Legacy File Systems

By Mark E. Donaldson

INTRODUCTION

At the BIOS level, a disk partition contains sectors numbered 0, 1, etc. Without additional support, each partition would be one large dataset. Operating systems add a directory structure to break the partition up into smaller files, assign names to each file, and manage the free space available to create new files. The directory structure and methods for organizing a partition is called a File System. Different File Systems reflect different operating system requirements or different performance assumptions.

Unix, for example, has the convention that lowercase and uppercase are different in file names, so "sample.txt" and "Sample.txt" are two different files. DOS and the systems that descend from it (Windows 95, OS/2, and Windows NT) ignore case differences when finding file names. Some File Systems work better on small machines, others work better on large servers. Each partition is assigned a type (in the MBR for primary partitions, in the Extended Partition directory for logical volumes). When the partition is formatted with a particular File System, the partition type will be updated to reflect this choice.

The same hard disk can have partitions with File Systems belonging to DOS, OS/2, NT, and Linux (or other Unix clones). Generally, an operating system will ignore partitions whose type ID represents an unknown file system type. It is fairly easy (given a big enough disk) to install all of the different operating systems and all of the File System types. There are a few rules to make things simple.

Each File System is described in detail in a separate section below:

- **The FAT** File system is used by DOS and is supported by all the other operating systems. It is simple, reliable, and uses little storage.

- **VFAT** is an alternate use of the FAT file system available in Windows 95 and Windows NT 3.5. It allows files to have longer names than the 8.3; convention adopted by DOS. VFAT stores extra information in the directory that older DOS and OS/2 systems can ignore.

- **HPFS** is used by OS/2 and is supported by Windows NT. It provides better performance than FAT on larger disk volumes and supports long file names. However, it requires more memory than FAT and may not be a reasonable choice on systems with only 8 megs of RAM.

- **NTFS** provides everything. It supports long file names, large volumes, data security, and universal file sharing. A departmental NT file server will probably have all its partitions formatted for NTFS. Because the other operating systems cannot use it, NTFS is less attractive on personal desktop workstations or portables.

FILE SYSTEMS AND DISK LETTERS

DOS and Windows 95 can only boot from the C: disk. Technically, the C: letter will be assigned to the first Primary Partition on the first hard disk that has a FAT file system. In no case can DOS boot from a second hard disk or a logical volume in the extended partition. However, if as the system comes up, the DOS boot sector and DOS files turn out to be on the second Primary Partition on the first hard disk, then this will not be a problem so long as the first partition has a non-FAT file system. DOS simply ignores primary partitions that are formatted for other operating systems. Some people exploit this feature. They put an HPFS or NTFS file system on the first Primary Partition, and a FAT file system on the second. This can produce confusion When the other operating system boots up, it will now assign letter C: to its first partition, and the disk that DOS calls "C" will become "D" on the other system. If the two systems share application programs, it becomes very difficult to configure INI files as the drive letter keeps changing back and forth. It is a simpler and safer strategy to accept the view that the first Primary

Legacy File Systems
By Mark E. Donaldson

Partition on the first hard disk should be formatted with the FAT file system and should be the C: drive in every operating system.

CHOOSING A FILE SYSTEM

The performance problems with FAT have been greatly reduced by various strategies to use Cache memory and to periodically DEFRAG the disk. FAT is the only system fully supported by DOS and Windows 95. It is also a perfectly acceptable choice under Windows NT and OS/2. FAT systems require the least memory and are the best choice on small machines. Although it is simpler to manage a few larger volumes, FAT performance degrades with volume size. The distance between the directory and the data increases the disk movement, and larger allocation units waste space. A good rule of thumb would limit FAT volumes to a maximum of 255 megabytes.

