Sistemas Operativos
Tema 9

Gestión de archivos

UNED

Manuel Fernández Barcell

http://www.mfbarcell.es

Objetivos docentes

- Conocer las posibles operaciones y características (tipos, atributos, estructuras internas y métodos de acceso) de los archivos soportadas por un sistema operativo.
- Conocer las principales estructuras de los directorios y las operaciones básicas sobre los mismos soportadas por un sistema operativo.
- Saber qué es un sistema de archivos y cuál es su estructura general.
- Saber qué es y cómo se implementa el montaje de un sistema de ficheros.
- Conocer cómo se puede implementar en un sistema de archivos la asignación de espacio, la gestión del espacio libre y los directorios.
- Conocer las principales inconsistencias que puede presentar un sistema de archivos y sus posibles soluciones.
- Conocer los principales métodos de recuperación de archivos.
- Saber cómo influye la gestión de los archivos que realiza el sistema operativo en la eficiencia del sistema informático.

Tipos de archivos

- Un archivo informático se puede definir como un conjunto de información relacionada que se almacena en memoria secundaria y que se identifica mediante un nombre, como una cadena de caracteres.
  - Un archivo contiene programas o datos.
- Dos de los tipos de archivos comunmente soportados por los sistemas operativos son los directorios y los archivos regulares u ordinarios.
  - Un directorio es un archivo que almacena una lista de los archivos y otros directorios que contiene.
- Un archivo regular puede ser un archivo ASCII o un archivo binario.
  - Un archivo binario
    - Contiene información de cualquier tipo codificado en binario con una estructura determinada que solo puede ser interpretada por los programas que los utilizan.
  - Un archivo ASCII
    - Está compuesto de líneas de caracteres ASCII codificados en binario que no requiere de un programa que las interprete.
- El nombre de un archivo es una cadena de caracteres.
  - Cada sistema de archivos soportado por el sistema operativo especifica la longitud máxima y el tipo de caracteres que puede tener.
    - La extensión del archivo proporciona información sobre el tipo de archivo.
      - Sistemas como UNIX ignoran las extensiones y otros como Windows las reconoce e interpretan asociándolas con el programa que los genera.

Atributos de un archivo

- La lista de atributos varía en función del sistema de archivos pero en general se tiene:
  - Tipo de archivo.
  - Tamaño, en bytes, palabras o bloques.
  - Localización, ubicación en memoria secundaria.
  - Creador y propietario. Identifica al usuario que lo creó y su actual propietario.
  - Permisos de acceso.
    - Para determinar quién puede acceder y qué puede hacer con el archivo.
  - Información asociada al tiempo.
    - Como fecha y hora de creación, modificación, último acceso, etc.

Estructura interna de un archivo

- La información contenida en un archivo se puede estructurar de tres formas posibles:
- **Secuencia de bytes**
  - Las operaciones de lectura y escrituras se realizan a nivel de byte.
  - El sistema operativo no tiene que interpretar la información contenida en cada byte; esa responsabilidad recae en los programas de aplicación que se ejecutan a nivel de usuario.
- **Secuencia de registros**
  - Cada registro, de igual longitud, posee su propia estructura interna.
  - Las operaciones de lectura y escritura también se realizan a nivel de registros.
- **Registros indexados**
  - Cada registro, de longitud variable, contiene un campo índice que permite identificarlo.
  - Se organiza en función de la clave de los registros que lo componen.
  - En una operación de lectura o escritura debe especificarse la clave de registro que se desea leer o escribir.
  - Al añadir un nuevo registro es necesario indicar su clave.
- Un archivo está formado por **bloques lógicos** o registros de igual o distinto tamaño.
  - En el caso de secuencia de bytes el tamaño del bloque es de un byte
- Un **bloque físico**, que comprende varios sectores, tiene asignada una dirección física B_F o número de bloque.
- Un archivo ocupará N bloques lógicos, existiendo fragmentación interna ya que el último bloque probablemente no sea ocupado al completo.

Métodos de acceso a un archivo

- Acceso secuencial.
  - Los bytes o registros se leen o escriben en orden comenzando desde el principio.
  - El sistema mantiene un puntero de lectura/escritura que indica la posición donde debe comenzar la siguiente operación.
  - Compiladores y procesadores de texto la utilizan.
- Acceso aleatorio.
  - También llamado acceso directo.
    - Los bytes o registros pueden ser leídos en cualquier orden.
  - Las operaciones de lectura/escritura pueden implementarse especificando el número de byte o registro al que se desea acceder en la misma instrucción o bien realizando primero una búsqueda y posteriormente realizar la operación de manera secuencial.

Operaciones sobre archivos

- Crear archivo
  - Crea un archivo sin datos, vacío, en la estructura de directorios y crea las entradas necesarias en el directorio correspondiente.
- Abrir archivo
  - Se busca en un directorio la entrada asociada al archivo y se carga en memoria principal toda la información que necesite sobre el archivo: atributos, posiciones de memoria secundaria donde se aloja, etc.
  - Si la operación se realiza con éxito se devuelve un entero positivo denominado identificador de archivo, que lo identifica en el conjunto de archivos abiertos por el proceso o por el conjunto de procesos.
- Posicionamiento o búsqueda, seek
  - Permite configurar el puntero de lectura/escritura para que apunte al comienzo del byte o registro donde se desea leer o escribir.
- Leer archivo
  - Permite la lectura del archivo desde la posición actual del puntero.
    - La llamada al sistema incluye como argumentos el identificador del archivo, el número de bytes o registros que se desean leer y la dirección de memoria del espacio de usuario del proceso donde colocar los datos leídos.
- Escribir archivo
  - Permite la escritura del archivo desde la posición actual del puntero.
  - La llamada debe incluir el identificador de archivo, el número de bytes o registros a escribir y la dirección de memoria del espacio de usuario del proceso desde el que tomar los datos.
  - Si los datos se escriben al final del archivo se provoca su aumento de tamaño.
- Renombrar archivo
- Cerrar archivo
  - La información que se mantiene en memoria principal asociada al archivo se descarta.
  - Si el archivo ha sido abierto por varios procesos no se cierra hasta que lo hace el último de los procesos.
- Borrar archivo.
  - Libera el espacio en memoria secundaria y elimina su entrada en el directorio.

Directorios
Estructura de los directorios

- Un directorio almacena una lista de archivos y subdirectorios.
- Estructura de directorios de un único nivel
  - Consta de un único directorio donde se almacena una lista con todos los archivos existentes, tanto de usuario como del sistema.
  - Todos los archivos deben tener nombres distintos.

