

LA FAT

Índice

Pág.

Introducción 1

1.- La estructura básica del sistema de archivos 1

2.- El sector de arranque 3

3.- La File-Allocation-Table (FAT) 5

La Fragmentación de archivos 5

Estructura de la FAT 6

Varias copias de la FAT 7

4.- El directorio raíz 7

La estructura de entradas de directorio 7

Los sub-directorios 8

Codificación de las entradas de la FAT 9

5.- La zona de datos 10

6.- Formatos de disquetes 12

7.- Bibliografía 13

Introducción. El sistema de archivos de MS-DOS

Desde el punto de vista del usuario se ve todo tan fácil: se copian archivos de un directorio a otro, se formatea un disquete o se borra un archivo, y todo esto con un solo comando. Pero detrás del telón existe un largo camino hasta que el DOS ha localizado la posición de un archivo o un espacio libre en el disco que se pueda emplear para guardar un archivo. Ya que las unidades de disquetes y discos duros no quieren saber nada de archivos y sub-directorios; sólo conocen sectores, pistas y cabezales; el DOS ha de equipar a cada almacenamiento masivo con un sistema de archivos, que haga posible el salto del nivel físico al nivel lógico. Este sistema de archivos se compone de una serie de estructuras de datos, que describen el tamaño del disquete o disco duro y su contenido. Cuál es la estructura del sistema de archivos de MS-DOS, lo vamos a ver a continuación.

1.-La estructura básica del sistema de archivos

En el centro del sistema de archivos de MS-DOS está el concepto de volumen. Cada disquete y disco duro que disponen de una letra identificativa propia, representan un volumen. Pero mientras que los disquetes no se

pueden dividir en varios volúmenes, en los discos duros esto sí es posible incluso habitual. Hasta la versión 4.0 del DOS, los volúmenes quedaban limitados a 32Mbytes, por lo que los discos duros con una capacidad superior a 32 Mbytes se habían de dividir en cualquier caso en varios volúmenes, si no se quería regalar el precioso espacio de disco.

Es importante anotar que cada volumen tiene una estructura homogénea bajo DOS, independientemente de si está en un disquete o en un disco duro. El tamaño no tiene ninguna importancia, ya que simplemente afecta al tamaño de las diferentes estructuras de datos, que se necesitan para la gestión del volumen.

División en sectores lógicos:

Desde el punto de vista del DOS, un volumen representa una secuencia de sectores, que acogen 512 bytes. Los diferentes sectores tienen un número de sectores lógicos y se enumeran desde 0 en adelante. Un volumen de 10Mbytes se compone, por ejemplo de 20480 sectores, que llevan los números 0 hasta 20479. El DOS no tiene ninguna influencia sobre el ordenamiento físico de los diferentes sectores, ya que esto es asunto únicamente del controlador de dispositivos, a través del cual se desarrolla toda la comunicación entre el DOS y el medio físico. Si el controlador de dispositivo dispersa los diferentes sectores lógicos aleatoriamente por el medio, o si los reparte según un orden estricto, no tiene ninguna importancia. Es importante solo que haya una asociación unívoca entre un sector lógico y uno físico. Así que mientras el MS-DOS opera en su interior con números de sectores lógicos, las diferentes funciones de archivos del DOS-API se refieren a archivos y sub-directorios. Por ello, el DOS ha de traducir el acceso a sectores individuales, antes de llamar a estas funciones. Para ello usa la estructura de directorios del volumen, así como de una estructura de datos denominada FAT.

Echemos un vistazo a la estructura básica de un volumen bajo el sistema de archivos de MS-DOS:

Estructura de un volumen

Sector de arranque
Primera tabla de localización de archivos (FAT)
Una o más copias de la FAT
Directorio raíz (eventualmente con etiqueta de volumen)
Zona de datos para archivos y sub-directorios

Como muestra el cuadro anterior, cada volumen se divide en diferentes zonas, que por una parte acogen las diferentes estructuras de datos del sistema de archivos, y por otra los diferentes archivos y sub-directorios. Estas estructuras se crean en el formateo de un volumen con ayuda del comando FORMAT.

No hemos comentado nada de los tamaños de las diferentes estructuras de datos y zonas; pero esto no es posible así por las buenas, ya que se adapta individualmente al tamaño del volumen correspondiente. Después del formateo de un volumen se puede averiguar el tamaño de las diferentes zonas, en base a las diferentes informaciones en el sector de arranque.

2.-El sector de arranque

Al formatear un volumen el sector de arranque se crea siempre como primer sector del volumen, para que sea fácil de localizar por el DOS. A parte de las informaciones sobre el tamaño y la estructura del volumen, contiene sobre todo el llamado Bootstrap-Loader, mediante el cual se puede arrancar el DOS. A esta circunstancia es la que le debe el nombre el sector de arranque. Al contrario de una creencia extendida, el DOS no se encuentra en memoria después del arranque del ordenador, sino que primero se ha de cargar en memoria y después se ha de ejecutar. Por ello el BIOS, que después de encender el ordenador se encarga de la

inicialización, carga el primer sector físico 0 de un disquete o del disco duro en memoria y comienza con la ejecución del código de programa allí contenido. Se tiene que tener en cuenta que se carga realmente el primer sector físico y no el primer sector lógico. Esto es muy importante, ya que con ello se determina que el primer sector lógico ha de coincidir con el primer sector físico. En el momento en el que se carga el sector de arranque aún no se encuentran en memoria los controladores de dispositivos, a través de los cuales se puede realizar el acceso al medio, aunque el controlador de dispositivo reparte los sectores aleatoriamente por el almacenamiento correspondiente. Así que el sector 0 lógico ha de coincidir con el físico.

