**Introducción a los directorios**   
  
El directorio puede considerarse como una tabla de símbolos que traduce los nombres de archivo a sus correspondientes entradas de directorio.  
  
**Directorio en dos niveles**  
  
Cada uno de los UFD tiene una estructura similar, pero sólo incluye los archivos de un único usuario. Cuando un trabajo de un usuario se inicia o cuan-do un usuario se conecta al sistema, se explora el directorio maestro de archivos del sistema. Por tanto, cada uno de los usuarios puede tener archivos con el mismo nombre, siempre y cuando los nombres de archivo dentro de cada UFD sean unívocos. La asignación de espacio de disco para las direcciones de usuario puede gestionarse mediante las técnicas explicadas en el Capítulo 11 para los propios archivos.  
  
Esta estructura aísla efectivamente a un usuario y otro y ese aislamiento es una ventaja cuando los usuarios son completamente independientes, pero puede llegar a ser una desventaja cuando los usuarios quieren cooperar en una cierta tarea y poder acceder a los archivos del otro. Algunos sistemas simplemente no permiten que un usuario acceda a los archivos locales de otro usuario. Para denominar a los archivos de manera unívoca en una estructura de directorio de dos niveles, debemos proporcionar tanto el nombre de usuario como el nombre del archivo. Los descendientes de los UFD son los propios archivos, que actúan como hojas del árbol.  
  
Cada sistema tiene su propia sintaxis para nombrar los archivos contenidos en los, directorios que no pertenecen al propio usuario. Un caso especial de esta situación es el que se refiere a los archivos del sistema. Los programas -proporcionados como parte del sistema están generalmente definidos como archivos. Cuando se proporcionan los comandos apropiados al sistema operativo, el cargador carga estos archivos y los ejecuta.  
  
Tal como hemos definido el sistema de directorios, este nombre de archivos se buscaría en el UFD actual. Se define un directorio de usuario especial para contener los archivos del sistema. La secuencia de directorios que se exploran cada vez que se proporciona un nombre de archivos se denomina ruta de búsqueda.

Si queremos permitir ese tipo de acceso, cada usuario debe tener la posibilidad de especificar un archivo que pertenezca al directorio de otro usuario. Para denominar a los archivos de manera unívoca en una estructura de directorio de dos niveles, debemos proporcionar tanto el nombre de usuario como el nombre del archivo. Un directorio en dos niveles puede verse como una estructura de árbol, o de árbol invertido, de altura 2. Por tanto, un nombre de usuario y un nombre de archivo definen un nombre de ruta.

Cada archivo del sistema tiene un nombre de ruta y para designar a un archivo de manera unívoca, el usuario debe conocer el nombre de ruta del archivo deseado. Por ejemplo, si el usuario A quiere acceder a su propio archivo denominada prueba, simplemente puede referirse a él como test. Se necesita una sintaxis adicional para especificar el volumen de un archivo. Algunos sistemas van todavía más allá y separan las partes de la especificación correspondientes al volumen, al nombre del directorio y al nombre del archivo.

Por ejemplo, en VMS, el archivo login. Otros sistemas simplemente tratan el nombre del volumen como parte del nombre de directorio. El primer nombre que se proporcione será el del volumen y el resto se referirá al directorio y al archivo. Por ejemplo, /u/pbg/ test podría especificar el volumen u, el directorio pbg y el archivo test.

Muchos -intérpretes de comandos simplemente tratan dicho comando como el nombre de un archivo que hay que cargar y ejecutar. Se define un directorio de usuario especial para contener los archivos del sistema .

Que el directorio que hay que borrar no está vacío, sino que contiene varios archivos o subdirec-3 torios. Alternativamente, el usuario B puede cambiar el subdirectorio actual para situarse en el directorio del usuario A y acceder al archivo simplemente proporcionando su nombre. El mismo archivo o subdirectorio puede estar en dos directorios diferentes. El grafo acíclico es una generalización natural del esquema de directorio con estructura de árbol.

Es importante observar que un archivo compartido no es lo mismo que dos copias del archivo. Con dos copias cada programador puede ver su propia copia en lugar del original, pero si un programador modifica el archivo, esos cambios no se reflejarán en la copia del otro. Con un archivo compartido, sólo existe un archivo real, por lo que cualquier cambio realiza-do por una persona será inmediatamente visible para la otra. El UFD de cada miembro del grupo contendrá como subdirectorio este directorio de archivos compartidos.

Incluso en el caso de un único usuario, la organización de archivos del usuario puede requerir que algún archivo sea colocado en diferentes subdirectorios. Por ejemplo, un programa escrito para un proyecto concreto podría tener que almacenarse tanto en el directorio correspondiente a todos los programas como en el directorio correspondiente a ese trayecto. Una manera bastante común, ejemplificada por muchos de los sistemas UNIX, consiste en crear una nueva entrada de directorio denominada enlace. Un enlace es, en la práctica, un puntero a otro archivo o subdirectorio.

