ubicación.

Sin importar la complejidad de la estructura del directorio, cada entrada de archivo en cualquier directorio contiene información que describe el archivo; se denomina **descriptor del archivo**. La información que suele incluirse en un descriptor del archivo incluye lo siguiente:

- Nombre del archivo: dentro de un directorio, los nombres de los archivos deben ser únicos. En algunos sistemas operativos, los nombres llevan mayúsculas y minúsculas.
- Tipo de archivo: la organización y el uso que son dependientes del sistema (por ejemplo, archivos y directorios).
- Tamaño del archivo: aunque es posible calcularlo a partir de otra información, el tamaño se guarda aquí por conveniencia.
- Ubicación del archivo: identificación del primer bloque físico (o de todos los bloques) donde está almacenado el archivo.

- Fecha y hora de creación.
- Propietario.
- Información de protección: restricciones de acceso, con base en a quién se permite tener acceso al archivo y el tipo de acceso permitido.
- Tamaño del archivo: su tamaño fijo o su tamaño máximo, dependiendo del tipo de registro.



(Figura 8.4)

Estructura de árbol del directorio de archivos. La "raíz" es el MFD que se muestra en la parte superior; cada nodo es un archivo del directorio y cada rama es una entrada del directorio que apunta a otro directorio o a un archivo real. Todos los programas y archivos de datos agregados más tarde al árbol son las hojas, representadas por círculos.

Convenciones de nomenclatura de archivos

El nombre de un archivo puede ser más largo de lo que cree el usuario. Dependiendo del nivel de sofisticación del administrador de archivos, el nombre puede tener desde dos hasta muchas componentes. Las dos componentes comunes a muchos nombres de archivos son el **nombre relativo del archivo** y una **extensión**.

Para evitar confusiones, en el siguiente análisis usaremos los términos "nombre completo del archivo" para identificar el **nombre absoluto del archivo** (es el nombre largo que incluye toda la información de ruta), y "nombre relativo" para indicar el nombre sin la información de ruta que aparece en listados de directorios y carpetas.

El nombre relativo del archivo es el nombre que lo diferencia de otros archivos en el mismo directorio. Algunos ejemplos pueden incluir DEPARTMENT ADDRESSES, TAXES_PHOTO, o AUTOEXEC. En general, el nombre relativo puede variar de longitud desde uno hasta muchos caracteres y puede incluir letras, así como dígitos. Sin embargo, todo sistema operativo posee reglas específicas que afectan la longitud del nombre relativo y los tipos de caracteres permitidos. La mayor parte de los sistemas operativos permiten nombres con

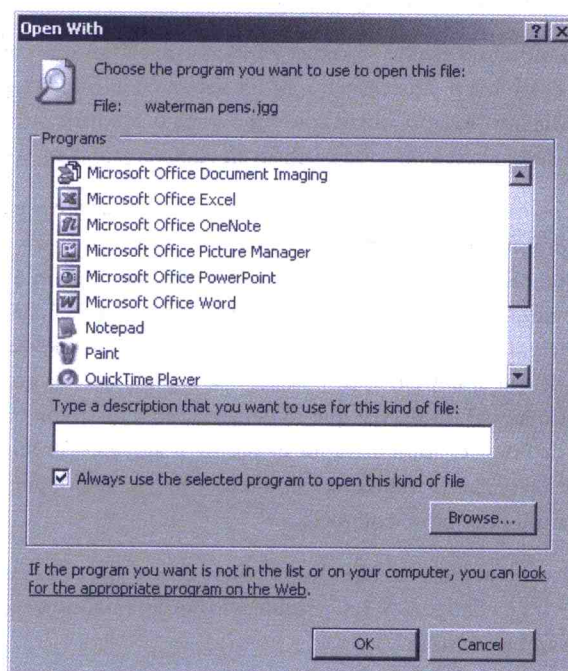
✓ El límite de caracteres para el nombre de un archivo (256 para el caso de algunos sistemas operativos de Windows) puede aplicarse a la ruta entera y no sólo al nombre del archivo relativo.

docenas de caracteres incluyendo espacios, guiones, subrayados y algunos caracteres del teclado.

Algunos sistemas operativos requieren una extensión que se agrega al nombre relativo del archivo. Suele ser de dos o tres caracteres de largo y está separado del nombre del archivo por un punto, y su propósito es identificar el tipo de archivo o su contenido. Por ejemplo, en un sistema operativo Windows, un nombre relativo de archivo típico con extensión sería BASIA_TUNE.MP3. En forma semejante, TAKE OUT MENU.RTF y TAKE OUT MENU.DOC indican que pueden ser abiertos con una aplicación de procesador de palabras. ¿Qué ocurre si una extensión es incorrecta o desconocida? La mayor parte solicita orientación de parte del usuario, como se muestra en la figura 8.5.

(Figura 8.5)

Para abrir un archivo con una extensión no reconocida, Windows pide al usuario que escoja una aplicación asociada con ese tipo de archivo.



Algunas extensiones (como EXE, BAT, COB y FOR) están restringidas por algunos sistemas operativos, ya que constituyen un conjunto estándar. Es decir, funcionan como una señal para que el sistema use un compilador o programa especial para ejecutar estos archivos.

Dependiendo del sistema operativo, puede haber otros componentes requeridos para el nombre completo de un archivo. He aquí cómo tres sistemas operativos diferentes identifican un archivo denominado INVENTORY_COST.DOC:

1. Usando un sistema operativo Windows y una computadora personal con tres unidades de disco, el nombre completo del archivo está compuesto por su nombre relativo y una extensión, precedida por la etiqueta de la unidad de disco y el nombre del directorio:

C:\IMFST\FLYNN\INVENTORY_COST.DOC

Esto indica al sistema que el archivo INVENTORY_COST.DOC requiere un programa de aplicación de un procesador de palabras y que puede encontrarse en el directorio; IMFST; subdirectorío FLYNN en el volumen que reside en la unidad C.

2. Un sistema UNIX o Linux podría identificar este archivo como:

```
/usr/imfst/flynn/inventory_cost.doc
```

La primera entrada está representada por la diagonal hacia delante (/), que representa un directorio maestro especial denominado *raíz*. Luego está el nombre del primer subdirectorío, *usr/imfst*, seguido por un sub-subdirectorío, */Flynn*, en este sistema de múltiples directoríos. La entrada final es el nombre relativo del archivo, *inventory_cost.doc*. (Observe que en UNIX y en Linux los nombres de archivo son sensibles y a menudo se escriben con minúsculas.)

Como puede ver, los nombres tienden a crecer en longitud, en tanto los administradores de archivos crecen en flexibilidad. Las carpetas en un sistema con interfaz gráfica de usuario, como Windows o Macintosh, en realidad son directoríos o subdirectoríos. Cuando alguien crea una carpeta, el sistema crea un subdirectorío en el directorío o en la carpeta actuales.

¿Por qué los usuarios no pueden ver el nombre completo del archivo cuando ingresan en éste? Primero, el administrador de archivos escoge un directorío para el usuario cuando empieza la sesión interactiva, de modo que todas las operaciones con archivos solicitadas por ese usuario empiezan desde este directorío “principal” o “base”. Segundo, a partir de este directorío principal, el usuario escoge un subdirectorío, que se denomina **directorío actual** o **directorío de trabajo**. A partir de entonces, se supone que los archivos están ubicados en este directorío actual. Cada vez que se tiene acceso a un archivo, el usuario escribe el nombre relativo de éste y el administrador de archivos agrega el prefijo idóneo. En tanto los usuarios hagan referencia a archivos en el directorío de trabajo, pueden tener acceso a sus archivos sin escribir el nombre completo.

El concepto de directorío actual está basado en la jerarquía subyacente de una estructura de árbol, como se muestra en la figura 8.4, y permite a los programadores recuperar un archivo al escribir sólo su nombre relativo. . .


```
INVENTORY_COST.DOC
```

. . . y no su nombre absoluto:

```
C:\IMFST\FLYNN\INVENTORY_COST.DOC
```

Organización de archivos

Cuando se analiza la organización de archivos se está hablando del arreglo de los registros en un archivo, ya que todos los archivos están compuestos por registros. Cuando un usuario emite un comando para modificar el contenido de un archivo, en realidad se trata de un comando para tener acceso a registros dentro del archivo.



Los sistemas operativos separan los elementos de los archivos con delimitadores. Por ejemplo, algunos usan una diagonal hacia delante (/) y otros, una hacia atrás (\).

Formato de los registros

Todos los archivos están compuestos por registros. Cuando un usuario emite un comando para modificar el contenido de un archivo, en realidad se trata de un comando para tener acceso a registros dentro del archivo. Dentro de cada archivo, se supone que todos los registros tienen el mismo formato; pueden ser de longitud fija o longitud variable, como se muestra en la figura 8.6. Y estos registros, sin importar su formato, pueden estar bloqueados o no.

