

Organización de Datos

Curso Servetto

Sistema de Archivo

El Sistema Operativo es responsable del manejo de recursos de una computadora. Los recursos pueden ser físicos (el hardware) o lógicos (carpetas y archivos). Una de sus funciones es proveer una interfaz controlada entre usuarios y aplicaciones, a nivel lógico, y dispositivos de almacenamiento secundario, a nivel físico: *usuarios y programas tienen una visión lógica de la información almacenada, y los dispositivos de almacenamiento tienen una visión física; el sistema operativo es el mediador.*

El conjunto de programas del sistema operativo encargados de proveer la visión lógica de la información almacenada a usuarios y programas conforman el *Sistema de Archivo*. Este sistema se encarga de proveer:

- *Identificación y localización de archivos:* mediante un sistema jerárquico de directorio o catálogo de archivos. El directorio es un archivo o conjunto de archivos cuyos registros comúnmente identifican tanto subdirectorios o carpetas como archivos con un nombre, y los caracterizan con propiedades como tipo, usuario responsable de su creación, fecha y hora de creación, fecha y hora de última modificación, tamaño en bytes (sólo para archivos), etc. Un subdirectorio o carpeta también puede tener registros que se refieran a un archivo o subdirectorio por el camino de acceso en el directorio general a su registro de especificación. Los registros del primer tipo, de especificación directa, se denominan *hard links*, y los del segundo tipo, de referencia a especificación, *soft links* o *accesos directos*.
 - *Seguridad:* los sistemas operativos que identifican a usuarios generalmente permiten que los mismos establezcan permisos de acceso a sus archivos para otros usuarios o grupos de usuarios, e invariablemente aíslan o encapsulan sus espacios de almacenamiento; en algunos casos permiten a los usuarios encriptar sus archivos, cuando no lo hacen por defecto.
 - *Asignación de espacio en dispositivos de almacenamiento:* para implementar archivos, el Sistema de Archivo debe llevar cuenta de las unidades de asignación que cada uno ocupa, mediante tablas de asignación generales (*FAT: File Allocation Table*) o individuales (*registros índice o inodes de archivos*) así como de las unidades de asignación libres en cada dispositivo de almacenamiento, mediante mapas de bits (un bit por unidad de asignación) o listas de unidades libres. La cantidad de registros físicos (sectores, en discos magnéticos) de las unidades de asignación depende de la capacidad del dispositivo y de la capacidad o tamaño de los campos de direccionamiento de unidades de asignación que emplee el sistema de archivo.
 - *Coordinación de transferencia:* tanto en los sistemas operativos multiusuario como en los sistemas monousuario pero multitarea, el sistema de archivo debe atender solicitudes simultáneas de acceso a un mismo dispositivo de almacenamiento de varios procesos; para optimizar sus servicios debe priorizar la atención de solicitudes de lectura o escritura de registros físicos que se encuentren en las pistas más próximas a la posición actual de los brazos de lectoescritura en el mismo sentido de desplazamiento actual (algoritmo del ascensor).
 - *Coordinación de comunicación entre la CPU y los dispositivos de almacenamiento:* debido a que la CPU es un dispositivo puramente electrónico y los dispositivos de almacenamiento son electromecánicos, sus velocidades de funcionamiento son
-

abismalmente distintas (como la de una liebre respecto a la de una tortuga); cuando un proceso solicita acceder a un dispositivo de almacenamiento, el sistema operativo no puede dejar que la CPU espere a que el dispositivo termine su tarea para continuar con su ejecución; cuando esto sucede el sistema suspende la ejecución del proceso para comprometer a la CPU en la ejecución de otro y así aprovechar su potencialidad al máximo (*la interacción entre la CPU y los dispositivos de almacenamiento es asíncrona*). Otra forma de optimizar la comunicación y administrar el acceso concurrente de procesos a un mismo registro físico, ya que los dispositivos de almacenamiento tienen acceso directo a la memoria RAM (son dispositivos *DMA: Direct Memory Access devices*) es el empleo de data caching o buffering, con bloqueo (denegación de acceso) de buffers.

Casos de Estudio

Unix BSD (*Berkeley Software Distribution*)

Identificación y localización de archivos: los registros de directorio son de longitud variable y sólo contienen el nombre del archivo o subdirectorio y el número del nodo índice (*inode*), un registro de una tabla general del sistema de archivo que lo caracteriza completamente y que además contiene su tabla de asignación de espacio en disco.

Seguridad: el nodo índice de un archivo contiene la identificación del usuario que es su dueño así como la del grupo de usuarios al que este pertenece, más los permisos de acceso para leer, escribir y / o ejecutar el archivo (tres bits para cada permiso) para el dueño, el grupo de usuarios del dueño y para los usuarios en general (nueve bits en total, codificados como tres dígitos octales). Como los directorios no pueden ser ejecutados, los permisos de ejecución para un directorio implican el derecho a buscar archivos en él.

Asignación de espacio en dispositivos de almacenamiento: cada Sistema de Archivo se organiza en unidades de asignación denominadas bloques, agrupados en secciones denominadas *Boot Block*, *Super Block*, *Inode Lists* y *Data Blocks*.

El *Boot Block* (Bloque Calzador) es una unidad de asignación que contiene el módulo de arranque del sistema operativo.

El *Super Block* (Súper Bloque) es otra unidad de asignación que contiene información de estado del sistema de archivo, y que se encuentra replicado convenientemente en distintas ubicaciones del sistema por para asegurar la integridad del mismo; contiene:

- El tamaño del sistema de archivo.
- El tamaño de bloques del sistema.
- Un mapa de bits (*bitmap*) para control de bloques libres.
- Cantidad de nodos índice libres.
- Lista de nodos índice libres.
- Puntero al primer nodo índice libre
- Campos para bloquear los mapas de bloques y la lista de nodos índice libres.
- Un campo para indicar que el súper bloque ha sido modificado.

Una *Inode List* (Lista de Nodos Índice) es un conjunto de unidades de asignación que contiene a los nodos índice de una sección de *Data Blocks*.

