

File Allocation Table

File Allocation Table (**FAT**) is a file system developed for personal computers. Originally developed in 1977 for use on floppy disks, it was adapted for use on hard disks and other devices. It is often supported for compatibility reasons by current operating systems for personal computers and many mobile devices and embedded systems, allowing interchange of data between disparate systems.^[3] The increase in disk drives capacity required three major variants: FAT12, FAT16 and FAT32. The FAT standard has also been expanded in other ways while generally preserving backward compatibility with existing software.

FAT is no longer the default file system for Microsoft Windows computers.^[4]

FAT file systems are still commonly found on floppy disks, flash and other solid-state memory cards and modules (including USB flash drives), as well as many portable and embedded devices. FAT is the standard file system for digital cameras per the DCF specification.

Contents

Overview

Concepts

Uses

Nomenclature

Types

Original 8-bit FAT

FAT12

Initial FAT16

Logical sectored FAT

Final FAT16

FAT32

Maximal sizes

Development

Extensions

Extended attributes

Long file names

Forks and alternate data streams

UMSDOS permissions and filenames

FAT+

FAT

Developer(s)	Microsoft, <u>NCR</u> , <u>SCP</u> , <u>IBM</u> , <u>Compaq</u> , <u>Digital Research</u> , <u>Novell</u> , <u>Caldera</u>
Full name	File Allocation Table
Variants	8-bit FAT, <u>FAT12</u> , <u>FAT16</u> , <u>FAT16B</u> , <u>FAT32</u> , <u>ExFAT</u> , <u>FATX</u> , <u>FAT+</u>
Introduced	1977 with <u>Standalone Disk BASIC-80</u>
Partition identifier	<u>MBR/EBR</u> : FAT12: <u>0x01</u> e.a. (Extended Attribute) FAT16: <u>0x040x060x0E</u> e.a. FAT32: <u>0x0B0x0C</u> e.a. <u>BDP</u> : EBD0A0A2-B9E5-4433-87C0-68B6B72699C7
Structures	
Directory contents	Table
File allocation	<u>Linked list</u>
Bad blocks	Cluster tagging
Limits	
Max. volume size	FAT12: 32 <u>MB</u> (256 MB for 64 KB clusters) FAT16: 2 GB (4 GB for 64 KB clusters) FAT32: 2 <u>TB</u> (16 TB for 4 KB sectors)
Max. file size	4,294,967,295 bytes (4 <u>GB</u> − 1) with

Derivatives

Turbo FAT

FATX

exFAT

Patents

Challenges and lawsuits

See also

Notes

References

External links

Overview

Concepts

The file system uses an index table stored on the device to identify chains of data storage areas associated with a file, the *File Allocation Table* (*FAT*). The FAT is statically allocated at the time of formatting. The table is a linked list of entries for each *cluster*, a contiguous area of disk storage. Each entry contains either the number of the next cluster in the file, or else a marker indicating the end of the file, unused disk space, or special reserved areas of the disk. The *root directory* of the disk contains the number of the first cluster of each file in that directory. The operating system can then traverse the FAT, looking up the cluster number of each successive part of the disk file as a *cluster chain* until the end of the file is reached. *Sub-directories* are implemented as special files containing the *directory entries* of their respective files.

Each entry in the FAT linked list is a fixed number of bits: 12, 16 or 32. The maximum size of a file or a disk drive that can be accessed is the product of the largest number that can be stored in the entries (less a few values reserved to indicate unallocated space or the end of a list) and the size of the disk cluster. Even if only one byte of storage is needed to extend a file, an entire cluster must be allocated to it, so large clusters waste much disk space if there are large numbers of small files.

Originally designed as an 8-bit file system, the maximum number of clusters must increase as disk drive capacity increases, and so the number of bits used to identify each cluster has grown. The successive major variants of the FAT format are named after the number of table element bits: 12 (FAT12), 16 (FAT16), and 32 (FAT32).

Uses

FAT16B and FAT32^[1]

Max. number of files FAT12: 4,068 for 8 KB clusters
FAT16: 65,460 for 32 KB clusters
FAT32: 268,173,300 for 32 KB clusters

Max. filename length 8.3 filename, or 255 UCS-2 characters when using LFN^[nb 1]

Features

Dates recorded Modified date/time, creation date/time (DOS 7.0 and higher only), access date (only available with ACCDATE enabled),^[2] deletion date/time (only with DELWATCH 2)

Date range 1980-01-01 to 2099-12-31 (2107-12-31)

Date resolution 2 seconds for last modified time, 10 ms for creation time, 1 day for access date, 2 seconds for deletion time

Forks Not natively

Attributes Read-only, Hidden, System, Volume, Directory, Archive

File system permissions FAT12/FAT16: File, directory and volume access rights for Read, Write, Execute, Delete only with DR-DOS, PalmDOS, Novell DOS, OpenDOS, FlexOS, 4680 OS, 4690 OS, Concurrent DOS, Multiuser DOS, System Manager,

The FAT file system has been used since 1977 for computers, and it is still frequently used in embedded systems. Compatible file systems make it easier to exchange data between, for example, desktop computers and portable devices. FAT file systems are the default for removable media such as floppy disks, super-floppies, memory and flash memory cards or USB flash drives. FAT is supported by portable devices such as PDAs, digital cameras, camcorders, media players, and mobile phones. While FAT12 is used on floppy disks, FAT16 and FAT32 are typically found on the larger media.

FAT was also used on hard disks throughout the DOS and Windows 9x eras. Microsoft introduced a new file system, NTFS, with the Windows NT platform in 1993, but FAT remained the standard for the home user until the introduction of the NT-based Windows XP in 2001. FAT is still used in hard drives expected to be used by multiple operating systems, such as in shared Windows, Linux and DOS environments.

Many operating systems provide support for FAT-formatted media through built-in or third-party file system handlers.

The DCF file system adopted by almost all digital cameras since 1998 defines a logical file system with 8.3 filenames and makes the use of either FAT12, FAT16, FAT32 or exFAT mandatory for its physical layer for compatibility. ^[5]

FAT is also used internally for the EFI system partition in the boot stage of EFI-compliant computers.^[6] Hidden FAT filesystems are also used in the UEFI boot partition on modern PCs.

For floppy disks, FAT has been standardized as ECMA-107^[7] and ISO/IEC 9293:1994^[8] (superseding ISO 9293:1987^[9]). These standards cover FAT12 and FAT16 with only short 8.3 filename support; long filenames with VFAT were partially patented.^[10]

Nomenclature

"FAT file system" refers to, FAT12, FAT16 and FAT32. Operating system utilities may not identify which version will be used to format a device.

"FAT16" refers to both the original group of FAT file systems with 16-bit wide cluster entries and also to later variants ("FAT16B") with 32-bit sector entries. Values stored in the disk parameter block can be used to identify the file structure.

"VFAT" is an optional extension for long file names, which can work on top of any FAT file system. Volumes using VFAT long-filenames can be read also by operating systems not supporting the VFAT extension.

The general type of file system (FAT12, FAT16 or FAT32) is determined by the width of the cluster entries in the FAT. Specific threshold values for the number of clusters, stored in the disk parameter block, define which FAT type is used. ^[6] Other properties of the storage device such as size, parameter block format, or file system name cannot reliably be used to derive the file system type.^[6]

REAL/32 (Execute right only with FlexOS, 4680 OS, 4690 OS; individual file / directory passwords not with FlexOS, 4680 OS, 4690 OS; World/Group/Owner permission classes only with multiuser security loaded)
FAT32: Partial, only with DR-DOS, REAL/32 and 4690 OS

Transparent compression FAT12/FAT16: Per-volume, SuperStor, Stacker, DoubleSpace, DriveSpace
FAT32: No

Transparent encryption FAT12/FAT16: Per-volume only with DR-DOS
FAT32: No

A FAT12 or FAT16 volume can be defined with a "FAT32 EBPB" normally used for FAT32 volumes.^[nb 2] Partition IDs are not used to specify a type of file system by themselves.^[11]

Types

Original 8-bit FAT

The original FAT file system (or *FAT structure*, as it was called initially) was designed and implemented by Marc McDonald,^[14] based on a series of discussions between McDonald and Bill Gates.^[14] It was introduced with 8-bit table elements^{[12][13][14]} (and valid data cluster numbers up to 0xBF^{[12][13]}) in a precursor to Microsoft's *Standalone Disk BASIC-80* for an 8080-based successor^[nb 3] of the NCR 7200 model VI data-entry terminal, equipped with 8-inch (200 mm) floppy disks, in 1977^[15] or 1978.^[nb 3] In 1978, *Standalone Disk BASIC-80* was ported to the 8086 using an emulator on a DEC PDP-10,^[16] since no real 8086 systems were available at this time. The FAT file system was also used in Microsoft's MDOS/MIDAS,^[14] an operating system for 8080/Z80 platforms written by McDonald since 1979. The *Standalone Disk BASIC* version supported three FATs,^{[12][13][17]} whereas this was a parameter for MIDAS. Reportedly, MIDAS was also prepared to support 10-bit, 12-bit and 16-bit FAT variants. While the size of directory entries was 16 bytes in *Standalone Disk BASIC*,^{[12][13]} MIDAS instead occupied 32 bytes per entry.

