文件分配表

维基百科，自由的百科全书

	FAT		
开发商	微软		
全名	文件分配表		
	（12位版本）	（16位版本）	（32位版本）
发布	1980年前（微软Disk BASIC）	1987年11月 (Compaq DOS 3.31)	1996年8月 (Windows 95 OSR2)
分区标识	0x01（MBR）	0x04, 0x06, 0x0E（MBR）	0x0B, 0x0C（MBR） EBD0A0A2-B9E5-4433-87C0-68B6B72699C7（GPT）
结构	**FAT12**	**FAT16**	**FAT32**
目录内容	表格		
文件配置	链表		
坏块	对含损坏扇区的簇加以标志		
限制	**FAT12**	**FAT16**	**FAT32**
最大文件大小	32 MB	2 GB	4 GB - 1 byte (2^{32}-1)
最大文件数量	4,077 (2^{12}-19)	65,517 (2^{16}-19)	268,435,437 (2^{28}-19)
最长文件名限制	8.3或者 长文件名255个字符		
最大卷大小	32 MB	2 GB, 使用64KB簇时可达4 GB，但非所有系统均支持	2 TB 使用32KB簇时可达8 TB
Features	**FAT12**	**FAT16**	**FAT32**
记录日期	创建、修改、访问		
日期范围	1980年1月1日至2107年12月31日		
日期分辨率	2 s		
岔流	非原生		
属性	只读，隐藏，系统，卷标，子目录，归档[1]		
透明压缩	Per-volume, Stacker，DoubleSpace，DriveSpace		无
透明加密	Per-volume only with DR-DOS		无

文件分配表（英语：File Allocation Table，首字母缩略字：FAT），是一种由微软发明并拥有部分专利[2]

的文件系统，供MS-DOS使用，也是所有非NT核心的Windows系统使用的文件系统。

FAT文件系统考虑当时计算机性能有限，所以未被复杂化，因此几乎所有个人计算机的操作系统都支持。这特性使它成为理想的软盘和存储卡文件系统，也适合用作不同操作系统中的数据交流。现在，一般所讲的FAT专指FAT32。

但FAT有一个严重的缺点：当文件删除后写入新数据，FAT不会将文件整理成完整片段再写入，长期使用后会使文件数据变得逐渐分散，而减慢了读写速度。碎片整理是一种解决方法，但必须经常重组来保持FAT文件系统的效率。

目录

历史

FAT文件系统遵行已用了多年的软件方法来进行规范。它在1977年由比尔·盖茨和马斯·麦当劳为了管理磁盘而发明，并在1980年被添·彼得逊的86-DOS操作系统采用。这也是86-DOS操作系统与CP/M操作系统最大的不

同点，若非此项差异，86-DOS操作系统与CP/M操作系统几乎可说完全相同。[3]

FAT12

初期的FAT就是现在俗称的FAT12。作为软盘的文件系统，它有几项限制：不支持分层性结构，簇定址只有12位（这使得控制FAT有些棘手）而且只支持最多32M（2^{16}）的分区。

当时入门级的磁盘是5.25"、单面、40磁道、每个磁道8个扇区、容量略少于160KB。上面的限制超过了这个容量一个或几个数量级，同时允许将所有的控制结构放在第一个磁道，这样在读写操作时移动磁头。这些限制在随后的几年时间里被逐步增大。

由于唯一的根目录也必须放在第一个磁道，能够存放的文件个数就限制在几十个。

目录

MS-DOS 2.0为了支持以内置10MB硬盘为特色的IBM PC XT，因此几乎与该计算机同时在1983年初发布。它引进了层次目录结构，除了允许更有效率地组织文件外，目录允许在硬盘上存储更多的文件，这是因为最大文件个数不再受制于（仍然是固定且有限的）根目录大小。这个数目现在能够等同于簇的数目（甚至更大，这是考虑到长度为0的文件并不占据任何FAT簇）。

FAT本身的格式并没有改变。PC XT的10MB的硬盘有4KB大小的簇。如果后来安装了一个20MB的硬盘，并且使用MS-DOS 2.0格式化，最后的簇大小将变为8KB，硬盘容量将变为15.9MB。

FAT16的开始

在1984年，IBM发布PC AT，内含一个20MB的硬盘。微软公司也同步发布了MS-DOS 3.0。簇集地址增加至16位，允许更大数量的簇（最大65,517），所以有更大的文件系统大小。但是，最大数量扇区及最大分区（相当于磁盘）的大小仍是32MB。所以，尽管技术上已经是“FAT16”，这种格式并不是我们今天常见到的这个名字所代表的格式。在MS-DOS 3.0格式化一个20 MB的硬盘，这硬盘将不能被MS-DOS 2.0或之前的版本所访问。当然，MS-DOS 3.0仍然可访问MS-DOS 2.0的格式（8KB簇的分区）。

MS-DOS 3.0也开始支持高密度1.2MB 5.25"磁盘，最著名的是每个磁道有15个扇区，这样就允许FAT有更大的空间。这或许促进了一个对于簇大小的不确定的优化，簇大小从2个扇区减到1个。这样做的最后结果是高密度磁盘比旧的*双密度*磁盘的速度大幅度降低。