Los *Data Blocks* (Bloques de Datos) son las unidades de asignación para almacenar directorios y archivos. El tamaño de las unidades de asignación se puede definir al crear el sistema de archivo, y debe ser una cantidad de sectores que sea potencia de

dos. El tamaño mínimo de bloque es de 4 Kb, y para minimizar la fragmentación interna de los archivos el sistema maneja fragmentos de bloque de tamaño submúltiplo del tamaño general, para asignarlos como bloques finales a los archivos; así, por ejemplo, un archivo de 5215 bytes ocupará un bloque de 4 Kb para sus primeros 4096 bytes, y asumiendo subbloques de 1 Kb, ocuparía dos subbloques para sus 1119 bytes restantes.

Es posible que un bloque tenga fragmentos de distintos archivos, pero un archivo no puede tener fragmentos en distintos bloques. El mapa de bits para control de bloques libres, por tanto, es de fragmentos.

Para optimizar la asignación de espacio y el acceso a los directorios y archivos, la sección de bloques de datos se divide en áreas conformadas por cilindros adyacentes, cada una de las cuales guarda, en posiciones distintas, un Super Block propio con información general del sistema redundante, para aumentar la seguridad. Cada grupo de cilindros tiene también su propia lista de nodos índice, y la política de asignación de espacio procura que los archivos queden localizados en una misma área.

Como los nodos índice de archivos dentro de un mismo directorio normalmente se acceden en forma grupal, el sistema trata de ubicar los nodos índice de archivos de un mismo directorio en una misma área. Los nuevos directorios se ubican en el área que tenga más cantidad de nodos índice libres y la menor cantidad de directorios.

El tamaño de la lista de nodos índice de un área de bloques de datos se establece por defecto: un nodo índice por cada 2 Kb.

La asignación de espacio para cada archivo o directorio se registra en una tabla individual para cada uno, localizada en su correspondiente nodo índice o inode. Los inodes son registros de longitud fija con campos que contienen la siguiente información:

- Identificación del propietario del archivo y del grupo de usuarios al que pertenece.
- Tipo de archivo: regular, directorio, especial de caracteres, FIFO (First In - First Out). Unix tiene tres tipos de archivo o subsistemas de entrada/salida (*I/O Systems*): Block I/O System (archivos normales en disco), character I/O System (terminales, impresoras, etc.) y Network I/O System (archivos FIFO, también llamados pipes o sockets).
- Permisos de acceso (ver *Seguridad* más arriba).
- Fechas de acceso: última modificación, último acceso, última modificación de inode.
- Número de links al archivo (cantidad de registros de directorio que referencian al inode).
- Lista de bloques de almacenamiento del archivo: los números de bloque 0 a 9 de la lista son de los primeros del archivo (direcciones directas), el número del undécimo bloque es el de uno de direcciones de bloques (dirección indirecta simple), el número del duodécimo es el de uno de direcciones de direcciones de bloques (dirección indirecta doble), y el número del décimo tercero es el de uno de direcciones de direcciones de direcciones de bloques (dirección indirecta triple).
- Tamaño del archivo en bytes.

Coordinación de comunicación entre la CPU y los dispositivos de almacenamiento: Unix efectúa la coordinación mediante una técnica denominada Buffer Caché. La cantidad de buffers es un parámetro de configuración del sistema operativo, y el control de los mismos se implementa mediante registros organizados en colas de dispersión (hash queues) doblemente encadenadas cuya cantidad también es configurable. Los registros de control de buffers,

además de estar enlazados para conformar una cola de dispersión, tienen enlaces para conformar una cola de buffers libres. La estructura de los registros de control de buffers es:

- Número de dispositivo.
- Número de bloque.
- Estado: bloqueado (*locked*), datos válidos, se debe escribir el buffer antes de reasignarlo (*delayed write*), se está leyendo o escribiendo actualmente, un proceso está esperando que se libere el buffer.
- Puntero a área de datos.
- Puntero al próximo registro en la cola de dispersión.
- Puntero al registro anterior en la cola de dispersión.
- Puntero al próximo buffer libre.
- Puntero al anterior buffer libre.

Las colas de dispersión y la de buffers libres tienen un registro especial de cabecera y son circulares: el último registro apunta al de cabecera como siguiente, y el de cabecera apunta al último como anterior. Esto permite una política de asignación de buffers LRU (*last recently used*), ya que toda vez que se reasigna un buffer se ubica al final de la cola de dispersión correspondiente al bloque que se asigna.

Para buscar un bloque en las colas de dispersión, se aplica una función de dispersión usando como argumentos el número de dispositivo y el número de bloque, que da como resultado unívoco el número de cola. Por ejemplo, una función posible sería:

$$ncola = (ndisp * nbloque) \bmod ccolas$$

donde *ncola* es el número de cola, *ndisp* el número de dispositivo lógico del sistema de archivo (con numeración a partir de 1), *nbloque* el número de bloque lógico en el dispositivo y *ccolas* la cantidad de colas de dispersión (sin contar la cola de buffers libres).

Los algoritmos para leer y escribir bloques de disco usan el algoritmo *getblk* para asignar buffers a un proceso. Una vez que un buffer ha sido asignado, el sistema operativo puede cargar el bloque correspondiente leyéndolo del disco y asignándolo al buffer para su manipulación, o puede modificar el contenido del buffer (escribir en él). El buffer queda bloqueado de manera que ningún otro proceso pueda acceder a él y modificar sus contenidos mientras esté ocupado; así se preserva la integridad de los datos en el buffer. Cuando el proceso al que se asignó el buffer termina de usarlo, el sistema lo libera de acuerdo al algoritmo *brlse*. Despierta a los procesos que hayan quedado suspendidos por solicitar ese buffer y a los procesos que hayan quedado suspendidos por falta de buffers libres. En ambos casos, la liberación de un buffer significa que queda disponible para uso de los procesos despertados (reactivados), pero el primero que lo tome con *getblk* volverá a bloquearlo impidiendo que el resto pueda acceder. El sistema operativo ubica al buffer al final de la cola de libres, a menos que haya ocurrido un error de entrada / salida o que lo haya marcado con datos válidos luego de una escritura a disco, casos en los cuales lo ubica al comienzo de la cola de libres.