FAT12

Between April and August 1980, while borrowing the FAT concept for SCP's own 8086 operating system QDOS 0.10,^[16] Tim Paterson extended the table elements to **12 bits**,^[18] reduced the number of FATs to two, redefined the semantics of some of the reserved cluster values, and modified the disk layout, so that the root directory was now located between the FAT and the data area for his implementation of **FAT12**. Paterson also increased the nine-character (6.3) filename^{[12][13]} length limit to eleven characters to support CP/M-style 8.3 filenames and File Control Blocks. The format used in Microsoft *Standalone Disk BASIC*'s 8-bit file system precursor was not supported by QDOS. By August 1980, QDOS had been renamed 86-DOS.^[19] Starting with 86-DOS 0.42, the size and layout of directory entries was changed from 16 bytes to 32 bytes^[20] in order to add a file date stamp^[20] and increase the theoretical file size limit beyond the previous limit of 16 MB.^[20] 86-DOS 1.00 became available in early 1981. Later in 1981, 86-DOS evolved into Microsoft's MS-DOS and IBM PC DOS.^{[14][18][21]} The capability to read previously formatted volumes with 16-byte directory entries^[20] was dropped with MS-DOS 1.20.

8-bit FAT

Developer(s)	<u>Microsoft</u> , <u>NCR</u> , <u>SCP</u>
Full name	8-bit File Allocation Table
Introduced	1977/1978: <u>NCR Basic +6</u> for NCR 1978: <u>Standalone Disk BASIC-80</u> (16-byte directory entries) ^{[12][13]} (1978: <u>Standalone Disk BASIC-86</u> internal only) 1979-06-04: <u>Standalone Disk BASIC-86</u> for SCP (16-byte directory entries) 1979: <u>MIDAS</u> (32-byte directory entries)
Limits	
Max. file size	8 MB
File size granularity	record-granularity (128 bytes) ^{[12][13]}
Max. filename length	<u>6.3 filename</u> (binary files), 9 characters (ASCII files) ^{[12][13]}
Max. directory depth	no sub-directories
Allowed characters in filenames	<u>ASCII</u> (0x00 and 0xFF not allowed in first character) ^{[12][13]}
Features	

FAT12 used 12-bit entries for the cluster addresses; some values were reserved to mark the end of a chain of clusters, to mark unusable areas of the disk, or for other purposes, so the maximum number of clusters was limited to 4078.^{[22][23]} To conserve disk space, two 12-bit FAT entries used three consecutive 8-bit bytes on disk, requiring manipulation to unpack the 12-bit values. This was sufficient for the original floppy disk drives, and small hard disk up to 32 megabytes. The FAT16B version available with DOS 3.31, supported 32-bit sector numbers and so increased the volume size limit.

All the control structures fit inside the first track, to avoid head movement during read and write operations. Any bad sector in the control structures area would make the disk unusable. The DOS formatting tool rejected such disks completely. Bad sectors were allowed only in the file data area. Clusters containing bad sectors were marked unusable with the reserved value 0xFF7.

While 86-DOS supported three disk formats (250.25 KB, 616 KB and 1232 KB with FAT IDs 0xFF and 0xFE on 8-inch (200 mm) floppy drives, IBM PC DOS 1.0, released with the original IBM Personal Computer in 1981, supported only an 8-sector floppy format with a formatted capacity of 160 KB (FAT ID 0xFE) for single-sided 5.25-inch floppy drives, and PC DOS 1.1 added support for a double-sided format with 320 KB (FAT ID 0xFF). PC DOS 2.0 introduced support for 9-sector floppy formats with 180 KB (FAT ID 0xFC) and 360 KB (FAT ID 0xFD).

86-DOS 1.00 and PC DOS 1.0 directory entries included only one date, the last modified date. PC DOS 1.1 added the last modified time. PC DOS 1.x file attributes included a hidden bit and system bit, with the remaining six bits undefined. At this time, DOS did not support sub-directories, but typically there were only a few dozen files on a diskette.

The PC XT was the first PC with a hard drive from IBM, and PC DOS 2.0 supported that hard drive with FAT12 (FAT ID 0xF8). The fixed assumption of 8 sectors per clusters on hard disks practically limited the maximum partition size to 16 MB for 512 byte sectors and 4 KB clusters.

The BIOS Parameter Block (BPB) was introduced with PC DOS 2.0 as well, and this version also added read-only, archive, volume label, and directory attribute bits for hierarchical sub-directories.^[24]

MS-DOS 3.0 introduced support for high-density 1.2 MB 5.25-inch diskettes (media descriptor 0xF9), which notably had 15 sectors per track, hence more space for the FATs.

FAT12 remains in use on all common floppy disks, including 1.44 MB and later 2.88 MB disks (media descriptor byte 0xF0).

Dates recorded	No
Attributes	Write protected, <u>EBCDIC</u> conversion, Read after write, Binary (random rather than sequential file) ^{[12][13]}

FAT12

<u>Developer(s)</u>	<u>SCP</u> , <u>Microsoft</u> , <u>IBM</u> , <u>Digital Research</u> , <u>Novell</u>
<u>Full name</u>	<u>12-bit File Allocation Table</u>
<u>Introduced</u>	1980-07 (<u>QDOS 0.10</u> , 16-byte directory entries) 1981-02-25 (<u>86-DOS 0.42</u> , 32-byte directory entries, several reserved sectors) c. 1981-08/10 (<u>PC DOS 1.0</u> , 32-byte directory entries, 1 reserved sector) 1982-03-03 (<u>MS-DOS 1.25</u> , 32-byte directory entries, 1 reserved sector)
<u>Partition identifier</u>	<u>MBR/EBR</u> : <u>FAT12</u> : <u>0x01</u> e.a. <u>BDP</u> : EBD0A0A2-B9E5-4433-87C0-68B6B72699C7
Limits	
<u>Max. volume size</u>	16 MB (with 4 KB clusters) 32 MB (with 8 KB clusters)
<u>Max. file size</u>	limited by volume size
<u>File size</u>	1 byte

Initial FAT16

In 1984, IBM released the PC AT, which required PC DOS 3.0 to access its 20 MB hard disk. ^{[25][26]} Microsoft introduced MS-DOS 3.0 in parallel. Cluster addresses were increased to 16-bit, allowing for up to 65,524 clusters per volume. However, the maximum possible number of sectors and the maximum partition size of 32 MB did not change. Although cluster addresses were 16 bits, this format was not what today is commonly understood as **FAT16**. A partition type 0x04 indicates this form of FAT16 with less than 65536 sectors (less than 32 MB for sector size 512). The benefit of FAT16 was the use of smaller clusters, making disk usage more efficient, particularly for large numbers of files only a few hundred bytes in size.

As MS-DOS 3.0 formatted all 16 MB-32 MB partitions in the FAT16 format, a 20 MB hard disk formatted under MS-DOS 3.0 was not accessible by MS-DOS 2.0. ^[27] MS-DOS 3.0 to MS-DOS 3.30 could still access FAT12 partitions under 15 MB, but required all 16 MB-32 MB partitions to be FAT16, and so could not access MS-DOS 2.0 partitions in this size range. MS-DOS 3.31 and higher could access 16 MB-32 MB FAT12 partitions again.

Logical sectored FAT

MS-DOS and PC DOS implementations of FAT12 and FAT16 could not access disk partitions larger than 32 megabytes. Several manufacturers developed their own FAT variants within their OEM versions of MS-DOS. ^[28]

Some vendors (AST and NEC ^[28]) supported eight, instead of the standard four, primary partition entries in their custom extended Master Boot Record (MBR), and they adapted MS-DOS to use more than a single primary partition.

Other vendors worked around the volume size limits imposed by the 16-bit sector entries by increasing the apparent *size* of the sectors the file system operated on. These *logical sectors* were larger (up to 8192 bytes) than the *physical sector* size (still 512 bytes) on the disk. The DOS-BIOS or System BIOS would then combine multiple physical sectors into logical sectors for the file system to work with.

These changes were transparent to the file system implementation in the DOS kernel. The underlying DOS-BIOS translated these logical sectors into physical sectors according to partitioning information and the drive's physical geometry.

The drawback of this approach was increased memory used for sector buffering and deblocking. Since older DOS versions could not use large logical sectors, the OEMs introduced new partition IDs for their FAT variants in order to hide them from off-the-shelf issues of MS-DOS and PC DOS. Known partition IDs for logical sectored FATs

granularity	
Max. number of files	4,068 for 8 KB clusters
Max. filename length	8.3 filename with OEM characters, 255 UCS-2 characters ^[nb 1] when using LFN
Max. directory depth	32 levels or 66 characters (with CDS), 60 levels or more (without CDS)
Features	
Dates recorded	Modified date (not with 86-DOS before 0.42), modified time (not with PC DOS 1.0 and 86-DOS), creation date/time (DOS 7.0 and higher only), access date (only available with <u>ACCDATE</u> enabled), ^[2] deletion date/time (only with <u>DELWATCH 2</u>)
Date range	<u>1980-01-01</u> to <u>2099-12-31</u> (<u>2107-12-31</u>)
Date resolution	2 seconds for last modified time, 10 ms for creation time, 1 day for access date, 2 seconds for deletion time
Attributes	<u>Read-only</u> (since DOS 2.0), <u>Hidden</u> , <u>System</u> , <u>Volume</u> (since MS-DOS 1.28 and PC DOS 2.0), <u>Directory</u> (since MS-DOS

include: 0x08 (Commodore MS-DOS 3.x), 0x11 (Leading Edge MS-DOS 3.x), 0x14 (AST MS-DOS 3.x), 0x24 (NEC MS-DOS 3.30^[28]), 0x56 (AT&T MS-DOS 3.x), 0xE5 (Tandy MS-DOS), 0xF2 (Sperry IT MS-DOS 3.x, Unisys MS-DOS 3.3 – also used by Digital Research DOS Plus 2.1).^[29] OEM versions like Toshiba MS-DOS, Wyse MS-DOS 3.2 and 3.3,^[30] as well as Zenith MS-DOS are also known to have utilized logical sectoring.^[31]

While non-standard and sub-optimal, these FAT variants are perfectly valid according to the specifications of the file system itself. Therefore, even if default issues of MS-DOS and PC DOS were not able to cope with them, most of these vendor-specific FAT12 and FAT16 variants can be mounted by more flexible file system implementations in operating systems such as DR-DOS, simply by changing the partition ID to one of the recognized types.^[nb 4] Also, if they no longer need to be recognized by their original operating systems, existing partitions can be "converted" into FAT12 and FAT16 volumes more compliant with versions of MS-DOS/PC DOS 4.0–6.3, which do not support sector sizes different from 512 bytes,^[32] by switching to a BPB with 32-bit entry for the number of sectors, as introduced since DOS 3.31 (see FAT16B below), keeping the cluster size and reducing the logical sector size in the BPB down to 512 bytes, while at the same time increasing the counts of logical sectors per cluster, reserved logical sectors, total logical sectors, and logical sectors per FAT by the same factor.