(Figura 8.6)

Cuando los datos se almacenan en campos de longitud fija (a), los datos que rebasan el tamaño fijo son truncados. Cuando los datos se almacenan en un registro con formato de longitud variable (b), el tamaño se agranda a fin de dar cabida al contenido, aunque para tener acceso al archivo se requiere más tiempo.

(a)	Dan	Whitesto	1243 Ele	Harrisbu	PA	412 683-
(b)	Dan	Whitestone	1243 Elementary Ave.		Harrisburg	PA

Los **registros de longitud fija** son los más comunes, ya que son a los que es más fácil acceder directamente. Es por esto que son ideales para archivos de datos. La parte crítica de los registros de longitud fija es el tamaño del registro. Si es demasiado pequeño –menor que el número de caracteres a almacenar en el registro– los caracteres que no caben son truncados. Pero si el tamaño del registro es demasiado grande –más grande que el número de caracteres a almacenar– se desperdicia espacio de almacenamiento.

Los **registros de longitud variable** no dejan espacio de almacenamiento vacío y no truncan ningún carácter, eliminando así las dos desventajas de los registros de longitud fija. Pero aunque es fácil leerlos (uno después de otro), es difícil tener acceso directo a ellos, ya que es difícil calcular exactamente el sitio en que se encuentra ubicado el registro. Es por ello que se usan principalmente en archivos que probablemente sean accedidos secuencialmente, como archivos de texto y archivos de programa o archivos que usan un índice para tener acceso a sus registros. Al formato del registro, la forma en que está bloqueado y otra información relacionada se mantienen en el descriptor del archivo.

La cantidad de espacio que realmente se usa para almacenar la información complementaria suele variar de un sistema a otro y se ajusta a las limitaciones físicas del medio de almacenamiento, como veremos más tarde en este capítulo.

Organización física de los archivos

La organización física de un archivo tiene que ver con la forma en que están dispuestos los registros y con las características del medio que se usa para almacenarlos.

En discos magnéticos (discos duros), los archivos pueden organizarse en una de tres formas: secuencial, directa o indexada secuencial. Para escoger la mejor de estas organizaciones de archivos, el programador o analista suele considerar las siguientes características prácticas:

- Volatibilidad de los datos: la frecuencia con que se hacen adiciones y borrados.
- Actividad del archivo: porcentaje de registros procesados durante una ejecución dada.
- Tamaño del archivo.
- Tiempo de respuesta: cantidad de tiempo que el usuario está dispuesto a esperar antes que se complete la operación solicitada. (Esto es especialmente importante cuando se realizan investigaciones sensibles al tiempo.)

La **organización secuencial de los registros** es con mucho la más fácil de implementar, porque los registros se almacenan y recuperan en serie, uno después de otro. Para encontrar un registro específico, el archivo se busca desde su inicio hasta que se encuentra el registro solicitado.

Para acelerar el proceso es posible integrar algunas características al sistema. Una consiste en escoger del registro un campo clave y luego ordenar los registros por ese campo antes de almacenarlos. Luego, cuando un usuario solicite un registro específico, el sistema sólo busca en el campo clave de cada registro en el archivo. Esta búsqueda termina cuando se encuentra una coincidencia exacta o el campo clave del registro solicitado es menor que el valor del último registro comparado, en cuyo caso se envía el mensaje “registro no encontrado” al usuario y termina la búsqueda.

Aunque esta técnica ayuda al proceso de búsqueda, complica el mantenimiento de los archivos, ya que el orden original debe preservarse siempre que se agreguen o borren registros. Y para preservar el orden físico, el archivo debe volver a escribirse por completo o mantenerse ordenado dinámicamente cada vez que se actualiza.

Una **organización de registros directa** usa archivos de acceso directo, lo que, por supuesto, sólo puede implementarse en dispositivos de almacenamiento con acceso directo (analizados en el capítulo 7). Estos archivos proporcionan al usuario la flexibilidad de tener acceso a cualquier registro en cualquier orden sin tener que comenzar a buscar desde el inicio del archivo para hacerlo. También se denomina “organización aleatoria” y los archivos se denominan “archivos de acceso aleatorio”.

Los registros se identifican por sus **direcciones relativas**: sus direcciones con respecto al principio del archivo. Estas **direcciones lógicas** se calculan cuando el registro se ha almacenado y luego otra vez cuando se recuperan los registros.

El método es bastante directo. El usuario identifica un campo (o una combinación de campos) en el formato del registro y lo designa como su **campo clave**, porque identifica de manera única cada registro. El programa usado para almacenar los datos sigue un conjunto de instrucciones, denominado **algoritmo hashing**, que transforma cada clave en un número: la dirección lógica del registro. Esto se proporciona al administrador de archivos, que lleva a cabo los pasos necesarios para transformar la dirección lógica en una dirección física (cilindro, superficie y números de registro) preservando la organización del archivo. Para recuperar un registro se usa el mismo procedimiento.



Las bases de datos usan una tabla hash para acelerar el acceso a los registros de datos. Al usar un algoritmo hash, cada registro se identifica de manera única, y la tabla hash contiene los apuntadores hacia cada registro.

Puede accederse de manera secuencial a un archivo de acceso directo, empezando en la primera dirección relativa y yendo a cada registro bajo la línea.

Los archivos de acceso directo pueden actualizarse más fácilmente que los ficheros secuenciales, porque los registros pueden ser reescritos rápidamente a sus direcciones originales después que se han introducido modificaciones. Y no hay necesidad de preservar el orden de los registros, para añadir o borrar éstos toma muy poco tiempo.

Por ejemplo, los datos para una empresa que recibe órdenes por correo telefónico deben ser ingresados rápidamente, así que la información de sus clientes debe ser recuperada con la misma rapidez. Suponga que usted está colocando una orden y que le piden el número de su código postal y el de su calle (por ejemplo, 15231 y 2737). El programa que recupera información del archivo de datos usa esta clave en un algoritmo hash para calcular la dirección lógica del sitio en que está almacenado su registro. Así, cuando el empleado que tomó la orden escribe 152132737, la pantalla muestra pronto una lista de clientes concurrentes cuyos números de cliente generaron la misma dirección lógica. Si usted está en la base de datos, el operador lo sabe al instante. En caso contrario, pronto lo estará.

El problema con los algoritmos hash es que varios registros con claves únicas (como los números de cliente) pueden generar la misma dirección lógica: entonces hay una colisión, como se muestra en la figura 8.7. Cuando esto ocurre, el programa debe generar otra dirección lógica antes de presentarla al administrador de archivos para su almacenamiento. Los registros que chocan se almacenan en un área de desbordamiento que se hizo a un lado cuando se creó el archivo. Aunque el programa hace todo el trabajo de vincular los registros desde el área de desbordamiento con su dirección lógica correspondiente, el administrador de archivos debe manejar la asignación física del espacio.

(Figura 8.7)

El algoritmo hashing provoca una colisión cuando el algoritmo, usando una combinación de dirección de calle y código postal, genera la misma dirección lógica (152132737) para tres registros diferentes.



El tamaño máximo del archivo se establece, cuando es creado, y finalmente cualquiera de los archivos se llena por completo o el número de registros almacenados en el área de desbordamiento puede hacerse tan grande que se pierde la eficiencia de recuperación. En este caso el archivo debe reorganizarse y volver a escribirse, lo cual requiere de la intervención del administrador de archivos.

La **organización secuencial indexada de registros** combina lo mejor del acceso directo y secuencial. Es creado y mantenido mediante un paquete de software denominado método de acceso secuencial indexado (ISAM, por sus siglas en inglés), que elimina la carga de manejar desbordamientos y preserva el orden de los registros en los hombros del programador.

Este tipo de organización no provoca colisiones, ya que no usa el resultado del algoritmo hash para generar la dirección de un registro. En lugar de eso, usa esta información para generar un archivo índice por medio del cual se recuperan los registros. Esta organización divide un archivo en orden secuencial en bloques del mismo tamaño. Su tamaño está determinado por el administrador de archivos, a fin de aprovechar los dispositivos de almacenamiento físico y optimizar las estrategias de recuperación. Cada entrada en el archivo índice contiene la clave del registro más alto y la ubicación física del bloque de datos donde están almacenados este registro y los registros con claves menores.

En consecuencia, para tener acceso a cualquier registro en el archivo, el sistema empieza por buscar en el archivo índice y luego se dirige a la ubicación física indicada en esa entrada. Puede decirse, entonces, que el archivo índice actúa como un apuntador hacia el archivo de datos. Un archivo secuencial indexado también tiene áreas de desbordamiento, aunque están dispersas en todo el archivo, tal vez muy pocos registros, de modo que puede ocurrir expansión de registros existentes, y nuevos registros pueden ubicarse en secuencia física próxima, así como en secuencia lógica. Otra área de desbordamiento está ubicada lejos del área de los datos principales, pero sólo se usa cuando las otras áreas de desbordamiento están completamente llenas. Este desbordamiento se denomina el último recurso.