FAT has proven to be quite reliable and is fairly immune to damage. When the system crashes, FAT can misplace disk space that was being allocated to a file. CHKDSK (or Microsoft's newer SCANDISK) will recover the missing space. Less frequently a really serious error could leave the same sector of disk space assigned to two different files. Such crosslinked files are damaged, and once this occurs the entire volume is suspect. The preferred recovery would be to back everything up, reformat the volume, and restore the data. Crosslinked files could be produced by a damaged operating system, or by a hardware problem in the disk subsystem itself.

HPFS is supported by OS/2 and Windows NT. Although it is not officially supported by DOS or Windows 95, there are shareware drivers (such as AMOS3) that can provide these systems with at least Read-Only access to HPFS files. Since OS/2 does not support VFAT, it cannot use long file names on a FAT volume. Many OS/2 software packages require long file names. An OS/2 system with enough memory and disk space should have at least one HPFS volume to support such packages.

Only Windows NT can use data on an NTFS volume. NTFS is required to provide full security on an NT File Server, and to support Macintosh datasets. On desktop workstations that run other operating systems as well as NT, NTFS is probably more trouble than it is worth.

A good general principle is to put FAT volumes first on a disk, then HPFS, and finally NTFS. All the systems will see the FAT volumes and will assign them disk letters. With device drivers for DOS, all the system will see the HPFS volumes as well. The NTFS volumes will only be available to Windows NT and will be ignored by the other systems.

THE FAT FILE SYSTEM

- Supported by all operating systems
- Minimal memory use
- Simple and reliable
- NAME.EXT file names
- Less efficient on partitions above 32 meg
- Not suitable for File Servers

The File Allocation Table was designed and coded in Feb., 1976 by a kid named Bill Gates during a five day stay at the Hilton Hotel in Albuquerque. He developed it for a version of Basic that could store programs and data on floppy disks. The FAT design was incorporated by Tim Patterson in an early version of an operating system for the Intel 8086 chip. Gates bought the rights to the system, then rewrote it to create the first version of DOS. As a direct result, Gates is the richest man in America.

Legacy File Systems
By Mark E. Donaldson

The FAT file system is simple and reliable. It does not lose data because the computer crashed in the middle of an update. It does not use a lot of memory. It does, however, do a lot of extra administrative I/O to different areas of the partition. The directory is allocated at the start of the partition and it contains the table of freespace. To write a new dataset, or to add data to an old one, the disk arm must be constantly moved between the location of the directory and the place where the data is being written. Without optimization, a file can end up fragmented into many small pieces.

When the system crashes, no data is lost. However, a FAT system may have removed disk area from the chain of free space, but may not yet have assigned it to any permanent new dataset. The CHKDSK (or on newer systems the SCANDISK) utility examines the FAT table to determine the status of every record on disk. The records which are not part of any dataset may be returned to the free space chain.

After CHKDSK finds unallocated sectors, it asks you whether they should be turned into files. If you were in the middle of creating a new file and the system crashed, and the data that was lost is extremely valuable, you might answer Yes. Then you can scan the recovered file scraps for the information you lost. Otherwise, answer No and the unallocated space will be recycled as freespace.

By design FAT supports a maximum of 64K allocation units. When the disk partition is 32 megabytes or less, then an allocation unit is a 512 byte sector. However, as the disk gets larger, the units get larger. A 64 megabyte disk partition has 1K allocation units. A 128 meg partition has 2K units. A 256 meg partition has 4K allocation units. Each file occupies one or more allocation unit. As the allocation units get large, any large number of small files wastes a lot of disk space.

The classical FAT directory structure (before NT and Windows 95) limits file names to eight characters with a three character extension (as in the name;AUTOEXEC.BAT;). This 8.3 naming convention was borrowed from earlier DEC minicomputers. The FAT structure also maintains for each file a set of attributes (is it a System dataset, should it be Hidden in the DIR display, should it be archived next time the disk is backed up, is it Read-only). There is also a data and time stamp when the file was last changed.

OS/2 allows a FAT file to have additional Extended Attributes. Since there is no room for these attributes in the FAT directory, OS/2 creates a separate hidden file on the disk volume named EA DATA. SF and stores the information there.

