



北京北亚数据恢复中心
http://www.raid-recovery.org



TEL: 4006-505-808

网站首页

实验室简介

客户服务

技术培训

服务报价

联系我们

您现在的位置: 数据恢复_北京数据恢复|北亚数据恢复中心 4006-505-808 >> 数据恢复文章 >> 数据恢复文
栏 >> 文章正文

报修电话: 4006-505-808

--站内文章检索----

数据恢复

OK

提示: 可按上面表单检索本站
所有文章及下载资源,
也可通过下面菜单进入文
章、
下载及其他类别!

最新动态

文章中心

下载中心

网络日志

咨询留言

交流论坛

北亚数据恢复中心服务
承诺:

FAT文件系统原理(四)

FAT文件系统原理(四)

更新时间:2004-4-20【字体: 小 大】

★FAT16存储原理:

当把一部分磁盘空间格式化为fat文件系统时, fat文件系统就将这个分区当成整块可分配的区域进行规划, 以便于数据的存储。一般来讲, 其划分形式如图7所示。我们把FAT16部分提取出来, 详细描述一下:

FAT16是Microsoft较早推出的文件系统, 具有高度兼容性, 目前仍然广泛应用于个人电脑尤其是移动存储设备中, FAT16简单来讲由图4. 3. 11所示的6部分组成(主要是前5部分)。引导扇区(DBR)我们已经说过, FAT16在DBR之后没有留有任何保留扇区, 其后紧随的便是FAT表。FAT表是FAT16用来记录磁盘数据区簇链结构的。像前面我们说过的例子一样, FAT将磁盘空间按一定数目的扇区为单位进行划分, 这样的单位称为簇。通常情况下, 每扇区512字节的原则是不变的。簇的大小一般是 2^n (n为整数)个扇区的大小, 像512B, 1K, 2K, 4K, 8K, 16K, 32K, 64K。实际中通常不超过32K。之所以簇为单位而不以扇区为单位进行磁盘的分配, 是因为当分区容量较大时, 采用大小为512b的扇区管理会增加fat表的项数, 对大文件存取增加消耗, 文件系统效率不高。分区的大小和簇的取值是有关系的, 见表9 (单击此处查看PDF版全文)

图4. 3. 11 Fat16的组织形式

引导扇区	FAT1	FAT2(重复的)	根文件夹	其他文件夹及所有文件	剩余扇区
1扇区	实际情况取大小	同FAT1	32个扇区	开始簇编号(从2开始)	不足一簇

- 免费检测
- 免费提供3天备份
- 专业数据恢复工程师提供服务
- 数据恢复前报价，客户确认后工程师开始数据修复
- 数据恢复不成功不收费
- 与客户签订保密协议，对客户的数据严格保密
- 整个恢复过程不会对客户的原盘有任何的写操作，以确保原盘的数据完全。

表9 FAT16分区大小与对因簇大小		
分区空间大小	每个簇的扇区	簇空间大小
0MB-32MB	1	512个字节
33MB-64MB	2	1k
65MB-128MB	4	2k
129MB-225MB	8	4k
256MB-511MB	16	8k
512MB-1023MB	32	16k
1024MB-2047MB	64	32k
2048MB-4095MB	128	64k

注意：少于32680个扇区的分区中，簇空间大小可最多达到每个簇8个扇区。不管用户是使用磁盘管理器来格式化分区，还是使用命令提示行键入format命令格式化，格式化程序都创建一个12位的FAT。少于16MB的分区，系统通常会将其格式化成12位的FAT，FAT12是FAT的初始实现形式，是针对小型介质的。FAT12文件分配表要比FAT16和FAT32的文件分配表小，因为它对每个条目使用的空间较少。这就给数据留下较多的空间。所有用FAT12格式化的5.25英寸软盘以及1.44MB的3.5英寸软盘都是由FAT12格式化的。除了FAT表中记录每簇链结的二进制位数与FAT16不同外，其余原理与FAT16均相同，不再单独解释。。。