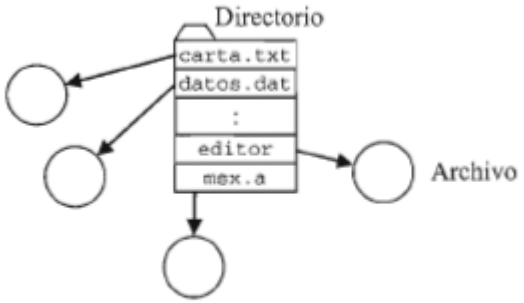

Figura 9.1 – Estructura de directorios de un único nivel

Estructura de directorios de dos niveles

- Consiste en un directorio raíz, o directorio de archivos maestro, que almacena una lista de directorios, uno por usuario.
- En su implementación más sencilla, un usuario sólo tiene acceso al contenido de su directorio.
- Para poder utilizar archivos de otros usuarios, además de que el usuario posea los permisos necesarios, el sistema debe manejar nombres de ruta.
  - Así, un archivo queda definido por la ruta del directorio donde se encuentra y su propio nombre.
- Este método también permite evitar la duplicidad de los archivos del sistema y de aplicación.
- El sistema crea un usuario ficticio especial y en su directorio almacena todos esos archivos.
- Cuando un usuario invoca un archivo primero se busca en su directorio de usuario y si no se encuentra se acude al directorio especial.

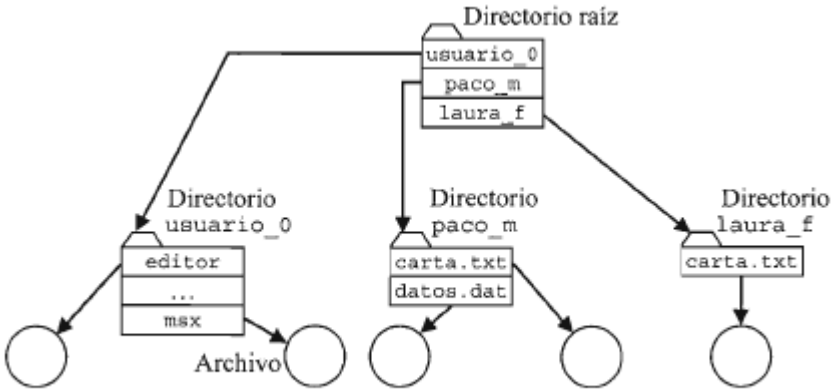

Figura 9.2 – Estructura de directorios de dos niveles

Estructura de árbol de directorios

- La lista almacenada en cada directorio puede contener entradas de otros directorios y archivos, pudiendo crearse múltiples niveles de directorios.
- Para acceder a un determinado archivo debe especificarse su ruta completa.
- La ruta puede ser absoluta, indicando el camino desde el directorio raíz, o relativa, desde el directorio actual o directorio de trabajo.
- Los sistemas que soportan una estructura de árbol crean dos entradas de directorio especiales:
  - La entrada punto ".",
    - Que hace referencia al directorio actual
  - La entrada dos puntos "..",
    - Que hace referencia al directorio padre.

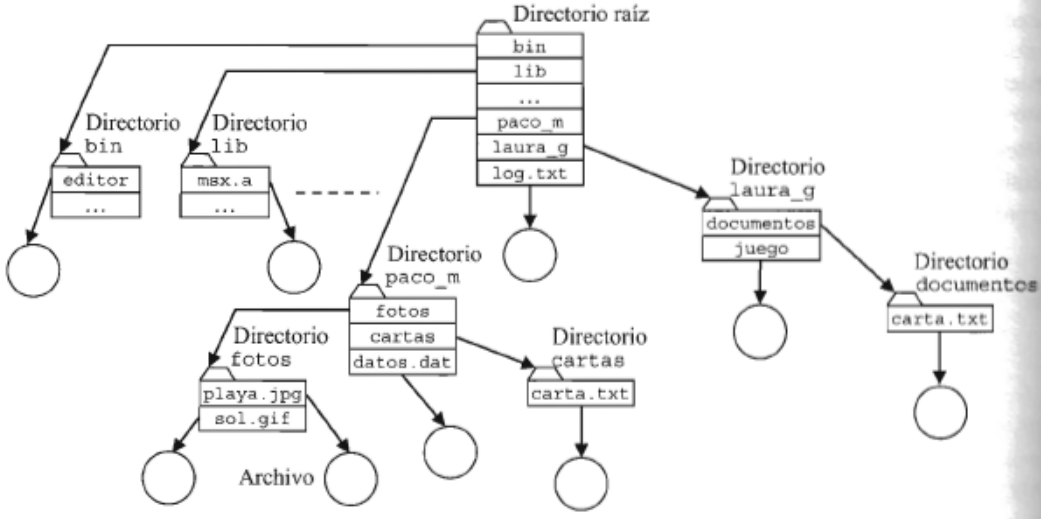

Figura 9.3 – Estructura de árbol de directorios

Estructura de directorios de gráfica acíclica

- Con el fin de implementar la compartición de archivos y directorios se emplea la estructura de directorios de gráfica acíclica, que permite que el mismo directorio o archivo pueda ser referenciado por dos o más directorios diferentes, siempre que no se produzcan ciclos.
- **Enlace duro**
  - Creando en el directorio origen una entrada que sea copia de la entrada del directorio destino asociada al archivo compartido.
  - Esta nueva entrada tendrá un atributo especial que indique que es un enlace
- **Enlaces simbólicos**
  - Un tipo especial de archivo que contiene la ruta absoluta o relativa del archivo o subdirectorio compartido
  - Los enlaces simbólicos permiten enlazar con cualquier archivo de la estructura de directorios local o remota.
  - El uso de enlaces simbólicos consumen más espacio que los enlaces duros, ya que ocupan una entrada de datos.
  - Su utilización aumenta el número de operaciones de E/S y la sobrecarga del sistema ya que es necesario buscar la entrada del enlace en el directorio correspondiente para conocer su ubicación en disco, leer del disco el archivo del enlace para conocer la ruta del elemento compartido y después seguir esa ruta para encontrar la entrada del directorio que le permita conocer la ubicación del elemento compartido