FAT has been around for a long time, so its problems have been carefully studied. Every version of DOS comes with a DEFRAG program that will reorganize the directory, files, and freespace to maximize performance. Effective use of RAM as a disk cache can address some of the I/O problems.

THE VFAT FILE SYSTEM

- Windows 95 and Windows NT use of FAT
- Supports long file names

Windows 95 and Windows NT support VFAT. Technically, VFAT is not a new File System. It uses the same directory structure, format, and partition type as ordinary FAT. VFAT is simply a way to store more information in the FAT directory.

The most important feature of VFAT is the ability to store long file names. Since it is built on ordinary FAT, each file has to have an 8 character name and 3 character extension. However, VFAT then allocates additional directory blocks to hold a longer file name. Programs running in DOS, OS/2, and

Linux (and old 16-bit Windows programs) will not see the longer file name. Only WIN32 programs running in NT or Windows 95 can make use of the longer name.

Because VFAT uses the old FAT directory to add some unusual new entries, the VFAT additions can be damaged if the disk is manipulated by a DOS or OS/2 disk utility that does not understand the new structure. Even a simple DEL command under OS/2 for a FAT dataset with a long file name can leave the extra blocks in the directory.

The Windows 95 SCANDISK utility can be run later to audit the VFAT structure and delete the extra directory blocks for files that no longer exist.

THE HPFS FILE SYSTEM

- Supports long file names with mixed case
- Directory positioned throughout the disk
- Much faster to create new files
- Allocates individual sectors (512 bytes)
- Uses more memory, bad for small PC

The High Performance File System was developed by Microsoft for OS/2 1.2, mostly to support the LAN Manager file server. The tables that describe the location of files and freespace are positioned at regular intervals throughout the dataset. New datasets are written where there is a large amount of freespace. This reduces fragmentation and keeps the disk arm from jumping around a large area. HPFS maintains a 512 byte allocation unit no matter how large the volume gets to be.

The HPFS directory allows file names to be long, to have multiple periods, and to have lowercase letters. Unlike NT and Windows 95, however, HPFS does not keep a separate 8.3 file name around for each dataset. If a DOS or Windows program running under OS/2 looks at a directory, it will not see the datasets that have long file names.

HPFS keeps information in a disk cache area of memory until it needs to be written to disk. To be sure that all the information has been properly written to disk, a user should try to shut the system down by command rather than just turning the power off. However, the system will occasionally crash. Any HPFS volume that was in use during a crash is marked dirty. Before it can be used, the next boot of the operating system must run the CHKDSK utility to examine the chains of freespace and file locations to correct any problems. As the disk volumes get larger, the CHKDSK after a crash can take a very long time.

It is difficult to generalize from individual examples, but the author had the experience of trying to install an application with a large number of files on a FAT disk. After 40 minutes the installation was 60% complete and had to be interrupted. Later on, the installation was repeated on an HPFS system and completed in 10 minutes.

The HPFS code and cache use a significant amount of memory. The loss of memory can effect performance, so HPFS is not recommended on small systems (say with 4-8 megabytes of RAM). OS/2 regards HPFS as its native file system and provides full support. Windows NT will read and write files in an HPFS partition, but it will not format new HPFS partitions. Linux has a read-only driver for HPFS.

Although DOS and Windows 95 do not formally support HPFS file systems, there is a very nifty driver named AMOS that provides read-only access to HPFS volumes. AMOS3 is installed as a TSR by the

AUTOEXEC file. It then supports access to the HPFS file structure much as the MSCDEX utility provided read-only access to files on the CDROM.

A FAT file system can be used by every operating system, but only Windows 95 and Windows NT support long file names on it. HPFS has very good performance and allows an NT or OS/2 system to download files with long names that can later be read by a Linux system. Since Unix files on the network all have tediously long file names, this is a particularly useful feature for part-time Linux hackers.