格式化FAT16分区时，格式化程序根据分区的大小确定簇的大小，然后根据保留扇区的数目、根目录的扇区数目、数据区可分的簇数与FAT表本身所占空间 来确定FAT表所需的扇区数目，然后将计算后的结果写入DBR的相关位置。

FAT16 DBR参数的偏移0x11处记录了根目录所占扇区的数目。偏移0x16记录了FAT表所占扇区的数据。偏移0x10记录了FAT表的副本数目。系统在得到这几项参数以后，就可以确定数据区的开始扇区偏移了。

FAT16文件系统从根目录所占的32个扇区之后的第一个扇区开始以簇为单位进行数据的处理，这之前仍以扇区为单位。对于根目录之后的第一个簇，系统并不编号为第0簇或第1簇（可能是留作关键字的原因吧），而是编号为第2簇，也就是说数据区顺序上的第1个簇也是编号上的第2簇。

FAT文件系统之所以有12，16，32不同的版本之分，其根本在于FAT表用来记录任意一簇链接的二进制位数。以FAT16为例，每一簇在FAT表中占据2字节（二进制16位）。所以，FAT16最大可以表示的簇号为0xFFFF（十进制的65535），以32K为簇的大小的话，FAT32可以管理的最大磁盘空间为：32KB×65535=2048MB, 这就是为什么FAT16不支持超过2GB分区的原因。

FAT表实际上是一个数据表，以2个字节为单位，我们暂将这个单位称为FAT记录项，通常情况其第1、2个记录项（前4个字节）用作介质描述。从第三个记录项开始记录除根目录外的其他文件及文件夹的簇链情况。根据簇的表现情况FAT用相应的取值来描述，见表10

表10 FAT16记录项的取值含义（16进制）	
FAT16记录项的取值	对应簇的表现情况
0000	未分配的簇
0002~FFEF	已分配的簇
FFF0~FFF6	系统保留

FFF7	坏簇
FFF8~FFFF	文件结束簇

看一幅在winhex所截FAT16的文件分配表，图10：

驱动器 L:

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	访问		
00000020	00	F8	FF	FF	FF	FF	2F	05	00	06	40	FF	5F	07	00	08	70	?
00000021	00	09	80	0A	90	0B	10	0C	11	0D	12	0E	13	0F	14	15	16		
00000022	00	11	00	12	00	13	00	14	00	15	00	16	00	17	00	18	00		
00000023	00	19	00	1A	00	1B	00	1C	00	1D	00	1E	00	1F	00	20	00		
00000024	00	21	00	22	00	23	00	24	00	25	00	26	00	27	00	28	00	!	".#\$.%&.'(. .
00000025	00	29	00	2A	00	2B	00	2C	00	2D	00	2E	00	2F	00	30	00)	.*+,.-.../0..
00000026	00	31	00	32	00	33	00	34	00	35	00	36	00	37	00	38	00	1	2.3.4.5.6.7.8..
00000027	00	39	00	3A	00	3B	00	3C	00	3D	00	3E	00	3F	00	40	00	9	.:;.<.=.>.?@..
00000028	00	41	00	42	00	43	00	44	00	45	00	46	00	47	00	48	00	A	B.C.D.E.F.G.H.
00000029	00	49	00	4A	00	4B	00	4C	00	4D	00	4E	00	4F	00	50	00	I	J.K.L.M.N.O.P.
0000002A	00	51	00	52	00	53	00	54	00	55	00	56	00	57	00	58	00	Q	R.S.T.U.V.W.X.
0000002B	00	59	00	5A	00	5B	00	5C	00	5D	00	5E	00	5F	00	60	00	Y	Z.[\].^_`.
0000002C	00	61	00	62	00	63	00	64	00	65	00	66	00	67	00	68	00	a	b.c.d.e.f.g.h.
0000002D	00	69	00	6A	00	6B	00	6C	00	6D	00	6E	00	6F	00	70	00	i	j.k.l.m.n.o.p.
0000002E	00	71	00	72	00	73	00	74	00	75	00	76	00	77	00	78	00	q	r.s.t.u.v.w.x.
0000002F	00	79	00	7A	00	7B	00	7C	00	7D	00	7E	00	7F	00	80	00	y	z.{ .}.~.€.
00000030	00	81	00	82	00	83	00	84	00	85	00	86	00	87	00	88	00	?	~~~~~}.~.€.
00000031	00	89	00	8A	00	8B	00	8C	00	8D	00	8E	00	8F	00	90	00	?	~~~~~}.~.€.
00000032	00	91	00	92	00	93	00	94	00	95	00	96	00	97	00	98	00	?	~~~~~}.~.€.
00000033	00	FF	FF	9A	00	FF	FF	9C	00	9D	00	FF	FF	9F	00	A0	00	?	?? ??
00000034	00	FF	FF	FF	FF	FF	FF	A4	00	A5	00	A6	00	A7	00	A8	00		?????
00000035	00	A9	00	AA	00	AB	00	AC	00	AD	00	AE	00	AF	00	B0	00	?	?????
00000036	00	B1	00	B2	00	B3	00	B4	00	B5	00	B6	00	B7	00	B8	00	?	?????
00000037	00	B9	00	BA	00	BB	00	BC	00	BD	00	BE	00	BF	00	C0	00	?	?????
00000038	00	C1	00	C2	00	C3	00	C4	00	C5	00	C6	00	C7	00	C8	00	?	?????