Los escenarios posibles en la asignación de buffers son:

1. *Se encuentra el bloque en la cola de dispersión correspondiente y su buffer está libre y tiene datos válidos:* se desencadena el bloque de la lista de libres y se cambia el estado del buffer a bloqueado. Si otro proceso intentare acceder al bloque mientras su buffer se encuentre bloqueado, se debería suspender hasta que el buffer se libere.

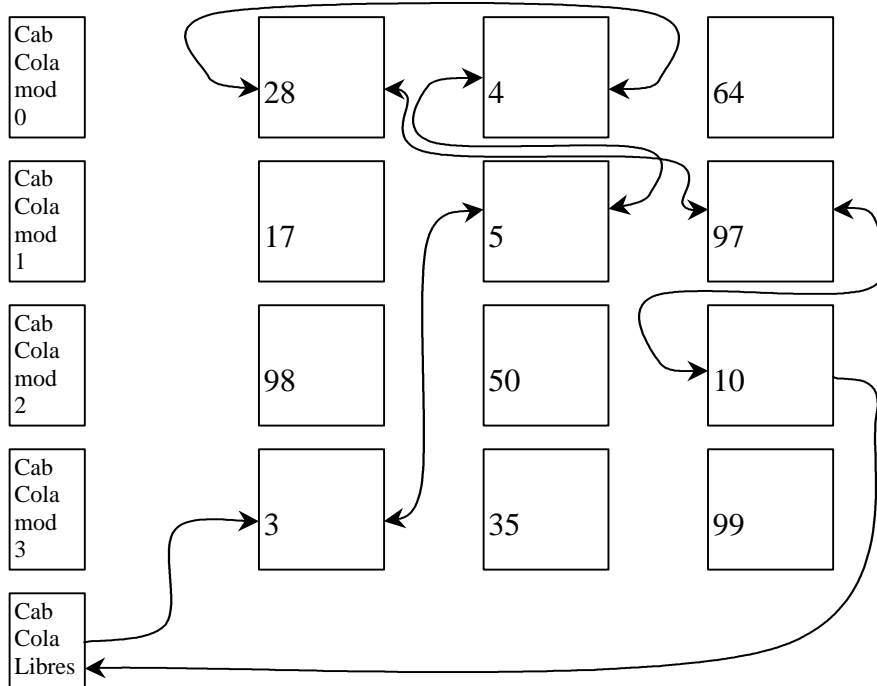
2. *No se encuentra el bloque en su cola de dispersión correspondiente y al buscar un buffer libre en la cola de libres se encuentra al primero con datos válidos:* se desencadena el primero de la cola de libres y se encadena al final de la cola de dispersión que corresponda al bloque a leer, cambiando su estado a bloqueado y fijando los números de dispositivo y de bloque correspondientes.
3. *No se encuentra el bloque en su cola de dispersión correspondiente y al buscar un buffer libre en la cola de libres se encuentran bloques con escritura diferida:* se debe desencadenar los buffers correspondientes de la lista de libres, cambiarles el estado para indicar que se están escribiendo, iniciar una escritura asincrónica de los bloques correspondientes y asignar el primer buffer con datos válidos de la cola como en el escenario anterior. Cuando se termina la escritura asincrónica de un bloque, su buffer se vuelve a ubicar primero en la cola de libres.
4. *No se encuentra el bloque en su cola de dispersión correspondiente y la cola de buffers libres está vacía:* se debe suspender el proceso a la espera de que algún otro libere un buffer.
5. *Se encuentra el bloque en la cola de dispersión correspondiente y su buffer está bloqueado:* se cambia el estado del buffer indicando que un proceso está esperando que se libere y se suspende el proceso a la espera de que el proceso que bloquea el buffer lo libere.

A continuación se grafican los escenarios descritos anteriormente (se asume que todos los bloques corresponden a un número de dispositivo lógico 1).