Esta área de desbordamiento del último recurso puede almacenar registros que se han agregado durante el tiempo de duración del archivo. El paquete de software mantiene los registros en orden lógico sin mucho esfuerzo de parte del programador. Por supuesto, cuando aquí se han agregado demasiados registros, el proceso de recuperación se vuelve lento, ya que la búsqueda por un registro debe ir desde el índice hasta el área de datos principales y terminar en el área de desbordamiento.

Cuando el tiempo de recuperación se hace demasiado lento, es necesario reorganizar el archivo. Se trata de un trabajo que, aunque no es tan tedioso como reorganizar archivos de acceso directo, suele ser realizado por un programa de mantenimiento.

Para la mayor parte de los archivos dinámicos, el acceso secuencial indexado es la organización a elegir, ya que permite tanto acceso directo a unos cuantos registros solicitados como acceso secuencial a muchos.

Asignación de almacenamiento físico

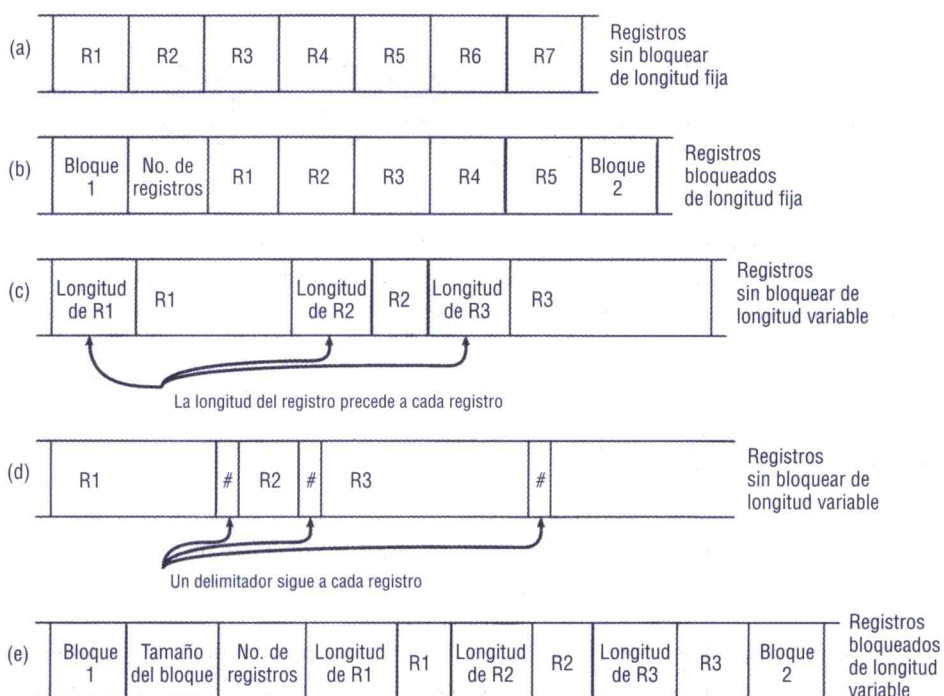
El administrador de archivos debe trabajar con archivos no sólo como unidades completas, sino también como unidades o registros lógicos. Los registros dentro de un archivo deben tener el mismo formato, aunque pueden variar en longitud, como se muestra en la figura 8.8.

A su vez, los registros se subdividen en campos. En la mayor parte de los casos, su estructura es administrada por programas de aplicación y no por el sistema operativo. Una excepción es la de los sistemas que están fuertemente orientados a aplicaciones de bases de datos, donde el administrador de archivos maneja la estructura de campo.

Así, cuando hablamos sobre almacenamiento de archivos, en realidad nos referimos a almacenamiento de registros. ¿Cómo se almacenan los registros dentro de un archivo? En este momento, el administrador de archivos y el administrador de dispositivos deben cooperar para asegurar el almacenamiento y la recuperación exitosas de registros. En el capítulo 7 sobre administración de dispositivos, presentamos el concepto de registros lógicos contra registros físicos, y este tema recurre aquí desde el punto de vista del administrador de archivos.

(Figura 8.8)

Todos los registros en un archivo deben tener el mismo formato, aunque sus tamaños pueden ser diferentes, como se muestra en estos cinco ejemplos de los formatos más comunes de registros. La información complementaria en (b, c, d y e) es proporcionada por el administrador de archivos, cuando se almacena el registro.



Almacenamiento contiguo

Cuando los registros usan **almacenamiento contiguo**, se almacenan unos después de otro. Este era el esquema usado en los primeros sistemas operativos. Es muy fácil de implementar y administrar. Cualquier registro puede encontrarse y leerse, una vez que se conocen su di-

rección inicial y tamaño, de modo que el directorio es muy alineado. Su segunda ventaja es la facilidad de acceso directo, porque cada una de las partes del archivo se almacena en la misma área compacta.

La desventaja más importante es que no es posible agrandar el archivo a menos que haya espacio vacío inmediatamente a continuación, como se muestra en la figura 8.9. En consecuencia, el espacio para la expansión debe tenerse en cuenta cuando se crea el archivo. En caso de que no haya suficiente espacio, todo el archivo debe volver a copiarse en una sección más grande del disco cada que se agregan registros. La segunda desventaja es la fragmentación (huecos de espacio no usado), lo cual puede resolverse al compactar y reordenar los archivos. Y, por supuesto, mientras se lleva a cabo la compactación no es posible tener acceso a los archivos.

(Figura 8.9)

Con un almacenamiento contiguo de archivos, el archivo A no puede agrandarse sin que vuelva a escribirse en un área de almacenamiento más grande. El archivo B puede agrandarse, pero sólo por un registro que sustituya el espacio libre que precede al archivo C.

Espacio libre	Archivo A del Registro 1	Archivo A del Registro 2	Archivo A del Registro 3	Archivo A del Registro 4	Archivo A del Registro 5	Archivo B del Registro 1	Archivo B del Registro 2	Archivo B del Registro 3	Archivo B del Registro 4	Espacio libre	Archivo C del Registro 1
---------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	---------------	--------------------------

El administrador de archivos sigue la pista de las áreas de almacenamiento vacías al tratarlas como archivos: se introducen en el directorio pero están indicadas para diferenciarlas de los archivos verdaderos. Normalmente, el directorio se mantiene en orden por número de sector, de modo que las áreas adyacentes vacías pueden combinarse en un espacio libre más grande.

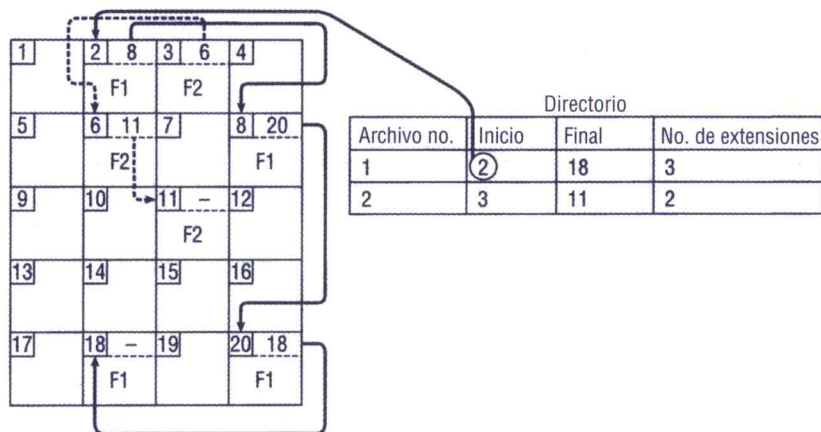
Almacenamiento no contiguo

La asignación de **almacenamiento no contiguo** permite que los archivos usen cualquier espacio de almacenamiento disponible en el disco. Los registros de un archivo se almacenan de manera contigua, sólo si hay suficiente espacio vacío. Cualquier registro restante y todas las demás adiciones al archivo se almacenan en otras secciones del disco. En algunos sistemas éstas se denominan **extensiones** del archivo y están vinculadas entre sí mediante apuntadores. El tamaño físico de cada extensión es determinada por el sistema operativo y suele ser 256; o alguna potencia de dos, bytes.

Las extensiones de los archivos suelen estar vinculadas de dos formas. La vinculación puede llevarse a cabo a nivel del almacenamiento, donde cada punto de la extensión apunta al siguiente en la secuencia, como se muestra en la figura 8.10. La entrada del directorio consta del nombre del archivo, la ubicación de almacenamiento de la primera extensión, la ubicación de la última extensión y el número total de extensiones sin contar la primera.

