Operating Systems

CS-384

File Systems

NTFS and FAT32

Submitted To: Dr. Chris Taylor

Submitted By: Aditya Sitani

Date: 02/3/2003

Table of Contents

Tables and Figures

Figures:

Tables:

Introduction

The world is revolving around computers and almost all the work in today's world is dependent on computers. An operating system is a program that acts as an intermediary between a computer user and the computer hardware. Just imagine how crazy a user would go if the computer would not be able to store data permanently, and all the data would get erased as soon as the system was rebooted or there was a power failure. Permanent data is stored on secondary storage devices like fixed disks, floppy disks, magnetic tapes and other types of storage devices. The file system management resides permanently on a secondary device. In an operating system, the file system manages the information that resides on the secondary storage device. A file system is the organization method of data on a hard disk volume. It consists of two distinct parts: collection of files, each storing related data, and a directory structure, which organizes and provides information about all the files in the system. Some file systems have a third part, partitions, which are used to separate physically or logically large collection of directories. The two most popular file systems used today are FAT32 (File Allocation Table) and NTFS (New Technology File System). Both the file systems have their own advantages and disadvantages. This report will discuss about these two different file systems and their working. To understand the concept of file systems, one should have a basic idea about physical disks.

Physical Disk

A hard disk uses round, flat disks called platters, coated on both sides with a special media material designed to store information in the form of magnetic patterns. These platters are mounted by cutting a hole in the center and stacking them onto a spindle. The platters rotate at a high speed, driven by a special spindle motor connected to the spindle. Special electromagnetic read/write devices called heads are mounted onto the platters and used to either record information onto the disk or read information from it. The sliders are mounted onto arms, all of which are mechanically connected into a single assembly and positioned over the surface of the disk by a device called actuators. A logic board controls the activity of the other components and communicates with the rest of the PC. A disk structure diagram is shown in figure 1.

Each surface of each platter on the disk can hold tens of billions of individual bits of data. Each platter has two heads, one on the top of the platter and one on the bottom, so a hard disk with three platters (normally) has six surfaces and six total heads. Each platter has its information recorded in concentric circles called tracks. Each track is further broken down into smaller pieces called sectors, each of which holds 512 bytes of information. Sectors (or blocks) are often grouped together to form a larger unit of storage called a cluster. Each of the file system uses different cluster size depending on the size of the partition.

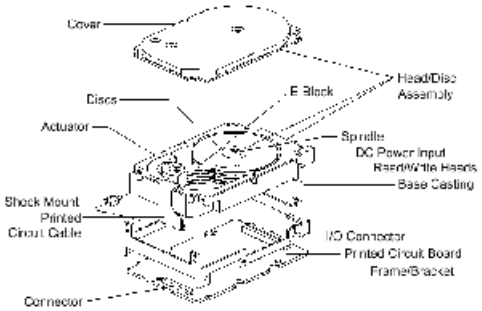

Fig: 1 Hard Disk Structure

Files

Files concept

The data is stored permanently in different storage media, such as magnetic tapes and optical disks. The operating system provides a uniform logical view of information storage. The operating system abstracts from the physical properties of its storage device to define a logical unit (file). Files are mapped by the operating system onto a physical device. The operating system attaches attributes to the file in order to keep track of and manage it.

File Attributes

The following are the most common file attributes:

Name: In order for us to use files and distinguish between them, files are given human readable names (an alpha numeric string), e.g. aditya.doc. Some systems such as UNIX use case sensitive names.

Identifier: This value allows the file to be kept track of within the file system. This field generally consists of an unseen numerical value, which the operating system assigns.

Type: This information is needed for those systems that support different file types. The type is decided on the basis of the extension of the file. It is responsible for associating the file with the right application, necessary for systems supporting multiple file types (e.g. ".txt" for text file, supported by a lot of text editors).

Location: This represents a pointer to the location on the device where the file is located.

Size: This information is to keep track of the current size of the file in bytes, words or blocks.

Protection: This attribute is associated with ownership, and privileges, in other words which user or user groups can read, write or execute the file.

Time and Date: This information keeps track of creation, last modification and last use.

