

SD 卡中 FAT32 文件格式快速入门（图文详细介绍）

目录

第一章 硬盘结构与 SD 卡结构.....	2
1.1 硬盘介绍.....	2
1.1 硬盘结构.....	2
1.2 MBR 分析.....	2
1.2 SD 结构.....	4
1.3 SD 卡存储结构.....	5
第二章 FAT32 文件系统介绍.....	6
2.1 FAT 文件系统简介.....	6
2.2 FAT32 文件系统结构.....	6
2.2.1 保留区（深绿色区域）.....	7
2.2.2 文件分区 FAT 表（黄色区域）.....	10
2.2.3 数据区（灰色区域）.....	12
附录.....	16

说明：

MBR：Master Boot Record (主引导记录)

DBR：DOS Boot Record (引导扇区)

FAT：File Allocation Table (文件分配表)

硬件：本文 SD 卡为 Kingston 4GB，FAT32 格式，簇大小 4KB，每扇区 512 字节。

第一章 硬盘结构与 SD 卡结构

1.1 硬盘介绍

1.1 硬盘结构

如果你熟悉硬盘结构跳过本节。下图是硬盘的结构，如果你只是为了学习 SD 卡 FAT32 文件系统的话，这里你只需要注意硬盘排序结构：主引导记录（MBR）—引导扇区—数据—引导扇区—数据。



图 1

1.2 MBR 分析

MBR (Main Boot Record 主引导记录区) 放置在硬盘物理地址 0 的地方。总共 512 字节的主引导扇区中，MBR 只占用了其中的 446 个字节，另外 64 个字节交给了 DPT (Disk Partition Table 硬盘分区表)，最后两个字节“55, AA”是分区的结束标志。DPT 由 4 个分区表组成，每个 16 字节。下图中以以硬盘的 MBR 图，粉红色为硬盘分区表。

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	访问 ▼
0000000000	33	C0	8E	D0	BC	00	7C	FB	50	07	50	1F	FC	BE	1B	7C	3欲屑。 鷗.P. .
0000000010	BF	1B	06	50	57	B9	E5	01	F3	A4	CB	BD	BE	07	B1	04	?PW瑰.螭私??
0000000020	38	6E	00	7C	09	75	13	83	C5	10	E2	F4	CD	18	8B	F5	8n. .u.廻.作?嫫
0000000030	83	C6	10	49	74	19	38	2C	74	F6	A0	B5	07	B4	07	8B	燼.It.8,t鰻??
0000000040	F0	AC	3C	00	74	FC	BB	07	00	B4	0E	CD	10	EB	F2	88	朕<.t ..??脰
0000000050	4E	10	E8	46	00	73	2A	FE	46	10	80	7E	04	0B	74	0B	N.鐵.s*樺.€~.t.
0000000060	80	7E	04	0C	74	05	A0	B6	07	75	D2	80	46	02	06	83	€~.t.柚.u襴F..
0000000070	46	08	06	83	56	0A	00	E8	21	00	73	05	A0	B6	07	EB	F..傳..?.s.柚. ë
0000000080	BC	81	3E	FE	7D	55	AA	74	0B	80	7E	10	00	74	C8	A0	紒>襖U縛.€~.t.莖
0000000090	B7	07	EB	A9	8B	FC	1E	57	8B	F5	CB	BF	05	00	8A	56	?肇熾.W嫫丝..癸V
00000000A0	00	B4	08	CD	13	72	23	8A	C1	24	3F	98	8A	DE	8A	FC	.??r#嫫\$?棟迄 Su
00000000B0	43	F7	E3	8B	D1	86	D6	B1	06	D2	EE	42	F7	E2	39	56	C縻嫫噫?翌B縻9VV
00000000C0	0A	77	23	72	05	39	46	08	73	1C	B8	01	02	BB	00	7C	.w#r.9F.s.?.?. .
00000000D0	8B	4E	02	8B	56	00	CD	13	73	51	4F	74	4E	32	E4	8A	嫫.媧.?sQ0tN2鑄S
00000000E0	56	00	CD	13	EB	E4	8A	56	00	60	BB	AA	55	B4	41	CD	V.?脛癸.`华U碯 í
00000000F0	13	72	36	81	FB	55	AA	75	30	F6	C1	01	74	2B	61	60	.r6伽U縻0隼.t+a`
0000000100	6A	00	6A	00	FF	76	0A	FF	76	08	6A	00	68	00	7C	6A	j.j. v. v.j.h.
0000000110	01	6A	10	B4	42	8B	F4	CD	13	61	61	73	0E	4F	74	0B	.j.落嫫?aaS.Ot..
0000000120	32	E4	8A	56	00	CD	13	EB	D6	61	F9	C3	49	6E	76	61	2鑄V.?脛a Invaa
0000000130	6C	69	64	20	70	61	72	74	69	74	69	6F	6E	20	74	61	lid partition ta
0000000140	62	6C	65	00	45	72	72	6F	72	20	6C	6F	61	64	69	6E	ble.Error loadin
0000000150	67	20	6F	70	65	72	61	74	69	6E	67	20	73	79	73	74	g operating syst
0000000160	65	6D	00	4D	69	73	73	69	6E	67	20	6F	70	65	72	61	em.Missing opera
0000000170	74	69	6E	67	20	73	79	73	74	65	6D	00	00	00	00	00	ting system.....
0000000180	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000000190	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00MBR引导代码...
00000001A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000001B0	00	00	00	00	00	2C	44	63	33	B1	33	B1	00	00	80	01,Dc3??,€.
00000001C0	01	00	07	FE	FF	7B	3F	00	00	00	3D	A8	DA	00	00	00DPT硬盘分区表
00000001D0	C1	7C	0F	FE	FF	7C	A8	DA	00	45	8F	1E	0D	00	00	00嫫.? Y.E?...
00000001E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00分区有效标志...
00000001F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	55	AAU