Directorio de gráfica acíclica

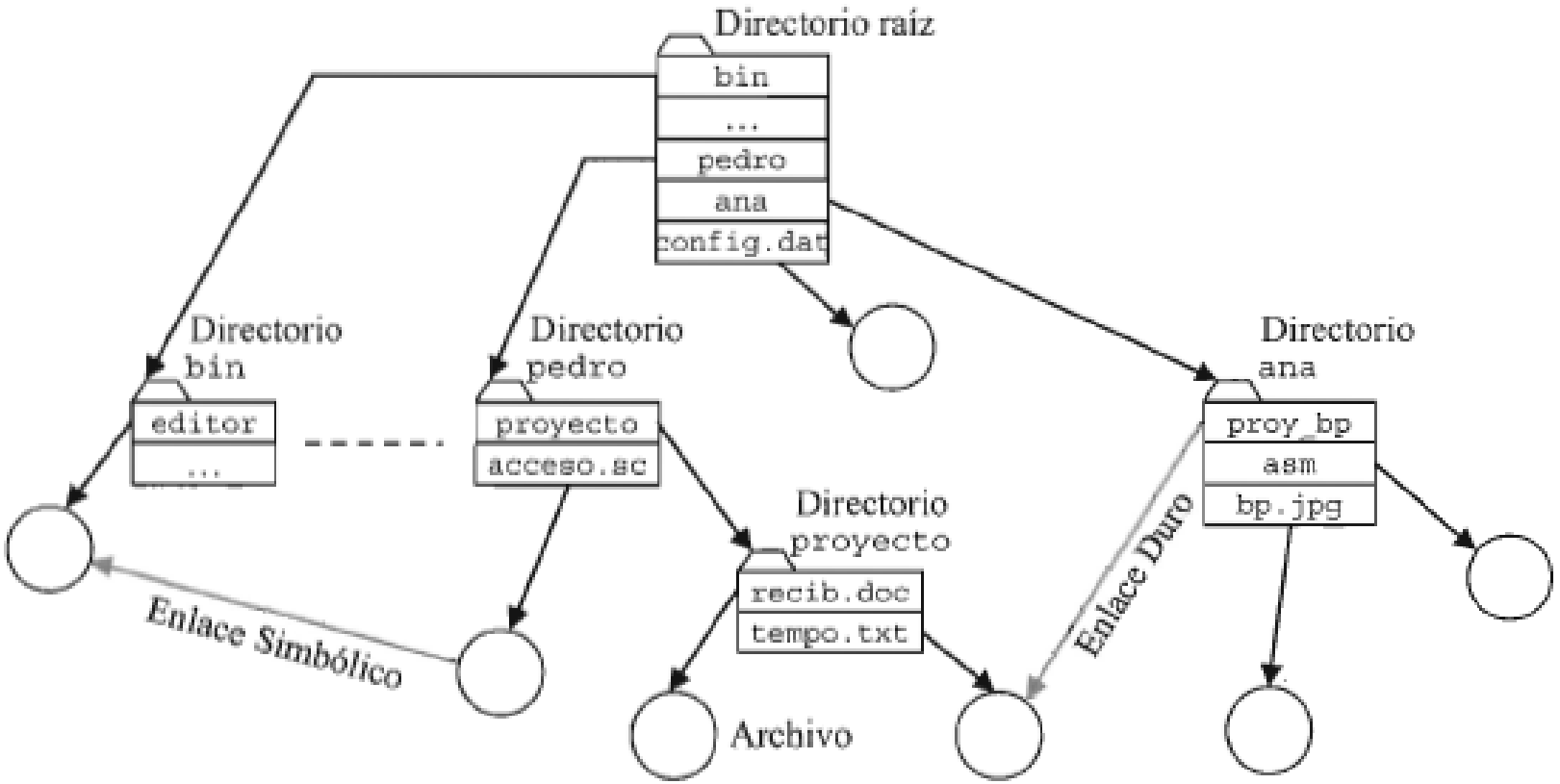

Figura 9.4 – Estructura de directorios de gráfica acíclica

Operaciones sobre directorios

- Crear directorio
  - Cuando se crea no contiene ninguna entrada, salvo “.” y “..”, si el sistema operativo las soporta.
- Borrar directorio.
  - En algunos casos sólo se permite si el directorio se encuentra vacío.
  - Si se permite, se eliminan también todos los elementos que contiene.
- Abrir directorio.
  - Consiste en encontrar en el directorio padre la entrada que se desea abrir y cargarla en memoria principal.
- Cerrar directorio.
  - Se libera la memoria principal que almacenaba la información del directorio.
- Leer directorio.
  - Permite leer el contenido de una entrada del directorio, necesario para operaciones de búsqueda y listado de contenido del directorio.
- Renombrar directorio.
  - Permite modificar su nombre.
- Enlazar.
  - Permite crear una entrada para enlace duro o simbólico.
- Desenlazar.
  - Permite eliminar un enlace.

SISTEMAS DE ARCHIVOS

Estructura de un sistema de archivos

- Un sistema de archivos define un esquema de almacenamiento de la información relativa a los archivos en un dispositivo de almacenamiento secundario.
- En un disco puede haber tantos sistemas de archivos distintos como particiones tenga.
- De forma general se pueden distinguir las siguientes áreas:
  - Bloque de arranque.
    - Se sitúa al comienzo de la partición y puede contener el código necesario para arrancar el sistema operativo.
  - Estructura de datos con metadatos del sistema de archivos.
    - Contiene información administrativa y estadística del sistema de archivos, como por ejemplo el identificador del tipo de sistema de archivos, número de bloques, número de bloques libres, etc.
    - En algunos sistemas se denomina superbloque.
      - Es copiado a memoria principal cuando se accede por vez primera al sistema de archivos.
  - Estructura de datos con información sobre los bloques libres en el sistema de archivos.
    - Generalmente denominada lista de bloques libres. Puede implementarse como mapa de bits o como lista enlazada.
  - Estructura de datos con información sobre bloques asignados a los archivos.
    - Algunas de las más utilizadas son:
      - Lista de nodos índice. O nodo-i,
        - » Estructura de datos utilizada para almacenar los atributos de un archivo y su ubicación en disco.
        - » Una entrada del nodo-i contiene la dirección física o el número de bloque donde se encuentra el nodo-i asociado a un determinado archivo.
    - Tabla de asignación de archivos. (*File Allocation Table*, FAT).
      - Tiene una entrada por cada bloque físico existente en la partición del sistema de archivos.
      - La entrada j de la tabla hace referencia al bloque físico j.
        - » Si el bloque físico j está asignado a un archivo, la entrada j en la FAT contiene la dirección física del siguiente bloque del archivo.
      - En este caso no es necesario mantener información sobre bloques libres.
  - Área de datos.
    - Contiene los bloques libres y bloques asignados a los archivos y directorios

Montaje de un sistema de archivos

- Para poder acceder a los contenidos de un determinado sistema de archivos ubicado en memoria secundaria este debe ser montado en algún punto dentro de la estructura de directorios.
  - A dicho punto se le denomina **punto de montaje**.
    - El sistema operativo realiza el montaje de los sistemas de archivos en el arranque o cuando se detecta un nuevo dispositivo de memoria secundaria conectado al computador.
- En sistemas Windows se trata cada sistema de archivos como un volúmen o unidad lógica, cada una de las cuales tiene su propia estructura de directorios.
  - Se asigna una letra a cada unidad.
- En sistemas Linux cada sistema de archivos se monta como un directorio del sistema de archivos principal, luego todos los sistemas de archivos quedan integrados dentro de la misma estructura de directorios.