A parallel development in MS-DOS / PC DOS which allowed an increase in the maximum possible FAT size was the introduction of multiple FAT partitions on a hard disk. To allow the use of more FAT partitions in a compatible way, a new partition type was introduced in PC DOS 3.2 (1986), the extended partition (EBR),^[14] which is a container for an additional partition called *logical drive*. Since PC DOS 3.3 (April 1987), there is another, optional extended partition containing the next *logical drive*, and so on. The MBR of a hard disk can either define up to four primary partitions, or an extended partition in addition to up to three primary partitions.

Final FAT16

In November 1987, Compaq Personal Computer DOS 3.31 (a modified OEM version of MS-DOS 3.3 released by Compaq with their machines) introduced what today is simply known as *the FAT16* format, with the expansion of the 16-bit disk sector count to 32 bits in the BPB. Although the on-disk changes were minor, the entire DOS disk driver had to be converted to use 32-bit sector numbers, a task complicated by the fact that it was written in 16-bit assembly language. The result was initially called the *DOS 3.31 Large File System*. Microsoft's DSKPROBE tool refers to type 0x06 as *BigFAT*,^[34] whereas some older versions of FDISK described it as *BIGDOS*. Technically, it is known as **FAT16B**.

	1.40 and PC DOS 2.0), <u>Archive</u> (since DOS 2.0)
<u>File system permissions</u>	File, directory and volume access rights for <u>Read</u> , <u>Write</u> , <u>Execute</u> , <u>Delete only with DR-DOS</u> , <u>PalmDOS</u> , <u>Novell DOS</u> , <u>OpenDOS</u> , <u>FlexOS</u> , <u>4680 OS</u> , <u>4690 OS</u> , <u>Concurrent DOS</u> , <u>Multiuser DOS</u> , <u>System Manager</u> , <u>REAL/32</u> (Execute right only with FlexOS, 4680 OS, 4690 OS; individual file / directory passwords not with FlexOS, 4680 OS, 4690 OS; <u>World/Group/Owner</u> permission classes only with multiuser security loaded)
<u>Transparent compression</u>	Per-volume, <u>SuperStor</u> , <u>Stacker</u> , <u>DoubleSpace</u> , <u>DriveSpace</u>
<u>Transparent encryption</u>	Per-volume only with <u>DR-DOS</u>

FAT16.

<u>Developer(s)</u>	<u>Microsoft</u> , <u>IBM</u> , <u>Digital Research</u> , <u>Novell</u>
<u>Full name</u>	<u>16-bit File Allocation Table</u> (with 16-bit sector entries)
<u>Introduced</u>	1984-08-14 (PC DOS 3.0)

Since older versions of DOS were not designed to cope with more than 65535 sectors, it was necessary to introduce a new partition type for this format in order to hide it from pre-3.31 issues of DOS. The original form of FAT16 (with less than 65536 sectors) had a partition type 0x04. To deal with disks larger than this, type 0x06 was introduced to indicate 65536 or more sectors. In addition to this, the disk driver was expanded to cope with more than 65535 sectors as well. The only other difference between the original FAT16 and the newer FAT16B format is the usage of a newer BPB format with 32-bit sector entry. Therefore, newer operating systems supporting the FAT16B format can cope also with the original FAT16 format without any necessary changes.

If partitions to be used by pre-DOS 3.31 issues of DOS need to be created by modern tools, the only criteria theoretically necessary to meet are a sector count of less than 65536, and the usage of the old partition ID (0x04). In practice however, type 0x01 and 0x04 primary partitions should not be physically located outside the first 32 MB of the disk, due to other restrictions in MS-DOS 2.x, which could not cope with them otherwise.

In 1988, the FAT16B improvement became more generally available through DR DOS 3.31, PC DOS 4.0, OS/2 1.1, and MS-DOS 4.0. The limit on partition size was dictated by the 8-bit signed count of sectors per cluster, which originally had a maximum power-of-two value of 64. With the standard hard disk sector size of 512 bytes, this gives a maximum of 32 KB cluster size, thereby fixing the "definitive" limit for the FAT16 partition size at 2 GB for sector size 512. On magneto-optical media, which can have 1 or 2 KB sectors instead of 0.5 KB, this size limit is proportionally larger.

Much later, Windows NT increased the maximum cluster size to 64 KB, by considering the sectors-per-cluster count as unsigned. However, the resulting format was not compatible with any other FAT implementation of the time, and it generated greater internal fragmentation. Windows 98, SE and ME also supported reading and writing this variant, but its disk utilities did not work with it and some FCB services are not available for such volumes. This contributes to a confusing compatibility situation.

Prior to 1995, versions of DOS accessed the disk via CHS addressing only. When MS-DOS 7.0 / Windows 95 introduced LBA disk access, partitions could start being physically located outside the first c. 8 GB of this disk and thereby out of the reach of the traditional CHS addressing scheme. Partitions partially or fully located beyond the CHS barrier therefore had to be hidden from non-LBA-enabled operating systems by using the new partition type 0x0E in the partition table instead. FAT16 partitions using this partition type are also named **FAT16X**.^[35] The only difference, compared to previous FAT16 partitions, is the fact that some CHS-related geometry entries in the BPB record, namely the number of sectors per track and the number of heads, may contain no or misleading values and should not be used.

	1984-08 (MS-DOS 3.0)
<u>Partition identifier</u>	MBR/EBR: FAT16: <u>0x04</u> e.a. BDP: EBD0A0A2-B9E5-4433-87C0-68B6B72699C7
Limits	
Max. file size	limited by volume size
File size granularity	1 byte
Max. number of files	65,536 for 32 KB clusters
Max. filename length	8.3 filename with OEM characters, 255 UCS-2 characters ^[nb 1] when using LFN
Max. directory depth	32 levels or 66 characters (with CDS), 60 levels or more (without CDS)
Features	
Dates recorded	Modified date/time, creation date/time (DOS 7.0 and higher only), access date (only available with ACCDATE enabled), ^[2] deletion date/time (only with DELWATCH 2)
Date range	1980-01-01 to 2099-12-31 (2107-12-31)
Date resolution	2 seconds for last modified time, 10 ms for creation time, 1 day for access date,

The number of root directory entries available for FAT12 and FAT16 is determined when the volume is formatted, and is stored in a 16-bit field. For a given number RDE and sector size SS, the number RDS of root directory sectors is $RDS = \text{ceil}((RDE \times 32) / SS)$, and RDE is normally chosen to fill these sectors, i.e., $RDE \times 32 = RDS \times SS$. FAT12 and FAT16 media typically use 512 root directory entries on non-floppy media. Some third-party tools, like mkdosfs, allow the user to set this parameter.^[36]

FAT32

In order to overcome the volume size limit of FAT16, while at the same time allowing DOS real-mode code to handle the format, Microsoft designed a new version of the file system, **FAT32**, which supported an increased number of possible clusters, but could reuse most of the existing code, so that the conventional memory footprint was increased by less than 5 KB under DOS.^[37] Cluster values are represented by 32-bit numbers, of which 28 bits are used to hold the cluster number.

Maximal sizes

The FAT32 boot sector uses a 32-bit field for the sector count, limiting the maximal FAT32 volume size to 2 terabytes with a sector size of 512 bytes. The maximum FAT32 volume size is 16 TB with a sector size of 4,096 bytes.^{[38][39]} The built-in Windows shell disk format tool on Microsoft Windows only supports volume sizes up to 32 GB, but larger FAT32 volumes can be created with the Command Prompt, PowerShell or third-party tools and read by Microsoft Windows.^[40]

The maximal possible size for a file on a FAT32 volume is 4 GB minus 1 byte or 4,294,967,295 ($2^{32} - 1$) bytes. This limit is a consequence of the 4-byte file length entry in the directory table and would also affect relatively huge FAT16 partitions enabled by a sufficient sector size.

Like FAT12 and FAT16, FAT32 does not include direct built-in support for long filenames, but FAT32 volumes can optionally hold VFAT long filenames in addition to short filenames in exactly the same way as VFAT long filenames have been optionally implemented for FAT12 and FAT16 volumes.

Development

FAT32 was introduced with MS-DOS 7.1 / Windows 95 OSR2 in 1996, although reformatting was needed to use it, and DriveSpace 3 (the version that came with Windows 95 OSR2 and Windows 98) never supported it. Windows 98 introduced a utility to convert existing hard disks from FAT16 to FAT32 without loss of data.