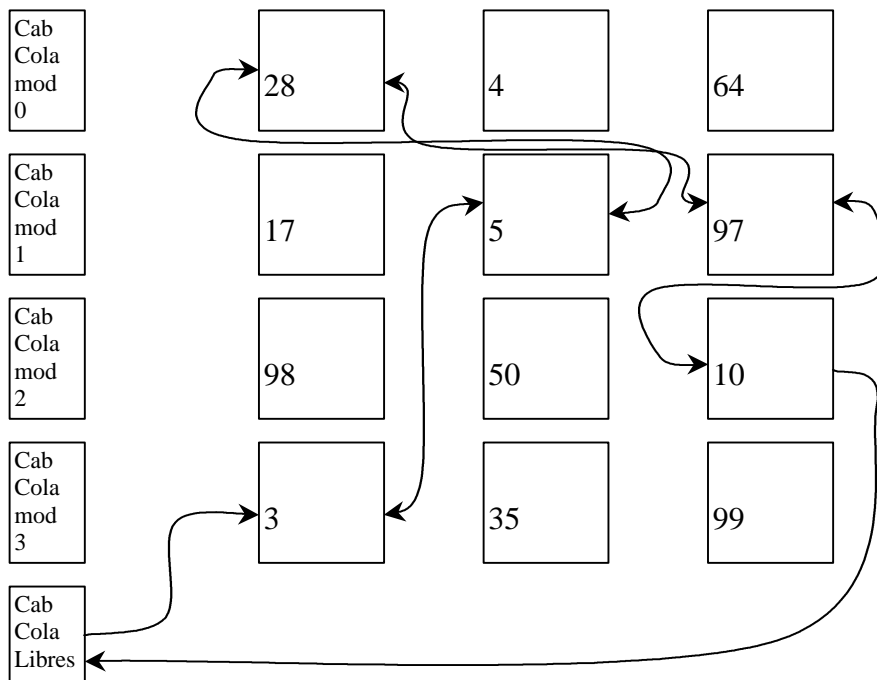
Escenario 1

Registros Cabecera de colas de dispersión y de cola de buffers libres

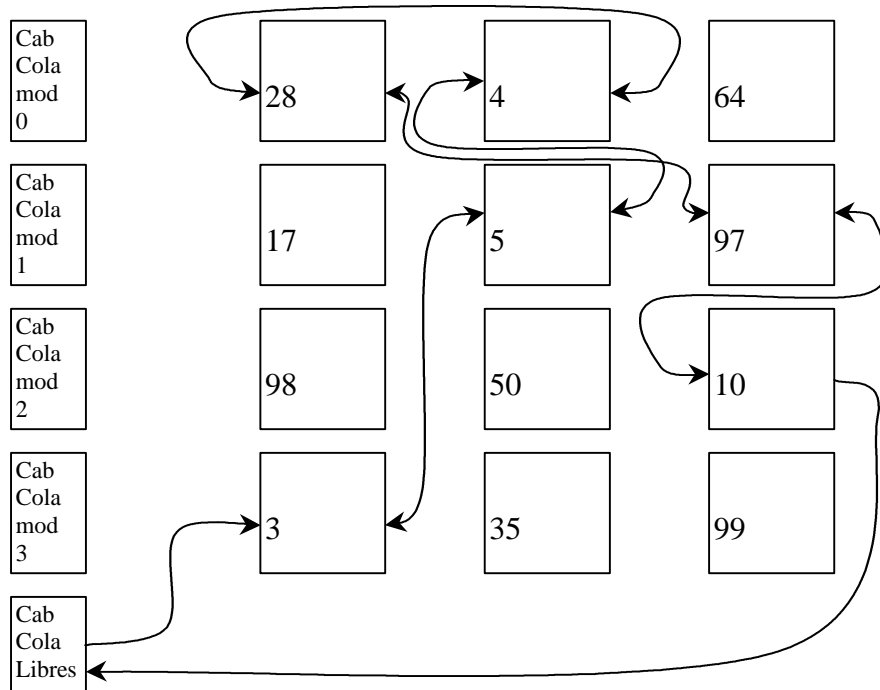
Registros de control de buffers en colas de dispersión (los registros en cada fila corresponden a la cola de dispersión correspondiente y están doblemente encadenados – sólo se denota el encadenamiento de registros correspondientes a buffers libres)



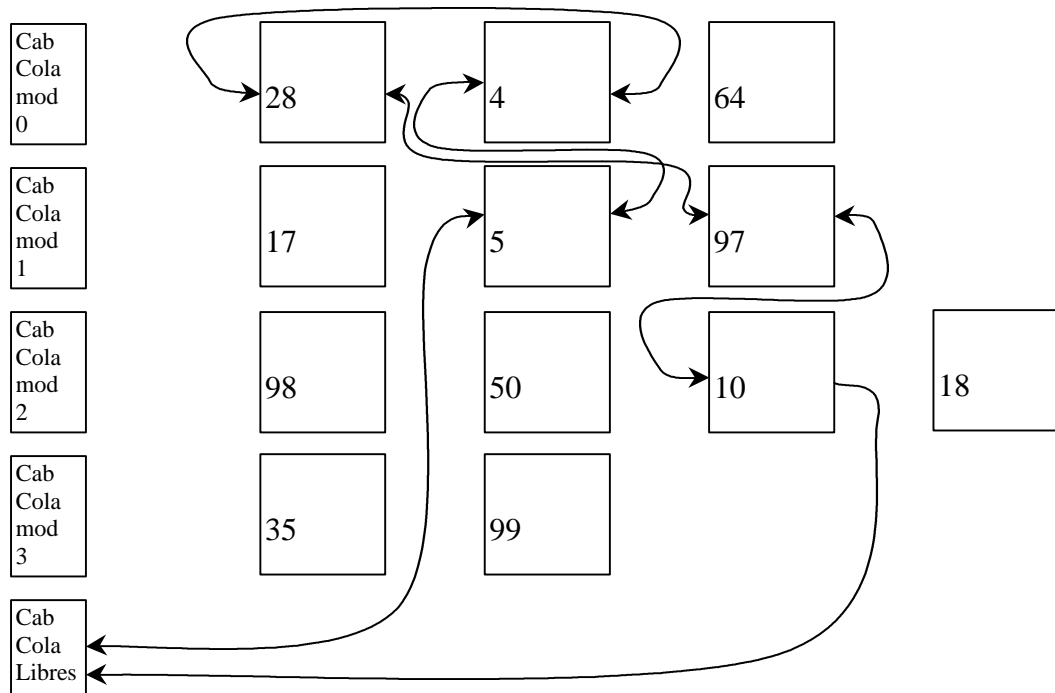
Búsqueda del bloque 4:



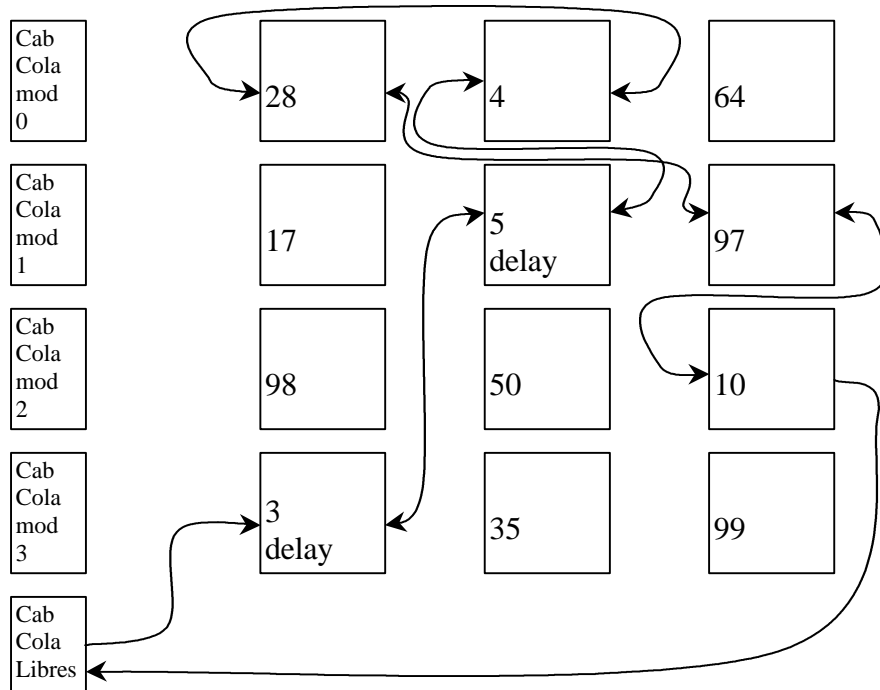
Escenario 2



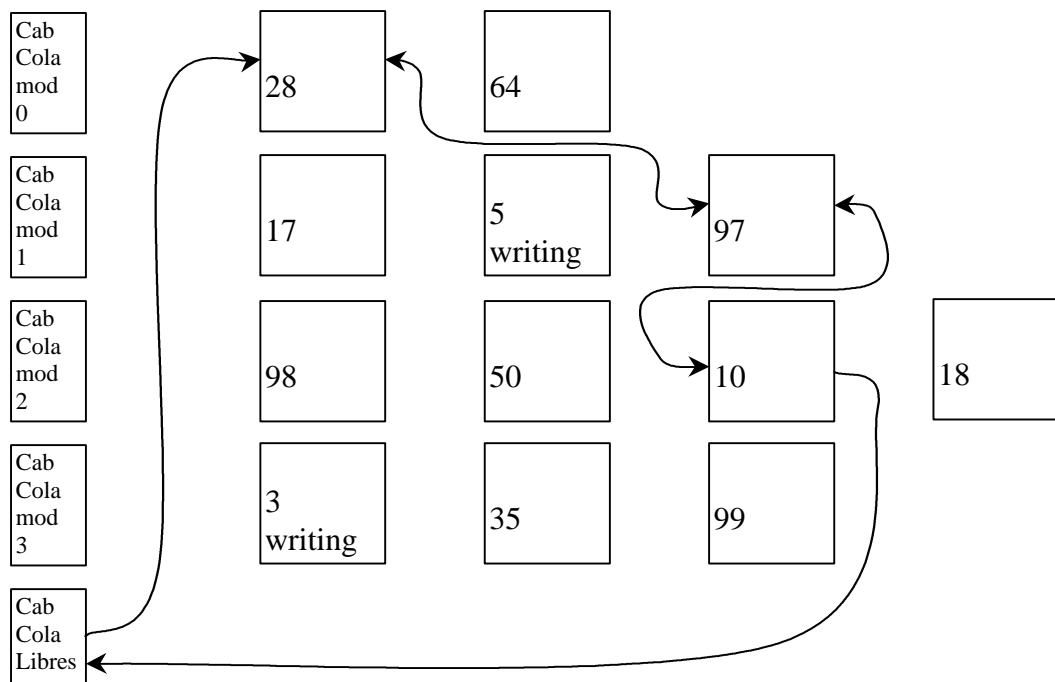
Búsqueda del bloque 18:



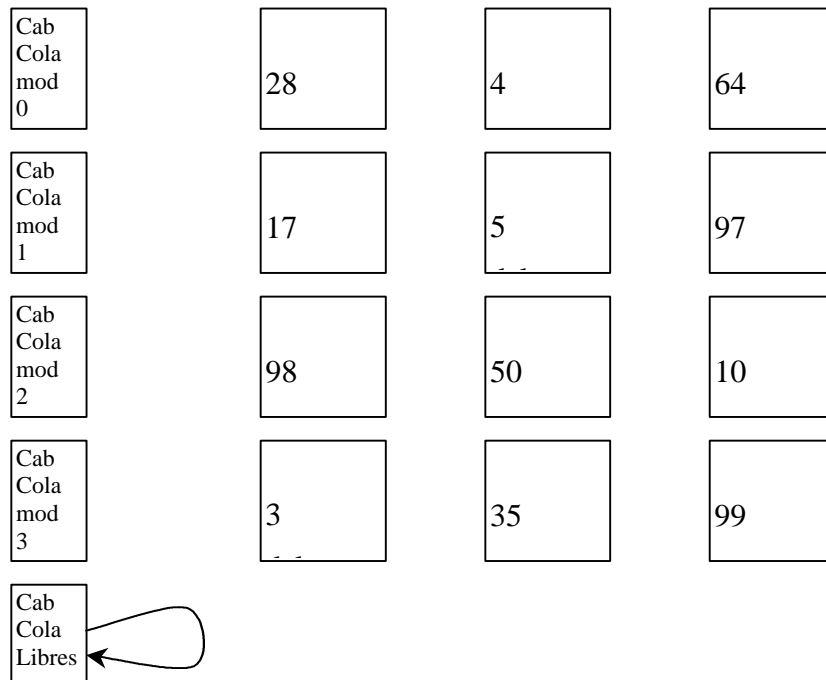
Escenario 3



Búsqueda del bloque 18:

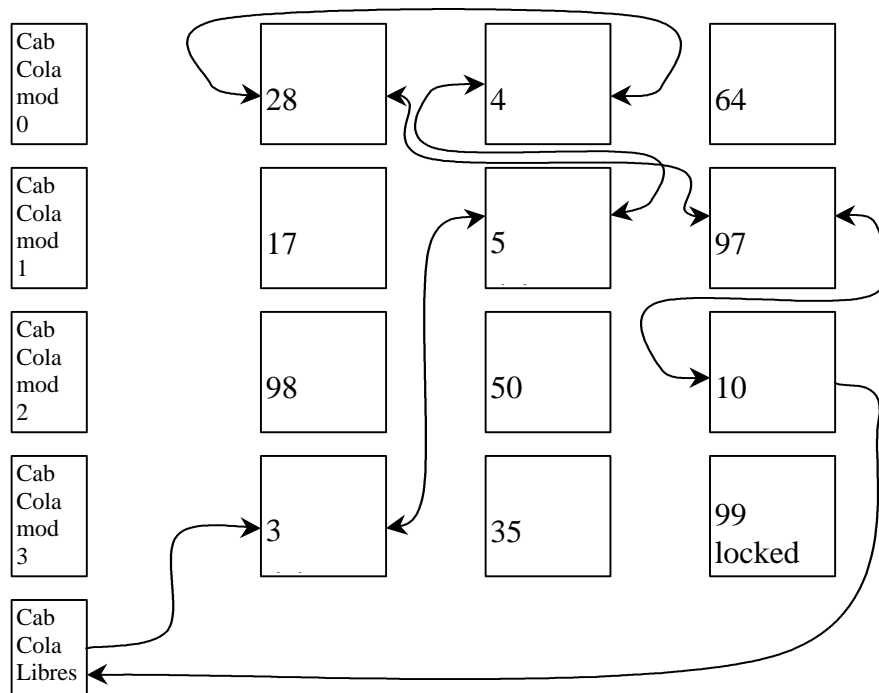


Escenario 4



Búsqueda del bloque 18

Escenario 5



Búsqueda del bloque 99

Manipulación de Archivos

Cuando un proceso se refiere a un archivo por su ruta de acceso y nombre (*pathname*), el sistema operativo chequea que el proceso tenga permiso para buscar (permiso de ejecución) en los directorios de la ruta de acceso (heredado del usuario responsable), y eventualmente

recupera el nodo índice del archivo. Cuando un proceso crea un nuevo archivo, el sistema operativo le asigna un nodo índice libre. Para la manipulación de un archivo, el sistema operativo debe tener a su nodo índice cargado en la memoria principal.