扩展分区和逻辑驱动器

除了改进FAT文件系统本身的结构之外，另一个提高FAT存储空间的方式是支持多个磁盘分区。最初，受限于主引导记录中文件分配表的固定结构一个硬盘最多只能切出多达4个分区。然而，由于DOS设计要求只能有一个分区标识为“活动的（Active）”，它也是主引导代码启动所用的分区。使用DOS工具不可能创建几个“主”DOS分区，并且第三方的工具也至少会警告这样一个机制将与DOS不兼容。

为了用一种兼容的方式使用更多的分区，一种新的分区类型被开发出来（1986年1月的MS-DOS 3.2），*扩展分区*它实际上是另外称为*逻辑分区*的一个容器。最初它里面只允许有一个逻辑分区、支持最大64MB的硬盘。在MS-DOS 3.3（1987年8月）这个限制更改到24个分区；它可能来自于强制性的C:-Z: 的磁盘命名规则。逻辑分区表使用盘上的数据结构来描述，可能是为了简化编码它与主引导记录非常相似，并且它们组织成类似于俄罗斯套娃那样的结构。一颗硬盘中只能有一个扩展分区。

在扩展分区观念导入之前，一些硬盘控制器（当时采用独立的硬盘控制卡，IDE标准尚未出现）能够将大硬

盘显示为两个独立的硬盘。

最终的FAT16

1987年11月，我们今天称之为*FAT*的格式最终到来，它在康柏DOS 3.31中去掉了磁盘扇区的16位计数器。这个结果曾经一度被称为*DOS 3.31大文件系统*。尽管看起来磁盘上的变动很小，这个DOS的磁盘代码都必须检查并转换到32位的扇区数，由于它完全是16位的汇编语言这样一个现实，这项工作就变得非常复杂。

1988年，这项改进通过MS-DOS 4.0得到广泛应用。现在分区大小受限于每个簇的8位有符号扇区计数，它最大能达到2的64次方，对于一个常用的有32KB个簇每扇区512字节的硬盘来说，将FAT16分区大小的“明显”限制扩充到2GB。在磁光盘媒体上，它能使用1或者2KB的扇区，这样大小限制也就成比例地增大。

后来，Windows NT通过将每个簇的扇区数当作无符号数将最大的簇大小增加到64KB。然而这个格式与当时其它任何格式的FAT都不兼容，并且这样的操作会产生大量的内部碎片。Windows 98也支持这种格式的读写操作，但是它的磁盘管理工具不支持这种格式。

长文件名（VFAT, LFN）

Windows 95设计人员的一个用户体验目标就是：除了传统的8.3文件名以外，在新操作系统中使用长文件名（LFN）。长文件名通过在目录条目排列时，使用一个工作区来实现（参见下面）。按照Windows 95VxD设备驱动程序的命名规则，这个新扩充的文件系统通常称为VFAT。

有意思的是，VFAT驱动在早于Windows 95的Windows for Groups 3.11中就已经出现，但它仅仅用于实现32位文件访问，一个绕过DOS的视窗自带高性能保护模式文件管理系统，它能够直接使用BIOS或者更好的32位磁盘访问，如Windows自带的保护模式磁盘驱动程序。它是一个后门；微软为Windows for Groups 3.11所作的广告说32位文件访问基于“芝加哥项目的32位文件系统”。

在Windows NT中，FAT文件系统对于长文件名的支持从3.5版就已经开始了。在MS-DOS 7.0以后的版本中，则可使用类似DOSLFN这样的软件使得DIR等命令显示出长文件名。

FAT32

为了解决FAT16对于卷大小的限制同时让DOS的实模式在非必要情况下不减少可用常规内存状况下处理这种格式，微软公司决定实施新一代的FAT，它被称为FAT32，带有32位的簇数，当前用了其中的28位。

理论上，这将支持总数达268,435,438（$<2^{28}$）的簇，允许磁盘容量达到8TB。然而，由于微软公司scandisk工具的限制，FAT32不能大于4,177,920（$<2^{22}$）个簇，这将卷的容量限制在了124.55GB，除非不再使用“scandisk”。[4]

FAT32随着Windows 95 OSR2发布，尽管需要重新格式化才能使用这种格式并且DriveSpace 3（Windows 95 OSR2和Windows 98所带版本）从来都不支持这种格式。Windows 98提供了一个工具用来在不丢失数据的情况下将现有的硬盘从FAT16转到FAT32格式。在NT产品线上对于它的支持从Windows 2000开始。

Windows 2000和Windows XP能够读写任何大小的FAT32文件系统，但是这些平台上的格式化程序只能创建最大32GB的FAT32文件系统。Thompson and Thompson（2003）写道“奇怪的是微软公司说这种现象是故意设计的”[5] 微软公司知识库文章184006[4]的确是这么说的，但是没有提出任何关于这个限制的合理解释。Peter Norton的观点是“微软公司在有意地削弱FAT32文件系统”[6]。

exFAT

在Windows Embedded CE 6.0中引入，Windows XP SP3 以及 Windows Vista SP1也引入了exFAT的支持。在很多方面exFAT有了相当大的改进，特别适合用于闪存。