(Figura 8.10)

Almacenamiento no contiguo de archivos donde la vinculación se lleva a cabo a nivel del almacenamiento. El archivo 1 empieza en la dirección 2 y continúa en las direcciones 8, 20 y 18. El directorio enlista la dirección inicial del archivo, su dirección final y el número de extensiones que usa. Cada bloque de almacenamiento incluye su dirección y un apuntador hacia el siguiente bloque para el archivo, así como los datos en sí.



La alternativa es que la vinculación se lleve a cabo a nivel del directorio, como se muestra en la figura 8.11. Cada extensión se enumera con su dirección física, su tamaño y un apuntador hacia la siguiente extensión. Un apuntador nulo, mostrado aquí como un guión (-), indica que es el último.

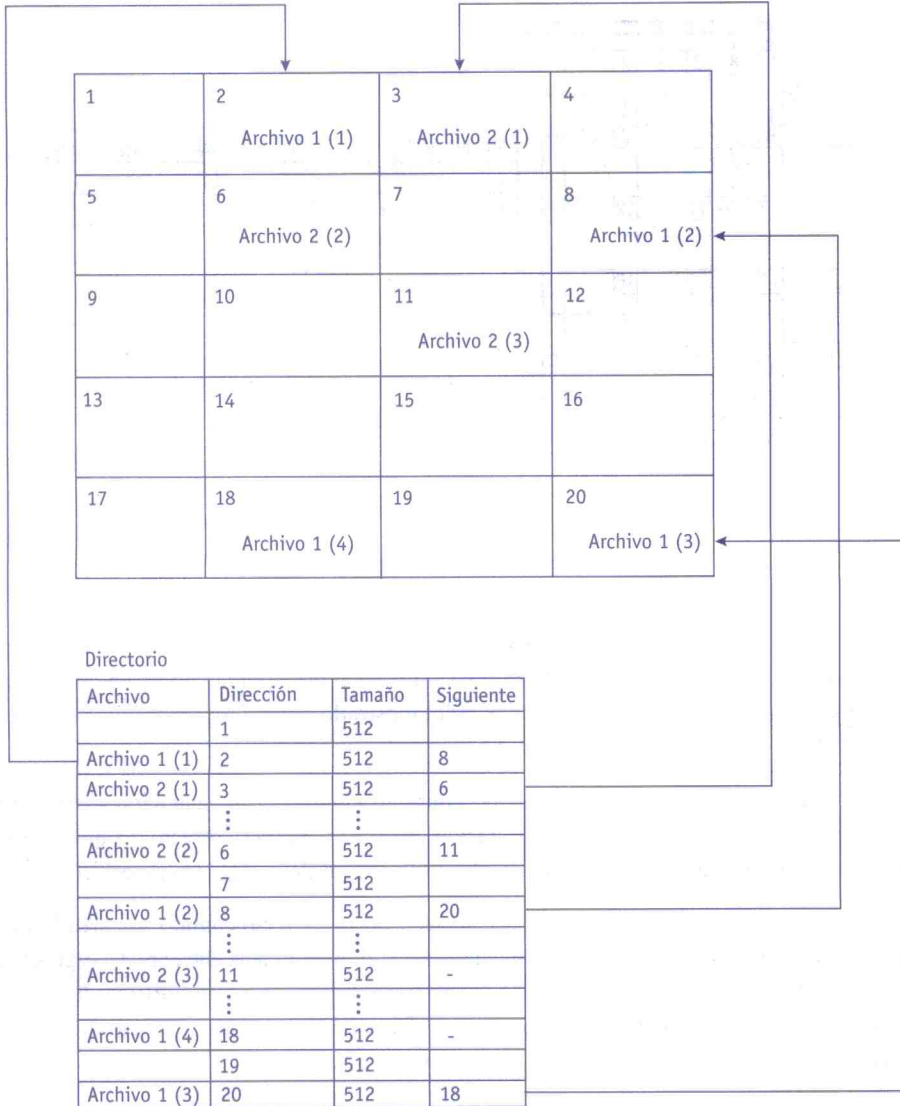
Aunque ambos esquemas de asignación no contiguo eliminan fragmentación externa de almacenamiento y la necesidad de compactación, no atienden acceso directo, ya que no hay ninguna forma sencilla para determinar la ubicación exacta de un registro específico.

Los archivos suelen declararse como de acceso secuencial o directo cuando son creados, de modo que el administrador de archivos puede escoger el método más eficiente de asignación de almacenamiento; contiguo para archivos con acceso directo y no contiguo para los archivos de acceso secuencial. Los sistemas operativos deben contar con la capacidad de atender ambos esquemas de asignación de almacenamiento.

Luego, los archivos pueden convertirse de un tipo a otro al crear un archivo del tipo deseado y copiando el contenido del archivo anterior en el archivo nuevo, usando un programa diseñado para este propósito especial.

Almacenamiento indexado

La asignación de almacenamiento indexado permite acceso directo a los registros, al reunir los apuntadores que vinculan todas las extensiones de ese archivo en un bloque de índice. Cada archivo tiene su propio bloque de índice, que consta de las direcciones de cada sector del disco que integran el archivo. El índice enumera cada entrada en el mismo orden en que



(Figura 8.11)

Asignación de almacenamiento no contiguo de archivos donde la vinculación se lleva a cabo a nivel del directorio para los archivos que se muestran en la figura 8.10.

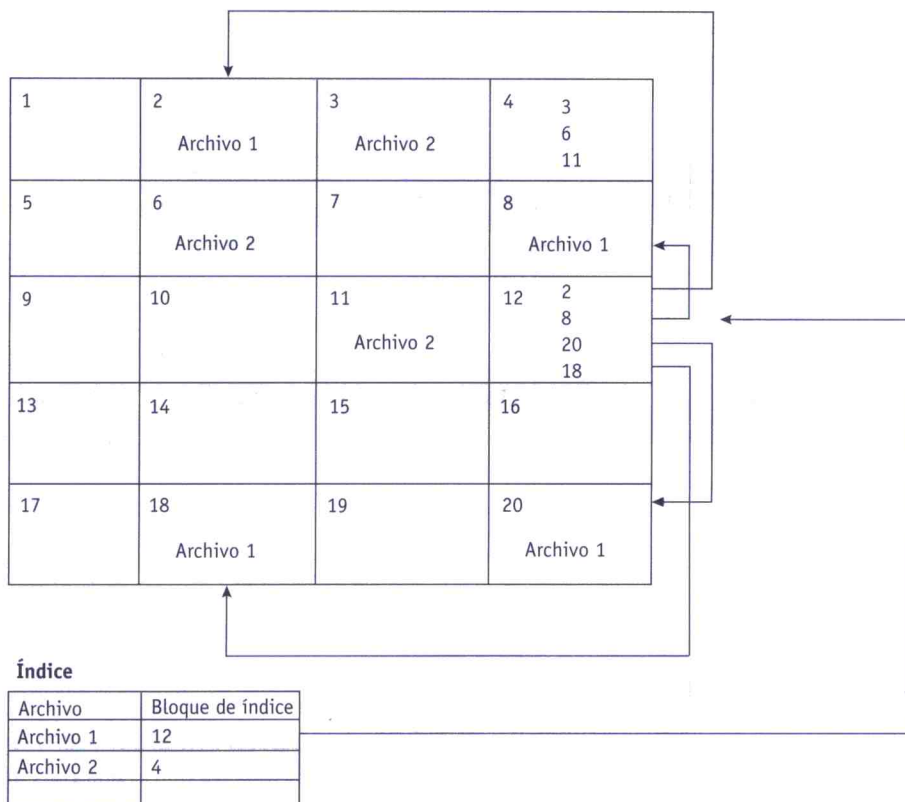
están vinculados los sectores, como se muestra en la figura 8.12. Por ejemplo, la tercera entrada en el bloque de índice corresponde al tercer sector que integra la lista.

Cuando se crea un archivo, todos los apuntadores en bloque de índice se habilitan en nulo. Luego, a medida que se llena cada sector, el apuntador se habilita en la dirección idónea del sector: para ser precisos, la dirección se elimina de la lista del espacio vacío y se copia en su posición en el bloque de índice.

Este esquema atiende acceso secuencial y directo, aunque no necesariamente mejora el uso del espacio de almacenamiento, ya que cada archivo debe tener un bloque de índice: en general del tamaño de un sector del disco. Para archivos más grandes con más entradas, es

(Figura 8.12)

Asignación de almacenamiento indexado con un índice de un nivel, permitiendo acceso directo a cada registro para los archivos que se muestran en las figuras 8.10 y 8.11.