La siguiente tabla aclara la estructura del sector de arranque de un volumen:

Desplazamiento	Contenido	Tipo	
+00h	Instrucción de salto a la rutina del BIOS	3 Bytes	
+03h	Fabricante y número de versión	8 Bytes	
+0Bh	Bytes por sector	1 Word	
+0Dh	Sectores por clúster	1 Byte	
+0Eh	Número de sectores reservador	1 Word	
+10h	Número de FAT's	1 Byte	
+11h	Número de entradas en el directorio raíz	1 Word	BIOS-Parameter-Block (BPB)
+13h	Número de sectores del volumen	1 Word	
+15h	Media Descriptor	1 Byte	
+16h	Número de sectores por FAT	1 Word	
+18h	Sectores por pista	1 Word	
+1Ah	Número de cabezales de lectura-escritura	1 Word	
+1Ch	Instancia de primer sector en el volumen del primer sector en el medio de almacenamiento	1 Word	
+1Eh-1FFh	Rutina de arranque	482 Bytes	
Longitud total:		512 bytes	

Siempre se comienza con la ejecución del código de programa en el sector de arranque con el primer byte, es decir, en la dirección de desplazamiento 00h. Por esta razón, el sector de arranque contiene siempre en ese sitio la instrucción de lenguaje ensamblador JUMP, después de la ejecución de la cual se pasa a la ejecución del programa a una posición posterior del sector de arranque. Esa instrucción JUMP puede ser una instrucción normal de salto, o un llamado short jump, cuyos códigos máquina tienen longitudes distintas. Como que el campo para la instrucción de salto siempre mide 3 bytes, pero un short jump sólo ocupa dos, a continuación viene la instrucción de lenguaje máquina NOP, que realmente no hace nada, pero que cierra el hueco al comienzo de los datos. Ya que al primer campo con la instrucción de salto le siguen toda una serie de campos que suministran diferentes informaciones sobre la estructura del volumen. El primero de estos campos comprende 8 bytes y contiene un nombre en forma de carácter ASCII. Por ejemplo, se puede tratar del nombre del fabricante de ordenadores, con cuya versión del DOS se formateó el disco. Pero también es posible el nombre del fabricante del software, si el volumen por ejemplo ha sido formateado con el programa de

En los siguientes campos se describe el formato del medio, es decir la cantidad de bytes por sector, el número de sectores por pista. A parte de lo también se graba la longitud de las diferentes estructuras del DOS, que se guardan en el medio. Sobre todas las informaciones físicas tienen importancia para las diferentes funciones del BIOS de la interrupción 13h, que son las que emplean los controladores de dispositivos habitualmente para leer y escribir sectores. Por ello esta zona se denomina como BIOS Parameter Block (BPB). Pero también el DOS utiliza estas informaciones para diferentes propósitos. Al BPB le siguen tres campos que no se utilizan por el DOS, pero que le pueden suministrar informaciones adicionales al controlador de dispositivos sobre el volumen. A esta zona le sigue finalmente la llamada rutina de Bootstrap, en la que la instrucción de salto se bifurca al inicio del sector de arranque. Se encarga de la carga y al ejecución del DOS.

Al sector de arranque le pueden seguir varios sectores reservados, que sirven para una ampliación del sector de arranque, si por ejemplo la rutina Bootstrap es demasiado extensa, como para se guardada en un solo sector. La cantidad de estos sectores está especificada en el sector de arranque en el campo a partir de la dirección de offset 0Eh. Siempre incluye el sector de arranque, de modo que un 1 en este campo muestra que no hay sectores reservados que sigan al sector de arranque. Así suele ser en la mayoría de los casos, ya que ninguna de las versiones actuales del DOS necesita un Bootstrap-Loader mayor que el que se puede guardar en el primer sector de arranque. Pero ya por la opción de extender el sector de arranque a varios sectores, resulta una cierta variabilidad en la ordenación de los diferentes componentes del sistema de archivos, ya que la FAT, sus copiar y el directorio raíz se desplazan naturalmente hacia atrás, si el sector de arranque le siguen otros sectores.

3.-La File-Allocation-Table (FAT)

Si el DOS quiere crear nuevos archivos, o ampliar archivos existente, ha de saber qué sectores de volumen correspondientes quedan libres. Estas informaciones las toma la llamada File-Allocation-Table (o FAT) que viene a continuación de la zona reservada. Cada entrada a esta tabla se corresponde con un número determinado de sectores, que son adyacentes lógicamente en el volumen. Uno de estos grupos de sectores se llama clúster. Para averiguar cuántos sectores se reúnen en un clúster, tenemos que mirar el valor de la posición 0Dh del sector de arranque como parte del BIOS Parameter Block. Sólo se permiten valores que representen una potencia de dos, es decir, 1,2,4,8,16,etc. En los discos duros del XT esta posición de memoria contiene habitualmente el valor 8, con lo que ocho sectores consecutivos forman habitualmente un clúster. En discos duros que se conectan a un AT, 386 y 486 normalmente solo son 4 sectores, que se reúnen en un clúster. Como muestra la siguiente tabla, según el medio de almacenamiento se almacenan un número distinto de sectores por clúster.