File Operation

There are six different operations that can be performed on a file. The operating system provides system calls to create, write, read, reposition, delete, and truncate files. These operations are described below:

Creating a file: A file is created in two steps. First, the space in the file system is found for the file and then an entry for the new file is made in the directory.

Writing to a file: The operating system passes the name of the file along with the data to be written using the write system call. Once the file is located, new data is added to the file and the write pointer is moved to the next location to write to. The write pointer is updated whenever a write occurs so that no data will be over written.

Reading from a file: A system call that specifies the name of the file to be read and where in the memory the next block of the file should be put. Again, the directory is searched for the associated directory entry, and the system needs to keep a read pointer to the location in the file where the next read is to take place.

Repositioning within a file: While repositioning a file, the directory is searched until the appropriate file is found. The pointer to the current file position is changed to some value desired by the operation. No input/output is required by this operation.

Deleting a file: The directory is searched for the named (e.g. file to be deleted) file. The space that the file has is then released, and the directory entry is erased.

Truncating a file: This function is used when the length of a file is to be changed and Other attributes have to remain the same. This function is important because without it, the operating system would have to recreate a file every time it is modified and delete the old one.

FAT32

The FAT (File Allocation Table) file system originated in the late 1970s and early 1980s and was the file system supported by the MS-DOS operating system. It was originally developed as a simple file system suitable for floppy disk drives less than 500K in size. Over time, it has been enhanced to support larger media. Currently there are three FAT file system types: FAT12, FAT16 and FAT32. The basic difference in these FAT sub types, and the reason for the names, is the size, in bits, of the entries in the actual FAT structure on the disk.

Origin Of FAT32

During late 1970s and early 1980s when FAT was introduced, MS-DOS was a 16-bit operating system with 16-bit addressing. The disk was split into 65,536 sectors. The maximum size of a hard drive with this system was 32MB if each sector was 512KB. This limitation was challenging since, if one had to make a larger hard drive, it would have to be partitioned equal to or less than 32MB. Thus, the need for expansion was realized.

FAT12 and FAT16 were introduced in response to the above problem. This system used clusters (group of sectors) instead of sectors when addressing. Now, instead of storing sector addresses in the FAT, cluster addresses would be stored. With this enhancement, FAT could now handle 65,535 clusters, rather than sectors, and thus could handle larger hard drives. FAT16 was used with windows 95 operating system, and could provide support up to 2GB(gigabytes). But the demand of an even larger drive was in the market and which would require further improvement over FAT16. With this, came up the FAT32, with the second edition of Windows 95. FAT32 also allocated the clusters in the same way as FAT16, but it used 32 bit addressing. This addressing size supported a limit of 2TB (terabyte) partitions.

Disk Structure

The basic FAT32 file system is characterized as file allocation table (FAT), which is really a table that resides at the very “top” of the volume. A section of disk at the beginning of each partition is set aside to contain the table. The table has one entry for each disk block, and is indexed by block numbers. The FAT is used much as a linked list. The directory entry contains the block number of the first block of the file. The table entry indexed by that block number then contains the block number of the next block in the file. The chain continues until the last block, which has a special end-of-file value as entry.