如图，FAT表以“F8 FF FF FF” 开头，此2字节为介质描述单元，并不参与FAT表簇链关系。小红字标出的是FAT扇区每2字节对应的簇号。

相对偏移0x4~0x5偏移为第2簇(顺序上第1簇)，此处为FF, 表示存储在第2簇上的文件(目录)是个小文件，只占用1个簇便结束了。

第3簇中存放的数据是0x0005，这是一个文件或文件夹的首簇。其内容为第5簇，就是说接下来的簇位于第5簇——> FAT表指引我们到达FAT表的第5簇指向，上面写的的数据是“FF FF”, 意即此文件已至尾簇。

第4簇中存放的数据是0x0006，这又是一个文件或文件夹的首簇。其内容为第6簇，就是说接下来的簇位于第6簇——> FAT表指引我们到达FAT表的第6簇指向，上面写的的数据是0x0007，就是说接下来的簇位于第7簇——> FAT表指引我们到达FAT表的第7簇指向……直到根据FAT链读取到扇区相对偏移0x1A~0x1B，也就是第13簇，上面写的的数据是0x000E，也就是指向第14簇——> 14簇的内容为“FF FF”，意即此文件已至尾簇。

后面的FAT表数据与上面的道理相同。不再分析。

FAT表记录了磁盘数据文件的存储链表，对于数据的读取而言是极其重要的，以至于Microsoft为其开发的FAT文件系统中的FAT表创建了一份备份，就是我们看到的FAT2。FAT2与FAT1的内容通常是即时同步的，也就是说如果通过正常的系统读写对FAT1做了更改，那么FAT2也同样被更新。如果从这个角度来看，系统的这个功能在数据恢复时是个天灾。

FAT文件系统的目录结构其实是一颗有向的从根到叶的树，这里提到的有向是指对于FAT分区内的任一文件(包括文件夹)，均需从根目录寻址来找到。可以这样认为：目录存储结构的入口就是根目录。

FAT文件系统根据根目录来寻址其他文件(包括文件夹)，故而根目录的位置必须在磁盘存取数据之前得以确定。FAT文件系统就是根据分区的相关DBR参数与DBR中存放的已经计算好的FAT表(2份)的大小来确定的。格式化以后，跟目录的大小和位置其实都已经确定下来了：位置紧随FAT2之后，大小通常为32个扇区。根目录之后便是数据区第2簇。