El sistema operativo contiene otras dos estructuras de datos: la tabla de archivos abiertos (*open file table*) y la tabla descriptora de archivos de usuario (*user file descriptor table*). La tabla de archivos abiertos es una estructura global, mientras que la descriptora de archivos de usuario es propia de cada proceso. Cuando un proceso abre o crea un archivo, el sistema le asigna una entrada en cada tabla, correspondiente al nodo índice del archivo. Las entradas en las tres estructuras mantienen el estado del archivo y los accesos del usuario a él. La tabla de archivos abiertos lleva cuenta del desplazamiento en bytes en el archivo donde comenzará la próxima lectura o escritura, y los derechos de acceso permitidos al proceso de apertura. La tabla descriptora de archivos de usuario identifica a todos los archivos abiertos por un proceso. El sistema operativo devuelve un descriptor de archivo (*file descriptor*) en las llamadas al sistema (*system calls*) *open* y *creat*, que es un índice en la tabla descriptora de archivos de usuario. Cuando ejecuta las llamadas *read* y *write*, el sistema operativo usa el descriptor de archivo para acceder a la tabla descriptora de archivos de usuario, sigue los punteros a las entradas a la tabla de archivos abiertos y a la tabla de nodos índice, y, desde el nodo índice, busca los datos en el archivo.

La copia en memoria de un nodo índice contiene los siguientes campos además de los campos de registro en disco:

- El estado del inode en memoria, indicando si
 - el inode está bloqueado,
 - un proceso está esperando que se libere el inode,
 - la copia en memoria del inode difiere de la copia en disco como resultado de un cambio de los datos en el inode,
 - la copia en memoria del inode difiere de la copia en disco como resultado de un cambio de los datos en el archivo de datos,
 - el archivo es un punto de montaje (conexión con otro dispositivo de almacenamiento o periférico).
- El número de dispositivo lógico del sistema de archivo que contiene al archivo.
- El número de inode. Como los inodes se almacenan en disco como un arreglo lineal, el sistema operativo los identifica en el disco por su posición en el arreglo. El inode en disco no necesita este campo.
- Punteros a otros inodes en memoria. El sistema operativo encadena los inodes en memoria en colas de dispersión y en una lista de buffers (de inodes) libres, igual que para los buffers de bloques. Las colas de dispersión se identifican en función del número de dispositivo lógico del inode y del número de inode en el dispositivo. El sistema puede tener a lo sumo una copia en memoria de un inode de un disco, pero los inodes pueden estar simultáneamente en una cola de dispersión y en la lista de libres, tal cual puede suceder con los buffers de bloques.
- Un contador de referencias, indicando el número de instancias del archivo correspondiente que están activas.

DOS (Disk Operating System)*

Identificación y localización de archivos: los directorios son archivos cuyos registros o “entradas” contienen el nombre, fecha y hora de creación o última actualización, y localización del archivo en el disco:

| <i>Byte</i> | <i>Propósito</i> |
|-------------|--|
| 0 – 7 | Nombre del archivo. El primer byte puede también indicar el estado: 00H El archivo nunca ha sido usado 05H EL primer carácter del nombre del archivo es E5H 2EH Subdirectorio E5H El archivo ha sido borrado |
| 8 - 10 | Extensión del nombre del archivo |
| 11 | Atributo del archivo que define su tipo. Un archivo puede tener más de un atributo, tal como 07H para un archivo de sistema de sólo lectura y oculto. 00H Normal 01H Sólo lectura 02H Oculto 04H De sistema 08H Etiqueta de volumen en los primeros 11 bytes 10H Subdirectorio 20H Modificado (indica si fue reescrito desde la última actualización) |
| 12 - 21 | Reservados |
| 22 - 23 | Hora del día en que fue creado o actualizado por última vez, en formato binario como hhhhhmmmmsssss. |
| 24 – 25 | Fecha en que fue creado o actualizado por última vez, en formato binario como aaaaaammmmdddd. El año comienza en 1980 y puede ser 0 aa 119, el mes puede ser 1 a 12, y el día 1 a 31. |
| 26 – 27 | Unidad de asignación inicial del archivo, desde 002 (en las unidades 00 y 01 se ubican indefectiblemente archivos del sistema). |
| 28 – 31 | Tamaño del archivo en bytes. |

Seguridad: el único mecanismo de seguridad asociado a los archivos es el control de sus atributos, que permite ocultarlos en las búsquedas en directorios.

Asignación de espacio en dispositivos de almacenamiento: la asignación de espacio de almacenamiento para archivos y directorios se hace por unidades de asignación o *clusters*, que dependiendo de la capacidad del disco y del tamaño de las direcciones de clusters pueden ser de uno, dos, cuatro, ocho, ... sectores (siempre una potencia de dos). Todo archivo requiere un mínimo de una unidad de asignación.

El primer sector del disco es el *boot record* o registro de calce del sistema operativo, que contiene instrucciones para ayudar al sistema a cargar del disco a memoria programas para

* De este sistema operativo hay versiones de IBM, MicroSoft y Digital Research.

controlar la entrada / salida de datos (BIOS: Basic Input / Output System) y para interpretar comandos (COMMAND), que indefectiblemente se ubican en las primeras unidades de asignación para archivos de datos. También contiene el nombre del fabricante (MicroSoft, Digital Research, IBM, etc.) y el número de versión del sistema y parámetros para el sistema básico de entrada / salida como bytes por sector, sectores por cluster, sectores reservados, sectores por pista y número de cabezas de lectura / escritura.