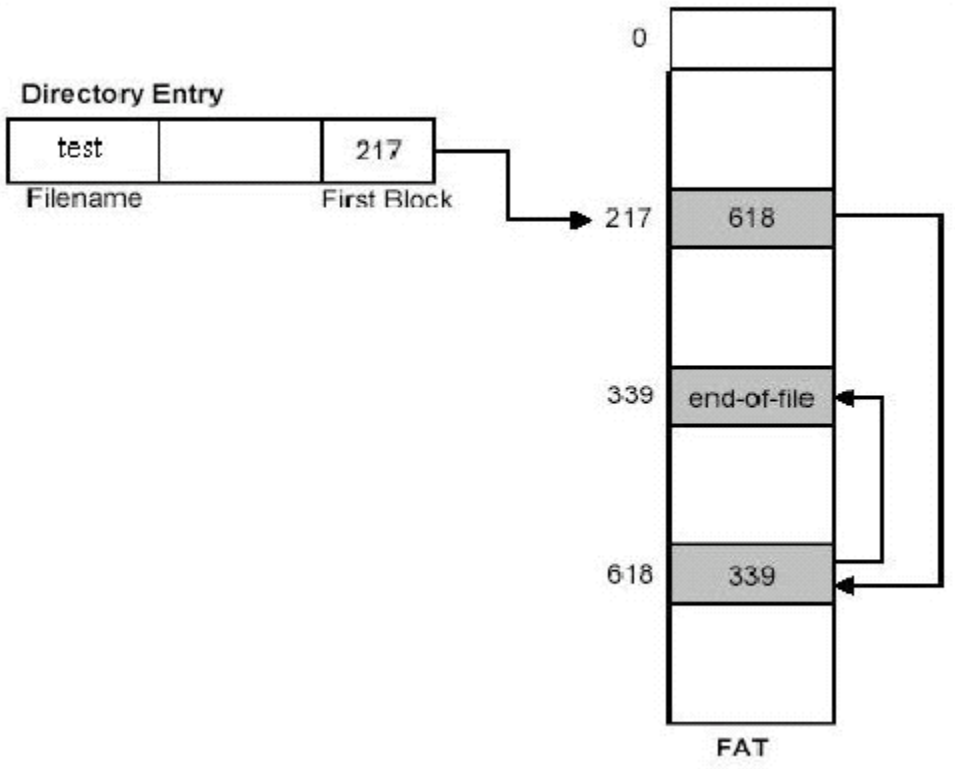

Fig: 2 The FAT file Structure

As can be seen from figure 2, the first block is 217 which again has a pointer to 618 which in turn points to block 339 where the is the end-of-file. The FAT allocation scheme can result in a significant number of disk head seeks, unless the FAT is cached.

FAT Naming convention

FAT uses the traditional 8.3 file naming convention and all filenames must be created with the ASCII character set. The name of a file or directory can be up to eight characters long, then a period (.) separator, and up to a three-character extension. The name must start with either a letter or number and can contain any characters except for the following:

. " / \ [ ] : ; | = ,

If any of the above characters are used, unexpected results may occur. The name cannot contain any spaces.

The following names are reserved:

CON, AUX, COM1, COM2, COM3, COM4, LPT1, LPT2, LPT3, PRN, NUL

Both FAT16 and FAT32 have the capability of VFAT. VFAT is a technical term for a long file name. VFAT allows up to 255 characters for a file name instead of the 8.3 file name as discussed.

Advantages of FAT32

FAT32 provides the following enhancements over previous implementations of the FAT file system:

- It supports up to 2 Terabytes in size.
- Uses space more effectively - FAT 32 uses smaller clusters (e.g. 4kb clusters for drives up to 8GB in size), resulting in 10 to 15% more efficient use of disk space relative to large FAT drives.
- It is more robust and is more flexible - the root directory on a FAT32 drive is now an ordinary cluster chain, so it can be arbitrarily large and located anywhere on the drive. In addition, FAT mirroring can be disabled, allowing a copy of the FAT other than the first one to be active. These features allow for dynamic resizing of FAT32 partitions.

Disadvantages of FAT32

Despite its popularity, the FAT32 File System is not a perfect file system. It suffers from three major problems:

- Fragmentation.
- Storage efficiency - The storage efficiency of the FAT32 File System degrades for larger partitions.
- It is not fault tolerant - The FAT32 File System is a corruptible file system where a computer crash, a hardware malfunction, or a programming glitch can destroy the file system.

NTFS File System

The NTFS File System was created in the early 1990s. Microsoft wanted to create a high-quality, high performance, reliable and secure operating system. All the operating systems Microsoft had before Windows XP or Windows 2000 had the power or features needed for Microsoft to take on UNIX or other serious operating systems. One of the biggest weaknesses of MS-DOS and Windows 3.x was that they relied on the FAT file system. FAT provided few of the features needed for data storage and management in a high-end, networked, corporate environment. To avoid crippling Windows NT, Microsoft had to create for it a new file system that was not based on FAT. The result was the New

Technology File System or NTFS. NTFS is definitely "new" from the standpoint that it is not based on the old FAT file system.