	2 seconds for deletion time
Attributes	<u>Read-only</u> , <u>Hidden</u> , <u>System</u> , <u>Volume</u> , <u>Directory</u> , <u>Archive</u>
File system permissions	File, directory and volume access rights for <u>Read</u> , <u>Write</u> , <u>Execute</u> , <u>Delete only with DR-DOS</u> , <u>PalmDOS</u> , <u>Novell DOS</u> , <u>OpenDOS</u> , <u>FlexOS</u> , <u>4680 OS</u> , <u>4690 OS</u> , <u>Concurrent DOS</u> , <u>Multiuuser DOS</u> , <u>System Manager</u> , <u>REAL/32</u> (Execute right only with <u>FlexOS</u> , <u>4680 OS</u> , <u>4690 OS</u> ; individual file / directory passwords not with <u>FlexOS</u> , <u>4680 OS</u> , <u>4690 OS</u> ; <u>World/Group/Owner</u> permission classes only with multiuser security loaded)
Transparent compression	<u>Per-volume</u> , <u>SuperStor</u> , <u>Stacker</u> , <u>DoubleSpace</u> , <u>DriveSpace</u>
Transparent encryption	<u>Per-volume only</u> with <u>DR-DOS</u>

FAT16B

Developer(s)	<u>Compaq</u> , <u>Digital Research</u> , <u>IBM</u> , <u>Microsoft</u> , <u>Novell</u>
Full name	<u>16-bit File Allocation Table</u> (with 32-bit sector entries)
Introduced	<u>1987-11</u> (<u>Compaq MS-DOS 3.31</u>) <u>1988-06-28</u> (<u>DR DOS 3.31</u>) <u>1988</u> (<u>IBM DOS 4.0</u>)

In the Windows NT line, native support for FAT32 arrived in Windows 2000. A free FAT32 driver for Windows NT 4.0 was available from Winternals, a company later acquired by Microsoft. The acquisition of the driver from official sources is no longer possible. Since 1998, Caldera's dynamically loadable DRFAT32 driver could be used to enable FAT32 support in DR-DOS.^{[41][42]} The first version of DR-DOS to natively support FAT32 and LBA access was OEM DR-DOS 7.04 in 1999. That same year IMS introduced native FAT32 support with REAL/32 7.90, and IBM 4690 OS added FAT32 support with version 2.^[43] Ahead Software provided another dynamically loadable FAT32.EXE driver for DR-DOS 7.03 with Nero Burning ROM in 2004. IBM PC DOS introduced native FAT32 support with OEM PC DOS 7.10 in 2003.

Two partition types have been reserved for FAT32 partitions, 0x0B and 0x0C. The latter type is also named **FAT32X** in order to indicate usage of LBA disk access instead of CHS.^{[41][44][45][46][47]} On such partitions, CHS-related geometry entries, namely the CHS sector addresses in the MBR as well as the number of sectors per track and the number of heads in the EBPB record, may contain no or misleading values and should not be used.^{[48][46][47]}

Extensions

Extended attributes

OS/2 heavily depends on extended attributes (EAs) and stores them in a hidden file called "EA[⋆]DATA[⋆]SF" in the root directory of the FAT12 or FAT16 volume. This file is indexed by two previously reserved bytes in the file's (or directory's) directory entry at offset 0x14.^[49] In the FAT32 format, these bytes hold the upper 16 bits of the starting cluster number of the file or directory, hence making it impossible to store OS/2 EAs on FAT32 using this method.

However, the third-party FAT32 installable file system (IFS) driver FAT32.IFS version 0.70 and higher by Henk Kelder & Netlabs for OS/2, eComStation and ArcaOS stores extended attributes in extra files with filenames having the string "EA[⋆]SF" appended to the regular filename of the file to which they belong. The driver also utilizes the byte at offset 0x0C in directory entries to store a special mark byte indicating the presence of extended attributes to help speed up things.^{[50][51]} (This extension is critically incompatible with the FAT32+ method to store files larger than 4 GB minus 1 on FAT32 volumes.)^[33]

	1988 (<u>OS/2 1.1</u>) 1988 (<u>MS-DOS 4.0</u>)
Partition identifier	<u>MBR/EBR</u> : <u>FAT16B</u> : <u>0x060x0E(LBA)</u> , e.a. <u>BDP</u> : <u>EBD0A0A2-B9E5-4433-87C0-68B6B72699C7</u>
Limits	
Min. volume size	8 <u>MB</u> (with 128 byte sectors) 32 MB (with <u>512</u> byte sectors) 256 MB (with <u>4 KB</u> sectors)
Max. volume size	2 <u>GB</u> (with 32 <u>KB</u> clusters) 4 GB (with 64 KB clusters) (<u>NT 4</u> , <u>PTS-DOS</u> , <u>EDR-DOS</u>) 8 GB (with 128 KB clusters and 1 or 2 KB sectors) (<u>NT 4</u> and <u>EDR-DOS</u> only) 8 GB (with 128 KB clusters and 512 byte sectors) (<u>EDR-DOS</u> only) 16 GB (with 256 KB clusters and 2 or 4 KB sectors) (<u>NT 4</u> only)
Max. file size	2,147,483,647 bytes (2 GB – 1) (without <u>LFS</u>) 4,294,967,295 bytes (4 GB – 1) (with <u>LFS</u>) limited by volume size only (with <u>FAT16</u> + ^[33])
File size granularity	1 byte
Max. number of files	65,460 for 32 KB clusters
Max. filename length	8.3 <u>filename</u> with <u>OEM</u> characters, 255 <u>UCS-2</u> characters ^[nb 1] when using <u>LFN</u>
Max. directory depth	32 levels or 66 characters (with <u>CDS</u>), 60 levels or more (without <u>CDS</u>)
Features	
Dates	Modified date/time, creation

Extended attributes are accessible via the Workplace Shell desktop, through REXX scripts, and many system GUI and command-line utilities (such as 4OS2).^[52]

To accommodate its OS/2 subsystem, Windows NT supports the handling of extended attributes in HPFS, NTFS, FAT12 and FAT16. It stores EAs on FAT12, FAT16 and HPFS using exactly the same scheme as OS/2, but does not support any other kind of ADS as held on NTFS volumes. Trying to copy a file with any ADS other than EAs from an NTFS volume to a FAT or HPFS volume gives a warning message with the names of the ADSs that will be lost. It does not support the FAT32.IFS method to store EAs on FAT32 volumes.

Windows 2000 onward acts exactly as Windows NT, except that it ignores EAs when copying to FAT32 without any warning (but shows the warning for other ADSs, like "Macintosh Finder Info" and "Macintosh Resource Fork").

Cygwin uses "EA ^sDATA . ^sSF" files as well.

Long file names

One of the user experience goals for the designers of Windows 95 was the ability to use long filenames (LFNs—up to 255 UTF-16 code units long),^[nb 1] in addition to classic 8.3 filenames (SFNs). For backward and forward compatibility LFNs were implemented as an optional extension on top of the existing FAT file system structures using a workaround in the way directory entries are laid out.

This transparent method to store long file names in the existing FAT file systems without altering their data structures is usually known as VFAT (for "Virtual FAT") after the Windows 95 virtual device driver.^[nb 5]

Non VFAT-enabled operating systems can still access the files under their short file name alias without restrictions; however, the associated long file names may be lost when files with long filenames are copied under non VFAT-aware operating systems.

In Windows NT, support for VFAT long filenames began with version 3.5.

Linux provides a VFAT filesystem driver to work with FAT volumes with VFAT long filenames. For some time, a UVFAT driver was available to provide combined support for UMSDOS-style permissions with VFAT long filenames.

recorded	date/time (<u>DOS 7.0</u> and higher only), access date (only available with <u>ACCDATE</u> enabled), ^[2] deletion date/time (only with <u>DELWATCH 2</u>)
Date range	<u>1980-01-01</u> to <u>2099-12-31</u> (<u>2107-12-31</u>)
Date resolution	2 seconds for last modified time, 10 ms for creation time, 1 day for access date, 2 seconds for deletion time
Attributes	<u>Read-only</u> , <u>Hidden</u> , <u>System</u> , <u>Volume</u> , <u>Directory</u> , <u>Archive</u>
File system permissions	File, directory and volume access rights for <u>Read</u> , <u>Write</u> , <u>Execute</u> , <u>Delete</u> only with <u>DR-DOS</u> , <u>PalmDOS</u> , <u>Novell DOS</u> , <u>OpenDOS</u> , <u>FlexOS</u> , <u>4680 OS</u> , <u>4690 OS</u> , <u>Concurrent DOS</u> , <u>Multituser DOS</u> , <u>System Manager</u> , <u>REAL/32</u> (Execute right only with <u>FlexOS</u> , <u>4680 OS</u> , <u>4690 OS</u> ; individual file / directory passwords not with <u>FlexOS</u> , <u>4680 OS</u> , <u>4690 OS</u> ; <u>World/Group/Owner</u> permission classes only with multituser security loaded)
Transparent compression	Per-volume, <u>SuperStor</u> , <u>Stacker</u> , <u>DoubleSpace</u> , <u>DriveSpace</u>
Transparent encryption	Per-volume only with <u>DR-DOS</u>

FAT32

Developer(s)	<u>Microsoft</u> , <u>Caldera</u>
Introduced	August 1996 (<u>Windows 95 OSR2</u>)
Partition identifier	<u>MBR/EBR</u> : <u>FAT32</u> : <u>0x0B0x0C</u> (<u>LBA</u>),e.a. <u>BDP</u> :

OS/2 added long filename support to FAT using extended attributes (EA) before the introduction of VFAT. Thus, VFAT long filenames are invisible to OS/2, and EA long filenames are invisible to Windows; therefore experienced users of both operating systems would have to manually rename the files.

Human68K supported up to 18.3 filenames and (Shift JIS) Kanji characters in a proprietary FAT file system variant.

In order to support Java applications, the FlexOS-based IBM 4690 OS version 2 introduced its own virtual file system (VFS) architecture to store long filenames in the FAT file system in a backwards-compatible fashion. If enabled, the virtual filenames (VFN) are available under separate logical drive letters, whereas the real filenames (RFN) remain available under the original drive letters.^[53]

Forks and alternate data streams

The FAT file system itself is not designed for supporting alternate data streams (ADS), but some operating systems that heavily depend on them have devised various methods for handling them on FAT volumes. Such methods either store the additional information in extra files and directories (classic Mac OS and macOS), or give new semantics to previously unused fields of the FAT on-disk data structures (OS/2 and Windows NT).