Asignación de espacio

Contigua

Enlazada

Indexada

Asignación contigua

- Se asigna a un archivo un conjunto de bloques físicos contiguos.
  - Así, si un archivo consta de N bloques y el primer bloque se almacena en B_{F0}, el segundo lo hará en B_{F0+1}, y el último ocupará el bloque físico B_{F0+N-1}.
- Este método minimiza las operaciones de búsqueda en disco y soporta tanto archivos de acceso secuencial como aleatorio.
- Presenta la desventaja de producir fragmentación externa.
  - Sucesivos procesos de creación y borrado de archivos provocará la aparición de huecos cada vez más pequeños, siendo necesario compactar el disco, esto es, trasladar todos los archivos a un lado de la partición y todos los huecos al otro.
- Otro inconveniente es que el sistema operativo debe conocer de antemano el espacio que va a ocupar el archivo.
  - Si se le asigna un hueco de tamaño superior al necesario se producirá fragmentación interna.
  - Si se le asigna un espacio inferior al que podría alcanzar tendrá que recurrirse a

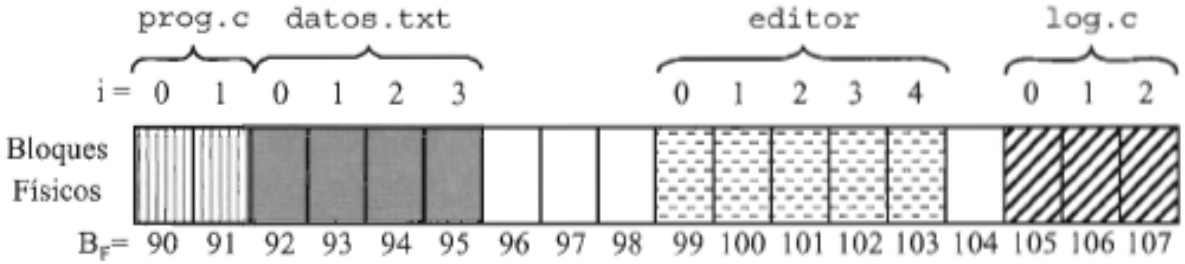

Figura 9.5 – Espacio del disco asignado mediante el método de asignación contigua

Asignación enlazada

- Consiste en almacenar al principio de cada bloque físico asignado a un archivo la dirección física del siguiente bloque físico del archivo.
  - Así, se organiza como una lista enlazada de bloques físicos.
- Permite usar cualquier bloque, por lo que no produce fragmentación externa y evita conocer por adelantado el tamaño del archivo.
- Resulta muy lento para su utilización con archivos de acceso aleatorio ya que para localizar un bloque hay que pasar por todos los anteriores.
- La utilización de algunos bytes del bloque para almacenar la dirección del bloque siguiente provoca que haya menos espacio disponible para datos y que este espacio no sea potencia de dos, dificultando el procesamiento y la eficiencia
- Una solución bastante utilizada es asignar el espacio en agrupamientos (clusters) de bloques físicos contíguos, almacenándose al comienzo la dirección del siguiente agrupamiento.
- Se puede utilizar una FAT, ubicada en sectores contiguos después del sector de arranque. La FAT tiene una entrada por cada bloque físico de la partición.

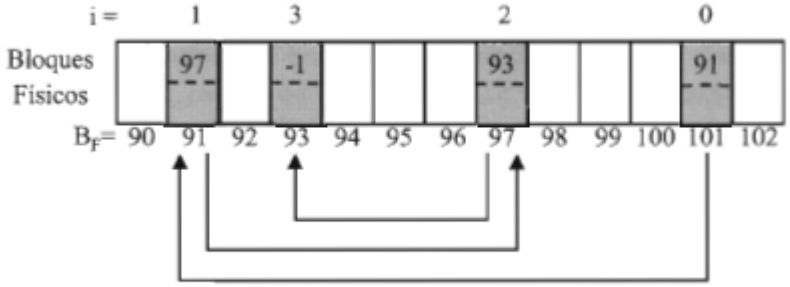

Figura 9.6 – Espacio del disco asignado mediante el método de asignación enlazada

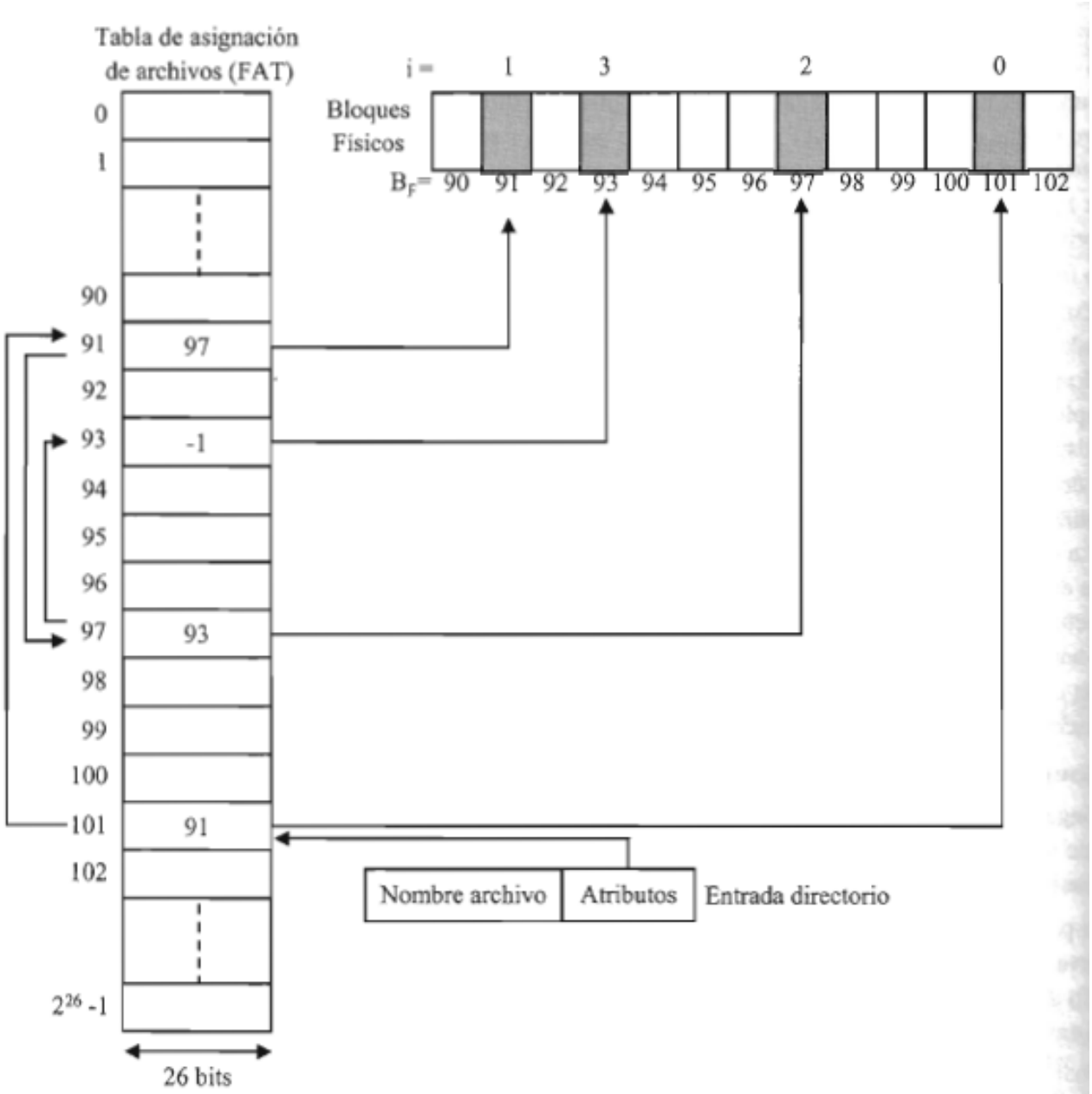

Figura 9.7 – Espacio del disco asignado mediante el método de asignación enlazada usando una FAT