第三方支持

其它IBM PC的可选操作系统，如Linux、FreeBSD和BeOS都支持FAT格式，并且大部分都在相应的Windows版本发布以后很快就支持VFAT和FAT32格式。早期的Linux发布版本还包括称为UMSDOS的格式，它是保存在一个独立的称为--linux-.---的带有Unix文件属性（如长文件名和访问许可）的FAT。UMSDOS在VFAT发布以后就不再使用。Linux内核从2.5.7开始就禁止了这项功能。Mac OS X操作系统在除启动盘之外的其它卷上也支持FAT文件系统。

FAT和其它数据流

FAT文件系统本身不是为支持ADS而设计的，但是一些高度依赖它们的操作系统创造出了不同的方法以在FAT驱动器上处理它们。这些方法或者在额外的文件或路径中存储附加的信息（Mac OS），或者给那些磁盘数据结构中以前没有使用的变量赋予新的含义（OS/2和Windows NT）。第二种设计，尽管想像起来会更有效率，但是它们不能被不认识这种格式的工具复制或者备份；使用不能识别这种格式的磁盘工具（如碎片整理或CHKDSK）操控这些磁盘时可能会破坏这些信息。

Mac OS使用PC Exchange存储不同的数据，文件属性和文件名存在一个名为FINDER.DAT的隐藏文件中，资源分支（ADS）存在名为RESSOURCE.FRK的子目录中，这些数据都存在使用它们的每个目录中。从PC Exchange 2.1开始，它们将Mac OS的长文件名保存为标准的FAT长文件名，并且将超过31个字符的FAT长文件名转换为唯一的31字符能够被Macintosh应用程序识别的文件名。

Mac OS X将元数据（资源分支、不同的ADS、文件属性）保存在与所有人相同并以“._”开始的名字的隐藏文件中，并且Finder将一些文件夹和文件元数据存在名为“.DS Store”的隐藏文件中。

OS/2高度依赖于扩展属性（EA）并且将它们存在位于FAT12或FAT16的根目录下名为“EA DATA. SF”的隐藏文件中。这个文件使用以前文件（或者目录）的目录清单中的两个保留字节索引。在FAT32格式中，这些字节中存有文件或者目录开始簇号的高 16位，这样就使它难于在FAT32上保存EA。扩展属性可以通过Workplace Shell桌面、REXX脚本、许多系统图形用户接口和命令行工具（如4OS2）来访问。

Windows NT支持HPFS、NTFS和FAT中所有扩展属性的处理（所用处理机制完全类似于OS/2），但是不能处理其它一些存于NTFS驱动器的ADS数据。试图从复制带有与NTFS驱动器属性不同扩展属性的ADS到FAT驱动器将报告一个警告信息提示ADS将会丢失。

Windows 2000以后产品的处理类似于Windows NT但复制到FAT32时它们没有显示任何警告信息直接丢弃扩展属性（但报告其他像“Macintosh Finder Info”和“Macintosh Resource Fork”这些ADS引起的警告）。

前景

微软公司最近获得了VFAT和FAT32的专利（但没有得到最初的FAT的专利），这引起了人们对于微软将会对Linux OS发布和初始化他们产品的媒体厂商收取专利费的担忧（参见下面的FAT授权协议）。尽管最初的裁定不利于微软公司，但是微软仍然获取了胜利并且得到了专利授权。

由于微软公司已经宣布不再开发基于MS-DOS作业系统Windows Me的后续版本，所以不再有可能会有新版的FAT。对于大多数用途来说，为Windows NT系列开发的NTFS文件系统从效率、性能、安全性及可靠性来说都

优于FAT；它的主要缺点是小容量文件所占的额外空间以及除了基于NT的Windows操作系统之外的很少有其他操作系统支持。由于确切的规范是微软公司的商业秘密，这就使得使用一个DOS软盘用于恢复目的很困难（根据微软MCSE训练教材说明此点是刻意保密，以确保NTFS文件系统不易被盗取数据）。微软公司提供了一个恢复界面来解决这个问题，由于安全的原因它严重限制了缺省情况下它所能解决的问题。

FAT仍然是移动媒体所常用的一种文件系统（CD和DVD是例外），软碟使用的是FAT12，其他多数活动媒体用的是FAT32（如用于数字相机的快闪存储卡和USBU盘，Windows格式化的默认选项仍为FAT32），除非其容量超出FAT32的限制。出于兼容性和存储空间利用率的考虑FAT仍然用在这些驱动器上，同时也是由于这些活动媒体上的文件的许可更容易遇到麻烦而不是更重要这样一个事实。

Windows 2000和XP支持的FAT32格式化的限制是32GB，这导致使用现代硬盘的用户必须要么使用NTFS要么使用其它程序格式化驱动器。一个解决的办法是使用从Linux移植到Windows平台的一个工具mkdosfs。