Mac OS using PC Exchange stores its various dates, file attributes and long filenames in a hidden file called "FINDER.DAT", and resource forks (a common Mac OS ADS) in a subdirectory called "RESOURCE.FRK", in every directory where they are used. From PC Exchange 2.1 onwards, they store the Mac OS long filenames as standard FAT long filenames and convert FAT filenames longer than 31 characters to unique 31-character filenames, which can then be made visible to Macintosh applications.

macOS stores resource forks and metadata (file attributes, other ADS) using AppleDouble format in a hidden file with a name constructed from the owner filename prefixed with "._", and Finder stores some folder and file metadata in a hidden file called ".DS_Store" (but note that Finder uses .DS_Store even on macOS' native filesystem, HFS+).

UMSDOS permissions and filenames

Early GNU/Linux distributions also supported a format known as UMSDOS, a FAT variant with Unix file attributes (such as long file name and access permissions) stored in a separate file called "--linux- - - -". UMSDOS fell into

EBD0A0A2-B9E5-4433-87C0-68B6B72699C7	
Limits	
Min. volume size	32 <u>MB</u> -4.5 <u>KB</u> (with 65525 clusters and 512 byte sectors) 256 MB-36 KB (with 65525 clusters and 4 <u>KB</u> sectors)
Max. volume size	2 <u>TB</u> (with 512 byte sectors) 8 TB (with 2 KB sectors and 32 KB clusters) 16 TB (with 4 KB sectors and 64 KB clusters)
Max. file size	2,147,483,647 bytes (2 GB – 1) (without <u>LFS</u>) 4,294,967,295 bytes (4 GB – 1) ^[1] (with <u>LFS</u>) 274,877,906,943 bytes (256 GB – 1) (only with FAT32+ ^[33])
File size granularity	16 byte
Max. number of files	268,173,300 for 32 KB clusters
Max. filename length	<u>8.3 filename</u> with <u>OEM</u> characters, 255 <u>UCS-2</u> characters ^[nb 1] when using <u>LFN</u>
Max. directory depth	32 levels or 66 characters (with <u>CDS</u>), 60 levels or more (without CDS)
Features	
Dates recorded	Modified date/time, creation date/time (DOS 7.0 and higher only), access date (only available with <u>ACCDATE</u> enabled), ^[2] deletion date/time (only with DELWATCH 2)
Date range	<u>1980-01-01</u> to <u>2099-12-31</u> (2107-12-31)
Date	2 seconds for last modified

disuse after VFAT was released and it is not enabled by default in Linux from version 2.5.7 onwards.^[54] For some time, Linux also provided combined support for UMSDOS-style permissions and VFAT long filenames through UVFAT.

FAT+

In 2007 the open **FAT+** draft proposed how to store larger files up to 256 GB minus 1 byte or 274,877,906,943 ($2^{38} - 1$) bytes on slightly modified and otherwise backward-compatible FAT32 volumes,^[33] but imposes a risk that disk tools or FAT32 implementations not aware of this extension may truncate or delete files exceeding the normal FAT32 file size limit. Support for **FAT32+** and **FAT16+** is limited to some versions of DR-DOS and not available in mainstream operating systems.^[55] (This extension is critically incompatible with the /EAS option of the FAT32.IFS method to store OS/2 extended attributes on FAT32 volumes.)

resolution	time, 10 ms for creation time, 1 day for access date, 2 seconds for deletion time
Attributes	<u>Read-only</u> , <u>Hidden</u> , <u>System</u> , <u>Volume</u> , <u>Directory</u> , <u>Archive</u>
File system permissions	Partial, only with <u>DR-DOS</u> , <u>REAL/32</u> and <u>4690 OS</u>
Transparent compression	yes

Derivatives

Turbo FAT

In its NetWare File System (NWFS) Novell implemented a heavily modified variant of a FAT file system for the NetWare operating system. For larger files it utilized a performance feature named Turbo FAT.

FATX

FATX is a family of file systems designed for Microsoft's Xbox video game console hard disk drives and memory cards,^{[56][57]} introduced in 2001.

While resembling the same basic design ideas as FAT16 and FAT32, the **FATX16** and **FATX32** on-disk structures are simplified, but fundamentally incompatible with normal FAT16 and FAT32 file systems, making it impossible for normal FAT file system drivers to mount such volumes.

The non-bootable superblock sector is 4 KB in size and holds an 18 byte large BPB-like structure completely different from normal BPBs. Clusters are typically 16 KB in size and there is only one copy of the FAT on the Xbox. Directory entries are 64 bytes in size instead of the normal 32 bytes. Files can have filenames up to 42 characters long using the OEM character set and be up to 4 GB minus 1 byte in size. The on-disk timestamps hold creation, modification and access dates and times but differ from FAT: in FAT, the epoch is 1980; in FATX, the epoch is 2000. On the Xbox 360, the epoch is 1980.^[58]

exFAT

exFAT is a file system introduced with Windows Embedded CE 6.0 in November 2006 and brought to the Windows NT family with Vista Service Pack 1 and Windows XP Service Pack 3 (and/or separate installation of Windows XP Update KB955704). It is loosely based on the File Allocation Table architecture, but incompatible, proprietary and protected by patents.^[59]

exFAT is intended for use on flash drives and memory cards such as SDXC and Memory Stick XC, where FAT32 is otherwise used. Vendors usually pre-format SDXC cards with it. Its main benefit is its exceeding of the 4 GB file size limit, as file size references are stored with eight instead of four bytes, increasing the limit to $2^{64}-1$ bytes.

Microsoft's GUI and command-line format utilities offer it as an alternative to NTFS (and, for smaller partitions, to FAT16B and FAT32). The MBR partition type is 0x07 (the same as used for IFS, HPFS, and NTFS). Logical geometry information located in the VBR is stored in a format not resembling any kind of BPB.

In early 2010, the file system was reverse-engineered by the SANS Institute.^[60] On August 28, 2019, Microsoft announced that it will be making the technical specification for exFAT publicly available so that it can be used in the Linux kernel and other operating systems.^[61]

Patents

Microsoft applied for, and was granted, a series of patents for key parts of the FAT file system in the mid-1990s. All four pertain to long-filename extensions to FAT first seen in Windows 95: U.S. patent 5,579,517,^[62] U.S. patent 5,745,902,^[63] U.S. patent 5,758,352,^[64] U.S. patent 6,286,013 (all expired since 2013).^[65]

On December 3, 2003, Microsoft announced^[66] that it would be offering licenses for use of its FAT specification and "associated intellectual property", at the cost of a US\$0.25 royalty per unit sold, with a \$250,000 maximum royalty per license agreement.^[67] To this end, Microsoft cited four patents on the FAT file system as the basis of its intellectual property claims.

In the EFI FAT32 specification^[6] Microsoft specifically grants a number of rights, which many readers have interpreted as permitting operating system vendors to implement FAT.^[68]

Non-Microsoft patents affecting FAT include: U.S. patent 5,367,671, specific to the OS/2 extended object attributes (expired in 2011).^[69]

Challenges and lawsuits

The Public Patent Foundation (PUBPAT) submitted evidence to the US Patent and Trademark Office (USPTO) in 2004 disputing the validity of U.S. patent 5579517,^[62] including prior art references from Xerox and IBM.^[70] The USPTO opened an investigation and concluded by rejecting all claims in the patent.^[71] The next year, the USPTO further announced that following the re-examination process, it affirmed the rejection of '517 and additionally found U.S. patent 5,758,352^[64] invalid on the grounds that the patent had incorrect assignees.

However, in 2006 the USPTO ruled that features of Microsoft's implementation of the FAT system were "novel and non-obvious", reversing both earlier decisions and leaving the patents valid.^[72]

In February 2009, Microsoft filed a patent infringement lawsuit against TomTom alleging that the device maker's products infringe on patents related to VFAT long filenames. As some TomTom products are based on GNU/Linux, this marked the first time that Microsoft tried to enforce its patents against the GNU/Linux platform.^[73] The lawsuit was settled out of court the following month with an agreement that Microsoft be given access to four of TomTom's patents, that TomTom will drop support for the VFAT long filenames from its products, and that in return Microsoft not seek legal action against TomTom for the five-year duration of the settlement agreement.^[74]

In October 2010, Microsoft filed a patent infringement lawsuit against Motorola alleging several patents (including two of the VFAT patents) were not licensed for use in the Android operating system.^[75] They also submitted a complaint to the ITC.^[76] Developers of open source software have designed methods intended to circumvent Microsoft's patents.^{[77][78]}

In 2013, patent EP0618540 "common name space for long and short filenames" (expired since 2014^[79]) was invalidated in Germany.^[80] After the appeal was withdrawn, this judgment became final on the 28th October 2015.^[81]

See also

- Comparison of file systems
- Design of the FAT file system
- Drive letter assignment
- List of file systems
- Transaction-Safe FAT File System

Notes

1. Since Windows 2000, Microsoft Windows uses UTF-16 instead of UCS-2 for the internal "Unicode". In UTF-16, a "character" (code point) may take up two code units.
2. Windows XP has been observed to create similar hybrid disks when reformatting FAT16B formatted ZIP-100 disks to FAT32 format. The resulting volumes were FAT32 by format, but still used the FAT16B EBPB. (It is unclear how Windows determines the location of the root directory on FAT32 volumes, if only a FAT16 EBPB was used.)
3. Sources differ in regard to the first NCR data entry terminal integrating support for the FAT file system. According to Stephen Manes and Paul Andrews, "Gates", development was for a NCR 8200 in late 1977, incorrectly classified as a floppy-based upgrade to the NCR 7200, which had been released in 1975-11 (model I and IV) and was built around an Intel 8080 8-bit processor, but was cassette-based only. However, the NCR Century 8200 was a 16-bit minicomputer, onto which several data entry terminals could be hooked up. Marc McDonald even remembered a NCR 8500, a mainframe of the Criterion series, which can be ruled out as well. Announced 1977-10 for shipment in 1978-02, NCR also introduced the NCR I-8100 series including the 8080-based NCR I-8130 and NCR I-8150 models of small business systems featuring dual floppy disks. Other sources indicate that either the NCR 7200 series itself or the successor series were the actual target platform. NCR Basic Plus 6 (based on Microsoft Extended BASIC-80) became available for the cassette-based NCR 7200 model VI in Q1/1977. The NCR 7500 series was released in 1978, based on a similar 8080 hardware, but now including NCR 7520 and 7530 models featuring 8-inch diskettes. NCR Basic +6, a precursor or adaptation of Standalone Disk BASIC-80 was available for them at least since 1979. One source claims that a special NCR 7200 model variant with two 8-inch diskettes and Microsoft BASIC existed and was imported by NCR Sydney into Australia the least.
4. DR-DOS is able to boot off FAT12/FAT16 logical sector media with logical sector sizes up to 1024 bytes.
5. A driver named VFAT appeared before Windows 95, in Windows for Workgroups 3.11, but this older version was only used for implementing 32-bit file access and did not support long file names.