FAT文件系统的的一个重要思想是把目录(文件夹)当作一个特殊的文件来处理，FAT32甚至将根目录当作文件处理(旁：NTFS将分区参数、安全权限等好多东西抽象为文件更是这个思想的升华)，在FAT16中，虽然根目录地位并不等同于普通的文件或者说是目录，但其组织形式和普通的目录(文件夹)并没有不同。FAT分区中所有的文件夹(目录)文件，实际上可以看作是一个存放其他文件(文件夹)入口参数的数据表。所以目录的占用空间的大小并不等同于其下所有数据的大小，但也不等同于0。通常是占很小的空间的，可以看作目录文件是一个简单的二维表文件。其具体存储原理是：

不管目录文件所占空间为多少簇，一簇为多少字节。系统都会以32个字节为单位进行目录文件所占簇的分配。这32个字节以确定的偏移来定义本目录下的一个文件(或文件夹)的属性，实际上是一个简单的二维表。

这32个字节的各字节偏移定义如表11：

表11 FAT16目录项32个字节的表示定义			
字节偏移(16进制)	字节数	定义	
0x0~0x7	8	文件名	
0x8~0xA	3	扩展名	
		属性	00000000(读写)
			00000001(只读)
			00000010(隐藏)
			00000100(系统)

0xB	1	字节	00001000 (卷标)
			00010000 (子目录)
			00100000 (归档)
0xC~0x15	10	系统保留	
0x16~0x17	2	文件的最近修改时间	
0x18~0x19	2	文件的最近修改日期	
0x1A~0x1B	2	表示文件的首簇号	
0x1C~0x1F	4	表示文件的长度	

对图10中的一些取值进行说明：

(1)、对于短文件名，系统将文件名分成两部分进行存储，即主文件名+扩展名。0x0~0x7字节记录文件的主文件名，0x8~0xA记录文件的扩展名，取文件名中的ASCII码值。不记录主文件名与扩展名之间的“.” 主文件名不足8个字符以空白符(20H)填充，扩展名不足3个字符同样以空白符(20H)填充。0x0偏移处的取值若为00H，表明目录项为空；若为E5H，表明目录项曾被使用，但对应的文件或文件夹已被删除。(这也是误删除后恢复的理论依据)。文件名中的第一个字符若为“.”或“..”表示这个簇记录的是一个子目录的目录项。“.”代表当前目录；“..”代表上级目录(和我们在dos或windows中的使用意思是一样的，如果磁盘数据被破坏，就可以通过这两个目录项的具体参数推算磁盘的数据区的起始位置，猜测簇的大小等等，故而是比较重要的)

(2)、0xB的属性字段：可以看作系统将0xB的一个字节分成8位，用其中的一位代表某种属性的有或无。这样，一个字节中的8位每位取不同的值就能反映各个属性的不同取值了。如00000101就表示这是个文件，属性是只读、系统。

(3)、0xC~0x15在原FAT16的定义中是保留未用的。在高版本的WINDOWS系统中有时也用它来记录修改时间和最近访问时间。那样其字段的意义和FAT32的定义是相同的，见后边FAT32。

(4)、0x16~0x17中的时间=小时*2048+分钟*32+秒/2。得出的结果换算成16进制填入即可。也就是：0x16字节的0~4位是以2秒为单位的量值；0x16字节的5~7位和0x17字节的0~2位是分钟；0x17字节的3~7位是小时。

(5)、0x18~0x19中的日期=(年份-1980)*512+月份*32+日。得出的结果换算成16进制填入即可。也就是：0x18字节0~4位是日期数；0x18字节5~7位和0x19字节0位是月份；0x19字节的1~7位为年号，原定义中0~119分别代表1980~2099，目前高版本的Windows允许取0~127，即年号最大可以到2107年。

(6)、0x1A~0x1B存放文件或目录的表示文件的首簇号，系统根据掌握的首簇号在FAT表中找到入口，然后再跟踪簇链直至簇尾，同时用0x1C~0x1F处字节判定有效性。就可以完全无误的读取文件(目录)了。