Asignación indexada

- En este método se almacena en un nodo-i los atributos de un archivo y las direcciones físicas de los primeros ocho o diez bloques de un archivo.
- También se almacenan las direcciones físicas de uno o varios bloques de indicrección simple, doble o triple.
  - Un bloque de indirección simple almacena direcciones físicas de bloques de archivo.
  - Un bloque de indirección doble almacena direcciones físicas de bloques de indirección simple.
  - Un bloque de indirección triple almacena direcciones físicas de bloques de indirección doble.
- Cada archivo tiene asociado un número entero positivo denominado número de nodo- i. Al principio de la partición se mantiene una lista con todos los nodos-i existentes.
  - El número de nodo-i asociado a un archivo se almacena en la entrada de directorio que contiene al archivo.
- Este método se puede usar tanto para archivos de acceso secuencial como aleatorio.
- No produce fragmentación externa y permite que el espacio de los bloques físicos que contienen datos puedan ser utilizados totalmente.
- Su implementación requiere de menos memoria principal que la FAT, ya que solo es necesario mantener en memoria los nodos-i de los archivos abiertos.
- La principal desventaja es que el tiempo de acceso a un bloque de archivo depende de la estructura interna del nodo-i.
  - En el mejor de los casos y suponiendo que el nodo-i del archivo ya ha sido cargado en memoria principal, si se desea acceder a uno de los primeros bloques del disco sólo hay que hacer una lectura para acceder a dicho bloque, pero si se trata de bloques a los que se accede a través de bloques de indirección simple, doble o triple, habrá que hacer una, dos o tres lecturas adicionales en disco.

Asignación indexada

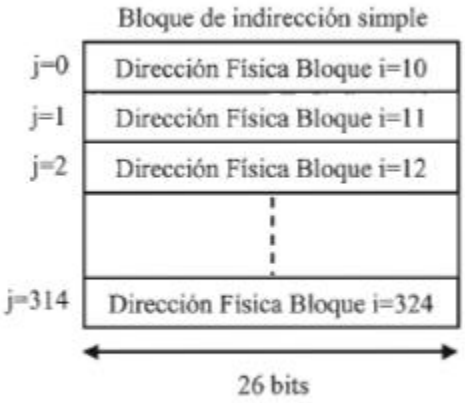

Figura 9.9 – Estructura del bloque de indirección simple del Ejemplo 9.11

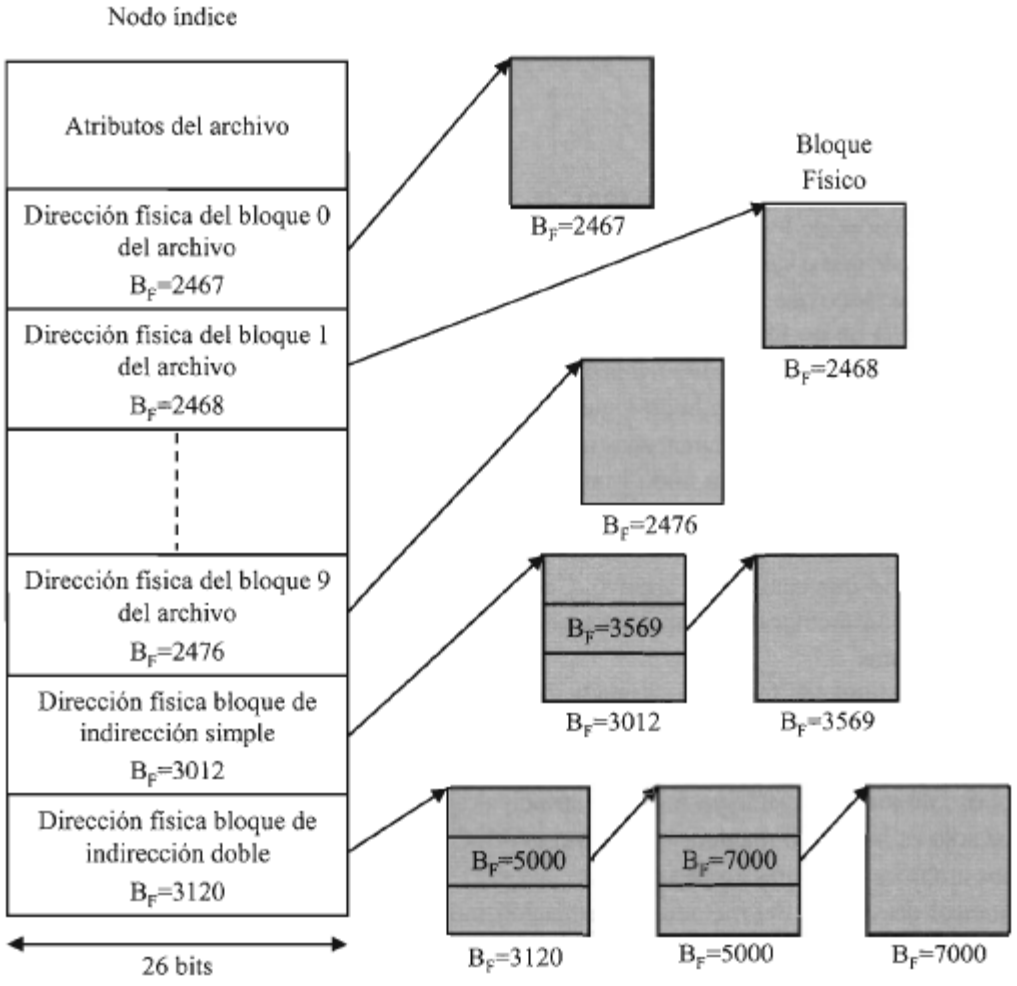

Figura 9.8 – Espacio del disco asignado mediante el método de asignación indexada

Gestión del espacio libre

Mapa de bits

- Cada bloque de disco tiene asignado un bit.
  - Si el bloque está ocupado se marca con un 0 y si está libre con un 1 (o viceversa).
- Si la partición ocupa N bloques, serán necesarios N bits.
- Su principal ventaja es la sencillez para encontrar el primer bloque libre o el primer conjunto de k bloques libres consecutivos.
- El principal inconveniente es que se precisa tener el mapa completo en memoria principal.