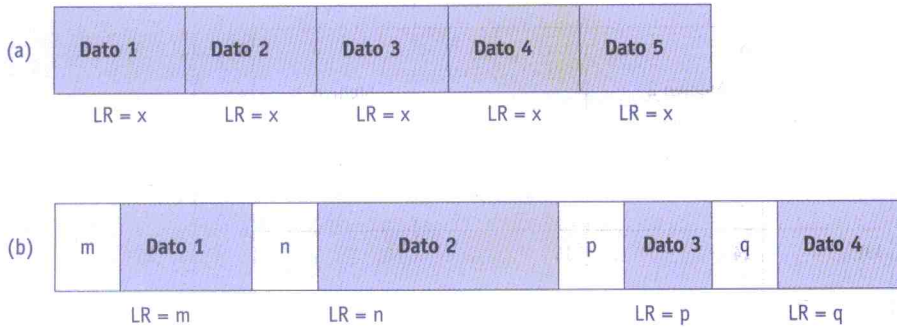
posible generar varios niveles de índices, en cuyo caso, para encontrar un registro deseado el administrador de archivos tiene acceso al primer índice (el nivel más alto), que apunta hacia un segundo índice (nivel más bajo), que apunta hacia un índice de nivel aún más bajo y finalmente hacia el registro de datos.

Métodos de acceso

Los métodos de acceso los determina una organización de archivos; se permite máxima flexibilidad con archivos secuenciales indexados y mínima, con secuenciales.

Un archivo que ha sido organizado de manera secuencial puede atender sólo acceso a sus registros, y estos registros pueden ser de longitud fija o variable, como se muestra en la figura 8.6. El administrador de archivos usa la dirección del último byte leído para tener acceso al siguiente archivo secuencial. En consecuencia, la **dirección del byte actual (CBA)** debe actualizarse cada vez que se tiene acceso a un registro, como cuando se ejecuta un comando READ.

La figura 8.13 muestra la diferencia entre el almacenamiento de registros de longitud fija y registros de longitud variable.



(Figura 8.13)

Archivos de longitud fija contra registros de longitud variable. a) Los registros de longitud fija tienen el mismo número de bytes, de modo que la longitud del registro (LR) es la constante x . b) Con registros de longitud variable, LR_k no es una constante. En consecuencia, se graba en los medios secuenciales, como cinta magnética, precediendo inmediatamente cada registro.

Acceso secuencial

Para *acceso secuencial de registros de longitud fija*, la dirección del byte actual (CBA) se actualiza simplemente al incrementarla por la longitud del registro (LR), que es una constante:

$$CBA = CBA + LR$$

Para *acceso secuencial de registros de longitud variable*, el administrador de archivos suma la longitud del registro (LR) más el número de bytes usados para contener la longitud del registro (N, el cual se mantiene constante, representado como m, n, p o q, en la figura 8.13).

$$CBA = CBA + N + LR_k$$

Acceso directo

Si un archivo está organizado de manera directa, es posible acceder fácilmente a él en orden directo o secuencial si los registros son de longitud fija. En el caso de *acceso directo con registros de longitud variable*, la dirección del byte actual (CBA) puede calcularse directamente a partir de la longitud del registro y del número de registro deseado NR (información proporcionada mediante el comando READ) menos uno:

$$CBA = (NR - 1) * LR$$

Por ejemplo, si se está buscando el comienzo del undécimo registro y la longitud fija de éste es de 25 bytes, la CBA sería:

$$(11 - 1) * 25 = 250$$

Sin embargo, si el archivo está organizado para *acceso directo con registros de longitud variable*, es virtualmente imposible tener acceso directo a un registro, ya que la dirección del registro deseado no puede calcularse fácilmente. En consecuencia, para tener acceso a un archivo, el

✓
Al principio, el acceso directo se denominaba "acceso aleatorio", ya que permitía acceso a los datos almacenados en orden aleatorio. No requiere almacenamiento secuencial.

El administrador de archivos debe efectuar una búsqueda secuencial en los registros. De hecho, se vuelve una lectura semisecucional a través del archivo, porque el administrador de archivos puede guardar la dirección del último registro al que se tuvo acceso, y cuando llega la siguiente solicitud, puede buscar hacia delante a partir de la CBA: si la dirección del registro deseado estaba entre la CBA y el final del archivo. En caso contrario, la búsqueda podría empezar desde el principio del archivo. Podría decirse que esta búsqueda semisecucional es sólo semiadecuada.

Una alternativa es que el administrador de archivos mantenga una tabla de números de registro y sus CBA. Después, para atender una solicitud, en esta tabla se busca la ubicación exacta de almacenamiento del registro deseado, de modo que el acceso directo se reduce a buscar en la tabla.

Para evitar tener que tratar con este problema, muchos sistemas obligan a los usuarios a tener organizados sus archivos para registros de longitud fija, si el acceso a los registros ha de ser directo.

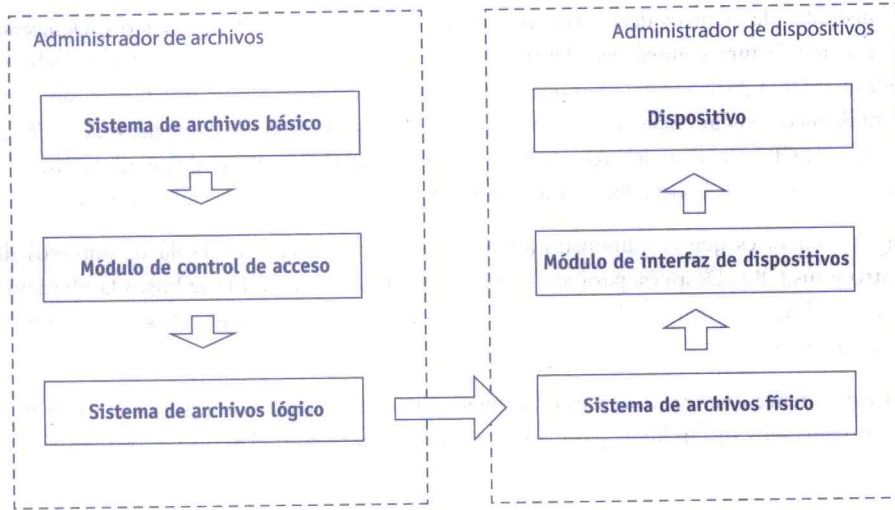
El acceso a los registros en un *archivo secuencial indexado* puede ser secuencial o directo, de modo que cualquiera de los procedimientos para calcular la CBA presentados en esta sección es válido, pero con un paso adicional: es necesario buscar en el archivo índice el apuntador hacia el bloque en que están almacenados los datos. Debido a que el archivo índice es más pequeño que el archivo de datos, puede mantenerse en la memoria principal y es posible realizar una búsqueda rápida para localizar el bloque donde se encuentra el registro deseado. Luego, el bloque puede recuperarse del almacenamiento secundario, y es posible calcular el byte inicial de la dirección del registro. En sistemas que atienden varios niveles de indexación para mejorar el acceso a archivos muy grandes, el índice en cada nivel debe buscarse antes que pueda efectuarse el cálculo de la dirección del byte actual. El punto de entrada a este tipo de datos suele ser mediante un archivo índice.

Como hemos visto, la organización de un archivo y los métodos usados para tener acceso a sus registros están estrechamente entrelazados, de modo que cuando se habla sobre un tipo de organización específica, ciertamente casi siempre se implica un tipo de acceso específico.

Niveles en un sistema de administración de archivos

La administración eficiente de archivos no puede separarse de la administración eficiente de los dispositivos que los alojan. Este capítulo y el capítulo previo sobre administración de dispositivos han presentado la amplia gama de funciones que es necesario organizar para que un sistema de E/S funcione de manera eficiente. En esta sección delinearemos una de las muchas jerarquías usadas para llevar a cabo estas funciones.

El módulo de nivel más alto se denomina “sistema de archivos básico” y pasa información a través del módulo de verificación y control de acceso al sistema de archivos lógico, que, a su vez, notifica al sistema de archivos físico, que trabaja con el administrador de archivos. La figura 8.14 muestra la jerarquía.



(Figura 8.14)
 Módulos típicos de un sistema de administración de archivos que muestra cómo se pasa la información desde el administrador de archivos en la parte superior de la jerarquía hasta el administrador de dispositivos en la parte inferior.

Cada nivel del sistema de administración de archivos se implementa usando técnicas de programación estructurada y modular que también establecen una jerarquía; es decir, los módulos ubicados en las posiciones superiores pasan la información a los módulos ubicados en las posiciones inferiores y la comunicación continúa siguiendo la cadena hasta el módulo más bajo, que se comunica con el dispositivo físico e interactúa con el administrador de archivos. Sólo entonces el registro está disponible para el programa del usuario.

Cada uno de los módulos puede dividirse aún más en tareas más específicas, como puede verse cuando esta instrucción de E/S se sigue a través del sistema de administración de archivos:

READ RECORD NUMBER 7 FROM FILE CLASSES INTO STUDENT

CLASSES es el nombre de un archivo con acceso directo abierto previamente para datos de entrada, y STUDENT es un registro de datos definido previamente dentro del programa y que ocupa ubicaciones específicas de la memoria.