设计

主磁盘结构

一个FAT分区或磁盘的结构布局							
内容	主引导区	文件系统信息扇区 (仅FAT32)	额外的保留空间(可选)	文件分配表#1	文件分配表#2 ...(可配置)	根目录(仅FAT12/FAT16)	其他所有数据...
大小(字节)	保留扇区数*扇区大小			文件分配表扇区数* 扇区大小	文件分配表扇区数* 扇区大小	根目录条目数量*32(按扇区大小向上取整)	剩下的磁盘空间

一个FAT文件系统包括四个不同的部分。

1. **保留扇区**，位于最开始的位置。 第一个保留扇区是引导扇区（*分区启动记录*）。它包括一个称为*基本输入输出参数块*的区域（包括一些基本的文件系统信息尤其是它的类型和其它指向其它扇区的指针），通常包括操作系统的启动调用代码。保留扇区的总数记录在引导扇区中的一个参数中。引导扇区中的重要信息可以被DOS和OS/2中称为*驱动器参数块*的操作系统结构访问。
2. **FAT区域**。它包含有两份*文件分配表*，这是出于系统冗余考虑，尽管它很少使用，即使是磁盘修复工具也很少使用它。它是分区信息的映射表，指示簇是如何存储的。
3. **根目录区域**。它是在根目录中存储文件和目录信息的*目录表*。在FAT32下它可以存在分区中的任何位置，但是在早期的版本中它永远紧随*FAT区域*之后。
4. **数据区域**。这是实际的文件和目录数据存储的区域，它占据了分区的绝大部分。通过简单地在FAT中添加文件链接的个数可以任意增加文件大小和子目录个数（只要有空簇存在）。然而需要注意的是每个簇只能被一个文件占有，这样的话如果在32KB大小的簇中有一个1KB大小的文件，那么 31KB的空间就浪费掉了。

启动扇区

格式如下

偏移（字节）	长度（字节）	说明
0x00	3	跳转指令（跳过开头一段区域）
0x03	8	OEM名称（空格补齐）。MS-DOS检查这个区域以确定使用启动记录中的哪一部分数据 [1] (https://groups.google.com/group/alt.msdos.programmer/msg/6b10a1ea602e61e)。常见值是IBM 3.3（在“IBM”和“3.3”之间有两个空格）和MSDOS5.0.
0x0b	2	每个扇区的字节数。*基本输入输出系统参数块*从这里开始。
0x0d	1	每簇扇区数
0x0e	2	保留扇区数（包括启动扇区）
0x10	1	文件分配表数目
0x11	2	最大根目录条目个数
0x13	2	总扇区数（如果是0，就使用偏移0x20处的4字节值）
0x15	1	介质描述 0xF8 单面、每面80磁道、每磁道9扇区 0xF9 双面、每面80磁道、每磁道9扇区 0xFA 单面、每面80磁道、每磁道8扇区 0xFB 双面、每面80磁道、每磁道8扇区 0xFC 单面、每面40磁道、每磁道9扇区 0xFD 双面、每面40磁道、每磁道9扇区 0xFE 单面、每面40磁道、每磁道8扇区 0xFF 双面、每面40磁道、每磁道8扇区 同样的介质描述必须在重复复制到每份FAT的第一个字节。有些操作系统（MSX-DOS 1.0版）全部忽略启动扇区参数，而仅仅使用FAT的第一个字节的介质描述确定文件系统参数。
0x16	2	每个文件分配表的扇区（FAT16）
0x18	2	每磁道的扇区
0x1a	2	磁头数
0x1c	4	隐藏扇区
0x20	4	总扇区数（如果超过65535，参见偏移0x13）
0x24	4	每个文件分配表的扇区（FAT32）。*扩展基本输入输出系统参数块*从这里开始。
0x24	1	物理驱动器个数（FAT16）
0x25	1	当前磁头（FAT16）
0x26	1	签名（FAT16）

0x27	4	ID（FAT16）
0x28	2	Flags（FAT32）
0x2a	2	版本号（FAT32）
0x2c	4	根目录启始簇（FAT32）
0x2b	11	卷标（非FAT32）
0x30	2	FSInfo扇区（FAT32）
0x32	2	引导扇区备份（FAT32）
0x34	12	保留未使用（FAT32）
0x36	8	FAT文件系统类型（如FAT、FAT12、FAT16）
0x3e	2	操作系统自引导代码
0x40	1	BIOS设备代号（FAT32）
0x41	1	未使用（FAT32）
0x42	1	标记（FAT32）
0x43	4	卷序号（FAT32）
0x47	11	卷标（FAT32）
0x52	8	FAT文件系统类型（FAT32）
0x1FE	2	扇区结束符（0x55 0xAA）

这里描述的启动扇区能在如OS/2 1.3的启动盘上看到。早期的版本使用一个较短的基本输入输出系统参数块，它们的启动代码在前面开始（如OS/2 1.1中是偏移0x2b）。