Lista enlazada

- Si un bloque de disco puede almacenar N_D direcciones de bloque, en cada bloque de la lista contendrá la dirección del siguiente bloque y N_{D-1} direcciones de bloques libres.
- La principal ventaja es que sólo es necesario mantener en memoria un bloque de la lista enlazada.

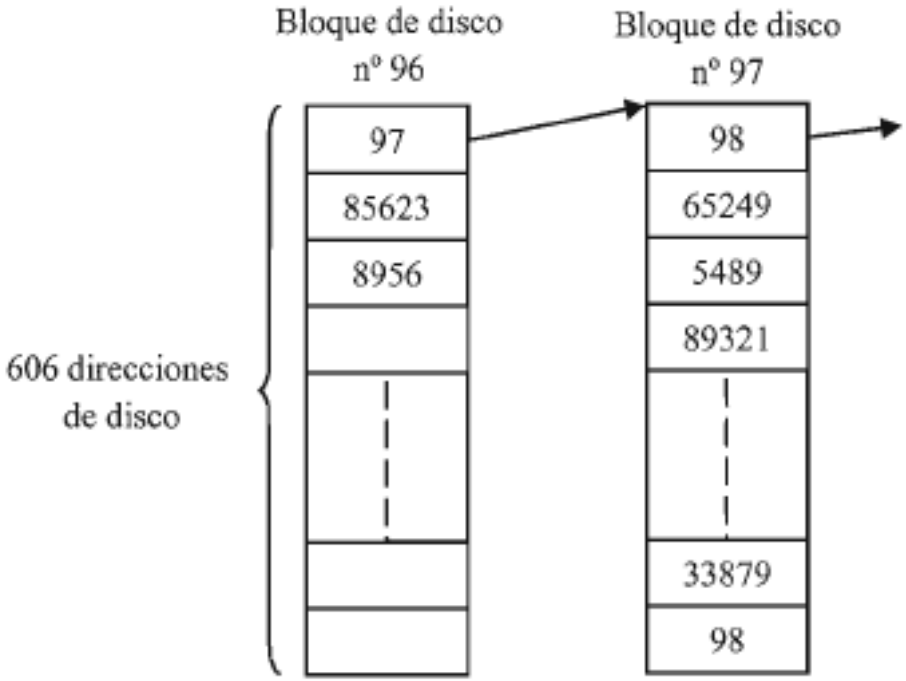

Figura 9.10 – Implementación de la lista de bloques libres mediante una lista enlazada

Implementación de directorios

- Un directorio es un archivo que almacena una lista de los archivos y subdirectorios que contiene.
  - Algunos sistemas, como FAT-32
    - Almacena en cada entrada de un directorio el nombre del archivo o subdirectorio y sus atributos, entre los que se encuentra la información para localizar los bloques de datos que lo contiene.
  - En UFS (UNIX) o ext2 de Linux,
    - Se almacena en cada entrada el nombre del archivo o subdirectorio y un puntero (número de nodo-i) a la estructura de datos (nodo-i) donde se almacenan los atributos del archivo.
    - Esta organización precisa de un acceso más a disco para copiar el nodo-i a memoria principal.

Entradas de Directorios

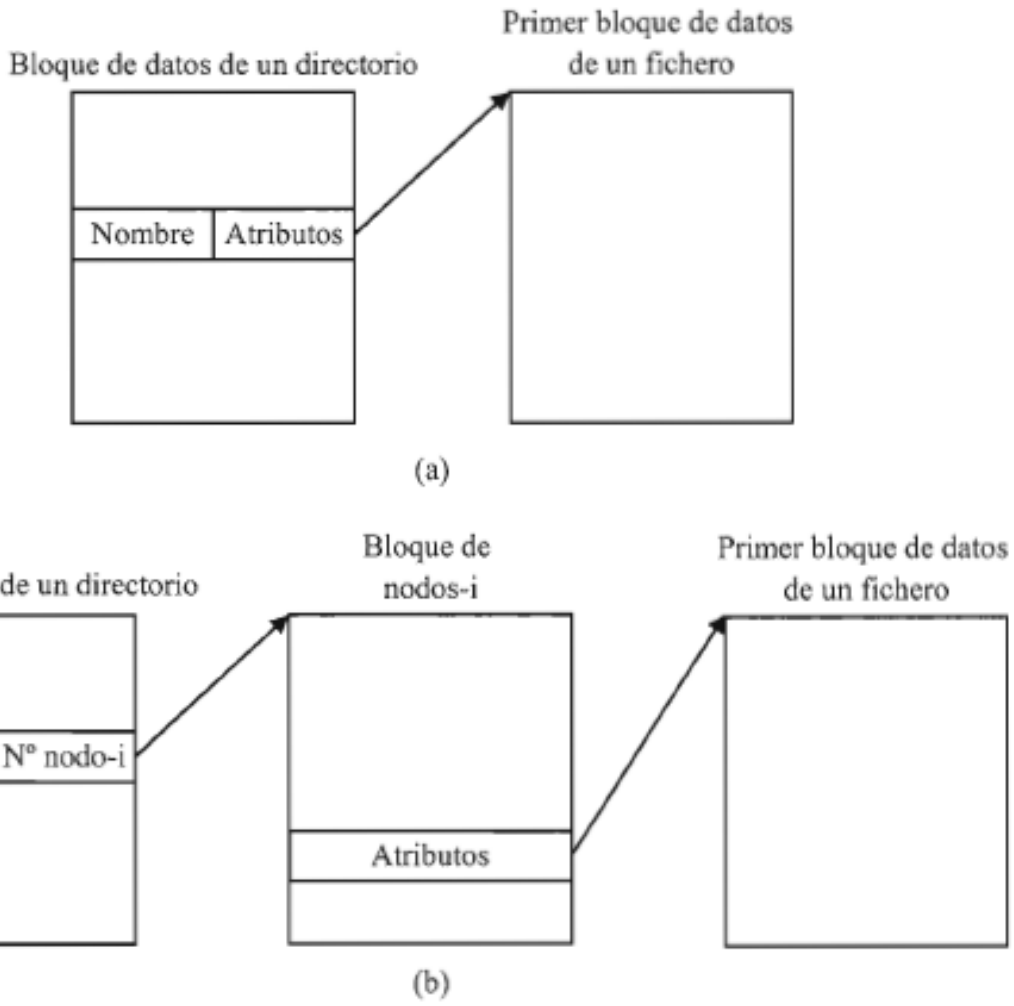

Figura 9.12 – Diferentes contenidos de la entrada de un directorio: a) Nombre y atributos. b) Nombre y número de nodo-i

Implementaciones

- **Directorios con entradas de igual tamaño**
  - En cada entrada se almacenan las informaciones asociadas al archivo y a continuación el nombre del archivo.
    - Es sencilla de gestionar pero para dar soporte de nombres largos cada entrada debe tener ese máximo, lo que supone un derroche de espacio.
- **Directorios con entradas de tamaño variable**
  - Primero se almacena el tamaño que ocupa la entrada, a continuación la información asociada a la entrada (atributos o nodo-i y finalmente el nombre de archivo (hasta un cierto tamaño) al que se le añade un carácter especial para marcar el final del nombre.
  - Cada entrada se rellena para que coincida con un entero positivo de palabras de memoria principal.
  - Como máximo se desperdicia casi una palabra.
  - Cuando se eliminan entradas quedan huecos de longitud variable, generando fragmentación externa que puede precisar de una compactación del directorio.
- **Directorios con entradas de igual tamaño y montículo (heap**)
  - Para almacenar los nombres de los archivos.
    - Se almacena un puntero al comienzo del nombre de archivo dentro del montículo y sus atributos o nodo-i.
  - Elimina la fragmentación externa y no es necesario incluir caracteres de relleno, pero es de administración más complicada al tener que gestionarse el montículo.
- La búsqueda de un archivo en un directorio
  - Se puede mantener **una cache** de directorios con las entradas recientemente accedidas o
  - Implementar cada directorio con **una tabla hash** y su lista de archivos y subdirectorios.