Cuando se realiza una referencia a un archivo, exploramos el directorio y, si la entrada del directorio está marcada como enlace, entonces se incluye el nombre del archivo real dentro de la información del enlace. Para resolver el enlace, utilizamos el nombre de ruta con el fin de localizar el archivo real. Los enlaces pueden identificarse fácilmente por el formato que tienen en la entrada de directorio y son, en la práctica, punteros indirectos nominados. El sistema operativo ignora estos enlaces a la hora de recorrer los árboles de directorio, con el fin de preservar la estructura acíclica del sistema.

Un doble clic sobre un archivo hace que se lea su atributo de creador y que se explore el archivo Desktop File en busca de una correspondencia. Si se encuentra una correspondencia, se inicia el programa ejecutable apropiado, utilizando como entrada el archivo sobre el que se ha hecho clic.

Directorios en un grafo acíclico

Como ambos programadores son igualmente responsables del proyecto, ambos quieren que el subdirectorio se encuentre dentro de su propio directorio. Cada directorio o archivo compartido existirá en el sistema de archivos en dos lugares simultáneamente.

Los archivos podrán tener ahora múltiples nombres de ruta absoluta. En consecuencia, puede haber nombres de archivo distintos que hagan referencia a un mismo archivo. Si estamos intentando recorrer el sistema de archivos completo este problema cobra una gran importancia, ya que no conviene recorrer las estructuras compartidas más de una vez. Una posibilidad consiste en eliminar el archivo cuan-do un usuario cualquiera lo borre, pero esta acción puede dejar punteros colgantes al archivo que ha dejado de existir.  
  
En un sistema en el que la compartición se implemente mediante enlaces simbólicos, la situación es algo más fácil de manejar. El borrado de un enlace no tiene por qué afectar al archivo original, ya que sólo se elimina el enlace. Si lo que se elimina es la propia entrada del archivo, se desasignará el espacio del archivo, dejando que los enlaces cuelguen. Podemos buscar estos enlaces y eliminarlos también, pero a menos que se mantenga con cada archivo una lista de los enlaces asociados esta búsqueda puede ser bastante costosa.  
  
En el caso de UNIX, cuando se borra un archivo se dejan los enlaces simbólicos y es responsabilidad del usuario darse cuenta de que el archivo original ya no existe o ha sido sustituido. Otra técnica de borrado consiste en preservar el archivo hasta que se borren todas las referencias al mismo. Para implementar esta técnica, debemos disponer de algún mecanismo para determinar que se ha borrado la última referencia al archivo. Podríamos mantener una lista de todas las referencias al archivo.  
  
Cuando se establece un enlace o una copia de la entrada de directorio, se añade una nueva entrada a la lista de referencias al archivo. El archivo se borrará cuando se vacíe su lista de referencias al archivo. El problema con esa técnica es el tamaño variable, y potencialmente grande, de la lista de referencias al archivo. Cuando el contador sea 0, podrá borrarse el archivo, ya que no habrá más referencias a él.  
  
El sistema operativo UNIX utiliza esta técnica para los enlaces no simbólicos, manteniendo un contador de referencias dentro del bloque de información del archivo. Es fácil darse cuenta de que la simple adición de nuevos archivos y subdirectorios a una estructura en árbol existente preserva la naturaleza y la estructura de árbol de ese directorio. La principal ventaja de un grafo acíclico es la relativa simplicidad de los algoritmos requeridos para recorrer el grafo y para determinar cuándo no existen ya más referencias a un archivo. Si acabamos de explorar un subdirectorio compartido de gran tamaño en busca de un archivo concreto sin haber llegado a encontrarlo, necesitamos evitar tener que volver a explorar dicho subdirectorio otra vez, ya que esa segunda búsqueda sería una pérdida de tiempo.  
  
Un problema similar existe cuando tratamos de determinar si un archivo puede ser borrado. Con las estructuras de directorio en grafo acíclico, un valor de 0 en el contador de referencia significará que ya no hay más referencias al archivo o directorio, y que ese archivo puede ser borra-do. Sin embargo, si existen ciclos, el contador de referencias puede no ser O incluso aunque ya no sea posible hacer referencia a un directorio de archivo. La recolección en memoria implica recorrer todo el sistema de archivos, marcando todos aquellos elementos a los que se pueda acceder.  
  
Sin embargo, la recolección de memoria para un sistema de archivos basado en disco consume muchísimo tiempo y raramente se la suele utilizar. Montarse para poder estar disponible para los procesos del sistema.  
  
Tura de directorios puede estar formada por múltiples volúmenes, que puede montarse para hacer que estén disponibles dentro del espacio de nombres del sistema de archivos. El proceso de montaje es bastante simple. " Normalmente, el punto de monta" je será un directorio vacío. Si montáramos ese sistema -de archivos bajo el directorio / users obtendríamos el nombre de ruta /users/jane, que podríamos utilizar para acceder al mismo directorio.  
  
A continuación, el sistema operativo verifica que el dispositivo contiene un sistema de archivos válido. Finalmente, el sistema operativo registra en su estructura de directorios que hay un sistema de archivo montado en el punto de montaje especificado. Este esquema permite al sistema operativo recorrer su estructura de directorios, pasando de un sistema de archivos a otro según sea necesario.