例外情况

Apricot PC的MS-DOS所用FAT的实现有一个不同的启动扇区组织以使用计算机与IBM不兼容的基本输入输出系统。跳转指令和OEM名被省略并且MS-DOS文件系统参数字于0x50（在标准扇区中偏移为0x0B - 0x17）。后来的Apricot MS-DOS版本除了Apricot特有的引导区之外也具有了读写标准启动分区的能力。

BBC Master 512上的DOS Plus根本就不使用传统的引导区。数据磁盘省略了引导区并且以一个单份的FAT开始（FAT的第一个字节用来确定磁盘容量），启动磁盘使用一个包含启动调用程序的小型ADFS文件系统，后面跟随一个单份的FAT。

文件分配表

一个分区分成同等大小的簇，也就是连续空间的小块。簇的大小随着FAT文件系统的类型以及分区大小而不同，典型的簇大小介于2KB到32KB之间。每个文件根据它的大小可能占有一个或者多个簇；这样，一个文件就由这些这些（称为单向链表）簇链所表示。然而，这些链并不一定一个接着一个在磁盘上存储，它们经常是在整个数据区域*零散*的储存。

文件分配表（FAT）是映射到分区每个簇的条目列表。每个条目记录下面五种信息中的一种。

- 链中下一个簇的地址
- 一个特殊的**簇链结束符（EOC**，End Of Cluster-chain，或称End Of Chain）符号指示链的结束
- 一个特殊的符号标示坏簇
- 一个特殊的符号标示保留簇
- 0来表示空闲簇

每个版本的FAT文件系统使用不同大小的FAT条目。这个大小已经由名字表示出来，例如FAT16文件系统的每个条目使用16位表示，32位文件系统使用32位表示。这个不同意味着FAT32系统的文件分配表能比FAT16映射更多的簇，它也允许FAT32有更大的分区大小。这也使得FAT32比 FAT16更能有效地利用磁盘空间，因为每个驱动器能够寻址更小的簇，这也就意味着更少的空间浪费。

FAT条目值：

FAT12	FAT16	FAT32	描述
0x000	0x0000	0x?0000000	空闲簇
0x001	0x0001	0x?0000001	保留簇
0x002 - 0xFEF	0x0002 - 0xFFEF	0x?0000002 - 0x?FFFFFEF	被占用的簇；指向下一个簇
0xFF0 - 0xFF6	0xFFF0 - 0xFFF6	0x?FFFFFF0 - 0x?FFFFFF6	保留值
0xFF7	0xFFF7	0x?FFFFFF7	坏簇
0xFF8 - 0xFFF	0xFFF8 - 0xFFFF	0x?FFFFFF8 - 0x?FFFFFFF	文件最后一个簇

注意FAT32只使用32位中的28位。高4位通常是0但它们是保留位，不要更改它们。在上面的表中它们用问号表示。

目录表

目录表是一个表示目录的特殊类型文件（现今通常称为文件夹）。它里面保存的每个文件或目录使用表中的32字节条目表示。每个条目记录名字、扩展名、属性（文件、目录、隐藏、只读、系统和卷）、创建的日期和时间、文件／目录数据第一个簇的地址，最后是文件／目录的大小。

除了FAT12和FAT16文件系统中的根目录表占据特殊的*根目录区域*位置之外，所有其它的目录表都存在数据区域。

合法的DOS文件名包括下面一些字符：

- 大写字母A-Z, 数字0-9
- 空格（尽管结尾的空格被作为填充而不是文件名的一部分）
- ! # $ % & ( ) - @ ^ _ ` { } ~ '
- 数值128-255

DOS文件名位于OEM字符集。

位于根目录区域和子目录区域的目录条目都是下面的格式：

<table>
<tr><th>字节偏移</th><th>长度</th><th>描述</th></tr>
<tr><td>0x00</td><td>8</td><td>DOS文件名（附加空格）
第一个字节可以是下面的特殊数值：
<table>
<tr><td>0x00</td><td>这个条目有用并且后面没有被占用条目</td></tr>
<tr><td>0x05</td><td>最初字符确实是0xE5</td></tr>
<tr><td>0x2E</td><td>'点'条目；'.'或者'..'</td></tr>
<tr><td>0xE5</td><td>这个条目曾经被删除不再有用。取消删除文件工具作为取消删除的一步必须使用一个正常的字符取代它。</td></tr>
</table></td></tr>
<tr><td>0x08</td><td>3</td><td>DOS文件扩展名（空格补齐）</td></tr>
<tr><td>0x0b</td><td>1</td><td>文件属性
第一个字节可以是下面一些特殊值：
<table>
<tr><th>位</th><th>掩码</th><th>描述</th></tr>
<tr><td>0</td><td>0x01</td><td>只读</td></tr>
<tr><td>1</td><td>0x02</td><td>隐藏</td></tr>
<tr><td>2</td><td>0x04</td><td>系统</td></tr>
<tr><td>3</td><td>0x08</td><td>卷标</td></tr>
<tr><td>4</td><td>0x10</td><td>子目录</td></tr>
<tr><td>5</td><td>0x20</td><td>文件</td></tr>
<tr><td>6</td><td>0x40</td><td>设备（内部使用，磁盘上看不到）</td></tr>
<tr><td>7</td><td>0x80</td><td>没有使用</td></tr>
</table>
属性值0x0F用来表示长文件名条目。</td></tr>
<tr><td>0x0c</td><td>1</td><td>保留，NT使用（参见后面）</td></tr>
<tr><td>0x0d</td><td>1</td><td>创建时间，最小时间分辨率：10ms单位，数值从0到199。</td></tr>
<tr><td>0x0e</td><td>2</td><td>创建时间。小时、分钟和秒根据后面的图示描述进行编码：
<table>
<tr><th>位</th><th>描述</th></tr>
<tr><td>15-11</td><td>小时（0-23）</td></tr>
<tr><td>10-5</td><td>分钟（0-59）</td></tr>
<tr><td>4-0</td><td>秒／2（0-29）</td></tr>
</table>
注意秒只保存了2秒的分辨率。更细分辨率的文件创建时间在偏移0x0d处。</td></tr>
</table>