Dispositivos	Sectores por clúster
Unidad de disquetes de 1 cara y unidad de disquetes de alta capacidad	1
Unidad de disquetes de doble cara	2
Disco duro en AT o mayor	4
Disco duro en XT	8

Hay que fijarse al ver la tabla anterior que el programa de formateo no está limitado a los valores indicados, sino que puede elegir el mismo tamaño de un clúster. Esto es tanto más cierto cuanto los volúmenes con más de 32 Mbytes a partir de la versión 4.0 del DOS se realizan mediante la agrupación de más sectores por clúster. Finalmente se rompe la frontera de 65536 clústers con más de 32Mbytes por volumen y una agrupación constante de 4 sectores por clúster, de modo que para coger el número de clúster se necesitan más de 16 bits. Pero agrupando cada vez más sectores en un clúster, la capacidad del sistema de archivos se puede aumentar casi libremente, sin tener que modificar nada en la estructura actual. La siguiente tabla muestra la agrupación que el DOS 4.0 emplea en volúmenes de más de 32 Mbytes. Los volúmenes se pueden expandir

hasta una capacidad de 2 Gigabytes de esta forma, aunque cada vez se regale más espacio en el último clúster de un archivo, ya que este sólo se aprovecha si el archivo lo rellena completamente y es un múltiplo del tamaño del clúster.

Tamaños de volúmenes y clústers a partir del DOS 4.0:

Tamaño del volumen hasta:	128Mbytes	256Mbytes	512Mbytes	1024Mbytes	2048Mbytes
Tamaño del clúster:	2Kbytes	4Kbytes	8Kbytes	16Kbytes	32Kbytes
Sectores por clúster:	4	8	16	32	64

Fragmentación de archivos

El sentido de resumir varios sectores en un clúster resulta de la lógica con la que el DOS escribe datos en un medio. Ya que para guardar un archivo no selecciona sectores adyacentes, sino que fragmenta el archivo de modo que quepa justo en los sectores que quedan libres. Todos los demás tampoco tienen sentido, ya que en el trabajo diario se están creando y borrando archivos continuamente. Mientras un volumen está vacío, los nuevos archivos y sus clústers se pueden crear en orden. Pero si se borra el archivo, se crea un hueco libre, y se tardaría demasiado en desplazar todos los demás clústers hacia adelante. Imaginemos lo que pasaría si borramos un archivo de 10Kbytes al principio del disco duro y hemos de desplazar 40 o 50 Mbytes hacia arriba. Por ello, el DOS crea el archivo en los huecos que han resultado del borrado de archivos anteriores. Lo deseable sería encontrar un hueco en el que quepa el archivo completo, o que al menos es mayor que el archivo, ya que así no se necesitaría fragmentar el archivo. Pero esto no se puede esperar en la práctica, ya que la probabilidad de encontrara tal agujero es demasiado baja. Por ello, el DOS fragmenta los archivos durante su creación, y los adapta a los huecos que encuentra. Aunque esta fragmentación reduce el acceso a un archivo sensiblemente, casi en cada proceso de lectura o escritura se ha de posicionar de nuevo el cabezal de lectura/escritura de la unidad. En caso de un almacenamiento continuo, este no sería el caso ya que un archivo ocuparía siempre pistas completas en el disquete o disco duro, que habitualmente se pueden leer de forma seguida. Para evitar una fragmentación excesiva del archivo, se resumen varios sectores contiguos en un clúster. Esto garantiza que en cualquier caso los sectores de un clúster contienen un trozo de un archivo, y se pueden leer en secuencia. Si se trabajara de forma orientada solo a sectores, un archivo de 24 sectores se podría colocar en un máximo de 24 sectores, posiblemente muy dispersos, de modo que el cabezal de lectura y escritura se habría de posicionar de nuevo un máximo de 24 veces para poder leer el archivo completo. En el principio del clúster se ahorra un montón de tiempo, ya que el archivo, en una distribución de 4 sectores por clúster solo se distribuye en seis clústers de modo que el cabezal de lectura/escritura sólo se habría de posiciones de nuevo un máximo de seis veces. Aunque todo esto tiene un inconveniente: dado que a un archivo siempre se le asigna un mínimo de un clúster, se puede llegar a desaprovechar mucho espacio. Hay que pensar por ejemplo en el fichero AUTOEXEC.BAT, que seguramente no medirá más de 150 bytes. Para su almacenamiento hay suficiente con un sector, y aún así se desperdician casi 400 bytes. En el caso de un agrupamiento de 4 sectores ocupa un clúster entero con 2048 bytes, con lo que quedan sin utilizar 1898Bytes del disco. En un disco duro con varios cientos de archivos pequeños, se juntan unas bonitas sumas de espacio desaprovechado. Además, la agrupación no nos puede ocultar que la fragmentación de archivos reduce sensiblemente la velocidad de acceso durante la lectura o la escritura. No en vano, los llamados defragmentadores tienen tanto éxito, y se han convertido en una parte fija de productos como PC-Tools o Norton Utilities. No tienen otra tarea que eliminar la fragmentación de archivos, reorganizando el medio y colocando todos los archivos en clústers físicamente adyacentes.