(7)、普通子目录的寻址过程也是通过其父目录中的目录项来指定的，与数据文件(指非目录文件)不同的是目录项偏移0xB的第4位置1，而数据文件为0。

对于整个FAT分区而言，簇的分配并不完全总是分配干净的。如一个数据区为99个扇区的FAT系统，如果簇的大小设定为2扇区，就会有1个扇区无法分配给任何一个簇。这就是分区的剩余扇区，位于分区的末尾。有的系统用最后一个剩余扇区备份本分区的DBR，这也是一种好的备份方法。

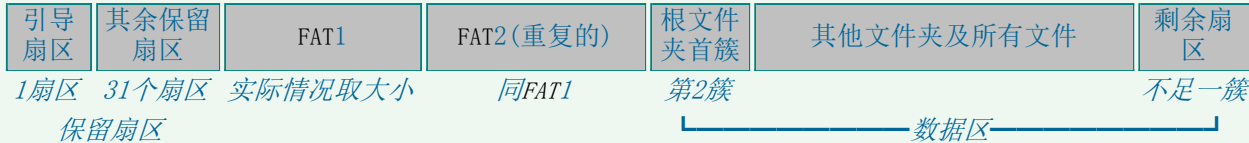
早的FAT16系统并没有长文件名一说，Windows操作系统已经完全支持在FAT16上的长文件名了。FAT16的长文件名与FAT32长文件名的定义是相同的，关于长文件名，在FAT32部分再详细作解释。

★FAT32存储原理：

FAT32是个非常有功劳的文件系统，Microsoft成功地设计并运用了它，直到今天NTFS铺天盖地袭来的时候，FAT32依然占据着Microsoft Windows文件系统中重要的地位。FAT32最早是出于FAT16不支持大分区、单位簇容量大以致空间急剧浪费等缺点设计的。实际应用中，FAT32还是成功的。

FAT32与FAT16的原理基本上是相同的，图4. 3. 12标出了FAT32分区的基本构成。

图4. 3. 12 Fat32的组织形式



FAT32在格式化的过程中就根据分区的特点构建好了它的DBR，其中BPB参数是很重要的，可以回过头来看一下表4和表5。首先FAT32保留扇区的数目默认为32个，而不是FAT16的仅仅一个。这样的好处是有助于磁盘DBR指令的长度扩展，而且可以为DBR扇区留有备份空间。上面我们已经提到，构建在FAT32上的win98或win2000、winXP，其操作系统引导代码并非只占一个扇区了。留有多余的保留扇区就可以很好的拓展OS引导代码。在BPB中也记录了DBR扇区的备份扇区编号。备份扇区可以让我们在磁盘遭到意外破坏时恢复DBR。

FAT32的文件分配表的数据结构依然和FAT16相同，所不同的是，FAT32将记录簇链的二进制位数扩展到了32位，故而这种文件系统称为FAT32。32位二进制位的簇链决定了FAT表最大可以寻址2T个簇。这样即使簇的大小为1扇区，理论上仍然能够寻址1TB范围内的分区。但实际中FAT32是不能寻址这样大的空间的，随着分区空间大小的增加，FAT表的记录数会变得臃肿不堪，严重影响系统的性能。所以在实际中通常不格式化超过32GB的FAT32分区。WIN2000及之上的OS已经不直接支持对超过32GB的分区格式化成FAT32，但WIN98依然可以格式化大到127GB的FAT32分区，但这样没必要也不推荐。同时FAT32也有小的限制，FAT32卷必须至少有65527个簇，所以对于小的分区，仍然需要使用FAT16或FAT12。

分区变大时，如果簇很小，文件分配表也随之变大。仍然会有上面的效率问题存在。既要有效地读写大文件，又要最大可能的减少空间的浪费。FAT32同样规定了相应的分区空间对应的簇的大小，见表12：

表12 FAT32分区大小与对因簇大小		
分区空间大小	每个簇的扇区	簇空间大小
<8GB	8	4k
>=8GB且<16GB	16	8k
>=16GB且<32GB	32	16k
>=32GB	64	32k

簇的取值意义和FAT16类似，不过是位数长了点罢了，比较见表13：

表13 FAT各系统记录项的取值含义(16进制)			