References

1. "File Systems" (<https://technet.microsoft.com/en-us/library/cc938937.aspx>). Microsoft TechNet.

2001. Retrieved July 31, 2011.
2. Microsoft (2006-11-15). Windows 95 CD-ROM CONFIG.TXT File (<http://c-bit.org/kb/135481/EN-US/>) Article 135481, Revision: 1.1, retrieved 2011-12-22: "For each hard disk, specifies whether to record the date that files are last accessed. Last access dates are turned off for all drives when your computer is started in safe mode, and are not maintained for floppy disks by default. Syntax: `ACCDATE=drive1+|- [drive2+|-]...`"
 3. "A brief introduction to FAT (File Allocation Table) formats" (<https://web.archive.org/web/20150925082826/http://www.wizcode.com/articles/comments/a-brief-introduction-to-fat-file-allocation-table/>). *www.wizcode.com*. Archived from the original (<http://www.wizcode.com/articles/comments/a-brief-introduction-to-fat-file-allocation-table/>) on September 25, 2015. Retrieved September 24, 2015.
 4. "Comparing NTFS and FAT file systems" (<http://windows.microsoft.com/en-us/windows-vista/comparing-ntfs-and-fat-file-systems>). Microsoft. Retrieved January 27, 2014.
 5. JEIDA/JEITA/CIPA (2010). "Standard of the Camera & Imaging Products Association, CIPA DC-009-Translation-2010, Design rule for Camera File system: DCF Version 2.0 (Edition 2010)" (https://web.archive.org/web/20130930190707/http://www.cipa.jp/english/hyoujunka/kikaku/pdf/DC-009-2010_E.pdf) (PDF). Archived from the original (http://www.cipa.jp/english/hyoujunka/kikaku/pdf/DC-009-2010_E.pdf) (PDF) on September 30, 2013. Retrieved 2011-04-13.
 6. "Microsoft Extensible Firmware Initiative FAT32 File System Specification, FAT: General Overview of On-Disk Format" (<http://download.microsoft.com/download/1/6/1/161ba512-40e2-4cc9-843a-923143f3456c/fatgen103.doc>). Microsoft. March 30, 2011. Retrieved December 21, 2018.
 7. "Volume and File Structure of Disk Cartridges for Information Interchange" (<http://www.ecma-international.org/publications/standards/Ecma-107.htm>). *Standard ECMA-107 (2nd ed., June 1995)*. ECMA. 1995. Retrieved July 30, 2011.
 8. "Information technology – Volume and file structure of disk cartridges for information interchange" (http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=21273). *ISO/IEC 9293:1994*. ISO catalogue. 1994. Retrieved January 6, 2012.
 9. "Information processing – Volume and file structure of flexible disk cartridges for information interchange" (http://www.iso.org/iso/iso_catalogue/catalogue_ics/catalogue_detail_ics.htm?csnumber=16948). *ISO 9293:1987*. ISO catalogue. 1987. Retrieved January 6, 2012.
 10. Reynolds, Aaron R.; Adler, Dennis R.; Lipe, Ralph A.; Pedrizetti, Ray D.; Parsons, Jeffrey T.; Arun, Rasipuram V. (May 26, 1998). "Common name space for long and short filenames" (<http://www.google.com/patents?id=bUohAAAAEBAJ>). *US Patent 5758352*. Retrieved January 19, 2012.
 11. Chappell, Geoff (January 1994). Schulman, Andrew; Pedersen, Amorette (eds.). *DOS Internals*. The Andrew Schulman Programming Series (1st printing, 1st ed.). Addison Wesley Publishing Company. ISBN 978-0-201-60835-9. (xxvi+738+iv pages, 3.5"-floppy [1] (https://web.archive.org/web/20190421200111/https://gopher.tildeverse.org/gopher.viste.fr/9/programming/PC/DOS/DOS%2520Internals/DOS_Internals.zip)[2] (https://web.archive.org/web/20200222111608/http://www.pcjs.org/pubs/pc/programming/DOS_Internals/) Errata: [3] (<https://web.archive.org/web/20200222111742/http://www.geoffchappell.com/notes/dos/internals/>)[4] (<https://web.archive.org/web/20200222111853/https://gopher.tildeverse.org/gopher.viste.fr/1/programming/PC/DOS/DOS%2520Internals/crtldr>)[5] (<https://web.archive.org/web/20200222111945/https://gopher.tildeverse.org/gopher.viste.fr/1/programming/PC/DOS/DOS%2520Internals/xmswatch>)
 12. *Xerox BASIC-80 – basic-80 reference manual* (http://bitsavers.trailing-edge.com/pdf/xerox/820-I/I/BASIC-80_5.0.pdf) (PDF). 5.0. Microsoft, Xerox. 1979. 610P70641. Retrieved June 2, 2014. (NB. For Microsoft (Standalone Disk / Disk / Extended / 8K) BASIC-80, (Standalone Disk / Extended) BASIC-86, BASIC Compiler, release 5.0)

13. *MICROSOFT BASIC-80 version 5.0 reference manual / BASIC-80 Interpreter and Compiler Addendum Release 5.1* (http://bitsavers.trailing-edge.com/pdf/xerox/820-II/BASIC-80_5.0.pdf) (PDF). 5.1. Microsoft. 1979. Retrieved June 2, 2014. (NB. For Microsoft (Standalone Disk / Disk / Extended / 8K) BASIC-80, (Standalone Disk / Extended) BASIC-86, BASIC Compiler, release 5.1)
14. Duncan, Ray; Bostwick, Steve; Burgoyne, Keith (1988). *The MS-DOS Encyclopedia: versions 1.0 through 3.2* (Completely reworked ed.). Redmond, Washington, USA: Microsoft Press. ISBN 1-55615-049-0. LCCN 87-21452 (<https://lccn.loc.gov/87-21452>). OCLC 16581341 (<http://www.worldcat.org/oclc/16581341>). (xix+1570 pages; 26 cm) (NB. This edition was published in 1988 after extensive rework of the withdrawn 1986 first edition by a different team of authors. [6] (<https://www.pcjs.org/pubs/pc/reference/microsoft/mspl13/msdos/encyclopedia/>))
15. Manes, Stephen; Andrews, Paul (1993). *Gates: How Microsoft's Mogul Reinvented an Industry —and Made Himself the Richest Man in America*. Doubleday. ISBN 0-385-42075-7.
16. Hunter, David (1983). "Tim Paterson – The roots of DOS" (<https://web.archive.org/web/20190906120449/http://www.patersonstech.com/dos/softalk.aspx>). *Softalk for the IBM Personal Computer* (March 1983). Archived from the original (<http://www.patersonstech.com/dos/softalk.aspx>) on September 6, 2019. Retrieved June 2, 2014.
17. Schulman, Andrew; Brown, Ralf D.; Maxey, David; Michels, Raymond J.; Kyle, Jim (1994) [November 1993]. *Undocumented DOS: A programmer's guide to reserved MS-DOS functions and data structures - expanded to include MS-DOS 6, Novell DOS and Windows 3.1* (https://archive.org/details/undocumenteddos00andr_0/page/11) (2 ed.). Reading, Massachusetts: Addison Wesley. p. 11 (https://archive.org/details/undocumenteddos00andr_0/page/11). ISBN 0-201-63287-X. (xviii+856+vi pages, 3.5"-floppy) Errata: [7] (<https://web.archive.org/web/20190417215556/http://www.cs.cmu.edu/afs/cs/user/ralf/pub/books/UndocumentedDOS/errata.ud2>)[8] (https://web.archive.org/web/20190417212906/https://www.pcjs.org/pubs/pc/programming/Undocumented_DOS/#errata-2nd-edition)
18. Paterson, Tim (September 30, 2007). "Design of DOS" (<http://dosmandrive1.blogspot.com/2007/09/design-of-dos.html>). *DosMan Drive1*. Retrieved July 4, 2011.
19. Seattle Computer Products (August 1980). "86-DOS - 8086 OPERATING SYSTEM - \$95" (http://archive.org/stream/byte-magazine-1980-08/1980_08_BYTE_05-08_The_Forth_Language#page/n173/mode/2up). *Byte* (Advertisement). Vol. 5 no. 8. BYTE Publications Inc. p. 173. ISSN 0360-5280 (<https://www.worldcat.org/issn/0360-5280>). CODEN BYTEDJ (https://cassi.ca.s.org/searching.jsp?searchIn=codens&exactMatch=y&c=Wly460-R_DY&searchFor=BYTEDJ). Retrieved August 18, 2013. [9] (<https://tech-insider.org/personal-computers/research/acrobat/8008-a.pdf>) (NB. The SCP advertisement already calls the product 86-DOS, but does not mention a specific version number. Version 0.3 is known to be called 86-DOS already, so the name change must have taken place either for version 0.2 or immediately afterwards in August 1980.)
20. Seattle Computer Products (1981). "SCP 86-DOS 1.0 Addendum" (http://bitsavers.informatik.uni-stuttgart.de/pdf/seattleComputer/86-DOS_1.0_Addendum.pdf) (PDF). Retrieved March 10, 2013.
21. Wallace, James; Erickson, Jim; (1992); *Hard Drive: Bill Gates and the Making of the Microsoft Empire*, John Wiley & Sons, ISBN 0-471-56886-4
22. Norton, Peter; (1986); *Inside the IBM PC, Revised and Enlarged*, Brady, ISBN 0-89303-583-1, p. 157
23. Jenkinson, Brian; Sammes, A. J. (2000). *Forensic Computing: A Practitioner's Guide (Practitioner Series)* (https://archive.org/details/forensiccomputin00samm_915). Berlin: Springer. p. 157 (https://archive.org/details/forensiccomputin00samm_915/page/n163). ISBN 1-85233-299-9. "... only 2¹² (that is, 4096) allocation units or clusters can be addressed. In fact, the number is less than this, since 000h and 001h are not used and FF0h to FFFh are reserved or used for other purposes, leaving 002h to FEFh (2 to 4079) as the range of possible clusters."