Estructura del FAT

El tamaño de las diferentes entradas de esta tabla en las versiones 1 y 2 del DOS es habitualmente 12 Bits. A partir de la versión del DOS 3.0, su tamaño depende del número de clústers. Si son más de 4096, se emplean 16 Bits, de lo contrario los habituales 12 Bits por entrada. Ya que con la anchura de la entrada de la FAT también aumenta el número clústers direccionables y con ello el tamaño del volumen que se puede gestionar

de esta forma. Al fin y al cabo, con una FAT de 12 Bits se puede gestionar justo 4096 clústers, y esto corresponde, con una agrupación de 4 sectores por clúster a una capacidad de 8 Mbytes. Naturalmente se podría expandir esta capacidad resumiendo más sectores en un clúster, pero esto no parece muy recomendable, en vistas al desperdicio que se hace de espacio de disco. Por ello, en los PC's modernos con discos duros de 40 Mbytes o más, sólo se encuentran FAT's de 16 Bits que permiten el direccionamiento de un máximo de 65536 clústers. Ya que la cantidad de Bits por FAT es un requisito para la interpretación correcta de la FAT, se ha de averiguar antes de acceder a la misma. También aquí nos pueden ayudar las informaciones del BIOS Parameter Block (BPB) en el sector de arranque. A partir de la posición de memoria 13h se encuentra el número total de sectores en el volumen descrito. Simplemente se ha de dividir ésta por el número de sectores por clúster, para obtener el número de clústers del volumen. Las dos primeras entradas de la FAT están reservadas exclusivamente y no tienen nada que ver con la ocupación de los diferentes clústers. Según el tamaño de la diferentes entradas de la FAT, se disponen de 24Bits (3Bytes) o 32Bits (4Bytes). El primer bytes contiene el llamado Media-Descriptor, mientras que los otros bytes suelen estar ocupados con el valor 255. El Media-Descriptor, que también se puede encontrar en el BPB en la dirección 15h, devuelve informaciones sobre el formato del volumen y de soporte de datos. Son posibles los siguientes códigos:

Códigos del Media-Descriptor:

Código	Medio
F0h	31/2-Disquete, 2 caras, 80 pistas, 18 sectores por pista
	31/2-Disquete, 2 caras, 80 pistas, 36 sectores por pista
F8h	Disco duro
F9h	51/4-Disquete, 2 caras, 80 pistas, 15 sectores por pista
	31/2-Disquete, 2 caras, 80 pistas, 9 sectores por pista
FAh	51/4-Disquete, 1 caras, 80 pistas, 8 sectores por pista
	31/2-Disquete, 1 caras, 80 pistas, 8 sectores por pista
FBh	51/4-Disquete, 2 caras, 80 pistas, 8 sectores por pista
	31/2-Disquete, 2 caras, 80 pistas, 8 sectores por pista
FCh	51/4-Disquete, 1 caras, 40 pistas, 9 sectores por pista
FDh	51/4-Disquete, 2 caras, 40 pistas, 9 sectores por pista
FEh	51/4-Disquete, 1 caras, 40 pistas, 8 sectores por pista
FFh	51/4-Disquete, 2 caras, 40 pistas, 8 sectores por pista

Varias copias de la FAT

Como el enlace de los diferentes clusters en los que se divide un archivo se realiza exclusivamente a través de la FAT, en el sistema de archivos del DOS tiene un papel muy importante. Éste se articula en forma de negativa siempre que un fallo del hardware, o por un virus, le cause daños a la FAT. En principio todos los archivos y sub-directorios siguen existiendo, pero son inalcanzables para el usuario, ya que el DOS no puede juntar los sectores como un todo.

El DOS le permite a un programa de formateo crear no sólo una, sino varias copias idénticas de la FAT. Su número se ha de guardar en el sector de arranque en la dirección de desplazamiento 10h. Si el DOS encuentra uno de estos medios, cuida todas las copias de la FAT simultáneamente, así que guarda allí los nuevos clusters ocupados o liberados al crear o borrar archivos.

Esto ofrece la ventana de que se puede sustituir la FAT primaria en caso de defecto por una de sus copias, para evitar la pérdida de datos. Esto es lo que hace el comando CHKDSK del DOS, que comprueba la integridad de la FAT primaria sustituyéndola en caso de error por una de sus copias.

4.-El directorio raíz

En el sistema de archivos del DOS, después de la FAT y sus copias viene el directorio raíz. Se compone de entradas de 32 bytes, que reflejan todas las informaciones relevantes sobre los archivos y sub-directorios allí contenidos. A ello también pertenece el nombre, tamaño, atributos, y fecha y hora de la última modificación o creación del archivo, y sobre todo la posición del archivo, es decir, el número de su primer cluster.