长文件名（LFN）使用一个技巧存储在FAT文件系统上——在目录表中添加假的条目。这些条目使用一个普通文件无法使用的卷标属性标识，普通文件无法使用是由于它们被大多数旧的MS-DOS程序忽略。很显然，一个只包含卷标的目录被当作空卷，这样就允许删除；使用长文件名创建的文件在从普通的DOS 删除就会发生这样的情形。

校验和也允许检验长文件名是否与8.3文件名匹配；当一个文件删除之后使用DOS在同一个目录位置重新创建之后就会出现不匹配现象。校验和使用下面的算法计算。（注意pFcbName是指向如正常目录条目中所显示的文件名的指针，例如前八个字符是文件名，最后三个是扩展名。点是隐含的。文件名中没有使用的空间将使用空格（ASCII 0x20）补齐。例如，“Readme.txt”将记录为"README TXT"。

```
unsigned char lfn_checksum (const unsigned char *pFcbName)
{
    int i;
    unsigned char sum=0;

    for (i=11; i; i--)
        sum = ((sum & 1) ? 0x80 : 0) + (sum >> 1) + *pFcbName++;
    return sum;
}
```

旧版的PC-DOS错误地将根目录中的长文件名当作卷标，这样它们就会显示错误的卷标。

每个假条目包含13UTF-16个字符（26字节），通过使用包含文件大小或者时间记录的区域获得除了旧的8+3之外的另外15个字节（但是出于安全和磁盘检查工具的考虑开始簇的区域没有使用保留值为0）。参见8.3中另外的解释。长文件名条目使用下面的格式：

字节偏移	长度	描述
0x00	1	序列号
0x01	10	名称字符（5个UTF-16字符）
0x0b	1	属性（永远是0x0F）
0x0c	1	保留（永远是0x00）
0x0d	1	DOS文件名校验和
0x0e	12	名称字符（6个UTF-16字符）
0x1a	2	第一个簇（永远是0x0000）
0x1c	4	名称字符（两个UTF-16字符）

如果一个文件名只包含小写字母、或者是一个小写字母的*名*加上大写扩展名的混合或者与此相反，没有特殊的字符并且满足8.3的限制，在Windows NT上就不创建VFAT的条目。相反，在目录条目的偏移0x0c处的没有说明的位用来指示文件名全部或者部分是小写字母。特别明确的是，位4意味着小写字母的扩展名，位3意味着*名*是小写字母，这样就允许如“example.TXT”和“HELLO.txt”这样的组合，但是不允许“Mixed.txt”这样的组合。很少有操作系统支持这种功能。非NT的Windows版本当这个扩展使用时将把文件名当作大写字母。缺省情况下，Linux的最近版本将认识这个扩展但是在写时并不使用它。