A continuación del *boot record*, a partir del sector 2, comienza la *FAT (File Allocation Table)*, que el sistema utiliza para asignar espacio en disco a los archivos y directorios. La FAT contiene una entrada por cada cluster (el uso de clusters con múltiples sectores reduce el número de entradas en la FAT y habilita al sistema a direccionar espacios en disco mayores). La FAT puede ser de entradas de 16 bits, o de entradas de 32 bits, según la versión del sistema operativo y / o la capacidad del disco. El registro de directorio de un archivo o directorio provee el cluster donde comienza, y cada entrada de la FAT indica el número relativo del siguiente cluster. El valor 0000H (asumiendo que la FAT tiene entradas de 16 bits) indica que el cluster está libre, el valor FFF7H indica que el cluster está dañado, y FFF8-FFFFH indican fin de archivo.

Luego de la FAT puede haber un duplicado de la misma, si el disco es de arranque del sistema, y a continuación una serie de sectores para el directorio raíz del sistema. El directorio raíz, por consiguiente, es el único con capacidad limitada, ya que el resto de los directorios se manejan como cualquier archivo.

Microsoft® NTFS (*New Technology File System*)*

Identificación y localización de archivos: los directorios no se implementan como archivos, sino como registros especiales de una tabla maestra del sistema, la MFT (*Master File Table*) que contiene una entrada para cada archivo y directorio del disco. El registro en la MFT de un directorio contiene una lista ordenada con los nombres e información estándar de cada uno de los archivos que contiene, de modo que para mostrar el contenido de un directorio sólo hay que acceder al registro del directorio en la MFT y no a los registros de cada archivo.

Seguridad: los registros de especificación de archivos de la MFT tienen información de control de acceso y auditoría sobre cada archivo, estructurada en un descriptor de seguridad con la siguiente estructura:

| |
|--|
| SID (System IDentifier) del propietario |
| SID del grupo |
| DACL (Discretionary Access Control List) |
| SACL (System Access Control List) |

- La DACL indica quién tiene permiso para hacer qué sobre el archivo.
- La SACL indica qué eventos quedarán registrados en un archivo de registro (log) de eventos de seguridad.

Las entradas de ambas listas tienen la forma

| |
|---|
| SID |
| Mask (GENERIC_EXECUTE, GENERIC_READ, GENERIC_WRITE, DELETE, WRITE_DAC, WRITE_OWNER) |

* Microsoft incluye este sistema de archivo en sus sistemas operativos NT, 2000, XP, etc.

| |
|--|
| Tipo de entrada [Acceso no permitido Acceso permitido Auditoría] |
| Bitfield de herencia |

Asignación de espacio en dispositivos de almacenamiento: los registros de especificación de archivos de la MFT tienen un mapa de asignación del archivo, con formato de lista de fragmentos. Los fragmentos son clusters contiguos del disco especificados por tres valores cada uno:

- VCN (*virtual cluster number*): Número relativo del primer cluster del fragmento en el archivo.
- LCN (*logical cluster number*): Número real del primer cluster del fragmento en el disco.
- Cantidad de clusters contiguos en disco del fragmento a partir del LCN.

Las direcciones de cluster son de 64 bits, que permiten manejar discos con mucha capacidad.

Si al formatear un disco no se especifica el tamaño de cluster, se toma una cantidad de sectores por defecto que depende del tamaño de la partición o del disco, según la siguiente tabla:

| Tamaño de partición o disco | Tamaño de Cluster |
|-----------------------------|-------------------|
| <=512Mb | 512 bytes |
| 513Mb a 1024Mb (1GB) | 1Kb |
| 1025Mb a 2048Mb (2GB) | 2Kb |
| >=2049Mb | 4Kb |

Al inicializar una partición NTFS se crean once archivos de metadatos que se usan para la administración del sistema:

| Archivo | Registro en la MFT | Descripción |
|-----------|--------------------|---|
| \$MFT | 0 | Master File Table. |
| \$MFTMIRR | 1 | Copia de los primeros 16 registros de la MFT. |
| \$LOGFILE | 2 | Archivo de log transaccional. Si el sistema falla, es posible restaurar un estado consistente del mismo a partir del log minimizando la pérdida de datos. |
| \$VOLUME | 3 | Numero de serie del volumen, y fecha y hora de creación. |
| \$ATTRDEF | 4 | Definiciones de atributos. |
| . | 5 | Directorio raíz del disco. |
| \$BITMAP | 6 | Mapa de clusters de la unidad (0: cluster libre, 1: en uso). |
| \$BOOT | 7 | Boot record de la unidad. |
| \$BADCLUS | 8 | Lista de clusters dañados. |
| \$QUOTA | 9 | Contiene, a partir de NT 5.0, información sobre el espacio máximo de disco que puede usar cada usuario. |
| \$UPCASE | 10 | Mapa de caracteres en mayúscula en función de sus correspondientes en minúscula. |

NT no muestra estos archivos pero pueden verse usando `dir /ah <nombre_arch>`

La MFT está organizada en registros (generalmente de 1Kb pero en discos grandes pueden ser de tamaño mayor) y la descripción de un archivo o directorio puede ocupar uno o mas registros. En estos registros se almacenan los atributos del archivo o directorio y su ubicación en el disco. Como la MFT se maneja como cualquier archivo, al ir creciendo y achicándose, se fragmenta, lo cual produce una baja en la velocidad de acceso, alargando los tiempos de acceso al file system. La solución propuesta para esto es predestinar una cantidad fija de clusters al inicio de la unidad para la MFT (Zona MFT) que queden sin asignar a ningún archivo ni directorio en tanto se cuente con otros clusters vacíos en disco. Cuando el disco se va llenando, se empiezan a usar esos clusters y aumenta la posibilidad de que la MFT se fragmente, por eso el acceso a disco se puede hacer mas lento. NTFS no permite que las herramientas de defragmentación actúen sobre la MFT.

Internamente NTFS identifica los directorios y archivos por el número de registro de la MFT donde se encuentra su registro inicial. Los archivos de metadatos tiene un número de registro base predeterminado (los registros para un determinado archivo no necesitan estar contiguos).