La cantidad máxima de las entradas en el directorio raíz se limita por un tamaño, que se fija en el sector de arranque a partir de la dirección 11h. El directorio raíz representa una estructura de datos estática, que no crece si se guardan más y más archivos o sub-directorios en el disco. Por ello puede generarse el caso de que el directorio raíz esté lleno y que al copiar archivos o crear sub-directorios aparezca el mensaje correspondiente en la pantalla.

Al formatear un volumen se selecciona el tamaño del directorio raíz en relación al tamaño del volumen, de forma que este caso ocurre pocas veces, a menos que se quieran guardar todos los programas de aplicación en el directorio raíz de un disquete o disco duro.

La cantidad de entradas del directorio raíz resulta de la multiplicación de los sectores reservados para él por 16, ya que en un tamaño de entrada de 32 Bytes se pueden colocar 16 entradas de directorio por sector.

La estructura de entradas de directorio

La siguiente tabla muestra la estructura de una entrada de directorio que puede describir tanto archivos como sub-directorios.

Desplazamiento	Contenido	Tamaño
+00h	Nombre del archivo	8 Bytes
+08h	Extensión del archivo	3 Bytes
+0Bh	Atributos del archivo	1 Byte
+0Ch	Reservado	10 Bytes
+16h	Hora de la última modificación	1 Word
+18h	Fecha de la última modificación	1 Word
+1Ah	Primer cluster del archivo	1 Word
+1Ch	Tamaño del archivo	2 Words
Longitud total		32 Bytes

Los primeros ocho Bytes contienen el nombre del archivo descrito. Si es más corto de ocho letras, se rellenan el resto de caracteres con espacios (carácter ASCII 32). Pero si la entrada del directorio no describe ningún archivo, sino un sub-directorio o etiqueta de volumen, se indica aquí el nombre de directorio o volumen. Si la entrada de directorio está vacía, o no describe un archivo de ninguna otra forma, el primer Byte del nombre de archivo contiene un código especial.

Código	Significado
00h	Última entrada del directorio
05h	El primer carácter del nombre de archivo tiene el código ASCII E5h.

2Eh	El archivo se refiere al directorio actual (.). Si el segundo carácter también es 2Eh, se refiere al directorio padre (..).
E5h	El archivo fue borrado

El segundo campo acoge la extensión del archivo, que está compuesta por tres caracteres. También ella se llena de espacios si no contiene tres caracteres. El punto entre el nombre del archivo y la extensión, que se ha de indicar en el DOS para la separación del nombre y la extensión, no se guarda. En sub-directorios y etiquetas de volumen este campo habitualmente permanece vacío y por ello se rellena con espacios. Solo muy pocos usuarios saben que también se puede dar una extensión a los nombres de directorio.

A continuación viene el campo de atributos que mide un Byte. Este campo indica si la entrada contiene un directorio, un archivo o una etiqueta de volumen. También podemos saber qué atributos tiene el archivo o directorio (Oculto, de Sistema y de Solo lectura).

Todo esto lo podemos comprobar en la siguiente figura:

Bits	Significado
0 (LSB)	1=Solo lectura. El archivo no puede ser eliminado ni sobrescrito.
1	1=Archivo oculto. El archivo habitualmente no se visualiza en un listado de archivos.
2	1=Archivo de sistema. El archivo forma parte del sistema operativo.
3	1=Etiqueta de volumen. Contiene el nombre de un volumen.
4	1=Sub-directorio.
5	1=Archivo.
6, 7 (MSB)	No utilizados

Mientras que el significado de los bits 0 a 4 debería quedar claro, el significado del bit 5 necesita una explicación adicional. El nombre de bit de archivo se lo debe a la circunstancia de que se emplea cada vez que hay que hacer un Backup. Cada vez que se crea o modifica un archivo, el DOS pone este bit automáticamente a 1. Si entonces se ejecuta un programa de copia de seguridad (por ejemplo el programa BACKUP del DOS), guarda el archivo y pone el bit de archivo a cero de nuevo. En la siguiente llamada puede saber con ayuda de este bit, si el archivo ha sido modificado desde su última llamada. Si el bit de archivo sigue teniendo el valor 0, no es así, y el archivo no se ha de guardar de nuevo. Si el bit de archivo, por el contrario, está a 1, el archivo ha sido modificado, y debería ser guardado de nuevo.

Detrás del campo de atributos se encuentra un campo reservado, que el DOS necesita para operaciones internas. Su significado no ha sido documentado oficialmente por Microsoft.

El campo de fecha y hora indica el momento en que se creó o modificó por última vez un archivo. Ambos están guardados como Word, y representan un campo de bits con diferentes informaciones.

