

NOCIONES SOBRE DISCOS DUROS

MBR, particiones, volúmenes, formatos, son conceptos que tendremos que tener claro si pretendemos gestionar adecuadamente nuestros discos duros.

La parte dura

Un disco duro es básicamente un conjunto de platos recubiertos con un material sensible a alteraciones magnéticas que giran a gran velocidad (cuya misión es la de almacenar los datos) y unos cabezales móviles de gran precisión (los encargados de leer y escribir en los platos magnéticos). Los dos estándares más utilizados son IDE y SCSI.

La secuencia de inicio

Una vez que la ROM ha detectado el disco duro principal lee el primer sector que se encuentra en éste, dicho sector es el Master Boot Record (MBR, literalmente registro maestro de arranque), que contiene un programa ejecutable y una tabla donde están definidas las particiones del disco. El ejecutable del MBR llama a su vez a otro programa contenido en el primer sector de la partición primaria activa que es el encargado de cargar el sistema operativo en memoria.

Particiones primarias y extendidas

Antes de comenzar a trabajar con un disco duro creamos en el una serie de particiones que contendrán los datos. Existen dos tipos de particiones: la **partición primaria** y la **partición extendida**.

La **partición primaria** es la única desde la que puede arrancar el sistema operativo, aquí es donde el MBR buscará el programa que controla el arranque (**boot manager**). En Dos podemos definir por cada disco hasta cuatro particiones primarias pero sólo una puede estar activa. Gracias a esto podrían convivir en nuestro equipo cuatro sistemas operativos distintos

La **partición extendida** no puede usarse como partición de arranque de un sistema operativo, pero tiene la ventaja de poder dividirse en varios volúmenes.

Sin embargo el usuario nunca trabaja directamente sobre las particiones de un disco duro, ni notará ninguna diferencia por trabajar en una primaria o en una extendida, para trabajar con los datos se define el concepto de **volumen**, cada volumen lleva asignada una letra de unidad que la identifica para poder trabajar con ella.

La utilidad "fdisk" del Dos, sólo permite crear una partición primaria y otra extendida, aunque con otras herramientas podremos crear varias (hasta un total de 4 que es lo máximo permitido)

Aunque se hayan definido las particiones en las que se va a estructurar el disco, todavía éstas no son accesibles al sistema operativo. Para ello tenemos que formatear cada una de las particiones con un formato reconocible por el sistema operativo que vayamos a instalar.

El proceso de formateo de un disco duro podemos clasificarlo en dos niveles, formateo a bajo nivel y a alto nivel. El primero creará en el disco duro las marcas de inicio y fin de los sectores, y por lo tanto los sectores en sí. Este tipo de formateo no debe ser realizado nunca, pues se corre el peligro de destruir de forma permanente al disco duro. Es mas, muchos fabricantes anulan la garantía de sus discos duros si éstos han sido reformateados a bajo nivel. El formateo a bajo nivel requiere software específico como "Hard disk manager".

El formateo a alto nivel sólo comprueba la integridad de todos los sectores buscando defectuosos, que en el caso de aparecer serán marcados como malos y así evitar su uso. También se crean los sectores de arranque, las tablas de asignación de ficheros (FAT) y el directorio raíz.

Una vez formateadas podemos definir el concepto de **volumen**.

Volúmenes

Los **volúmenes** o **unidades lógicas** son porciones de uno o varios discos duros que el sistema operativo trata como si fueran una sola unidad física, como si dijéramos un disco duro virtual. De manera que podemos escribir y leer datos en ellas, formatearlas, buscar errores, etc.

Un volumen puede abarcar:

Una partición primaria completa, la partición primaria sólo puede albergar un volumen.

Un fragmento de una partición extendida, en una partición extendida podemos incluir tantos volúmenes como queramos ;-), este tipo de volumen recibe el nombre de "unidad lógica".

Porqué particionar un disco duro

El hacer particiones en un disco duro puede representar interesantes ventajas:

Flexibilidad: si tenemos en nuestro sistema sólo un disco con una sola partición primaria no podremos acceder a las interesantes opciones que nos permite el administrador de discos, ya que esta partición albergará el sistema y no puede ser modificada. Si tenemos uno o varios discos con unas cuantas particiones podremos crear nuevos volúmenes, ampliar el espacio en las unidades existentes, crear discos espejo, etc.

Organización: es adecuado tener separados los grandes bloques de datos en distintas particiones de manera que nuestro árbol de directorio sea menos complicado y más intuitivo.

Rendimiento: en discos duros de gran tamaño formateados con FAT ganaremos espacio si hacemos un adecuado número de particiones.

SISTEMA DE FICHEROS FAT

El sistema de ficheros FAT es el empleado en DOS. Debe su nombre al uso de una tabla de ubicación de ficheros, con un formato rígido, que permite una manipulación sencilla del disco, a costa de una pérdida de rendimiento, debida en gran medida a la fragmentación de ficheros.

Un disco duro es un dispositivo de almacenamiento secuencial, lo que implica que los datos deben ser escritos y leídos de uno en uno. La primera medida que podría mejorar el rendimiento, sería el uso de varios platos, lo que permitiría la lectura de datos en paralelo. Pero la realidad nos dice que el proceso más lento de acceso a los datos de un disco, es el posicionamiento de los cabezales. Por lo tanto, si el fichero se encuentra en el disco en bloques contiguos, el rendimiento general será mayor.

El sistema FAT es el mismo que se emplea en los disketes. El disco está formado por un sector de inicio conteniendo un programa cargador del sistema operativo (boot), y a continuación, una tabla de asignación de ficheros o FAT. En un disco, el hardware debe acceder a sectores físicos, y ésta será la mínima cantidad de información que es posible leer o escribir. Típicamente un sector son 512 bytes.