24. Paterson, Tim (1983). "An Inside Look at MS-DOS" (<https://web.archive.org/web/20110720115141/http://patersonstech.com/Dos/Byte/InsideDos.htm>). Byte. Archived from the original (http://www.patersonstech.com/dos/Byte/InsideDos.htm#InsideDos_44) on July 20, 2011. Retrieved 2011-07-18. "The numbering starts with 2; the first two numbers, 0 and 1, are reserved."
25. IBM (1984). *IBM PC DOS 3.0 announcement letter*.
26. IBM (1985). *IBM PC DOS Technical Reference*. First Edition, P/N 6024181, dated February 1985.
27. Microsoft Knowledge Base article: "MS-DOS Partitioning Summary" (<http://c-bit.org/kb/69912/EN-US/>)
28. FYI – Installing DR DOS on NEC DOS 3.3 Partitions (http://cd.textfiles.com/netwares/NOV_INFO/RNW93/10JAN93.MON), Novell, January 5, 1993, FYI.M.1101, retrieved August 12, 2014
29. Brouwer, Andries. "List of partition identifiers for PCs" (http://www.win.tue.nl/~aeb/partitions/partition_types-1.html).
30. "Q78407: Wyse DOS 3.3 Partitions Incompatible with MS-DOS 5.x and 6.x" (<ftp://ftp.microsoft.com/misc1/PEROPSYS/MSDOS/KB/Q78/4/07.TXT>). Microsoft. December 17, 2000. Alt URL (<https://archive.org/download/ftp.microsoft.com/ftp.microsoft.com.zip/ftp.microsoft.com%2FMISC1%2FPEROPSYS%2FMSDOS%2FKB%2FQ78%2F4%2F07.TXT>)
31. "Q68176: Upgrading Pre-4.0 Systems with Logical Drive(s) > 32 MB" (<ftp://ftp.microsoft.com/misc1/PEROPSYS/MSDOS/KB/Q68/1/76.TXT>). Microsoft. December 17, 2000. Alt URL (<https://archive.org/download/ftp.microsoft.com/ftp.microsoft.com.zip/ftp.microsoft.com%2FMISC1%2FPEROPSYS%2FMSDOS%2FKB%2FQ68%2F1%2F76.TXT>)
32. Brouwer, Andries. "Properties of partition tables" (http://www.win.tue.nl/~aeb/partitions/partition_types-2.html#ss2.6).
33. Kuhnt, Udo; Georgiev, Luchezar I.; Davis, Jeremy (2007). "FAT+ draft revision 2" (<https://web.archive.org/web/20150219123449/http://www.fdos.org/kernel/fatplus.txt>) (2 ed.). Archived from the original (<http://www.fdos.org/kernel/fatplus.txt>) (FATPLUS.TXT) on February 19, 2015. Retrieved August 5, 2015.
34. "Dskprobe Overview: Data Recovery" ([https://technet.microsoft.com/en-us/library/cc736327\(v=ws.10\).aspx](https://technet.microsoft.com/en-us/library/cc736327(v=ws.10).aspx)). Microsoft TechNet. March 28, 2003. Retrieved August 3, 2011.
35. "Errors Creating Files or Folders in the Root Directory" (<http://c-bit.org/kb/120138/EN-US/>). Microsoft Help and Support. December 16, 2004. Retrieved October 14, 2006.
36. "mkdosfs man page" (<http://www.die.net/doc/linux/man/man8/mkdosfs.8.html>).
37. "Windows 98 Resource Kit – Chapter 10 – Disks and File Systems" (<https://technet.microsoft.com/en-us/library/cc768180.aspx>). Microsoft TechNet. 1998. Retrieved July 16, 2012.
38. "Limitations of FAT32 File System" (<http://c-bit.org/kb/184006/EN-US/>). Microsoft Knowledge Base. March 26, 2007. Retrieved August 21, 2011. "Clusters cannot be 64 kilobytes (KB) or larger"
39. "Limitations of the FAT32 File System in Windows XP" (<http://c-bit.org/kb/314463/EN-US/>). Microsoft Knowledge Base. December 1, 2007. Retrieved August 21, 2011.
40. Glenn, Walter. "How to Format USB Drives Larger Than 32GB With FAT32 on Windows" (<http://www.howtogeek.com/316977/how-to-format-usb-drives-larger-than-32gb-with-fat32-on-windows/>). How-To Geek. Retrieved January 26, 2021.
41. *README.TXT – Caldera DR-DOS FAT32 Enabled Boot Disk (DRFAT32)*. Caldera, Inc. July 24, 1998.
42. *DRFAT32.SYS R1.00 INT 13h Interface for FAT32 Redirector*, Caldera, Inc., September 11, 1998
43. IBM; *4690 OS User's Guide Version 5.2*, IBM document SC30-4134-01, 2008-01-10 ([10] (ftp://ftp.software.ibm.com/software/retail/pubs/sw/opsys/4690/ver5r2/bsf1_UG_mst.pdf))

44. Karpowitz, Christina (September 23, 1998). "PowerQuest PartitionMagic 4.0 now available" (<https://web.archive.org/web/19990208204638/http://www.powerquest.com/press/PM4available.html>). PowerQuest. Archived from the original (<http://www.powerquest.com/press/PM4available.html>) on February 8, 1999. Retrieved April 17, 2015.
45. Livingston, Brian (October 28, 1998). "FAT-32X may operate differently than FAT-32 on large hard drives" (<http://brianlivingston.com/windowmanager/archive/cgi-bin/new/livingst/981026bl.htm>). *InfoWorld*. Vol. 20 no. 43. Retrieved April 17, 2015.
46. Duitz, Neal (July 17, 2001). "Can anyone explain FAT32X?" (<https://web.archive.org/web/20040613232209/http://www.win98private.net/fat32x.htm>). Win98 Private FAQ, Windows 98 Consumer Preview Program. Archived from the original (<http://www.win98private.net/fat32x.htm>) on June 13, 2004. Retrieved April 17, 2015.
47. Costanzo, Lance (May 14, 1998). "FAT32X" (<https://web.archive.org/web/19980521015015/http://lance.advantweb.com/fat32x/>). Archived from the original (<http://lance.advantweb.com/fat32x/>) on May 21, 1998. Retrieved April 17, 2015.
48. Steinberg, David (May 1, 1998). "What is a FAT32X partition?" (<http://www.sysopt.com/showthread.php?88915-What-is-FAT32X&p=532052&viewfull=1#post532052>). *Tech Tip / FAQ*. PowerQuest Technical Support. Retrieved April 17, 2015.
49. Eager, Bob; Tavi Systems (October 28, 2000); *Implementation of extended attributes on the FAT file system* ([11] (<http://www.tavi.co.uk/os2pages/eadata.html>) Archived (<https://web.archive.org/web/20060613212300/http://www.tavi.co.uk/os2pages/eadata.html>) June 13, 2006, at the Wayback Machine)
50. Kelder, Henk; (2003); *FAT32.TXT for FAT32.IFS version 0.9.13.* ([12] (<http://svn.netlabs.org/repos/fat32/branches/fat32-0.9/src/fat32.txt>)): "This byte [...] is not modified while running Windows 95 and neither [sic] by SCANDISK or DEFRAG. [...] If another program sets the value to 0x00 for a file that has EAs these EAs will no longer be found using DosFindFirst/Next calls only. The other OS/2 calls for retrieving EAs (DosQueryPathInfo, DosQueryFileInfo and DosEnumAttribute) do not rely on this byte. Also the opposite could [...] occur. [...] In this situation only the performance of directory scans will be decreased. Both situations [...] are corrected by CHKDSK".
51. Kelder, Henk; *FAT32.TXT for FAT32.IFS version 0.74* ("Archived copy" (<https://web.archive.org/web/20120330171510/http://macarlo.com/fat32v074.htm>). Archived from the original (<http://macarlo.com/fat32v074.htm>) on March 30, 2012. Retrieved January 14, 2012.). Comment: This older version of the README file still discusses the old 0xEA and 0xEC magic values.
52. Eager, Bob (October 28, 2000). "Implementation of extended attributes on the FAT file system" (<https://web.archive.org/web/20060613212300/http://www.tavi.co.uk/os2pages/eadata.html>). *Tavi OS/2 pages*. Archived from the original (<http://www.tavi.co.uk/os2pages/eadata.html>) on June 13, 2006. Retrieved October 14, 2006.
53. IBM; *4690 OS Programming Guide Version 5.2*, IBM document SC30-4137-01, 2007-12-06 ([13] (ftp://ftp.software.ibm.com/software/retail/pubs/sw/opsys/4690/ver5r2/bsi1_PG_mst.pdf))
54. "Release notes for v2.5.7" (<https://www.kernel.org/pub/linux/kernel/v2.5/ChangeLog-2.5.7>). The Linux Kernel archives. March 12, 2002. Retrieved October 14, 2006.
55. Kuhnt, Udo (July 21, 2011). "DR-DOS/OpenDOS Enhancement Project" (<http://www.drDOSprojects.de/>). Retrieved April 20, 2015.
56. "FATX Specification" (<http://www.free60.org/wiki/FATX>). free60 wiki. Retrieved August 16, 2011.
57. de Quincey, Andrew; Murray-Pitts, Lucien (August 29, 2008). "Xbox partitioning and file system details" (https://web.archive.org/web/20100617020539/http://www.xbox-linux.org/wiki/Xbox_Partitioning_and_Filesystem_Details). 0.13. Xbox-Linux project. Archived from the original (http://www.xbox-linux.org/wiki/Xbox_Partitioning_and_Filesystem_Details) on June 17, 2010. Retrieved May 25, 2014.