Ejemplos

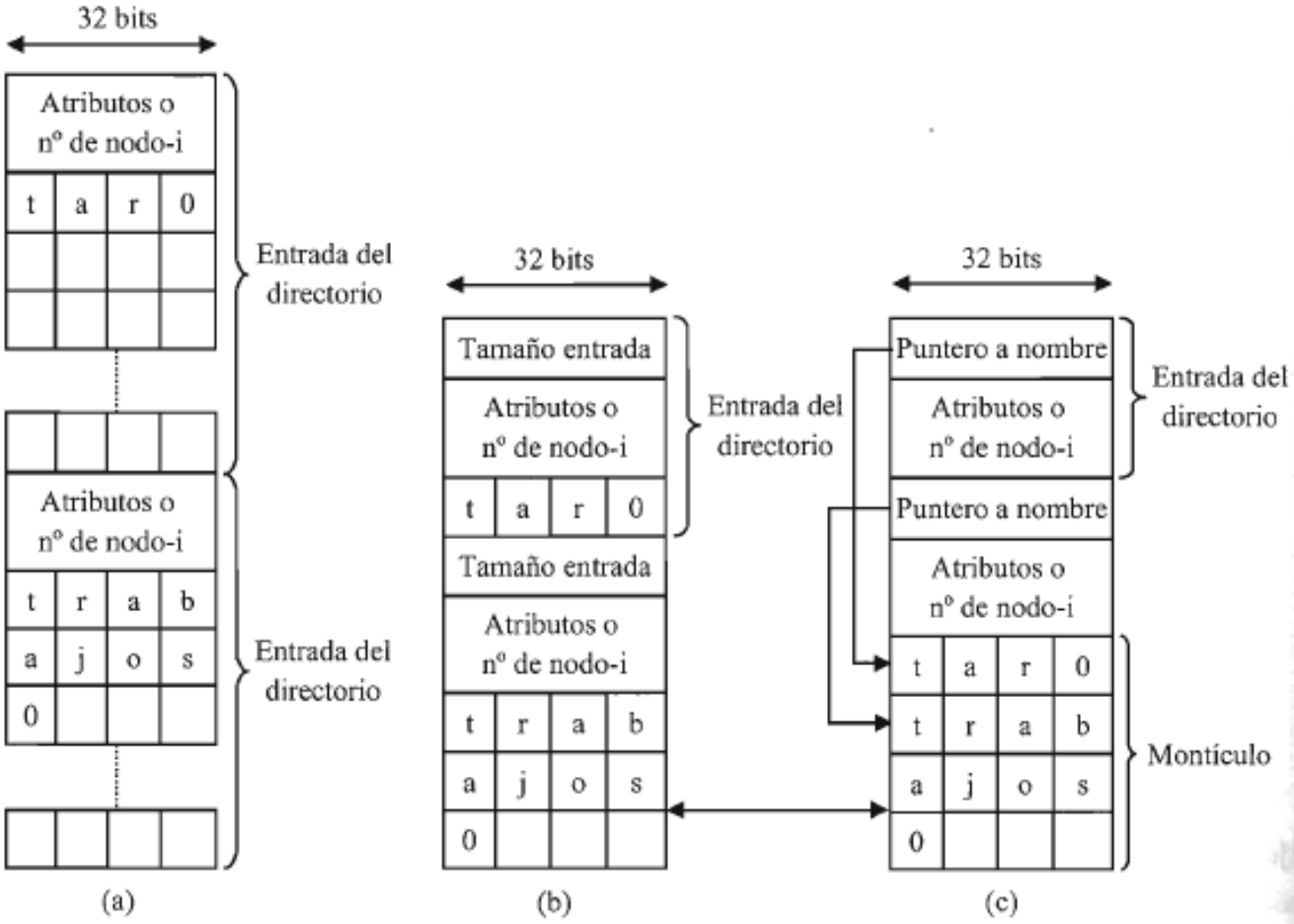

Figura 9.13 – Posibles implementaciones de un directorio: a) Entradas de igual tamaño. b) Entradas de tamaño variable. c) Entradas de igual tamaño y montículo de nombres.