¿ cómo se va almacenando la información dentro de un disco?. Pues de la forma más lógica, es decir, uno detrás de otro. Pero ¿qué ocurre si borramos un fichero y a continuación grabamos otro?. Pues lo lógico sería que el espacio dejado por el fichero borrado quede disponible para el nuevo, aunque si este segundo es mayor que el primero, no cabra en el hueco y habrá que fragmentarlo. Claro está, que si un fichero se ha fragmentado, tendrá que existir un mecanismo para luego poder volver a pegarlo.

Con FAT, se emplea una tabla donde se guarda información relativa al uso de las unidades de asignación del disco. Estas unidades de asignación pueden coincidir con un sector (512 bytes), aunque en la práctica se emplean tamaños mayores. Las unidades de asignación reciben el nombre de "cluster". En un diskette, un cluster, equivale a un sector, pero en un disco duro son frecuentes valores de 4, 8 ó mas sectores por cluster. La FAT tiene una importancia vital en el acceso a los datos del disco, por lo que existe una copia de seguridad que se actualiza de forma permanente. Esto evitaría la pérdida de los datos en el caso que se cortara la luz en el instante de producirse la escritura de la tabla FAT, ya que se destruiría la primera copia, pero no la segunda.

Los disketes emplean el formato FAT 12, lo que quiere decir que se usan 12 bits para codificar los datos de la tabla. Con 12 bit, tenemos 4096 unidades de asignación, con lo que podríamos asignar hasta 2.097.152. ¿Y si quisierámos formatear un disco de 2.8MB? Pues podríamos recurrir a aumentar el tamaño del cluster. Si empleamos cluster a 4 sectores (2048bytes), podremos disponer de una capacidad de hasta 8MB. Aunque seguiríamos disponiendo de sólo 4096 unidades de asignación, y cada fichero por pequeño que sea gasta como mínimo una.

Contenido	Significado
000H	Cluster disponible
FFFH	Último cluster del fichero
FF7H	Cluster defectuoso
XXXH	Número del siguiente cluster del fichero

En la tabla podemos ver el significado de los bytes de la FAT 12.

Para aumentar la cantidad de espacio direccionable aparece el FAT 16, que empla 16bits para cada entrada, lo que aumenta a 65536, el número de entradas admitidas. Si se admite que como máximo cada cluster tendrá un tamaño de 64 sectores(32kBytes), tendremos una capacidad máxima de 2Gbytes.

El sistema de ficheros con FAT a 32 bits, aumenta el tamaño de cada entrada en la FAT, pero permite capacidades de disco de hasta 2Tbyte, con tamaños de cluster de tan sólo 512 bytes. En la práctica se prefiere emplear . tamaños de cluster mayores entre 2 y 4k.

EL DIRECTORIO

A diferencia de otros sistemas operativos, el acceso a ficheros no está restringido. Esto es así, porque el DOS es un sistema monousuario, y por lo tanto supone al usuario como dueño de todo. Esto además simplifica la estructura de los directorios.

En todo disco, debe existir al menos un directorio raíz. Este directorio raíz tiene un sitio y tamaños fijos dentro del disco y estos valores no pueden ser cambiados una vez que el disco ha sido formateado. A modo de ejemplo, citar que el directorio raíz de un disquete “sólo” admite 224 entradas.

Cada entrada del directorio ocupa 32 bytes, y en ellos se almacena el nombre, el tamaño, la fecha, el tipo de archivo (oculto, del sistema..), y el cluster donde se encuentra la primera parte del fichero.

Si la entrada corresponde a un directorio, el cluster correspondiente será una nueva estructura de directorio. Los directorios que cuelgan del raíz no tienen limitación en cuanto al número de entradas.

Versiones de MS-Dos

La tabla de particiones de un disco duro está localizada en la posición 01BE (446) del primer sector del disco. Cada entrada requiere 16 bytes, y solo se admiten 4, es decir, 64 bytes. Existe un byte que describe el tipo de partición.

0x1C2 = Partición 1
0x1D2 = Partición 2
0x1E2 = Partición 3
0x1F2 = Partición 4

El sector MBR siempre acaba con los bytes “55AA”. MS-DOS comenzó a dar soporte a discos duros en la versión 2.0.

MS-DOS Version 2.x

MS-DOS 2.x soportaba un único tipo de partición, con un tamaño inferior a 15 megabytes (MB) , con un sistema de archivos FAT (file allocation table) de 12 bit. Fdisk creaba una única partición por drive.

MS-DOS 3.0

MS-DOS 3.0 soporta particiones mayores 15 MB usando una FAT de 16-bit, la cual emplea un tamaño de cluster menor, y por lo tanto consigue mayor eficacia en el aprovechamiento del disco. Sólo se permite una partición por drive.

MS-DOS 3.3

MS-DOS 3.3 introduce soporte para más de una partición por drive. Aparecen las particiones extendidas, las cuales pueden crecer como los directorios de una forma ilimitada. Este tipo de particiones no pueden tener autoarranque.

MS-DOS 4.0

MS-DOS 4.0 y posteriores da soporte a drives de mas de 32MB. The following table explains.

En la tabla siguiente tenemos un resumen de los tipos de particiones:

Partition Type	Fdisk Reports	Starting in Size	FAT Type	MS-DOS version
01	PRI DOS	0-15 MB	12-Bit	2.0 (a)
04	PRI DOS	16-32 MB	16-Bit	3.0
05	EXT DOS	0-2 GB (b)	n/a	3.3
06	PRI DOS	32 MB-2 GB (b)	16-bit	4.0

(a) 15-MB size limitation extended in version 3.0.