NTFS was designed to meet a number of specific goals. In no particular order, the most important of these are:

- Reliability - NTFS implements specific features to allow important transactions to be completed as an integral whole, to avoid data loss, and to improve fault tolerance.
- Security and Access Control - One of the most important advantages that users gain when choosing the NTFS file system over older file systems such as FAT, is greater control over who can perform what sorts of operations on various data within the file system.
- Storage Efficiency - Again, at the time that NTFS was developed, most PCs used FAT16, which results in significant disk space due to slack. NTFS avoids this problem by using a very different method of allocating space to files than FAT does.
- Breaking Size Barriers - Unlike FAT where maximum partition was up to 4GB, NTFS allows larger partition size of 16 exabyte(10^{18}).
- Long File Name - NTFS allows file names upto 255 characters instead of 8.3 characters limitation of conventional FAT.

NTFS Disk Structure

Formatting a volume with the NTFS file system results in the creation of several system files and the Master File Table (MFT), which contains information about all the files and folders on the NTFS volume.

The first information on an NTFS volume is the Partition Boot Sector, which starts at sector 0 and can be up to 16 sectors long. The first file on an NTFS volume is the Master File Table (MFT).

The following figure (Figure 3) illustrates the layout of an NTFS volume when formatting has finished.

Fig: 3 Layout of an NTFS volume

Partition Boot Sector

The first piece of information on the partition is the boot sector. The partition boot sector contains information, which the file system uses to access the volume. The boot sector is up to 8KB in length. Partition Boot Sector consists of two different primary structures:

- BIOS parameter block - This has information about the partition and about the volume name, size and location of metafiles.

- Volume Boot Code - This is a small block of program code that instructs the system on how to load the operating system.

Master File Table (MFT)

The NTFS file system contains a file called the master file table, or MFT. There is at least one entry in the MFT for every file on an NTFS volume, including the MFT itself. All information about a file, including its size, time and date stamps, permissions, and data content, is stored either in MFT entries, or in space outside the MFT that is described by MFT entries. As files are added to an NTFS volume, more entries are added to the MFT and the MFT increases in size. When files are deleted from an NTFS volume, their MFT entries are marked as free and may be reused. However, disk space that has been allocated for these entries is not reallocated, and the size of the MFT does not decrease. File attributes that do not require much space are stored in the MFT and they are called resident attributes. When all of the information for a file is too large to fit in the MFT file record, some of its attributes are non-resident. The non-resident attributes are allocated one or more clusters of disk space elsewhere in the volume.

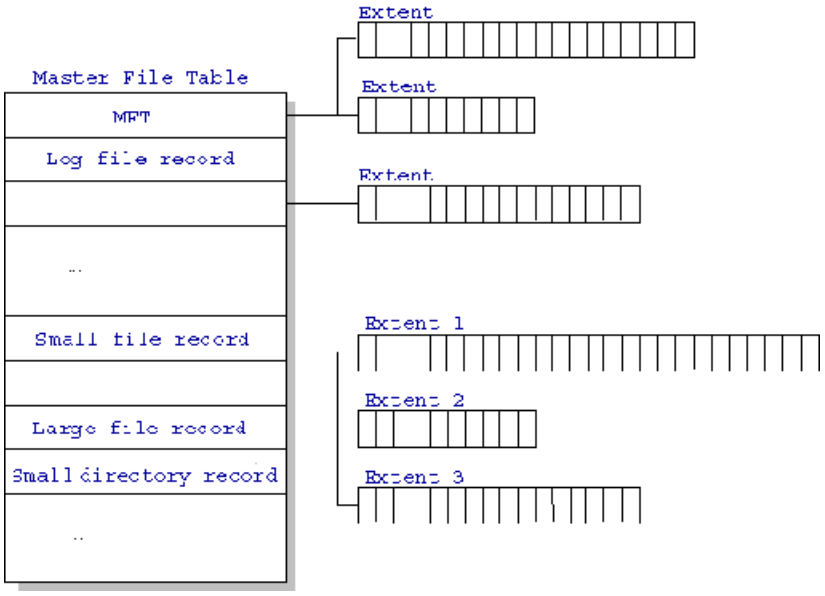

Fig: 4 Master file table

The table below lists all of the file attributes currently defined by the NTFS file system. This list is extensible, meaning that other file attributes can be defined in the future.