Debido a que ya se ha abierto el archivo, ya se ha buscado en el directorio de archivos para verificar la existencia de CLASSES, e información pertinente sobre el archivo ya se ha llevado a la tabla de archivos activos del sistema operativo. Esta información incluye su tamaño de registro, la dirección de su primer registro físico, su protección e información de control de acceso, como se muestra en el listado del directorio UNIX de la tabla 8.1.

Control de acceso	No. enlaces	Grupo	Propietario	No. de bytes	Fecha	Hora	Nombre del archivo
drwxrwxr-x	2	journal	comp	12820	10 ene	19:32	ArtWarehouse
drwxrwxr-x	2	journal	comp	12844	15 dic	09:59	bus_transport
-rwxr-xr-x	1	journal	comp	2705221	6 mar	11:38	CLASSES
-rwxr-r-	1	journal	comp	12556	20 feb	18:08	PAYroll
-rwx---	1	journal	comp	8721	17 ene	07:32	supplier

(Tabla 8.1)

Lista típica de los archivos almacenados en el directorio denominado Journal.

Esta información se usa en el sistema de archivos básico, que activa el módulo de verificación de control de acceso para verificar que ese usuario está autorizado para efectuar esta operación con este archivo. Si se permite el acceso, la información y el control pasan al sistema de archivos lógico. En caso contrario, al usuario se le envía un mensaje que dice “acceso denegado”.

Al usar la información pasada por el sistema de archivos básico, el sistema de archivos lógico transforma el número de registro en su byte de dirección usando la fórmula conocida:

$$CBA = (NR - 1) * LR$$

Este resultado, junto con la dirección del primer registro físico y, en el caso donde los registros están bloqueados, el tamaño del bloque físico, se envían al sistema de archivos físico, que calcula la ubicación donde reside físicamente el registro deseado. Si en ese bloque hay más de un registro, calcula el desplazamiento del registro dentro de ese bloque usando estas fórmulas:

$$\text{número de bloque} = \text{enteros} \left[\frac{\text{byte de dirección}}{\text{tamaño del bloque físico}} \right] + \text{dirección del primer registro físico}$$

$$\text{desplazamiento} = \text{residuo} \left[\frac{\text{byte de dirección}}{\text{tamaño del bloque físico}} \right]$$

Esta información se pasa al módulo de interfaz de dispositivos, que, a su vez, transforma el número de bloque en la combinación real cilindro/superficie/registro necesaria para recuperar la información del dispositivo de almacenamiento secundario. Una vez recuperada, aquí es donde entran en escena los algoritmos de planificación de dispositivos, ya que la información es colocada en un búfer y el control regresa al sistema de archivos físico, que copia la información en la ubicación deseada de la memoria. Finalmente, cuando la operación está completa, a todos los otros módulos se envía el mensaje “todo bien”.

Aunque para nuestro ejemplo usamos un comando READ, un comando WRITE se maneja exactamente igual hasta que el proceso llega al manejador de dispositivos. En ese momento, la porción del módulo de interfaz de dispositivos que maneja la asignación de espacio libre, el módulo de asignación, entra en acción, ya que es responsable de seguir la pista de las áreas sin usar en cada dispositivo de almacenamiento.


Debemos observar aquí que la verificación, el proceso de asegurarse que una solicitud es válida, ocurre a todos los niveles del sistema de administración de archivos. La primera verificación se lleva a cabo a nivel del directorio cuando el sistema comprueba la existencia del archivo solicitado. La segunda ocurre cuando el módulo de verificación de control de acceso determina si se permite el acceso. La tercera ocurre cuando el sistema de archivos lógico comprueba si el byte de dirección está dentro de los límites del archivo. Finalmente, el módulo de interfaz de dispositivos comprueba si existe el dispositivo de almacenamiento.

En consecuencia, la operación correcta de un simple comando de usuario requiere el esfuerzo coordinado del sistema de administración de archivos.

Módulo de verificación de control de acceso

Los primeros sistemas operativos no podían prestar el servicio de compartición de archivos entre los usuarios. Por ejemplo, los primeros sistemas necesitaban 10 copias de un compilador para atender a 10 usuarios. Los sistemas actuales sólo requieren una copia para atender a todos, sin importar el número de programas activos que haya en el sistema. De hecho, cualquier archivo puede compartirse; desde archivos de datos y archivos de programas propiedad del usuario hasta sistemas de archivos. Las ventajas de compartir archivos son numerosas. Además de ahorrar espacio, permite la sincronización de actualizaciones de datos, como cuando dos aplicaciones actualizan el mismo archivo de datos. También mejora la eficiencia de los recursos del sistema, ya que si se comparten los archivos en la memoria principal entonces hay una reducción de operaciones de E/S.

Sin embargo, como suele ocurrir, el progreso conlleva problemas. La desventaja de compartir archivos es que debe salvaguardarse la integridad de cada archivo; eso requiere el control sobre a quién se le permite el acceso al archivo y qué tipo de acceso se permite. Hay cinco acciones posibles que pueden realizarse sobre un archivo: la capacidad de sólo LECTURA, de sólo ESCRITURA, de sólo EJECUTAR, de sólo BORRAR o alguna combinación de las cuatro. Cada sistema de administración de archivos tiene su propio método para controlar el acceso a archivos.



El control de acceso es un elemento crítico de la seguridad del sistema. Identifica a quién se le permite el acceso a qué archivos y qué operaciones se le permite realizar a ese usuario.

Matriz de control de acceso

Intuitivamente, la **matriz de control de acceso** es atractiva y fácil de implementar, pero por su tamaño, sólo funciona bien para sistemas con pocos archivos y pocos usuarios. En la matriz, cada columna identifica un usuario y cada renglón identifica un archivo. La intersección de un renglón y una columna contiene los derechos de acceso para ese usuario a ese archivo, como se ilustra en la tabla 8.2.

(Tabla 8.2)

Matriz de control de acceso que muestra los derechos de acceso para cada usuario a cada archivo. Al usuario 1 se le permite acceso ilimitado al archivo 1 pero sólo para leer y ejecutar el archivo 4, y se le niega el acceso a los otros tres archivos.

R = acceso para lectura,
W = acceso para escritura,
E = acceso para ejecutar,
D = acceso para borrar, y
un guión (-) = acceso no concedido.

	Usuario 1	Usuario 2	Usuario 3	Usuario 4	Usuario 5
Archivo 1	RWED	R-E-	----	RWE	---E-
Archivo 2	----	R-E-	R-E-	--E-	----
Archivo 3	----	RWED	----	--E-	----
Archivo 4	R-E-	----	----	----	RWED
Archivo 5	----	----	----	----	RWED

En la implementación real, las letras RWED están representadas por bits uno y cero: un uno indica que se ha concedido el acceso, y un cero, que se ha negado el acceso. En consecuencia, como se muestra en la tabla 8.3, el código para el usuario 2 para el archivo 1 se leería "1010" y no "R-E-".

(Tabla 8.3)

Los cinco códigos de acceso para el usuario 2 de la tabla 8.2. El código resultante para cada archivo es creado al asignar un 1 a cada arca de verificación y un 0 a cada espacio en blanco.

Acceso	R	W	E	D	Código resultante
R-E-	✓		✓		1010
R-E-	✓		✓		1010
RWED	✓	✓	✓	✓	1111
----					0000
----					0000

Como puede ver, la matriz de control de acceso es un método simple, pero a medida que aumentan el número de usuarios y archivos, la matriz se vuelve extremadamente grande; algunas veces demasiado grande para almacenarse en la memoria principal. Otra desventaja es que se desperdicia mucho espacio, ya que muchas de las entradas son nulas, como en la tabla 8.2, donde al usuario 3 no se permite acceder a la mayor parte de los archivos, y el archivo 5 está restringido a todos menos a un usuario. Un esquema que conservaba espacio sólo tendría una entrada para el usuario 3 o una para el archivo 5, pero esto es incompatible con el formato de la matriz.

Listas de control de acceso

La lista de control de acceso es una modificación de la matriz de control de acceso. Cada archivo se escribe en la lista y contiene los nombres de los usuarios a quienes se les ha permitido el acceso al archivo, así como el tipo de acceso que se permite a cada uno. Para acortar la lista, sólo se menciona a quienes pueden usar el archivo; a quienes se les ha negado el acceso se agrupan bajo un encabezado global como WORLD, como se muestra en la tabla 8.4.