(b) 2 GB (gigabytes) includes a limit of 1024 cylinders per drive imposed by the standard AT ROM BIOS interrupt 13 protocol.

MS-DOS 5.0

MS-DOS 5.0 y posteriores permiten utilizar mas de 8 discos duros. and later support up to eight physical hard disks. Share.exe is not required for full large-drive support; this support is included in the MS-DOS kernel.

Existen versiones de Dos que dan soporte a mas de 4 particiones , así como a otros nuevos tipos de particiones.

Windows 95

Windows 95 soporta 2 nuevos tipos de partición (0E y 0F) que permiten emplear los LBA (logical block addressing), que son extensiones de la INT3h, que es incapaz de manejar discos duros con más de 1024 cilindros.

Partition Type	Fdisk Reports	Size	FAT Type	Starting in version
01	PRI DOS	0-15 MB	12-Bit	MS-DOS 2.0
04	PRI DOS	16-32 MB	16-Bit	MS-DOS 3.0
05	EXT DOS	0-2 GB	n/a	MS-DOS 3.3
06	PRI DOS	32 MB-2 GB	16-bit	MS-DOS 4.0
0E	PRI DOS	32 MB-2 GB	16-bit	Windows 95
0F	EXT DOS	0-2 GB	n/a	Windows 95

Types 0E and 0F require extended Int13 support.

NOTA: El tipo 0E es el mismo que el 06, y el 0F el mismo que el 05. Sin embargo, las aplicaciones deben tener soporte de (LBA) INT13h extensiones, para poder acceder a los cilindros altos del disco duro.

Windows 95 OEM Service Release 2 y Windows 98

Windows 95 OEM Service Release 2 (OSR2) y Windows 98 soportan dos nuevos tipos de partición (0B y 0C) para el sistema de ficheros FAT32. TITLE : Description of FAT32 File System

Las particiones soportadas son:

Partition	Type	Fdisk Reports	Size	FAT Type	Starting in version
01		PRI DOS	0-15 MB	12-Bit	MS-DOS 2.0
04		PRI DOS	16-32 MB	16-Bit	MS-DOS 3.0
05		EXT DOS	0-2 GB	n/a	MS-DOS 3.3
06		PRI DOS	32 MB-2 GB	16-bit	MS-DOS 4.0
0E		PRI DOS	32 MB-2 GB	16-bit	Windows 95
0F		EXT DOS	0-2 GB	n/a	Windows 95
0B		PRI DOS	512 MB - 2 terabytes	32-bit	OSR2
0C		EXT DOS	512 MB - 2 terabytes	32-bit	OSR2

Types 0E, 0F, and 0C require extended Int13 support.

TAMAÑOS DE LOS CLUSTERS. -

El sistema operativo es quien define cuál será el tamaño para los clusters de un disco duro. Esta decisión es efectuada a partir del tamaño del disco duro. De forma estándar, para unidades de almacenamiento con formato FAT y VFAT, que son las más comunes en el mundo de las computadoras personales, se tiene una tabla de tamaños predefinida, la misma se muestra en la figura 1. Está por demás decir que en un disco con una capacidad de 4GB, cualquier archivo con un tamaño menor a 64KB ocupa 64KB, aunque sea de 1KB. El desperdicio es realmente notable, razón por la cual se aconseja realmente particionar los discos duros en subunidades más pequeñas a fin de reducir los efectos de la fragmentación interna.

Sobre el tema se han realizado numerosos estudios y pruebas, en las que en base a programas hechos en lenguajes de bajo nivel como C, se puede analizar realmente la eficiencia de almacenamiento de un disco. La tabla de la figura inferior resume esta información. Ciertamente que se puede ver que mientras más chica es la unidad más se aprovecha el disco duro, aunque por otro lado la FAT crece de forma proporcional a la eficiencia. A través de esta información no resulta demasiado difícil inferir cuál es el tamaño más adecuado para una partición de un disco duro dado por el punto de equilibrio del tamaño de la FAT y de la eficiencia alcanzada: aproximadamente 512KB.

CAPACIDAD DEL D.D.	TAMAÑO DEL CLUSTER	TAMAÑO FAT (Bytes)	EFICIENCIA
16MB a 128MB	2 KB	213324	97.7 %
128MB a 256MB	4 KB	110050	96.2 %
256MB a 512MB	8 KB	58686	93.6 %
512MB a 1GB	16 KB	33292	89.6 %
1GB a 2GB	32 KB	20988	84.8 %
2GB a 4GB	64 KB	15144	79.4 %

Fig. 2 - Rendimiento y eficiencia de los Discos Duros

libre de la unidad. Además, FAT sólo admite los atributos de archivo de sólo lectura, oculto, de sistema y modificado.

Convención de nomenclatura FAT

FAT utiliza la convención de nomenclatura tradicional de archivos de 8.3 y todos los nombres de archivo se deben crear con el juego de caracteres ASCII. El nombre de un archivo o un directorio puede tener una longitud máxima de ocho caracteres, un punto (.) de separación y una extensión de hasta tres caracteres. El nombre debe empezar por una letra o un número, y puede contener cualquier carácter excepto los siguientes:

." \ [] : ; | = ,

Si se utiliza alguno de estos caracteres, es posible que se produzcan resultados inesperados. El nombre no puede contener espacios.

Los nombres siguientes están reservados:

CON, AUX, COM1, COM2, COM3, COM4, LPT1, LPT2, LPT3, PRN, NUL
Todos los caracteres se convierten a mayúsculas.

Ventajas de FAT

El sistema de archivos FAT es el mejor para unidades o particiones con menos de 200 MB, porque FAT se inicia con muy poca carga de trabajo.