Codificación de la hora	
Bits	Significado
0 – 4 (LSB)	Segundos / 2
5 – 10	Minuto
11 – 15 (MSB)	Hora
Codificación de la fecha	
Bits	Significado
0 – 4 (LSB)	Día del mes

5 – 8	Mes
9 – 15 (MSB)	Año (relativo a 1980)

A la hora y la fecha le siguen el campo decisivo, que indica el número de cluster que contiene los primeros sectores del archivo. Con ello también se apunta a la entrada de la FAT, mediante la cual se puede averiguar el siguiente cluster de un archivo. Por favor tenga en cuenta que este campo no tiene ninguna importancia en una etiqueta de volumen, ya que no está enlazado con ningún archivo, sino que sólo indica el nombre del volumen. La cosa es ligeramente distinta en el caso de subdirectorios, como se demostrará a continuación. El tamaño de archivo en Bytes se guarda como DWord y sigue al número de clusters. El Lo-Word le precede al Hi-Word como siempre. Mediante la siguiente fórmula se puede calcular el tamaño de archivo con ayuda de estos dos Words:

Tamaño de archivo = Word1 + Word2 * 65536

Los sub-directorios

Si el bit 4 del Byte de atributos está activado, con esta entrada se está describiendo un subdirectorio. En este tipo de entradas el campo con el nombre de archivo y su extensión contiene el nombre de subdirectorio. El campo de fecha y hora contiene el momento de su creación. La longitud de un subdirectorio siempre se indica con 0.

El campo que normalmente apunta al primer cluster del archivo indica aquí el cluster que contiene las entradas de directorio de este subdirectorio. Ya que los subdirectorios se guardan como los archivos normales los clusters cualesquiera, que no son necesariamente contiguos, se encadenan al igual que los clusters de un archivo, mediante las entradas de la FAT.

Allí se guardan las mismas entradas de 32 Bytes que también se pueden encontrar en el directorio raíz de un volumen. Al contrario que el directorio raíz, los subdirectorios se gestionan de forma dinámica. Si se crea un subdirectorio, DOS reserva de momento 1 cluster, y crea en el entradas de directorio vacías. Estas entradas acogen la descripción de archivos y subdirectorios, que cree el usuario. Si se han llenado todas las entradas, en la siguiente creación de un archivo o de un subdirectorio se abre un nuevo cluster para acoger más entradas, y se enlaza con el primer cluster mediante la FAT.

Por otra parte; el DOS vuelve a liberar clusters de directorio si se quedan vacíos, porque se han borrado los archivos que había en ellos. Y lo mismo es lo que hace el DOS en la gestión de subdirectorios que se crean dentro de otros subdirectorios. El número de entradas en los subdirectorios no está limitado, al contrario que en el directorio raíz, sino que se adapta a las necesidades del usuario.

Durante la creación de un subdirectorio se crean directamente dos entradas con los nombres «. » o «.. », que no se pueden eliminar, y que seguramente ya habrá visto al observar listados de directorio. No se pueden borrar y sólo desaparecen al deshacer el subdirectorio completo con ayuda del comando RD.

La primera de las dos entradas apunta al subdirectorio actual, su campo de cluster contiene el número del primer cluster del subdirectorio actual. La segunda= entrada apunta al llamado directorio-padre, que en el árbol de directorios se encuentra en un nodo antes que el directorio actual. Si el directorio-padre es el directorio raíz, el campo de clusters contiene el valor 0. Mediante esta entrada se puede recorrer el camino de vuelta al directorio raíz, ya que con cada directorio-padre, uno se acerca al directorio raíz.

En el método descrito anteriormente para crear y gestionar subdirectorios se puede ver que el DOS realiza una separación de los diferentes archivos en una unidad de almacenamiento según sus directorios, pero que no almacena los archivos de un subdirectorio en una zona de disco particular, sino que los reparte por todo el disco.

Codificación de las entradas de la FAT

A lo mejor nos preguntamos por qué las diferentes entradas de la FAT tienen una anchura de 12 o 16 Bits, si sólo se ha de mostrar si un clúster está ocupado o no. Para ello, basta con un Bit, que podría contener un 1 si el clúster asociado está ocupado, o un 0 indicando que el clúster está libre.

Pero las diferentes entradas de la FAT no sólo indican si el clúster está ocupado o libre sino que a través de ellas se guardan los diferentes clústers en que se guardarán los datos de un archivo. Y esto se hace así: mediante una entrada de directorio de un archivo, el DOS averigua en qué clúster están guardados los primeros sectores de un archivo. El número de este clúster corresponde con el número de la entrada correspondiente de la FAT. En esta entrada de la FAT está el número de clúster en el que están guardados los siguientes sectores del archivo. Y de la entrada FAT de este clúster, se puede averiguar de nuevo en qué clúster están los siguientes sectores de archivo, y así sucesivamente, hasta alcanzar el último clúster de un archivo. Sin embargo, los números de clúster no se refieren al inicio del volumen, sino al inicio de la zona de datos, como veremos más adelante. Para aclararlo, un pequeño ejemplo en este punto: Supongamos que a la entrada de directorio de un archivo se nos indica que el inicio del archivo está en el cuarto clúster del volumen. Para obtener ahora el número del siguiente clúster del archivo, se ha de leer la cuarta entrada de la FAT (atención: una entrada de la FAT puede tener 12 o 16 Bits). En él está el número del siguiente clúster del archivo.