Archivo Acceso

Archivo 1	USUARIO1 (RWED), USUARIO2 (R-E-), USUARIO4 (RWE-), USUARIO5 (--E-), WORLD (----)
Archivo 2	USUARIO2 (R-E-), USUARIO3 (R-E-), USUARIO4 (--E-), WORLD (----)
Archivo 3	USUARIO2 (RWED), USUARIO4 (--E-), WORLD (----)
Archivo 4	USUARIO1 (R-E-), USUARIO5 (RWED), WORLD (----)
Archivo 5	USUARIO5 (RWED), WORLD (----)

(Tabla 8.4)

Lista de control de acceso que muestra a cuáles usuarios se les ha permitido el acceso a cada archivo. Este método requiere menos espacio de almacenamiento que una matriz de control de acceso.

Algunos sistemas acortan aún más la lista de control de acceso al colocar a cada usuario en una categoría: sistema, propietario, grupo y mundo. SYSTEM o ADMIN está diseñado para personal del sistema que tiene acceso ilimitado a todos los archivos en el sistema. El OWNER tiene control absoluto sobre todos los archivos creados en la cuenta del propietario. Un propietario puede crear un archivo GROUP de modo que todos los usuarios que pertenecen al grupo idóneo tienen acceso al archivo. WORLD está integrado por todos los otros usuarios en el sistema; es decir, los que no están en ninguna de las otras tres categorías. En este sistema, el administrador de archivos designa tipos de acceso predeterminados a todos los archivos en el momento de su creación, y es responsabilidad del propietario modificarlos según sea necesario.

Listas de capacidad

Una lista de capacidad muestra la información de control de acceso desde una perspectiva diferente. Enumera todos los usuarios y los archivos a los que cada uno tiene acceso, como se muestra en la tabla 8.5.

Usuario Acceso

Usuario 1	Archivo 1 (RWED), Archivo 4 (R-E-)
Usuario 2	Archivo 1 (R-E-), Archivo 2 (R-E-), Archivo 3 (RWED)
Usuario 3	Archivo 2 (R-E-)
Usuario 4	Archivo 1 (RWE-), Archivo 2 (--E-), Archivo 3 (--E-)
Usuario 5	Archivo 1 (--E-), Archivo 4 (RWED), Archivo 5 (RWED)

(Tabla 8.5)

Una lista de capacidad que muestra los archivos para cada usuario requiere menos espacio de almacenamiento que una matriz de control de acceso y es más fácil de mantener que una lista de control de acceso cuando se agregan o eliminan usuarios del sistema.

De los tres esquemas descritos hasta el momento, el de uso más común es la lista de control de acceso. Sin embargo, las listas de capacidad están ganando popularidad, ya que en sistemas operativos como Linux o UNIX, controlan el acceso a dispositivos, así como también a archivos.

Aunque parece que ambos métodos son iguales, hay algunas diferencias sutiles que se explican mejor con una analogía. Una lista de capacidad puede igualarse a boletos de un concierto específico donde sólo se ponen a disposición de personas cuyo nombre aparece en la lista. Por otra parte, una lista de control de acceso puede igualarse a una lista de reservaciones en un restaurante con aforo limitado, donde cada sitio se asigna a una persona específica.

Compresión de datos

Los algoritmos de **compresión de datos** son de dos tipos: algoritmos fuertes que típicamente son utilizados para archivos de texto o aritméticos, los que retienen los datos en el archivo a lo largo del proceso de compresión-descompresión; y algoritmos débiles que generalmente son utilizados para archivos de datos y video y para borrar datos de forma permanente. A primera vista, no podemos pensar que una pérdida de datos puede ser tolerable, pero cuando los datos borrados son un ruido indeseable, tonos más allá de la capacidad del oído humano o espectros de luz que no podemos ver, eliminar estos datos puede ser indetectable y por tanto aceptable.

Compresión de texto

Para comprimir un texto en una base de datos, aquí se describen brevemente tres métodos: registros con caracteres repetidos, términos repetidos y compresión *front-end*.

Registros con caracteres repetidos: los datos en un campo de longitud fija pueden incluir un nombre corto seguido por muchos caracteres en blanco. Esto puede sustituirse por un campo de longitud variable y un código especial para indicar cuántos espacios en blanco fueron truncados.

Por ejemplo, suponga que la cadena original, ADAMS, se ve como sigue cuando se almacena sin comprimir en un campo de 15 caracteres de ancho (*b* representa un caracter en blanco):

ADAMSbbbbbbbbbb

Una vez codificada se ve como sigue:

ADAMSb10

De manera semejante, es posible acortar números con muchos ceros, también con un código para indicar cuántos ceros deben agregarse para recrear el número original (en este caso con el signo #). Por ejemplo, si la entrada original es este número:

300000000

la entrada codificada es ésta:

3#8

Términos repetidos pueden comprimirse usando símbolos para representar cada una de las palabras más usadas en la base de datos. Por ejemplo, en la base de datos de un estudiante universitario, cada una de las palabras comunes como *estudiante*, *curso*, *maestro*, *aula*, *calificación* y *departamento* pueden representarse con un solo carácter. Por supuesto, el sistema debe poder distinguir entre datos comprimidos y no comprimidos.

La *compresión front-end* trabaja con base en el elemento de datos previos. Por ejemplo, la base de datos del estudiante donde los nombres de los estudiantes se guardan en orden alfabético puede comprimirse, como se muestra en la tabla 8.6.

Lista original	Lista comprimida
Smith, Betty	Smith, Betty
Smith, Gino	7Gino
Smith, Donald	7Donald
Smithberger, John	5berger, John
Smithbren, Ali	6ren, Ali
Smithco, Rachel	5co, Rachel
Smither, Kevin	5er, Kevin
Smithers, Renny	7s, Renny
Snyder, Katherine	1nyder, Katherine

(Tabla 8.6)

Cada entrada asume un número dado de caracteres de la entrada previa que tienen en común y agrega los caracteres que lo hacen único. Así, "Smithbren, Ali" usa los seis primeros caracteres de "Smithbren, John" y agrega "ren". En consecuencia, la entrada es 6ren, Ali.

Hay una negociación: se gana espacio de almacenamiento, pero se pierde tiempo de procesamiento. Recuerde que para todos los esquemas de compresión de datos, el sistema debe poder distinguir entre datos comprimidos y no comprimidos.

Otros esquemas de compresión

La compresión débil permite una pérdida de datos del archivo original para admitir una compresión significativa. Esto significa que el proceso de compresión es irreversible y que el archivo original no puede ser reconstruido. Las especificaciones de los algoritmos de compresión dependen en gran medida del tipo de archivo que se va a comprimir, con JPEG como una opción po-

pular para conservar imágenes y MPEG para imágenes y video. Para archivos de video y música, la International Organization for Standardization (ISO, por sus siglas en inglés) ha emitido normas MPEG que “son normas internacionales relacionadas con la compresión, descompresión, procesamiento y representación codificada de fotografías móviles, audio y su combinación”.

ISO es el desarrollador mundial líder de normas internacionales. Para más información sobre normas de compresión actuales y otras normas industriales, le pedimos al lector visitar la página <http://www.iso.org>.

Conclusión

El administrador de archivos controla todos los archivos en el sistema y procesa los comandos del usuario (leer, escribir, modificar, crear, borrar, etc.) para interactuar con cualquier archivo en el sistema. También administra los procedimientos de control de acceso para mantener la integridad y la seguridad de los archivos bajo su control.

Para alcanzar sus objetivos, el administrador de archivos debe poder acomodar una variedad de organizaciones de archivos, esquemas de asignación de almacenamiento físico, tipos de registros y métodos de acceso. Y, como hemos visto, esto requiere software de administración de archivos cada vez más complicado.

En este capítulo hemos analizado:

- La organización de archivos secuencial, directa y secuencial indexada.
- La asignación de almacenamiento contiguo, no contiguo e indexado de archivos.
- Registros de longitud fija contra registros de longitud variable.
- Los tres métodos de control de acceso.
- Técnicas de compresión de datos.

Para obtener el máximo de un administrador de archivos, es importante que los usuarios conozcan los puntos fuertes y débiles de sus segmentos: cuáles métodos de acceso son permitidos sobre cuáles dispositivos y con qué estructuras de archivos, así como las ventajas y desventajas de cada una en eficiencia total.

Términos clave

Algoritmo *hashing*: Conjunto de instrucciones usadas para realizar una transformación clave donde un campo clave del registro determina su ubicación.

Almacenamiento contiguo: Tipo de almacenamiento de archivos donde toda la información se guarda en ubicaciones adyacentes en un medio de almacenamiento.

Almacenamiento no contiguo: Tipo de almacenamiento donde la información se guarda en ubicaciones no adyacentes en un medio de almacenamiento.

Archivo: Grupo de registros relacionados que contiene información que los programas de aplicación específicos usan para generar reportes.

Archivo de datos: Un archivo que contiene sólo datos.

Archivo de programa: Archivo que contiene instrucciones para la computadora.

Base de datos: Grupo de archivos relacionados interconectados a varios niveles para proporcionar a los usuarios flexibilidad de acceso a los datos almacenados.