Desventajas de FAT

Preferiblemente, con unidades o particiones de más de 200 MB no se debería utilizar el sistema de archivos FAT ya que a medida que aumenta el tamaño del volumen, el rendimiento disminuye rápidamente. No es posible establecer permisos en archivos que se encuentran en particiones FAT. Las particiones FAT tienen un tamaño máximo de 4 gigabytes (GB) en Windows NT y de 2 GB en MS-DOS.

INTRODUCCIÓN A HPFS

El sistema de archivos HPFS se presentó por primera vez con OS/2 1.2 para permitir mejor acceso a las unidades de disco duro grandes que empezaban a aparecer en el mercado. Adicionalmente, era necesario un nuevo sistema de archivos para ampliar el sistema de nombres, la organización y la seguridad de las demandas crecientes del mercado de los servidores de red. HPFS mantiene la organización de directorios de FAT, pero agrega la ordenación automática del directorio por nombres de archivo. Los nombres de archivo se amplían hasta 254 caracteres de doble byte. HPFS también permite que un archivo esté compuesto de "datos" y de atributos especiales que permiten una mayor flexibilidad en cuanto a la compatibilidad con otras convenciones de nomenclatura y con la seguridad. Además, la unidad de asignación se cambia de clústeres a sectores físicos (512 bytes), lo que reduce la pérdida de espacio en disco. En HPFS, las entradas de directorio contienen más información que en FAT. Además de los atributos de archivo, incluye información acerca de la fecha y hora de modificación, creación y acceso. En lugar de señalar al primer clúster del archivo, las entradas de directorio en HPFS señalan a

FNODE.

FNODE puede contener los datos del archivo o punteros que pueden señalar a los datos del archivo o a otras estructuras que, eventualmente, señalan a los datos del archivo.

- HPFS intenta asignar en sectores contiguos tantas partes del archivo como sea posible. Esto se realiza para aumentar la velocidad cuando un archivo se procesa secuencialmente.
- HPFS organiza una unidad en conjuntos de bandas de 8 MB y, siempre que es posible, un archivo se encuentra en una de estas bandas. Entre cada banda hay mapas de bits de asignación de 2 K, que realizan un seguimiento de qué sectores están asignados dentro de una banda. El sistema de bandas incrementa el rendimiento porque el cabezal de la unidad no tiene que volver al principio lógico (normalmente, el cilindro 0) del disco, sino al mapa de bits de asignación de bandas más cercano para determinar dónde está almacenado un archivo.
- Además, HPFS incluye dos objetos de datos exclusivos especiales:

Superbloque

El superbloque se encuentra en el sector 16 lógico y contiene un puntero al FNODE del directorio raíz. Uno de los mayores peligros de HPFS es que si se pierde o se daña el superbloque debido a un sector defectuoso, también se pierde el contenido de la partición, incluso si el resto de la unidad está bien. Sería posible recuperar los datos de la unidad si se copia todo a otra unidad con un sector 16 correcto y se reconstruye el superbloque. Sin embargo, esta tarea es muy compleja.
Bloque de reemplazo

El bloque de repuesto se encuentra en el sector 17 lógico y contiene una tabla de "correcciones activas" y el bloque de directorio de repuesto. En HPFS, cuando se detecta un sector defectuoso, se utiliza la entrada de "correcciones activas" para apuntar de forma lógica a un sector correcto existente en lugar de apuntar al sector defectuoso. Esta técnica de tratamiento de errores de escritura se denomina corrección activa.

La corrección activa es una técnica mediante la cual, si se produce un error debido a un sector defectuoso, el sistema de archivos mueve la información a otro sector y marca el sector original como defectuoso. Esto se realiza de forma transparente para cualquier aplicación que efectúe operaciones de entrada y salida de disco (es decir, la aplicación nunca sabe que ha habido problemas en el disco duro). El uso de un sistema de archivos que admite la corrección activa elimina los mensajes de error como "¿Anular, reintentar o ignorar?" de FAT cuando se encuentra un sector defectuoso.

Nota: La versión de HPFS incluida con Windows NT no admite la corrección activa.

Ventajas de HPFS

HPFS es el mejor sistema para unidades de 200 a 400 MB.

Desventajas de HPFS

Debido a la carga que implica HPFS, no es una elección muy eficaz para un volumen de menos de 200 MB. Además, con volúmenes mayores de 400 MB, se degrada algo el rendimiento.

INTRODUCCIÓN A NTFS

Desde el punto de vista del usuario, NTFS organiza los archivos en directorios que, al igual que en HPFS, están ordenados. Sin embargo, a diferencia de FAT o HPFS, no hay objetos "especiales" en el disco y no existe ninguna dependencia del hardware subyacente, como los sectores de 512 bytes. Además, no hay ubicaciones especiales en el disco, como ocurre en las tablas FAT o los superbloques HPFS.

Los objetivos de NTFS consisten en proporcionar:

- Fiabilidad, lo que es muy deseable para sistemas de altas prestaciones y servidores de archivos
- Una plataforma para funcionalidad agregada.
- Supresión de las limitaciones de los sistemas de archivos FAT y HPFS

Fiabilidad

Para asegurar la fiabilidad de NTFS, se ha hecho hincapié en tres áreas principales: recuperación, supresión de los errores graves de sectores individuales y corrección activa. NTFS es un sistema de archivos recuperable porque realiza un seguimiento de las transacciones efectuadas en el sistema de archivos. Cuando se ejecuta CHKDSK en FAT o HPFS, se comprueba la coherencia de los punteros en el directorio, la asignación y las tablas de archivos. En NTFS, se mantiene un registro de las transacciones efectuadas en estos componentes de modo que CHKDSK sólo necesita deshacer las transacciones hasta el último punto de confirmación para recuperar la coherencia del sistema de archivos.