Se crea una cadena mediante la cual se pueden localizar los diferentes clusters del archivo en el orden adecuado. A través del número de cluster resulta el número lógico de los diferentes sectores, que DOS ha de comunicarle al controlador de dispositivos si quiere leer o escribir estos sectores. Al fin y al cabo, los controladores de dispositivos del DOS no trabajan a nivel de clusters, sino a nivel de sectores. La conversión entre clusters y sectores lógicos es muy sencilla, ya que el número de cluster simplemente se ha de multiplicar con el número de sectores por cluster.

La entrada de la FAT que corresponde con el último cluster de un archivo naturalmente ha de contener un código especial, que le señale al DOS que el archivo termina aquí. Esto se señala con un número de cluster que en una FAT de 12 bits sea mayor que FF8h y en una FAT de 16 bits mayor que FFF8h.

Pero estos no son los únicos números de cluster que se reservan para fines especiales, como se muestra en la siguiente tabla:

Código de cluster en FAT de 12 bits	Significado
000h	Cluster libre
FF0h – FF6h	Cluster reservado
FF7h	Cluster defectuoso, no se utiliza
FF8h – FFFh	Último cluster de un archivo
xxxh (Cualquier otro valor)	Siguiente cluster de un archivo
Código de cluster en FAT de 16 bits	Significado
0000h	Cluster libre
FFF0h – FFF6h	Cluster reservado
FFF7h	Cluster defectuoso, no se utiliza
FFF8h – FFFFh	Último cluster de un archivo
xxxxh (Cualquier otro valor)	Siguiente cluster de un archivo

Así por ejemplo, los números de cluster FF0h a FF6h o FFF0h a FFF6h están por clusters reservados, indican que esos clusters se usan para el directorio raíz del volumen, que tiene una posición y tamaño prefijados y cuyos clusters no se han de encadenar mediante la FAT.

Un significado especial también lo tiene el código de clúster FF7h o FFF7h. Marca un cluster en el que se encuentren sectores defectuosos, para que estos no se empleen para guardar archivos, evitando así una posible pérdida de datos.

Los clusters no ocupados finalmente se marcan con el código de cluster 0h. Este código también está reservado para el primer cluster del volumen, pero al igual que el primer cluster no representa sectores, ya que allí se encuentra almacenado el Media-Type del volumen. Esto también es la razón por la que el número de cluster, al hacer la conversión de números de cluster a números de sector, se ha de reducir en dos, como veremos después.

5.-La zona de datos

Volvamos a la estructura del sistema de archivos del DOS. A1 directorio raíz le sigue la zona de datos cuyo cluster se emplea para guardar archivos y subdirectorios. Ocupa el espacio libre completo del volumen,

Para cada cluster en esta zona naturalmente existe una entrada en la FAT, que se puede modificar mediante la manipulación de archivos en esta zona. Si por ejemplo se aumenta un archivo, el DOS reserva un cluster aún libre, para guardar más datos del archivo. La entrada de la FAT del último cluster contenía hasta ahora la marca de fin, pero ahora apunta al nuevo cluster, que a su vez contiene la nueva marca de fin. A1 buscar un cluster que deba acoger los nuevos datos del archivo, las versiones 1 y 2 del DOS buscaban en la FAT desde el principio, y reservaban el primer cluster que no estuviera marcado como reservado.

A partir de la versión 3 se utiliza un método algo más decidido, para seleccionar un cluster que a ser posible esté en las cercanías del otro cluster, para mantener mínimo el tiempo de acceso.

Para aclarar la interrelación de las diferentes estructuras de datos que forman el sistema de archivos del DOS, queremos seguir el camino con ustedes que lleva desde la FAT a los diferentes sectores de un archivo.

Comencemos en el punto en que el DOS busca en la entrada de directorio de un archivo el primer cluster, y con ello la entrada de la FAT para el siguiente cluster. Desde este valor de cluster ahora hay que calcular el sector correspondiente. Para ello se ha de restar 2 al número de cluster, concretamente porque las dos primeras entradas de la FAT se encuentran ocupadas por el Media-Descriptor, y a causa de ello la segunda entrada de la FAT corresponde al cluster número cero del medio. Después se ha de multiplicar el número de clusters con la cantidad de sectores por cluster. Estas informaciones el DOS las puede obtener del BIOS-Parameter-Block, que ha cargado junto con el sector de arranque.

Pero aún con esto el DOS no tiene el número de sector buscado. Ya que los números de clusters en la entrada de directorio y en la FAT no se refieren al primer sector del volumen, sino al primer sector de la zona de datos. Pero esta no comienza en el sector cero, sino detrás del directorio raíz. Por ello primero se ha de calcular el número de sector lógico en el que comienza la zona de datos, que se ha de añadir al número del sector previamente calculado.

Con qué sector lógico comienza la zona de datos, se puede calcular con ayuda de las informaciones del BIOS-Parameter-Block. Como sabe, al sector de datos del volumen le preceden primero el sector de arranque, la FAT y sus copias, y finalmente el directorio raíz. Por ello se ha de calcular la longitud de estas zonas y sumarlas. La cantidad de sectores reservados (incluido el sector de arranque) el DOS lo toma de la posición de memoria 0Eh del sector de arranque. Allí se encuentra en la posición de memoria 16h la cantidad de sectores por FAT. Se ha de multiplicar con la cantidad de FAT de la posición de memoria 10h. A ello se suma la cantidad de sectores reservadas.