Los registros de la MFT se componen de un encabezado (*header*), y de un conjunto de atributos específicos según el tipo de archivo, delimitados por un encabezado identificador de cada atributo. Los valores de los atributos pueden residir directamente en el registro, como los de información estándar y de seguridad, o estar referenciados en el registro, como los datos del archivo.

El encabezado de un registro contiene:

- Nro. de secuencia usado para verificación de integridad
- Un puntero al primer atributo en el registro
- Un puntero al primer byte de espacio libre en el registro
- Si no es el registro base del archivo en la MFT, el número del registro base.

Hay 14 tipos de atributo:

| | |
|------------------------|---|
| \$VOLUME_VERSION | Versión del disco |
| \$VOLUME_NAME | Nombre del disco |
| \$VOLUME_INFORMATION | versión de NTFS |
| \$FILE_NAME | Nombre del directorio o archivo |
| \$STANDARD_INFORMATION | Fecha y hora de creación, modificación, oculto, de sistema, de sólo lectura |
| \$SECURITY_DESCRIPTOR | Información de seguridad |
| \$DATA | Datos del archivo |
| \$INDEX_ROOT | Contenido del directorio |
| \$INDEX_ALLOCATION | Contenido del directorio |
| \$BITMAP | Bitmap del contenido del directorio |
| \$ATTRIBUTE_LIST | Headers de atributos no residentes. |
| \$SYMBOLIC_LINK | Path Acceso Directo |
| \$EA_INFORMATION | Atributos de compatibilidad OS/2 |
| \$EA | Atributos de compatibilidad OS/2 |

Recuperación de fallas mediante el uso de logs

Cada unidad o partición NTFS tiene un archivo de log con registros de dos tipos: REDO y UNDO. Los registros de tipo UNDO se usan para acciones, por ejemplo, de incorporación o modificación de datos en un archivo; si hay una falla y la operación no se completó correctamente, hay que deshacer todas las actualizaciones que se hicieron. Los registros de tipo REDO se usan, por ejemplo, para la eliminación de archivos.

Las actualizaciones se hacen en memoria y se va actualizando el log. Cada cinco segundos se produce un checkpoint en el cual se registran los cambios de todas las estructuras en disco y si se completan con éxito, se eliminan los registros correspondientes del log evitando que crezca desmesuradamente.

IBM OS/2 HPFS (*High Performance File System*)

Identificación y localización de archivos: los directorios están formados por uno a varios bloques de directorio de 2 kb (de 4 sectores contiguos). Siempre que sea posible, los bloques de directorio se asignan en un área del disco exclusiva para los directorios, denominada Banda de Bloques de Directorio (es una sección especial de sectores contiguos de 8 Mb que se encuentra cerca del centro del disco), pero cuando se encuentre llena, se asignan donde se encuentre lugar.

Una entrada de directorio se compone de los siguientes campos:

- Longitud de la entrada
- Fecha y Hora de creación y modificación
- Longitud del nombre del archivo o subdirectorio
- Nombre del archivo o subdirectorio
- Puntero al fnode (registro de especificación del archivo)
- Puntero a bloque de directorio sucesor (para estructuración del directorio como árbol B)

Las entradas de directorio están ordenadas alfabéticamente.

Si un directorio es demasiado largo para ser descrito dentro de un bloque, se le asignan más bloques y se organiza como un árbol B.

Seguridad: los registros de especificación de archivos y directorios (*fnodes*) contienen listas de control de acceso al archivo.

Asignación de espacio en dispositivos de almacenamiento: un archivo o directorio se identifica por una estructura llamada fnode (*file node*), que ocupa un sector del disco y se guarda próximo al archivo o directorio que describe y tiene la siguiente estructura:

- Longitud del nombre del archivo
- Primeros 15 caracteres del nombre del archivo
- Información de control
- Historia de accesos
- Atributos extendidos
- Listas de Control de Acceso
- Punteros al archivo o puntero al bloque de directorio raíz

Un fnode puede almacenar hasta ocho punteros a fragmentos del archivo de hasta 16Mb cada uno. Cada puntero emplea treinta y dos bits para la dirección del sector inicial de cada

fragmento, y otros treinta y dos bits para especificar la cantidad de sectores contiguos del fragmento. Los archivos cuyo tamaño y fragmentación lo permitan, son descritos completamente dentro del fnode. Aquellos que sean más grandes o estén muy fragmentados, se representan utilizando una estructura de árbol: en lugar de tener punteros a fragmentos, en el fnode se tienen punteros a otros nodos denominados alnodes, con capacidad para cuarenta punteros a fragmento.

Todo volumen HPFS se estructura de la siguiente manera:

- *Bloque de Carga (Boot block)*: sectores 0 a 15. Contiene la identificación del volumen y el programa de carga del sistema operativo (*bootstrap*).
- *Súper Bloque*: sector 16. Contiene:
 - Versión de HPFS
 - Puntero al fnode del directorio raíz
 - Cantidad de sectores del volumen
 - Cantidad de sectores dañados
 - Puntero a los bitmaps de espacio libre
 - Puntero a la lista de bloques dañados
 - Puntero a la Banda de Bloques de Directorio
 - Fecha de la última vez que se ejecutó un chequeo (CHKDSK)
- *Bloque de Reserva (Spare Block)*: sector 17. Contiene campos indicadores y otros punteros, como por ejemplo, uno a un conjunto de bloques libres de reserva por si se agota la banda de bloques de directorio.
- *Bandas*: el resto del disco se divide en bandas de 8Mb. Cada una tiene un mapa de bits de espacio libre que se ubica alternativamente al inicio o al final de la banda. Si una banda tiene su *bitmap* al principio, la siguiente lo tendrá al final. De esta forma se puede disponer de un espacio máximo de 16Mb contiguos para asignar a archivos. El tamaño de la banda es una decisión de implementación. Una de las bandas es la de bloques de directorio.

Índice

| | |
|--|----|
| Sistema de Archivo..... | 1 |
| Casos de Estudio..... | 2 |
| Unix BSD (<i>Berkeley Software Distribution</i>)..... | 2 |
| DOS (Disk Operating System) | 11 |
| MicroSoft® NTFS (<i>New Technology File System</i>)..... | 12 |
| IBM OS/2 HPFS (<i>High Performance File System</i>)..... | 15 |
| Índice | 16 |