En FAT o HPFS, si se produce un error en un sector que es la ubicación de uno de los objetos especiales del sistema de archivos, se produce un error de un único sector. NTFS impide esta situación de dos formas: en primer lugar, no utiliza objetos especiales en el disco y realiza un seguimiento y protege todos los objetos que hay en el disco. En segundo lugar, en NTFS, se mantienen múltiples copias de la tabla maestra de archivos (el número depende del tamaño del volumen). Al igual que las versiones OS/2 de HPFS, NTFS admite la corrección activa.

Funcionalidad agregada

Uno de los objetivos de diseño principales de Windows NT en todos los niveles es proporcionar una plataforma ampliable sobre la que se pueda construir, y NTFS no es una excepción. NTFS proporciona una plataforma rica y flexible para se puedan utilizar otros sistemas de archivos. Además, NTFS es

totalmente compatible con el modelo de seguridad de Windows NT y admite múltiples secuencias de datos. Un archivo de datos ya no es un único flujo de datos. Por último, en NTFS los usuarios pueden agregar sus propios atributos personalizados a un archivo.

Supresión de limitaciones

En primer lugar, NTFS ha incrementado enormemente el tamaño de los archivos y de los volúmenes, por lo que ahora pueden ser de hasta 2^{64} bytes (16 exabytes o 18.446.744.073.709.551.616 bytes). NTFS también ha vuelto al concepto FAT de los clústeres para evitar el problema de HPFS de un tamaño fijo de sector. Esto se ha hecho porque Windows NT es un sistema operativo portátil y es probable encontrar en algún punto una tecnología de disco distinta. Por tanto, se consideró que los 512 bytes tenían muchas posibilidades de no ser siempre una buena opción para la asignación. Esto se consigue al permitir que el clúster se defina como múltiplos del tamaño de asignación natural del hardware. Por último, en NTFS todos los nombres de archivo se basan en Unicode y se mantienen los nombres de archivo 8.3 así como los nombres largos.

Ventajas de NTFS

NTFS es el mejor sistema para volúmenes de 400 MB o más. Esto se debe a que en NTFS no se degrada el rendimiento con tamaños de volúmenes mayores, como sucedía en FAT. El sistema de recuperación diseñado en NTFS es tal que nunca debería ser necesario ejecutar un programa de reparación de discos en una partición NTFS. Para obtener información acerca de las ventajas adicionales de NTFS, consulte:

Desventajas de NTFS

No se recomienda utilizar NTFS en un volumen que sea menor de 400 MB, aproximadamente, debido a la carga de espacio que supone NTFS. Esta carga de espacio son los archivos de sistema NTFS que normalmente utilizan al menos 4 MB de espacio de la unidad en una partición de 100 MB. Actualmente, no se ha integrado en NTFS el cifrado de archivos. Por tanto, cualquiera puede iniciar en MS-DOS, u otro sistema operativo, y utilizar un programa de modificación de discos de nivel inferior para ver los datos almacenados en un volumen NTFS. No es posible dar formato a un disquete con el sistema de archivos NTFS. Windows NT da formato a los disquetes con el sistema de archivos FAT debido a que la carga que implica NTFS no cabría en un disquete.

Convenciones de nomenclatura de NTFS

Los nombres de archivos y de directorios pueden tener una longitud de hasta 255 caracteres, incluida la extensión. Los nombres conservan las mayúsculas y las minúsculas, pero no distinguen entre mayúsculas y minúsculas. NTFS no distingue los nombres de archivo según el uso de mayúsculas o minúsculas. Los nombres pueden contener cualquier carácter excepto los siguientes:

? " / \ < > * | :

Nota: Actualmente, desde la línea de comandos, sólo se pueden crear nombres de archivo de hasta 253 caracteres.

EL SISTEMA DE ARCHIVOS EXT2

El sistema operativo Linux nace como una variante de UNIX para funcionar sobre PC's, por lo que debe adoptar lo mejor del UNIX sin olvidar al DOS para permitir algún tipo de compatibilidad ya que la mayoría de PC's funcionan con DOS.

El sistema de ficheros

Un fichero es una secuencia de octetos repartida en sectores dispuestos aleatoriamente en la superficie de un disco. En su acepción más amplia, el sistema de ficheros es un servicio que presta el sistema operativo a las aplicaciones para que estas vean el fichero como una secuencia continua de octetos,

identifiquen el fichero mediante un nombre y acceda a este a través de primitivas sencillas de posicionamiento en la secuencia de octetos, lectura y escritura. El sistema de ficheros aísla a las aplicaciones de las características físicas del dispositivo, por ejemplo los mandatos de control del adaptador de disco, y protege los ficheros de un usuario frente al resto de usuarios. También se utiliza la denominación de sistema de ficheros a la disposición en el disco de los ficheros que soporta el sistema de sistema de ficheros.