第三方扩展

在微软公司添加长文件名和创建／访问时间戳之前，其它的操作系统使用目录表字节0x0C-0x15存储其它的元数据。它们包括：

<table>
<tr><th>字节偏移</th><th>长度</th><th>系统</th><th>描述</th></tr>
<tr><td>0x0C</td><td>2</td><td>RISC OS</td><td>文件类型，0x000 - 0xFFF</td></tr>
<tr><td>0x0C</td><td>1</td><td>DOS Plus</td><td>用户定义文件属性F1-F4<table><tr><th>位</th><th>掩码</th><th>描述</th></tr><tr><td>7</td><td>0x80</td><td>F1</td></tr><tr><td>6</td><td>0x40</td><td>F2</td></tr><tr><td>5</td><td>0x20</td><td>F3</td></tr><tr><td>4</td><td>0x10</td><td>F4</td></tr></table></td></tr>
<tr><td>0x0D</td><td>1</td><td>DR-DOS</td><td>被删除文件名最初第一个字符</td></tr>
<tr><td>0x0E</td><td>2</td><td>DR-DOS和FlexOS</td><td>加密文件密码</td></tr>
<tr><td>0x10</td><td>4</td><td>DR-DOS 7</td><td>被删除文件最初的文件时间和日期；被删除文件有设置到删除时间的正常时间和日期</td></tr>
<tr><td>0x12</td><td>2</td><td>DR-DOS 6和FlexOS</td><td>文件所有者身份</td></tr>
<tr><td>0x14</td><td>2</td><td>DR-DOS和FlexOS</td><td>文件许可位（只有FlexOS使用执行许可）：<table><tr><th>位</th><th>掩码</th><th>描述</th></tr><tr><td>0</td><td>0x0001</td><td>所有者删除需要密码</td></tr><tr><td>1</td><td>0x0002</td><td>所有者执行需要密码</td></tr><tr><td>2</td><td>0x0004</td><td>所有者写需要密码</td></tr><tr><td>3</td><td>0x0008</td><td>所有者读需要密码</td></tr><tr><td>4</td><td>0x0010</td><td>组删除需要密码</td></tr><tr><td>5</td><td>0x0020</td><td>组执行需要密码</td></tr><tr><td>6</td><td>0x0040</td><td>组写需要密码</td></tr><tr><td>7</td><td>0x0080</td><td>组读需要密码</td></tr><tr><td>8</td><td>0x0100</td><td>全部用户删除需要密码</td></tr><tr><td>9</td><td>0x0200</td><td>全部用户执行需要密码</td></tr><tr><td>10</td><td>0x0400</td><td>全部用户写需要密码</td></tr><tr><td>11</td><td>0x0800</td><td>全部用户读需要密码</td></tr></table></td></tr>
</table>

FAT授权

在二十世纪九十年代中期微软公司申请并获得了一系列的FAT文件系统核心部分的专利。由于具有广泛的兼容性和易于理解，FAT经常被选为用于数字相机和个人数码助理中闪存的数据交换格式。

2003年12月3日微软公司宣布使用FAT规范和“相关智能产权”将需要得到 授权 (http://www.microsoft.com/about/legal/intellectualproperty/search/details.mspx?ip_id=IDAEYWHD&techType=Any&ipCat=Patents&feeStructure=Any&keywords=&ipVenture=false)，每个销售单元支付0.25美元的著作权费，每个授权协议最多250,000美元的著作权费。

为了这个目的，微软公司提及了四个关于FAT文件系统的专利作为它的知识产权主张的基础。所有这些与长文件名扩展有关的FAT首先出现在Windows 95中：

- 美国专利 5,745,902 (http://www.google.com/patents?vid=5745902) - 访问使用不同文件名格式的文件名的方法和系统。1992年7月6日备案。这包括生成、联系一个8.3兼容格式的短文件名和长文件名的方法（如“Microsoft.txt”和“MICROS~1.TXT”），以及列举相互冲突的短文件名的方法（如“MICROS~2.TXT”和“MICROS~3.TXT”）。现在还不清楚这个专利是否覆盖不具有显式长文件名能力的FAT实现。Unix文件系统中的硬链接看起来不是先行者：从长文件名删除一个FAT文件也将删除它的短文件名。将一个文件重命名为一个“短”文件名也将一致地更改长文件名；同样，将一个文件重命名为“长”文件名也将重新生成一个“短”文件名。在NTFS中，硬链接和两个名字是不同的概念，并且每个硬链接都有两个名字。最后，在API的层面上，当在系统中进行目录搜索时两个文件名都会出现；它们看起来不是两个独立的文件并且它们也没有有必要去“映射”确定同一个文件。
- 美国专利 5,579,517 (http://www.google.com/patents?vid=5579517) - 长、短文件名公用的名字空间。1995年4月24日备案。这包括将多个连续8.3目录条目链接在一起支持长文件名的方法，其中一些条目特殊进行标记阻止可能引起混淆地早期的不支持长文件名的FAT实现。
  - 公共专利基金会成功地对这项专利发起了挑战；这个专利申请由于所申请的技术在专利美国专利 5,307,494 (http://www.google.com/patents?vid=5307494)和美国专利 5,367,671 (http://www.google.com/patents?vid=5367671)中的 先期发现 (https://archive.is/20130102084746/http://news.com.com/Microsoft+FAT+patent+falls+flat/2100-1014_3-5390138.html) 在2004年9月14日被 驳回 (http://www.pubpat.org/Microsoft_517_Rejected.htm)。这个决定后来在2006年1月10日被美国专利局所推翻。
- 美国专利 5,758,352 (http://www.google.com/patents?vid=5758352) - 长、短文件名公用的名字空间。1996年9月5日备案。它非常类似于5,579,517。
  - 公共专利基金会成功地对这项专利发起了挑战；美国专利商标局基于 "the six assignees names were incorrect" (http://www.theregister.co.uk/2005/10/05/microsoft_patent/) [2] (http://www.out-law.com/default.aspx?page=6202) 在2005年10月5日驳回了这项专利。这个决定也在后来的2006年1月10日被美国专利局推翻。
- 美国专利 6,286,013 (http://www.google.com/patents?vid=6286013) - 在操作系统中为长、短文件名提供一个公用的名字空间的方法和系统。1997年1月28日备案。它所申请的内容包括Windows 95、Windows 98和Windows Me的长文件名提供给它们MS-DOS兼容层所用的方法。它看起来不影响非微软的FAT实现。