FAT12记录项的取值	FAT16记录项的取值	FAT32记录项的取值	对应簇的表现情况
000	0000	00000000	未分配的簇
002~FFF	0002~FFEF	00000002~FFFFFFEF	已分配的簇
FF0~FF6	FFF0~FFF6	FFFFFFF0~FFFFFFF6	系统保留
FF7	FFF7	FFFFFFF7	坏簇
FF8~FFF	FFF8~FFFF	FFFFFFF8~FFFFFFF	文件结束簇

FAT32的另一项重大改革是根目录的文件化，即将根目录等同于普通的文件。这样根目录便没有了FAT16中512个目录项的限制，不够用的时候增加簇链，分配空簇即可。而且，根目录的位置也不再硬性地固定了，可以存储在分区内可寻址的任意簇内，不过通常根目录是最早建立的(格式化就生成了)目录表。所以，我们看到的情况基本上都是根目录首簇占簇区顺序上的第1个簇。在图4. 3. 12中也是按这种情况制作的画的。

FAT32对簇的编号依然同FAT16。顺序上第1个簇仍然编号为第2簇，通常为根目录所用(这和FAT16是不同的，FAT16的根目录并不占簇区空间，32个扇区的根目录以后才是簇区第1个簇)

FAT32的文件寻址方法与FAT16相同，但目录项的各字节参数意义却与FAT16有所不同，一方面它启用了FAT16中的目录项保留字段，同时又完全支持长文件名了。

对于短文件格式的目录项。其参数意义见表14：

表14 FAT32短文件目录项32个字节的表示定义			
字节偏移(16进制)	字节数	定义	
0x0~0x7	8	文件名	
0x8~0xA	3	扩展名	
0xB*	1	属性字节	00000000(读写)
			00000001(只读)
			00000010(隐藏)
			00000100(系统)
			00001000(卷标)
			00010000(子目录)
			00100000(归档)
0xC	1	系统保留	

0xD	1	创建时间的10毫秒位
0xE~0xF	2	文件创建时间
0x10~0x11	2	文件创建日期
0x12~0x13	2	文件最后访问日期
0x14~0x15	2	文件起始簇号的高16位
0x16~0x17	2	文件的最近修改时间
0x18~0x19	2	文件的最近修改日期
0x1A~0x1B	2	文件起始簇号的低16位
0x1C~0x1F	4	表示文件的长度

* 此字段在短文件目录项中不可取值0FH, 如果设置为0FH, 目录段为长文件名目录段

说明:

- (1)、这是FAT32短文件格式目录项的意义。其中文件名、扩展名、时间、日期的算法和FAT16时相同的。
- (2)、由于FAT32可寻址的簇号到了32位二进制数。所以系统在记录文件(文件夹)开始簇地址的时候也需要32位来记录，FAT32启用目录项偏移0x12~0x13来表示起始簇号的高16位。
- (3)、文件长度依然用4个字节表示，这说明FAT32依然只支持小于4GB的文件(目录)，超过4GB的文件(目录), 系统会截断处理。

FAT32的一个重要的特点是完全支持长文件名。长文件名依然是记录在目录项中的。为了低版本的OS或程序能正确读取长文件名文件，系统自动为所有长文件名文件创建了一个对应的短文件名，使对应数据既可以用长文件名寻址，也可以用短文件名寻址。不支持长文件名的OS或程序会忽略它认为不合法的长文件名字段，而支持长文件名的OS或程序则会以长文件名为显式项来记录和编辑，并隐藏起短文件名。

当创建一个长文件名文件时，系统会自动加上对应的短文件名，其一般有的原则：

- (1)、取长文件名的前6个字符加上“~1”形成短文件名，扩展名不变。
- (2)、如果已存在这个文件名，则符号“~”后的数字递增，直到5。
- (3)、如果文件名中“~”后面的数字达到5，则短文件名只使用长文件名的前两个字母。通过数学操纵长文件名的剩余字母生成短文件名的后四个字母，然后加后缀“1”直到最后(如果有必要，或是其他数字以避免重复的文件名)。
- (4)、如果存在老OS或程序无法读取的字符，换以“_”