Finalmente el DOS necesita la cantidad de sectores que utiliza el directorio raíz. Por ello toma del Word a partir de la dirección de offset 11h en el sector de arranque, la cantidad de entradas del directorio raíz. Este

valor se ha de multiplicar por 32 (Bytes por entrada) y después dividir entre 512 (Bytes por sector). El resultado se ha de sumar a la longitud del sector de arranque y al tamaño total de la FAT, para obtener finalmente el número del primer sector en la zona de datos.

Naturalmente el DOS no realiza este cálculo en cada acceso a un volumen, sino sólo al arrancar o cambiar el medio. En una unidad de disquetes este cálculo se ha de hacer a cada cambio de disco.

Dado que el DOS suma el número del primer sector de la zona de datos al primer número de sector del archivo direccionado, obtiene su número de sector lógico, que puede pasarle al controlador de dispositivo para acceder al archivo.

Si no se ha de acceder sólo al principio del archivo, sino también a los siguientes clusters, el DOS ha de leer el número del siguiente cluster de la FAT. Cómo ocurre esto depende mucho de la anchura de las diferentes entradas de la FAT. La lectura del siguiente cluster es especialmente fácil si se trata de una FAT de 16 bits. En ese caso sólo se ha de multiplicar el número de cluster con 2. El resultado de esta multiplicación representa la dirección de offset, relativa al inicio de la FAT en el disco, en la que se puede encontrar el siguiente cluster.

Si el DOS está tratando con una FAT de 12 bits, el número de cluster se ha de multiplicar por 1,5. La parte entera de este producto se utiliza como offset en la FAT, y se lee el Word de esta posición de disco. Si el producto es entero, el Word leído se ha de operar con el valor 0FFFh la operación lógica Y, para obtener el número de siguiente cluster. Pero si el producto no es entero, la operación con Y no se realiza, y en vez de ello el Word se desplaza 4 bits a la derecha, es decir que se divide entre 16.

En ambos casos el DOS no ha de cargar primero la FAT en la memoria, ya que siempre tiene disponible una copia de esta estructura de datos, para ahorrar tiempo.

Ahora, el DOS conoce el siguiente cluster del archivo. Si en una FAT de 12 bits se encuentra en la zona entre FF8 h y FFFh, y en una FAT de 16 bit en la zona entre FFF8h y FFFFh, se alcanzó el final del archivo. De lo contrario se repite el proceso completo.

6.-Formatos de disquetes

Mientras que el DOS le deja mano libre a los programas de formateo de discos duro, en los disquetes prevé determinados formatos, que determinan exactamente las diferentes características de un volumen. Disquetes con otros formatos no se pueden procesar así como así por el DOS, si no se le ayuda con un controlador de dispositivo especial.

La siguiente tabla muestra los datos del formato para disquetes de 5,25":

Introducido en versión del DOS	1.0	1.10	2.0	2.0	3.0
Media-Descriptor	FEh	FFh	FCh	FDh	F9h
Número de cabezales lectura/escritura	1	2	1	2	2
Número de pistas por cabezal	40	40	40	40	80
Número de sectores por pista	8	8	9	9	15
Número de bytes por sector	512	512	512	512	512
Número de sectores por cluster	1	2	1	2	1
Número de sectores reservados	1	1	1	1	1
Número de sectores por FAT	1	1	2	2	7
Número de FAT's	2	2	2	2	2
Sectores en directorio raíz	4	7	4	7	14

Entradas en el directorio raíz	64	112	64	112	224
Número total sectores	320	640	360	720	2400
Sectores libres para archivos	313	630	351	708	2371
Número clusters	313	315	351	354	2371
Capacidad total	160 Kb.	320 Kb.	180 Kb.	360 Kb.	1,2 Mb.
Capacidad total para archivos	156 Kb.	315 Kb.	175 Kb.	354 Kb.	1,185 Mb.

La siguiente tabla muestra los datos del formato para disquetes de 3,5":

Introducido en versión del DOS	3.0	3.30
Media-Descriptor	F9h	F0h
Número de cabezales lectura/escritura	2	2
Número de pistas por cabezal	80	80
Número de sectores por pista	9	18
Número de bytes por sector	512	512
Número de sectores por cluster	2	1
Número de sectores reservados	1	1
Número de sectores por FAT	3	9
Número de FAT's	2	2
Sectores en directorio raíz	7	14
Entradas en el directorio raíz	112	224
Número total sectores	1440	2880
Sectores libres para archivos	1426	2847
Número clusters	720	2880
Capacidad total	720 Kb.	1,5 Mb.
Capacidad total para archivos	713 Kb.	1,44 Mb.

7.Bibliografía

- PC Interno 2.0, Michael Tischer, Ed. Marcombo 1995
- Programación en Turbo Pascal , Luís Joyanes Aguilar, Ed. Mc Graw Hill 1993

Descriptor de medios

Entrada directorio del archivo Format-EXE

Entrada

FAT 12 Bits

00

`F'

`O'

`R'

`M'

0

FDFh

1

FFFh

04

`A'

`T'

32

32

2

17Fh

3

002h

08

`E'

`X'

`E'

0

4

5

006h

0C