Attribute Type	Description
Standard Information	Includes information such as timestamp and link count.
Attribute List	Lists the location of all attribute records that do not fit in the MFT record.
File Name	A repeatable attribute for both long and short file names. The long name of the file can be up to 255 Unicode characters. The short name

	is the 8.3, case-insensitive name for the file. Additional names, or hard links, required by POSIX can be included as additional file name attributes.
Security Descriptor	Describes who owns the file and who can access it.
Data	Contains file data. NTFS allows multiple data attributes per file. Each file typically has one unnamed data attribute. A file can also have one or more named data attributes, each using a particular syntax.
Object ID	A volume-unique file identifier. Used by the distributed link tracking service. Not all files have object identifiers.
Logged Tool Stream	Similar to a data stream, but operations are logged to the NTFS log file just like NTFS metadata changes. This is used by EFS.
Reparse Point	Used for volume mount points. They are also used by Installable File System (IFS) filter drivers to mark certain files as special to that driver.
Index Root	Used to implement folders and other indexes.
Index Allocation	Used to implement folders and other indexes.
Bitmap	Used to implement folders and other indexes.
Volume Information	Used only in the $Volume system file. Contains the volume version.
Volume Name	Used only in the $Volume system file. Contains the volume label.

Table 1: NTFS File Attributes

Metadata

NTFS stores both user data and internal management data, in the form of files. The most important of these are a set of special system files, which are also called metadata files. Metadata files contain internal information about the data in NTFS Volume. These metadata files are automatically formed when the NTFS partition is formatted, and is placed at the top of the partition. The MFT is actually one of these metadata files, but it also contains descriptions of the other metadata files. In fact, the first 16 records of the MFT are reserved for metadata files.

The table below provides the important information about the metadata files, including their English names, file names, MFT record numbers and a brief description of what each does:

$MFT	Itself MFT
$MFTmirr	copy of the first 16 MFT records placed in the middle of the disk
$LogFile	journaling support file (see below)
$Volume	housekeeping information - volume label, file system version, etc.
$AttrDef	list of standard files attributes on the volume
$.	root directory
$Bitmap	volume free space bitmap
$Boot	boot sector (bootable partition)
$Quota	file where the users rights on disk space usage are recorded (began to work only in NT5)

$Upcase	File - the table of accordance between capital and small letters in files names on current volume. It is necessary because in NTFS file names are stored in Unicode that makes 65 thousand various characters and it is not easy to search for their large and small equivalents.

Table 2: Meta Data Table

Features of NTFS

NTFS multiple data streams

NTFS supports multiple data streams, where the stream name identifies a new data attribute on the file. A handle can be opened to each data stream. A data stream, then, is a unique set of file attributes. Streams have separate opportunistic locks, file locks, and sizes, but common permissions.

NTFS Compressed Files

One of the most attractive features of NTFS is file-based compression. With compression it is possible to store most files in a way that they take up less space on the disk than they normally would. Most of the operating systems allow compression for the entire disk volume, but NTFS allows almost all individual files and folders within the NTFS partition to compress. The compression is handled by the operating system during writes, and decompression is automatic whenever an application needs to read the file. Performance when working with compressed files can be degraded compared to regular files, because of the overhead required to compress and decompress the file on a regular basis--it takes CPU time.

NTFS Security and Encryption

All the recourses in a system (e.g. disks, folders, and files) are treated as objects. Any user who wants access to these objects should have the right to do so. This feature cannot be overridden in any way, including attempt to view the volume through another operating system. One of the security features is encryption.

The Encrypting File System (EFS) provides the core file encryption technology used to store encrypted files on NTFS volumes and thus cannot be read by another NT installation. In combination with a standard and very much safe password on the system itself, this possibility provides the safety of files selected by the user. If an intruder tries to access, open, read, copy, move or rename the encrypted file or folder, he/she receives an “access denied” message.