Campo: Grupo de bytes relacionados que el usuario puede identificar con un nombre, tipo y tamaño. Un registro está compuesto por campos.

Campo clave: 1) Campo único o combinación de campos única en un registro que identifica de manera única ese registro; o 2) es el campo que determina la posición de un registro en una secuencia ordenada.

Compresión de datos: Procedimiento usado para reducir la cantidad de espacio requerido para almacenar datos al reducir, codificar o abreviar términos o caracteres repetitivos.

Descriptor de archivos: Información que se mantiene en el directorio para describir un archivo o su extensión

Dirección del byte actual (CBA): Dirección del último byte leído. Es usada por el administrador de archivos para tener acceso a registros en almacenamiento secundario y es necesario actualizarlo cada que se accede a un registro.

Dirección lógica: El resultado de una transformación clave.

Dirección relativa: En un entorno de organización directa, indica la posición de un registro en relación con el inicio de un archivo.

Directorio: Área de almacenamiento en un volumen de almacenamiento secundario (disco, plato de disco, etc.), que contiene información sobre los archivos almacenados en ese volumen.

Directorio actual: el directorio o subdirectorío en que está trabajando el usuario.

Directorio de trabajo: El directorio o subdirectorío donde trabaja actualmente el usuario.

Directorio maestro de archivos (MFD): Es un archivo almacenado inmediatamente después del descriptor de volumen. Enumera los nombres y las características de cada campo contenido en ese volumen.

Extensión: En algunos sistemas operativos, es la parte del nombre del archivo que indica el compilador o paquete de software necesario para ejecutar los archivos. En UNIX y Linux es opcional y se denomina *sufijo*.

Extensiones: Cualquier registro restante y todas las otras adiciones al archivo que están almacenadas en otras secciones del disco.

Independiente del dispositivo: Programas desprovistos de instrucciones detalladas necesarias para interactuar con cualquier dispositivo de E/S presente en el sistema de cómputo.

Lista de capacidad: Método de control de acceso que enumera cada usuario, los archivos a los que tiene acceso cada uno y el tipo de acceso que se permite a estos archivos.

Lista de control de acceso: Método de control que enumera cada archivo, los nombres de los usuarios a quienes se permite tener acceso al archivo, y el tipo de acceso que se permite a cada uno.

Matriz de control de acceso: Método de control que usa una matriz con cada archivo, cada usuario y el tipo de acceso que se permite a cada usuario para cada archivo.

Nombre absoluto del archivo: Nombre de un archivo, otorgado por el usuario, precedido por el directorio (o directorios) donde se encuentra el archivo y, cuando es necesario, la etiqueta del dispositivo específico.

Nombre relativo del archivo: Nombre y extensión de un archivo que lo distingue de otros archivos en el mismo directorio.

Organización de archivos directos: Archivos guardados en un dispositivo de almacenamiento con acceso directo y organizados para proporcionar a los usuarios la flexibilidad de tener acceso aleatorio a cualquier registro, sin importar su posición en el archivo.

Organización indexada secuencial de registros: Forma para organizar datos en un dispositivo de almacenamiento con acceso directo. El objetivo de crear un índice es mostrar dónde se almacenan los registros de datos. Cualquier registro de datos puede recuperarse al consultar primero el índice.

Organización secuencial de registros: Organización de registros en una secuencia específica. Los registros en un archivo secuencial deben procesarse uno después de otro.

Palabra de bloqueo: Secuencia de letras y/o números que se proporcionan a los usuarios para evitar manipulación peligrosa con sus archivos.

Registro: Grupo de archivos relacionados tratados como una unidad. Un archivo es un grupo de registros relacionados.

Registro de longitud fija: Registro que siempre contiene el mismo número de caracteres.

Registro de longitud variable: Registro que no es de longitud uniforme, no deja espacio de almacenamiento vacío y no trunca ningún carácter, eliminando así las dos desventajas de los registros de longitud fija.

Subdirectorio: Directorio creado por el usuario dentro de los límites de un directorio existente. Algunos sistemas operativos lo denominan carpeta.

Volumen: Cualquier unidad de almacenamiento secundario, como discos duros, paquetes de discos, CD, DVD, discos removibles o cintas.

Búsquedas de interés

- Almacenamiento de archivos en internet (cloud computing)
- Políticas de respaldo de archivos
- Técnicas de compresión de archivos
- Auditoría de archivos de acceso
- Limitaciones del nombre de archivos
- Función Hash

Ejercicios

Temas de investigación

- A. Investigue el tamaño del software del sistema operativo al encontrar la cantidad de espacio de almacenamiento secundario (disco) requerido por versiones diferentes del mismo sistema operativo o diferentes sistemas operativos. Si los tamaños son sustancialmente diferentes, explique por qué ocurre esto, como cuestiones de plataformas, características, etc. Cite sus fuentes.
- B. Consulte literatura actual o internet para investigar convenciones de nomenclatura de archivos para cuatro sistemas operativos diferentes (sin incluir UNIX, MS-DOS, Windows o Linux). Observe el intervalo aceptable de caracteres, la longitud máxima la sensibilidad al tipo de letra, etc. Proporcione ejemplos de nombres de archivos aceptables e inaceptables. Para crédito extra, explique cómo el administrador de archivos de estos sistemas operativos acorta nombres de archivos largos (en caso de hacerlo) en sus listas internas para facilitar su manipulación. Cite sus fuentes.

Ejercicios

1. Con sus propias palabras, explique por qué la desubicación de archivos es importante y qué puede suceder si esto no ocurre en una base regular.
2. Describa cómo el administrador de archivos ubica un archivo para un usuario único. Enliste los pasos que usted piensa que deben seguirse y explique su razonamiento.
3. ¿La independencia del dispositivo es importante para el administrador de archivos? ¿Por qué sí? ¿Por qué no? Describa las consecuencias si este no fuera el caso.
4. ¿Piensa que la recuperación de archivos es diferente en un sistema guiado por menú y en uno guiado por comandos? Explique su respuesta y describa las diferencias entre los dos. Proporcione un ejemplo de cuándo es preferible uno sobre el otro.
5. Imagine un ejemplo real para cada uno: un volumen de múltiarchivo y un archivo multivolumen. Incluya una descripción del medio utilizado para almacenamiento y una descripción general de los datos en el archivo.
6. Como se describió en este capítulo, los archivos pueden formatearse con campos de longitud fija o de longitud variable. En su opinión, ¿es posible combinar ambos formatos en un solo disco? Explique las razones de su respuesta.
7. Explique por qué es difícil soportar accesos directos a archivos con registros de longitud variable. Sugiera un método para manejar este tipo de archivo si se requiere un acceso directo.
8. Proporcione un ejemplo de los nombres de tres archivos de su propia computadora que no residan en el directorio maestro o raíz. Para cada archivo, liste su nombre relativo y su nombre completo.
9. En sus propias palabras, describa el propósito del directorio de trabajo y cómo puede acceder de forma rápida o lenta a los archivos. En su opinión, ¿puede existir más de un directorio de trabajo? Explique.

10. Para cada uno de los siguientes elementos en una matriz de control de acceso para User 2010, proporcione el tipo de acceso permitido para cada uno de los siguientes archivos:
- a. Archivo1 --E-
 - b. Archivo2 RWED
 - c. Archivo3 R-E-
 - d. Archivo4 R---
11. Para cada uno de los elementos en una lista de control, proporcione el tipo de acceso permitido para cada usuario del archivo221 y describa quién está incluido en la categoría WORLD:
- a. User2010 R-E-
 - b. User2014 -E-
 - c. User2017 RWED
 - d. WORLD R---
12. Ideé una forma de comprimir la siguiente lista de nombres utilizando una técnica fuerte similar a la mostrada en la tabla 8.6. Describa su método y muestre la lista comprimida. Explique por qué su técnica es fuerte (y no débil).
- POWE
 POWELL
 POWER
 POWERS
 PORUN

Ejercicios avanzados

13. Si usted está diseñando el control de acceso a archivos para un entorno altamente seguro y pudiera escoger entre el establecimiento de muchas categorías de acceso y apenas unas cuantas, ¿qué escogería y por qué?
14. Compare y contraste asignación de memoria dinámica y la asignación de archivos en almacenamiento secundario.
15. ¿Cuándo la compactación del almacenamiento secundario es benéfica desde la perspectiva del administrador de archivos? Proporcione varios ejemplos. Enumere algunos problemas que podrían presentarse como resultado de la compactación y explique cómo podrían evitarse.
16. Aunque administradores de archivos sofisticados implementan la compartición de archivos al permitir que varios usuarios tengan acceso a una sola copia de un archivo al mismo tiempo, otros implementan la compartición de archivos al proporcionar una copia del archivo a cada usuario. Enumere las ventajas y desventajas de cada método.