Zonas y bloques

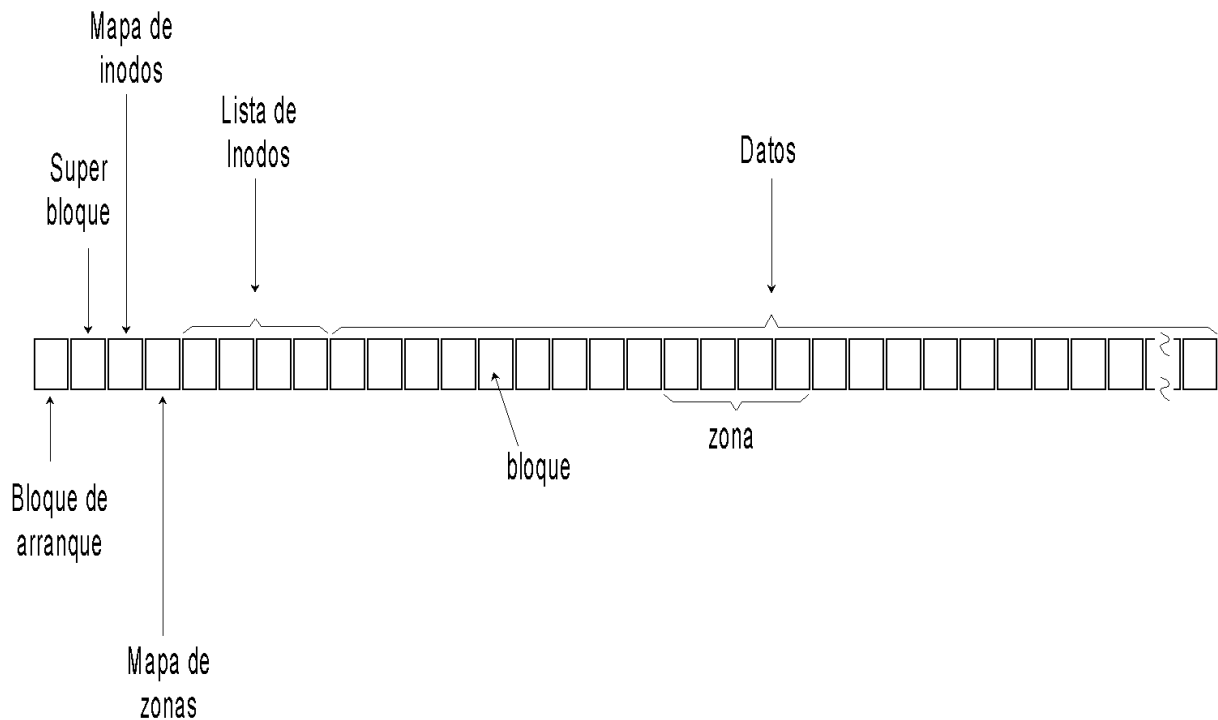


Figura: Muestra de un dispositivo de bloques en un sistema de ficheros Linux.

Por dispositivo entendemos un disquete o una partición del disco duro. El sistema de ficheros comienza en el segundo bloque de la partición o disquete, ya que el primer bloque se reserva como bloque de arranque. Antes de hablar de zonas hay que hablar de bloques. Un disco es realmente una secuencia de sectores. El sector es la unidad física direccionada por el hardware.

El bloque es una entidad lógica que consiste en un número determinado de sectores consecutivos. En Linux un bloque son dos sectores pero otros sistemas operativos tienen otro tamaño de bloque. El tamaño del bloque es un parámetro de operación del sistema de ficheros. Este lee y escribe los ficheros del disco siempre en términos de bloques, de modo que un tamaño de bloque grande propicia menos transferencias entre disco y memoria, ya que una transferencia contiene más octetos. Por otra parte, cuanto mayor sea el tamaño del bloque, más espacio se desperdicia en el disco, sobre todo por parte de los ficheros pequeños. La zona no es más que una extensión del concepto de bloque. Pues bien, una zona es un número consecutivo de bloques, generalmente una potencia de 2. Si un bloque se define como la unidad de transferencia, la zona es la unidad de asignación de espacio en disco. Los ficheros se extienden en el disco en forma de zonas.

Supongamos un tamaño de bloque de 1K, un tamaño de zona de cuatro bloques y un disco de 8Gb. Este disco dispone de $8\text{Gb} = 2^3 \times 2^{30} = 2^{33}$ octetos y $2^{33}/(4 \times 2^{10}) = 2^{33}/2^{12} = 2^{21} = 2\text{M}$ zonas. La versión de

Linux utiliza, no obstante, una zona de 1K, es igual que el bloque, y utiliza direcciones de zona de 32 bits. Estas direcciones son de tipo `"..\..\Src\include\sys\types.h.doc"` para la versión V1 del sistema de ficheros y `"..\..\Src\include\sys\types.h.doc"` para la versión V2. La configuración de 32 bits posibilita sistemas de ficheros de hasta $2^{32} \times 2^{10} = 2^2 \times 2^{40} = 4 \text{ Tb}$.

La zona no es concepto asociado al disco, sino más bien al sistema de ficheros. El espacio de referencias ocupa un número entero de zonas. Los datos comienzan alineados en la siguiente zona. El mapa de zonas es un mapa de bits. Cada bit se corresponde con una zona. Si el bit es cero, la zona está libre y puede utilizarse para almacenar parte de un fichero. Si es uno, está ocupada. Los dos primeros bits del mapa de zonas siempre están a 1. ¿Por qué? Cuando se crea el sistema de ficheros, todos los bits están a cero excepto los dos primeros. La razón es que la primera zona ocupa su espacio correspondiente, pero no se utiliza.

La segunda zona se utiliza para albergar un fichero que está en el disco desde el instante de creación del sistema de ficheros, es el directorio raíz.