NTFS Sparse File

A sparse file has an attribute that causes the I/O subsystem to allocate only meaningful (nonzero) data. Nonzero data is allocated on disk, and non-meaningful data (large strings of data composed of zeros) is not. When a sparse file is read, allocated data is returned, as it was stored; non-allocated data is returned, by default, as zeros. NTFS deallocates sparse data streams and only maintains other data as allocated. When a program accesses a sparse file, the file system yields allocated data as actual data and deallocated data as zeros.

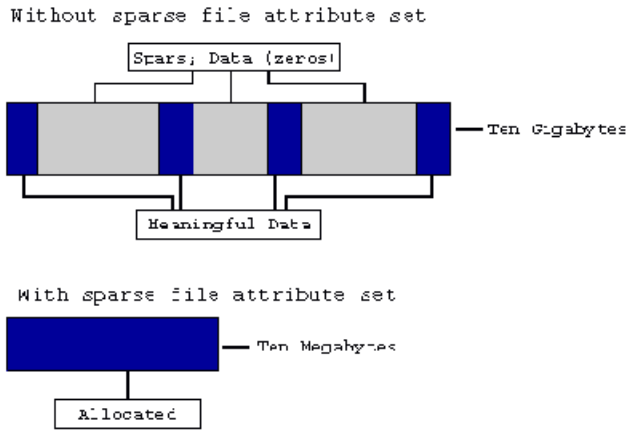

Fig 5: Sparse File

Hard Links

A hard link is when the same file has two names. (e.g. adi.txt and adi1.txt), if a user deletes the first file (adi.txt), the second file (adi1.txt) remains. If he deletes the second file, the first one will remain. Although both the files are same, they are independent of each other.

FAT32 vs. NTFS

Comparing FAT32 and NTFS often becomes quite challenging as both have their own advantages and disadvantages. The table given below shows the pluses and minuses of the widespread file system used (e.g. FAT32 and NTFS) used today.

	FAT32	NTFS
Systems	Windows98, NT5	NT4, NT5

Maximal volume size	Near by unlimited	Near by unlimited
Maximal files count	Near by unlimited	Near by unlimited
File name	Long names (up to 255 chars), system character set.	Long names (up to 255 chars), unicode character set.
File attributes	Basic set	All that programmers want
Security	No	Yes (capability of physical encryption, starting from NT5.0)
Compression	No	Yes
Fault tolerance	low	Fully (automatic)
Economy	Improved (small clusters)	Maximal
Performance	as for FAT, but also additional penalty for big volumes	very effective for all volumes

Table 3: Comparison between FAT32 and NTFS

Conclusion

Different operating systems use different file systems. The two popular systems are FAT32 and NTFS. It is desirable to use NTFS if ones operating system is Windows NT (windows 2000), as the choice of any other file system would limit the convenience and flexibility of the operating system. The decision of choice isn't really all that difficult these days. NTFS has a number of advantages over FAT with regard to file security,

encryption, disk compression, support for large disk, file stream and hard links. To make the computer a multi-boot system, one might want to consider FAT32. If one is concerned about being able to see files across partitions, the shared partitions FAT32 can be made. It can be clearly seen that NTFS has far more advantages over FAT32. The older operating system uses FAT32 file system but with the new and upcoming operating systems, NTFS is getting much more popular.

References

Silberschatz, Galvin, Gagne. *Operating System Concepts*, Sixth Edition. John Wiley & Sons, Inc.

Milenkovic, Milan. *Operating System Concepts and Design.* McGraw – Hill Book Company.

Claybrook, Billy G. *File Management Techniques.* John Wiley & Sons,

Digit- life.com, 1997, "NTFS file system" . Retrieved January 30,2003. http://www.digit-life.com/articles/ntfs/

ntfs.com,2002. Retrived January 21,2003. http://www.ntfs.com

Tech TV inc, 2003, "NTFS vs FAT32" . Retrieved january 21,2003 http://www.techtv.com/screensavers/windowstips/story/0,24330,3201552,00.html

Microsoft Product Support Services, Feb 19,2002. Retrieved January 21,2003. http://support.microsoft.com/default.aspx?scid=kb;EN-US;q100108

The PC Guide, april 17, 2001. "New Technology File System". Retrieved 30,2003 http://www.pcguide.com/ref/hdd/file/ntfs/index.htm