THE NTFS FILE SYSTEM

- Supports long file names
- Access control by directory or file
- Can compress individual files or directories
- Share directories with Mac users from NT Server
- Efficient for larger partitions
- Add space when partition fills up

NTFS is a new file system for Windows NT. NTFS is designed to be all things to all people and to include all the features of every other file system in common use. While HPFS also supports long file names, NTFS support long UNICODE file names. In theory, an NTFS file can have its name in Chinese or Hebrew characters. At the same time, NTFS maintains an 8.3 name for the file so that it can be used by a DOS program. NTFS also supports case-sensitive file access (for Unix programs) and case-insensitive file access (for DOS, OS/2, and Windows programs).

NTFS supports a variety of multi-user security models. There is native Windows NT security established by File Manager based on the groups to which a user's account belongs. NT Advanced Server also supports a Macintosh security model that simulates an Apple File Server. Unix applications will see security that obeys the Posix model. A Windows NT Server cannot share disk space with Macintosh computers unless the volume is in NTFS format. NTFS should also be used for BackOffice applications such as SQL Server and SMS. However, a Windows NT Workstation that does not function as a central server can function quite well with VFAT and HPFS volumes.

NTFS writes updates to a log area of each volume. After a system crash, this log area can be used to cleanup problems almost instantaneously, producing a much faster recovery than with HPFS. NTFS supports volume sets where a single disk letter is associated with a volume created from a number of separate free space areas scattered across several disks. If an NTFS volume fills up, it can be dynamically expanded by adding an extra chunk of free disk space from the same or from another hard disk.

While DOS has drivers that allow an entire disk to be compress, NTFS allows infrequently used files or directories to be individually selected for compression. In File Manager highlight the dataset or director, then from the File menu pulldown select Compress. Files are automatically decompressed as they are used, and new files are compressed if they are stored in a compressed directory. Frequently used files can be left uncompressed to avoid slowing the system.

There are no funny drivers that will allow other operating systems to access data in an NTFS volume. Anything stored in the NTFS file system is unavailable when DOS, Windows 95, or OS/2 is booted on the same machine. However, a Windows NT Server can share NTFS volumes through the network and the other operating systems can then access the files. Therefore, NTFS is strongly recommended as the

only good choice for all disk space used on a dedicated Windows NT Backoffice Server but may not be a good choice for the local files on workstations that boot multiple operating systems.

The following table compares the various operating system file system attributes, limitations, and capabilities:

	FAT 16	FAT 32	HPFS	NTFS	LINUX
Sector Size	**512 bytes**	**512**	**512 B**	**512**	**Uses Blocks**
Maximum/Min Cluster Size	**32,768 B**	**512/Z:n B**	**N/A**	**64 K**	**Uses Blocks**
Maximum Clusters Allowed	**65,525**	**Drive Limited**	**N/A**	**Drive Limited**	**Uses Blocks**
Maximum File Size	**2 GB**	**7.8 TB**	**2 GB**	**7.8 TB**	**Drive Limited**
Maximum Partition Size	**2 GB**	**7.8 GB**	**Drive Limited**	**7.8 GB**	**Drive Limited**
Maximum Drive Size	**2 GB**	**2 TB**	**64 GB**	**2 TB**	**Drive Limited**
Maximum Root Entries Allowed		**No Limit**	**No Limit**	**No Limit**	**No Limit**
Maximum File Name Characters	**11**	**256**	**254**	**256**	**256**
Naming Convention	**8.3**	**Flexible**	**Flexible**	**Flexible**	**Flexible**

The MS-DOS 16 bit FAT (FAT 16) limitations are imposed by the 16 bit cluster address numbers. The FAT file system, is limited to 65,525 clusters (16 bits to the power of 2) and a cluster size of 32,768 (32 K) (a cluster must be a power of 2 and less than 65,536 bytes. Multiplying the maximum number of clusters (65,525) by the maximum cluster size (32,768) equals 2 GB.