Inodos

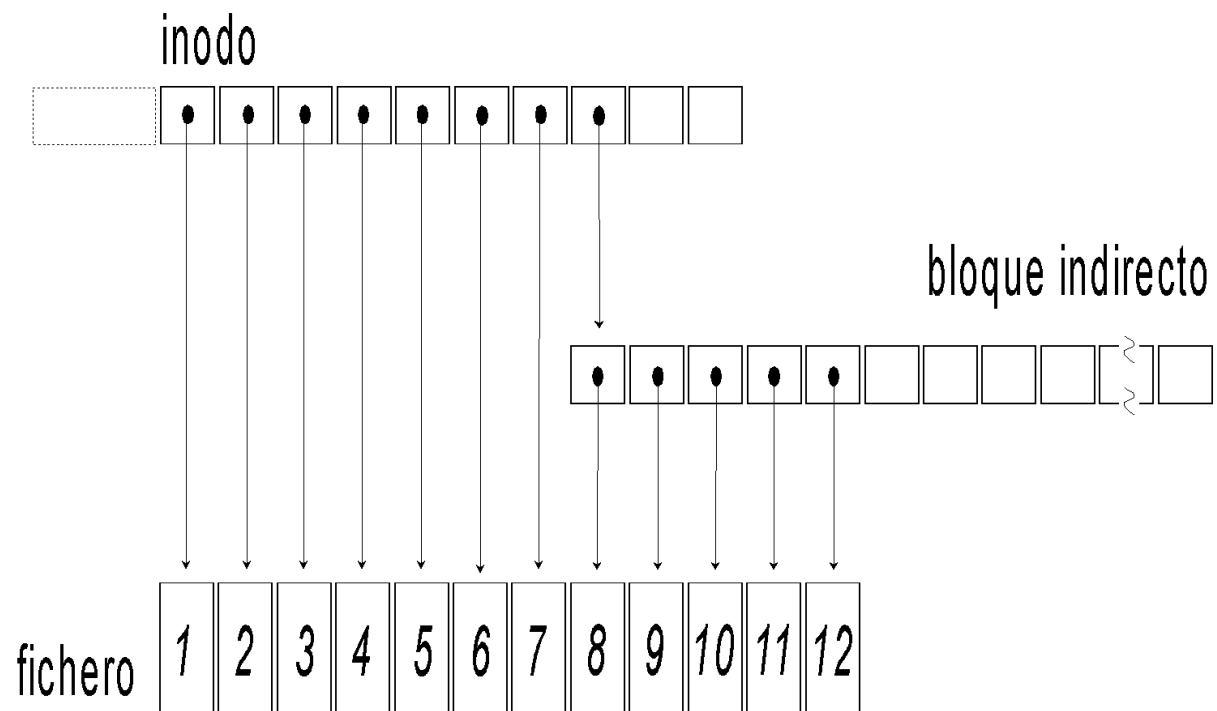


Figura: Implementación de un inodo.

¿Qué es un inodo? Cada fichero ocupa una serie de zonas en el disco. Cada fichero dispone de un inodo, descrito en `"..\..\Src\fs\Type.h.doc"` (`d1_inode`) para la versión V1 y `"..\..\Src\fs\Type.h.doc"` (`d2_inode`) para la versión V2. El inodo tiene dos partes. La primera son los atributos del fichero y la segunda el conjunto de direcciones de las zonas que ocupa el fichero.

¿Qué es una dirección de zona? Depende. En Linux, una dirección de zona es la dirección absoluta en el dispositivo del primer bloque de la zona. Un dispositivo puede ser una partición o un disquete. En la versión V2 del sistema de ficheros, existen en total diez. Las siete primeras son direcciones directas, las dos siguientes son indirectas y la última doblemente indirecta. Una dirección directa es la dirección de una zona del fichero. Una dirección indirecta apunta a bloque zona auxiliar que contiene direcciones de zona. La versión V2 tiene por defecto una zona de un bloque, de modo que la zona auxiliar puede almacenar $1\text{Kb}/(4\text{b/dir}) = 256$ direcciones de zona.

Cada inodo ocupa **64 octetos** en el disco.

Los inodos de un sistema de ficheros se agrupan como entradas de la tabla o lista de inodos. Esta tabla tiene un tamaño fijo. Algunos inodos están usados y otros no. Esta condición se marca en el mapa de inodos, un mapa de bits. Como las zonas, los inodos también tienen un mapa de bits. Los dos primeros bits del mapa siempre están a 1. Cuando se crea el sistema de ficheros, todos los bits están a cero excepto los dos primeros. La razón es que el primer inodo ocupa su espacio correspondiente, pero no se utiliza. El segundo inodo corresponde al directorio raíz del sistema de ficheros.

Cuando un fichero se abre, el contenido del inodo pasa a formar parte de una estructura en memoria denominada `"..\\..\\Src\\fs\\Inode.h.doc"` (inode). Esta estructura tiene dos partes. La primera consiste en el contenido del inodo en el disco y la segunda discrimina el inodo de entre todos los inodos de todos los sistemas de ficheros y contiene asimismo información de control de las operaciones sobre el fichero.