长文件名的实现有赖于目录项偏移为0xB的属性字节，当此字节的属性为：只读、隐藏、系统、卷标，即其值为0FH时，DOS和WIN32会认为其不合法而忽略其存在。这正是长文件名存在的依据。将目录项的0xB置为0F，其他就任由系统定义了，Windows9x或Windows 2000、XP通常支持不超过255个字符的长文件名。系统将长文件名以13个字符为单位进行切割，每一组占据一个目录项。所以可能一个文件需要多个目录项，这时长文件名的各个目录项按倒序排列在目录表中，以防与其他文件名混淆。

长文件名中的字符采用unicode形式编码(一个巨大的进步哦)，每个字符占据2字节的空间。其目录项定义如表15。

表15 FAT32长文件目录项32个字节的表示定义		

字节偏移 (16进制)	字节数	定义		
0x0	1	属性 字节 位 意 义	7	保留未用
			6	1表示长文件最后一个目录项
			5	保留未用
			4	顺序号数值
			3	
			2	
			1	
			0	
0x1~0xA	10	长文件名unicode码①		
0xB	1	长文件名目录项标志，取值0FH		
0xC	1	系统保留		
0xD	1	校验值(根据短文件名计算得出)		
0xE~0x19	12	长文件名unicode码②		
0x1A~0x1B	2	文件起始簇号(目前常置0)		
0x1C~0x1F	4	长文件名unicode码③		

系统在存储长文件名时，总是先按倒序填充长文件名目录项，然后紧跟其对应的短文件名。从表15可以看出，长文件名中并不存储对应文件的文件开始簇、文件大小、各种时间和日期属性。文件的这些属性还是存放在短文件名目录项中，一个长文件名总是和其相应的短文件名一一对应，短文件名没有了长文件名还可以读，但长文件名如果没有对应的短文件名，不管什么系统都将忽略其存在。所以短文件名是至关重要的。在不支持长文件名的环境中对短文件名中的文件名和扩展名字段作更改(包括删除，因为删除是对首字符改写E5H)，都会使长文件名形同虚设。长文件名和短文件名之间的联系光靠他们之间的位置关系维系显然远远不够。其实，长文件名的0xD字节的校验和起很重要的作用，此校验和是用短文件名的11个字符通过一种运算方式来得到的。系统根据相应的算法来确定相应的长文件名和短文件名是否匹配。这个算法不太容易用公式说明，我们用一段c程序来加以说明。

假设文件名11个字符组成字符串shortname[], 校验和用chknum表示。得到过程如下：

```
int i, j,chknum=0;
for (i=11; i>0; i--)
    chksum = ((chksum & 1) ? 0x80 : 0) + (chksum >> 1) + shortname[j++];
```

如果通过短文件名计算出来的校验和与长文件名中的0xD偏移处数据不相等。系统无论如何都不会将它们配对的。依据长文件名和短文件名对目录项的定义，加上对簇的编号和链接，FAT32上数据的读取便游刃有余了。

五、结束。

本文出自数据恢复网(www.sjhf.net), 疏漏在所难免, 希望指正。若需转载请保留此信息; 若需修改, 请用以下方式与作者取得联系

- 1、<http://www.sjhf.net>
- 2、zymail@vip.sina.com
- 3、sjhf@sjhf.net

[单击此处查看PDF版全文](#)

- 上一篇文章: [FAT文件系统原理\(三\)](#)
- 下一篇文章: [反黑行动之数据恢复](#)

【关闭窗口】

©2001-2006 北京
北亚数据恢复中心

 站内文章搜索

文章标题 ▼

所有栏目 ▼

关键字

搜索

[首页](#) [联系我们](#) [加入收藏](#) [版权申明](#) [文章地图](#) [下载地图](#) [RSS生成](#) [XML生成](#) [友情链接](#)

北亚数据恢复中心

全国统一客服电话: 4006-505-808

北京部: 北京市海淀区中关村大街11号E世界A座832B

京ICP备09039053号