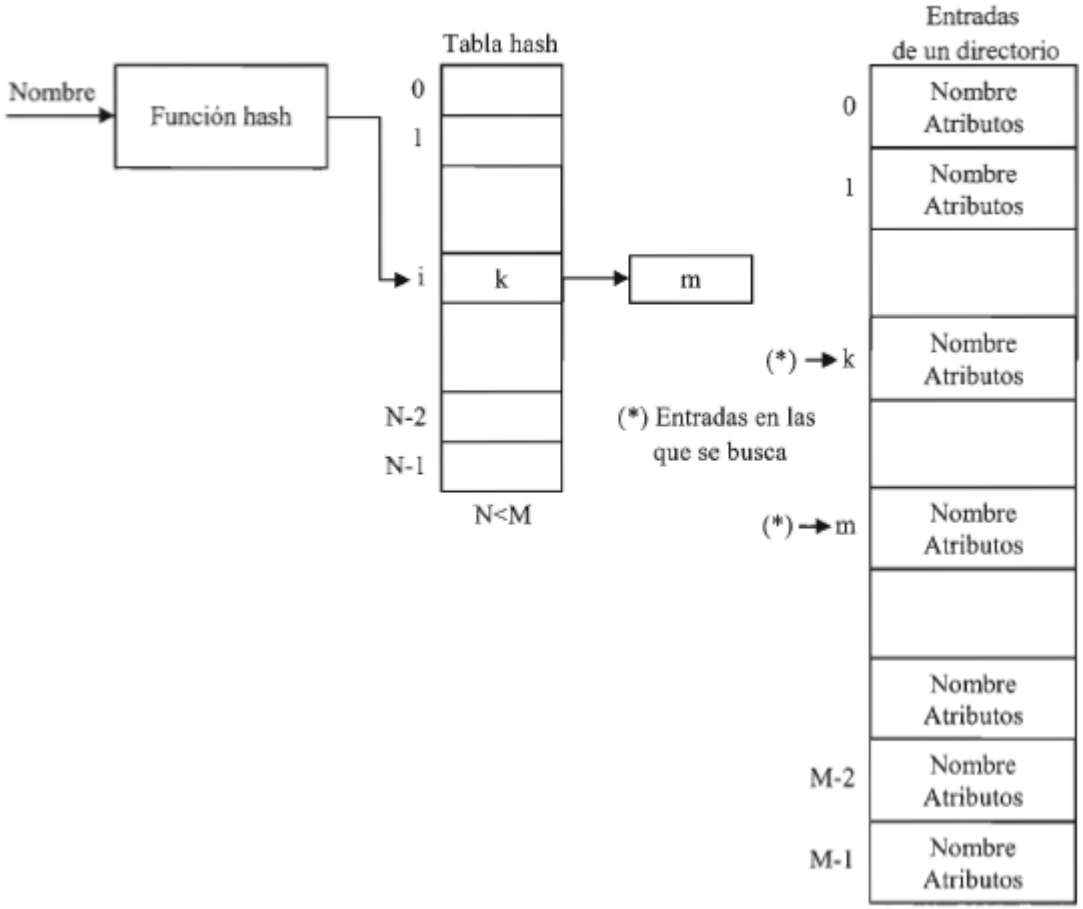

Figura 9.14 – Implementación de un directorio con una tabla hash para acelerar la búsqueda de archivos

Consistencia

- Un bloque no aparece en la lista de bloques libres ni está asignado a ningún archivo.
  - Se soluciona añadiendo el bloque a la lista de bloques libres.
- Un bloque figura en la lista de bloques libres y también está asignado a un archivo.
  - Se elimina el bloque de la lista de bloques libres.
- Un bloque figura varias veces en la lista de bloques libres.
  - Solo aparece si se implementa mediante lista enlazada de bloques libres y se soluciona reconstruyendo la lista.
- Un bloque está asignado a N archivos, siendo N > 1.
  - Es la peor situación y puede solucionarse copiado ese bloque a N - 1 bloques libres.
  - Se recupera la consistencia, pero probablemente después existan N - 1 archivos con errores.
- Para evitar el uso de verificadores los sistemas de archivos modernos (NTFS, ext4, HFS+) implementan la técnica del registro diario o journaling, que consiste en:
  - Almacenar un informe de las operaciones a realizar antes de hacerlas y eliminar el informe una vez realizadas con éxito.
  - Si al reiniciar el sistema el informe sigue existiendo significa que ha existido un error y vuelve a repetirse las operaciones pendientes para dejar el sistema de archivos en un estado Consistente
- Fsck: programa comprobador de consistencia

Ejemplos de consistencia

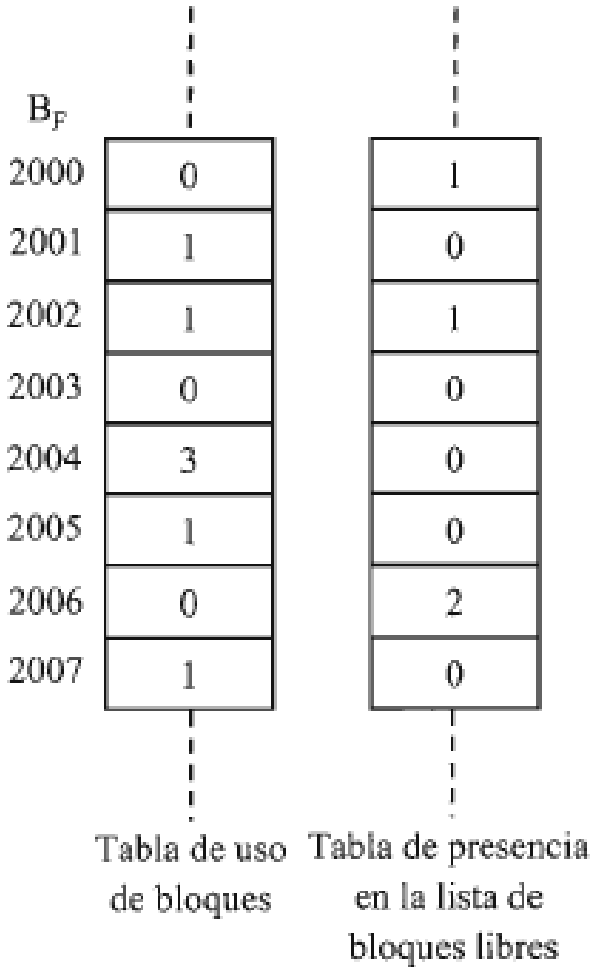

Figura 9.15 – Contenido asociado a los bloques $B_F = 2000$ a $B_F = 2007$ de la tabla de uso de bloques y la tabla de presencia en la lista de bloques libres construidas por `fsck` en el Ejemplo 9.15

Recuperación de archivos

Copias de seguridad

- Copia de seguridad lógica.
  - Únicamente contiene los directorios y archivos que el administrador o el usuario desea copiar al medio de respaldo.
  - Permite acceder a los contenidos de forma individualizada.
  - Estas copias pueden ser de tres tipos:
    - Copia completa,
      - con todos los archivos seleccionados.
    - Copia diferencial,
      - únicamente los archivos nuevos o modificados desde la última completa determinada.
    - Copia incremental,
      - únicamente los archivos creados o modificados desde la última completa o incremental.
- Copia de seguridad física
  - También conocida como imagen de disco.
  - Consiste en copiar bloque a bloque la partición de disco.
  - Son más simples de realizar pero requieren montar la imagen para iniciar la recuperación.

Instantáneas

- ZFS, de Solaris, utiliza la técnica de copiar al escribir, copy-on-write.
- En esta técnica si el sistema operativo precisa modificar el contenido de un bloque en el disco se localiza un bloque libre y se copia el contenido modificado en ese bloque, modificando los punteros para que apunten a ese nuevo bloque.
- El sistema agrupa las modificaciones y las realiza cada cierto tiempo (valor típico 30 segundos).
- Cada modificación se etiqueta con un número de versión diferente.
- La versión actual coexiste con versiones anteriores del sistema de archivos.
- Cada nueva versión se denomina instantánea (snapshot).
- Cada instantánea puede montarse como un archivo de solo lectura para recuperar datos individualmente o en modo de escritura para restaurar el sistema completo a un estado anterior.
- Las instantáneas no son una copia de seguridad.
- Las diferencias fundamentales son:
  - Las instantáneas se toman de forma prácticamente inmediata y no ocupan espacio en el momento de su creación.
    - Sólo se requiere cambiar el número de versión actual.
  - Las instantáneas pueden ser restauradas con gran rapidez.
    - Tan solo es necesario volver a montar el sistema de archivos a partir de la versión anterior.
  - Las instantáneas residen en el mismo disco físico que los datos que respaldan, por lo que no son eficaces frente a daños físicos en el disco y deben complementarse con algún método de copia de seguridad.