El superbloque

El bloque de arranque está formado por el sector de arranque y un sector que tradicionalmente no se utiliza. La nomenclatura ahora se vuelve un tanto confusa. El primer bloque del sistema de ficheros se denomina el superbloque. El superbloque contiene una estructura de datos, también denominada superbloque, que proporciona la descripción de la disposición de los elementos del sistema de ficheros. Esta estructura es la primera parte de `super_block`, una estructura aumentada utilizada por el sistema de ficheros. `Super_block` tiene dos partes, la primera es la que reside en el disco, en el superbloque, y la segunda se crea cuando se monta el sistema de ficheros y, a partir de este momento, reside siempre en memoria. La primera parte ocupa sus campos de zonas e inodos. `"..\\..\\Src\\fs\\Super.h.doc"` (`N_inodes`) es el número de inodos disponibles en el sistema de ficheros, el tamaño de la tabla de inodos menos uno. La razón es que el primer inodo, si bien ocupa espacio en el disco, no se utiliza. `"..\\..\\Src\\fs\\Super.h.doc"` (`S_nzones`) es el número total de zonas del dispositivo que alberga el sistema de ficheros contando los datos y el resto. Este campo sólo se utiliza cuando el sistema de ficheros fue creado por una versión V1 de Linux. `"..\\..\\Src\\fs\\Super.h.doc"` (`S_imap_blocks`) es el número de bloques que ocupa el mapa de inodos. `"..\\..\\Src\\fs\\Super.h.doc"` (`Z_imap_blocks`) es el número de bloques que ocupa el mapa de zonas. `"..\\..\\Src\\fs\\Super.h.doc"` (`S_firstdatazone`) es la primera zona del dispositivo que contiene datos. `"..\\..\\Src\\fs\\Super.h.doc"` (`S_log_zone_size`) representa el número de bloques que abarca una zona. Realmente es el logaritmo base dos de este número. Por ejemplo, si hay cuatro bloques por zona, `s_log_zone_size` vale 2, si hay 8 vale 3, etc. `"..\\..\\Src\\fs\\Super.h.doc"` (`S_max_size`) tamaño máximo que puede alcanzar un fichero en este sistema de ficheros. `"..\\..\\Src\\fs\\Super.h.doc"` (`S_magic`) es un código mágico de superbloque. `"..\\..\\Src\\fs\\Super.h.doc"` (`S_zones`) es equivalente al campo `s_nzones` para versiones V2 del sistema de ficheros.

Un comentario sobre las versiones V1 y V2. Linux fue concebido en el año 1985. El sistema de ficheros original de Linux tenía direcciones de disco de 16 bits. Con 16 bits se pueden direccionar 64K zonas. Una zona era un bloque, lo que significaba que un sistema de ficheros podía tener un tamaño de hasta 64Mb, un tamaño más que suficiente para los discos duros de entonces.

El crecimiento previsible de la capacidad de los discos podía ser soportado aumentando el número de bloques por zona. Además, con direcciones de 16 bits se podía disponer de inodos de sólo 32 octetos. Esta versión del sistema de ficheros se conoce como V1. La capacidad de la versión V1 pronto se mostró insuficiente. Linux introduce la versión V2, con direcciones de disco de 32 bits.

Los directorios

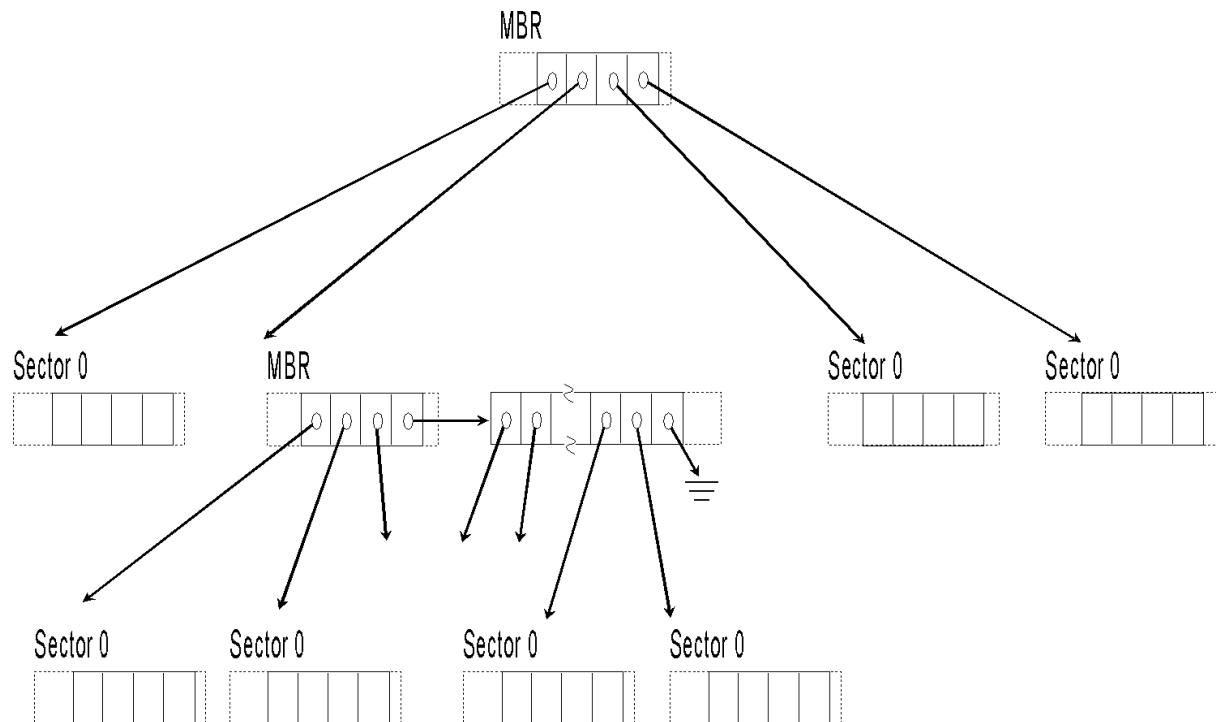


Figura: Desglose de las MBR de los directorios de un disco duro

¿Qué es un directorio? No es más que un fichero convencional con un contenido estructurado. Un directorio consiste en uno o más bloques que albergan una tabla. Cada entrada describe uno de los hijos del directorio, que puede ser un directorio o un fichero. La entrada se describe en la estructura "..\..\Src\include\sys\dir.h.doc" (direct). Tiene dos partes, una es el número de inodo del hijo y otra una cadena de hasta 14 octetos que contiene el nombre del hijo.