许多技术评论断言这些专利仅仅涵盖了支持长文件名的FAT实现，那些只使用短名字的移动固态媒体和消费设备将不受影响。

另外，在微软2000年12月6日出版的"Microsoft Extensible Firmware Initiative FAT 32 File System Specification，FAT: General Overview of On-Disk Format"，微软公司明确地给出了一些授权，许多读者将它认为是微软允许操作系统厂商实现FAT。

控诉

由于人们广泛要求重新审查这些专利，公众专利基金会向美国专利和商标局（USPTO）提出了一些证据争辩这些专利的有效性，其中包括施乐公司和IBM的早期参考资料。美国专利商标局承认这些证据提出了“可专利性的实质性的新问题”并且对于微软公司FAT专利的有效性展开调查。

2004年9月30日，美国专利商标局主要基于公共专利基金会所提供的证据驳回了美国专利 5,579,517（http://www.google.com/patents?vid=5579517)的专利主张。这个基金会的执行总裁Dan Ravicher说“现在专利局只不过是确认了我们已经知道了一段时间的事情，微软公司的专利是假的。”

PUBPAT的新闻发布会说，“微软公司仍然有机会回应专利局的驳回。有代表性的是第三方的重新审查要求如PUBPAT提供的资料成功地减小了专利的范围或者有70%的机会完全驳回专利。”

2005年10月5日，专利局宣布随着调查的深入它驳回了专利5.579,517的专利主张，另外它发现专利美国专利 5,758,352（http://www.google.com/patents?vid=5758352)有错误的专利受益人而无效。

最后在2006年1月10日，专利局裁定微软公司的FAT系统的实现特点是“新颖和非显然的”，推翻了早期的两个非最终裁决。

注解

1. Archive（A）比特：文件创建或更改后会被设成1，经操作系统内的backup做文件备份后会被设成0。如只需备份上次备份后有修改的文件，可借此比特识别。
2. 专利申请在于文件系统中支持长文件名的技术，而不是文件系统核心本身。
3. Duncan, Ray（1989）.Design goals and implementation of the new High Performance File System (http://cd.textfiles.com/megademo2/INFO/OS2_HPFS.TXT). *Microsoft Systems Journal* **4**（5）.
4. FAT32 文件系统的限制 (http://support.microsoft.com/kb/184006)，微软支持知识库文章
5. Thompson, Robert Bruce and Barbara Fritchman Thompson, *PC Hardware in a Nutshell, 3rd Edition,*, O'Reilly, ISBN 0-596-00513-X（p. 506 re Microsoft "bizarrely" saying 32 GB limitation is by design
6. Norton, Peter（2002页面文件）

参见

- NTFS
- 文件系统的对比
- 驱动器字母分配
- 软件专利
- 文件系统列表
- Rock Ridge和Joliet：象FAT上的VFAT一样为CD添加长文件名的系统。

外部链接

- News article about final patent ruling (https://archive.is/20130425112950/http://news.com.com/Microsofts+file+system+patent+upheld/2100-1012_3-6025447.html?part=rss&tag=6025447&subj=news)
- Microsoft's statement on "FAT File System Technology and Patent License" (https://web.archive.or

g/web/20060213224425/http://www.microsoft.com/mscorp/ip/tech/fat.asp)
- FAT File System (https://web.archive.org/web/20121114212231/http://www.zeeis.com/fat-file-system/)
- Slashdot discussion on Microsoft's claims of FAT-related patents (http://slashdot.org/article.pl?sid=03/12/04/1318212)
- Microsoft Extensible Firmware Initiative FAT 32 File System Specification, FAT: General Overview of On-Disk Format (https://web.archive.org/web/20071013033802/http://www.microsoft.com/whdc/system/platform/firmware/fatgen.mspx)
- Understanding FAT32 Filesystems (explained for embedded firmware developers) (http://www.pjrc.com/tech/8051/ide/fat32.html)
- Microsoft's war on GPL dealt patent setback (http://www.theregister.co.uk/2004/06/14/ms_fat_patent_reexamined/)
- A Short History of MS-DOS (https://web.archive.org/web/20130801102940/http://patersontech.com/Dos/Byte/History.html)，by Tim Paterson
- Detailed Explanation of FAT Boot Sector (http://support.microsoft.com/support/kb/articles/Q140/4/18.asp) - Microsoft Knowledge Base Article 140418
- At PUBPAT's Request, Patent Office Rejects Microsoft's FAT Patent: All Claims of Reynolds '517 Patent Ruled Invalid (http://www.pubpat.org/Microsoft_517_Rejected.htm)
- Volume and file size limits of FAT filesystems (https://web.archive.org/web/20050907181002/http://www.microsoft.com/resources/documentation/Windows/XP/all/reskit/en-us/prkc_fil_tdrn.asp)