58. Steil, Michael (February 26, 2008) [2003]. "Differences between Xbox FATX and MS-DOS FAT" (https://web.archive.org/web/20100617022009/http://www.xbox-linux.org/wiki/Differences_between_Xbox_FATX_and_MS-DOS_FAT). Xbox-Linux project. Archived from the original (http://www.xbox-linux.org/wiki/Differences_between_Xbox_FATX_and_MS-DOS_FAT) on June 17, 2010. Retrieved May 25, 2014.
59. Microsoft. "exFAT File System Intellectual Property licensing program" (<https://web.archive.org/web/20130507183540/http://www.microsoft.com/en-us/legal/intellectualproperty/IPLicensing/Programs/exFATFileSystem.aspx>). Archived from the original (<http://www.microsoft.com/en-us/legal/intellectualproperty/IPLicensing/Programs/exFATFileSystem.aspx>) on May 7, 2013. Retrieved 2013-04-23.
60. <https://www.sans.org/reading-room/whitepapers/forensics/reverse-engineering-microsoft-exfat-file-system-33274>
61. "exFAT in the Linux kernel? Yes!" (<https://cloudblogs.microsoft.com/opensource/2019/08/28/exfat-linux-kernel/>). *Microsoft Open Source Blog*. August 28, 2019. Retrieved October 6, 2019.
62. US 5579517 (<https://worldwide.espacenet.com/textdoc?DB=EPODOC&IDX=US5579517>), Reynolds, Aaron R.; Dennis R. Adler & Ralph A. Lipe et al., "Common name space for long and short filenames", issued 1996
63. US 5745902 (<https://worldwide.espacenet.com/textdoc?DB=EPODOC&IDX=US5745902>), Miller, Thomas J. & Gary D. Kimura, "Method and system for accessing a file using file names having different file name formats", issued 1998
64. US 5758352 (<https://worldwide.espacenet.com/textdoc?DB=EPODOC&IDX=US5758352>), Reynolds, Aaron R.; Dennis R. Adler & Ralph A. Lipe et al., "Common name space for long and short filenames", issued 1998
65. US 6286013 (<https://worldwide.espacenet.com/textdoc?DB=EPODOC&IDX=US6286013>), Reynolds, Aaron R.; Dennis R. Adler & Ralph A. Lipe et al., "Method and system for providing a common name space for long and short file names in an operating system", issued 1996
66. Microsoft.com (<http://www.microsoft.com/presspass/press/2003/dec03/12-03ExpandIPPR.msp>) Archived (<https://web.archive.org/web/20090822044026/http://www.microsoft.com/presspass/press/2003/dec03/12-03ExpandIPPR.msp>) August 22, 2009, at the *Wayback Machine*
67. "FAT File System" (<http://web.archive.loc.gov/all/20160921194326/http://www.microsoft.com/iplicensing/productDetail.aspx?product%20title=FAT%20File%20System>). *Intellectual Property Licensing*. Microsoft. Archived from the original (<http://www.microsoft.com/iplicensing/productDetail.aspx?product%20title=FAT%20File%20System>) on September 21, 2016.
68. Garrett, Matthew (January 19, 2012). "EFI and Linux: the future is here, and it's awful" (<https://www.youtube.com/watch?v=V2aq5M3Q76U>). *linux.conf.au*. YouTube. Retrieved January 12, 2014.
69. US 5367671 (<https://worldwide.espacenet.com/textdoc?DB=EPODOC&IDX=US5367671>), Feigenbaum, Barry A. & Felix Miro, "System for accessing extended object attribute (EA) data through file name or EA handle linkages in path tables", issued 1994
70. Ravicher, Daniel B. (April 15, 2004). "PUBPAT's Request for Reexamination of Microsoft's FAT Patent" (http://www.pubpat.org/assets/files/MicrosoftFAT/Reynolds_517_Reexam_Request.pdf) (PDF). *Public Patent Foundation*. Retrieved January 12, 2014.
71. USPTO (September 30, 2004). "Patent Office's Office Action Rejecting Microsoft FAT Patent" (http://www.pubpat.org/assets/files/MicrosoftFAT/Reynolds_517_Rejected_040916.PDF) (PDF). *Public Patent Foundation*. Retrieved January 12, 2014.
72. Broache, Anne (January 10, 2006). "Microsoft's file system patent upheld" (<https://www.cnet.com/news/microsofts-file-system-patent-upheld/>). CNET News.
73. Paul, Ryan (February 25, 2009). "Microsoft suit over FAT patents could open OSS Pandora's Box" (<https://arstechnica.com/microsoft/news/2009/02/microsoft-sues-tomtom-over-fat-patents-in-linux-based-device.ars>). *arstechnica.com*. Retrieved February 28, 2009.

74. Fried, Ina (March 30, 2009). "Microsoft, TomTom settle patent dispute" (http://news.cnet.com/8301-13860_3-10206988-56.html). cnet.com. Retrieved August 22, 2009.
75. "Microsoft Motorola Patent Suit" (<https://www.scribd.com/doc/38550703/Microsoft-Motorola-Patent-Suit>). October 1, 2010. Retrieved October 2, 2010.
76. Protalinski, Emil (October 1, 2010). "Microsoft sues Motorola, citing Android patent infringement" (<https://arstechnica.com/microsoft/news/2010/10/microsoft-sues-motorola-citing-android-patent-infringement.ars>). arstechnica.com. Retrieved October 2, 2010.
77. Paul, Ryan (July 2, 2009). "New Linux patch could circumvent Microsoft's FAT patents" (<https://arstechnica.com/information-technology/2009/07/vfat-linux-patch-could-circumvent-microsofts-patent-claims/>). ArsTechnica.com. Retrieved October 30, 2013.
78. Brown, Eric (July 2, 2009). "Can FAT patch avoid Microsoft lawsuits?" (<https://web.archive.org/web/20130131034455/http://www.desktoplinux.com/news/NS4980952387.html>). DesktopLinux.Com. Archived from the original (<http://www.desktoplinux.com/news/NS4980952387.html?kc=rss>) on January 31, 2013. Retrieved August 23, 2009.
79. [14] (<https://patents.google.com/patent/EP0618540B1/en>), "Common name space for long and short filenames", issued 1994-03-31
80. Müller, Florian (December 5, 2013). "Federal Patent Court of Germany invalidates Microsoft FAT patent, appeals court may disagree" (<http://www.fosspatents.com/2013/12/federal-patent-court-of-germany.html>). FOSS Patents. Retrieved January 12, 2014.
81. Labesius, Stefan (May 28, 2017). "Bundespatentgericht erklärt VFAT-Patent erneut für nichtig" (<https://www.ifross.org/artikel/bundespatentgericht-erkl-rt-vfat-patent-erneut-f-r-nichtig>). ifrOSS. Retrieved August 2, 2019.

External links

- *Description of the FAT32 File System* (<http://support.microsoft.com/kb/154997/>): Microsoft Knowledge Base Article 154997
- *MS-DOS: Directory and Subdirectory Limitations* (<https://jeffpar.github.io/kbarchive/kb/039/Q39927/>): Microsoft Knowledge Base Article 39927
- *Overview of FAT, HPFS, and NTFS File Systems* (<http://support.microsoft.com/kb/100108/>): Microsoft Knowledge Base Article 100108
- Microsoft Technet; *Volume and file size limits of FAT file systems* (<https://web.archive.org/web/20060307082555/http://www.microsoft.com/technet/prodtechnol/winxppro/reskit/c13621675.msp>), copy made by Internet Archive Wayback Machine (<https://archive.org/>) of an article with summary of limits in FAT32 which is no longer available on Microsoft website.
- Chen, Raymond; *Microsoft TechNet: A Brief and Incomplete History of FAT32* (<https://web.archive.org/web/20081118122857/http://www.microsoft.com/technet/technetmag/issues/2006/07/WindowsConfidential/>)
- *Fdisk does not recognize full size of hard disks larger than 64 GB* (<http://support.microsoft.com/kb/263044>): Microsoft Knowledge Base Article 263044
- *Microsoft Windows XP: FAT32 File System* (https://web.archive.org/web/20050319235548/http://www.microsoft.com/resources/documentation/Windows/XP/all/reskit/en-us/prkc_fil_cycz.asp), copy made by the Internet Archive's Wayback Machine of an article with summary of limits in FAT32 which is no longer available on Microsoft website.

Retrieved from "https://en.wikipedia.org/w/index.php?title=File_Allocation_Table&oldid=1025271537"

This page was last edited on 26 May 2021, at 17:33 (UTC).

Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia

Foundation, Inc., a non-profit organization.