图 2

对于我们来说，更关注硬盘分区表中红色区域：下图为硬盘分区表详细说明

起始地址	字节数	描述
0x1BE	1	可引导标志，0x00 不可引导，0x80 可引导
0x1BF~0x1C1	3	分区起始 CHS 地址(CHS=磁头、柱面、扇区)，起始地址
0x1C2	1	分区类型
0x1C3~0x1C5	3	分区结束 CHS 地址
0x1C6	4	从磁盘开始到该分区开始的偏移量（分区起始 LBA 地址 Little-endian 顺序）
0x1CA	4	总扇区数（Little-endian 顺序）

表1 图2分区表第一字段			
字节位移	字段长度	值	字段名和定义
0x01BE	BYTE	0x80	引导指示符 (Boot Indicator) 指明该分区是否是活动分区。
0x01BF	BYTE	0x01	开始磁头 (Starting Head)
0x01C0	6位	0x01	开始扇区 (Starting Sector) 只用了0~5位。后面的两位 (第6位和第7位) 被开始柱面字段所使用
0x01C1	10位	0x00	开始柱面 (Starting Cylinder) 除了开始扇区字段的最后两位外, 还使用了1位来组成该柱面值。开始柱面是一个10位数, 最大值为1023
0x01C2	BYTE	0x07	系统ID (System ID) 定义了分区的类型, 详细定义, 请参阅图4
0x01C3	BYTE	0xFE	结束磁头 (Ending Head)
0x01C4	6位	0xFF	结束扇区 (Ending Sector) 只使用了0~5位。最后两位 (第6、7位) 被结束柱面字段所使用
0x01C5	10位	0x7B	结束柱面 (Ending Cylinder) 除了结束扇区字段最后的两位外, 还使用了1位, 以组成该柱面值。结束柱面是一个10位的数, 最大值为1023
0x01C6	DWORD	0x0000003F	相对扇区数 (Relative Sectors) 从该磁盘的开始到该分区的开始的位移量, 以扇区来计算
0x01CA	DWORD	0x00DAA83D	总扇区数 (Total Sectors) 该分区中的扇区总数

图 3

1.2 SD 结构

SD 卡没有分区，默认就是一个分区。首先用 WinHex 文件打开 SD 所在的盘符，显示如下图：

文件名	扩展	大小	创建时间	修改时间	访问时间	属性	第1扇区	哈希集
分区 1		3.7 GB					8192	
剩余扇区		0.6 MB					7743330	
非分区空间		4.0 MB					0	

图 4

① **非分区空间（红线所示）**：起始扇区 0，我认为这个分区就是硬盘上的 MBR 所在区域，打开后如下图所示,也可以称为 SD 卡的 MBR 区域。

文件名	扩展	大小	创建时间	修改时间	访问时间	属性	第1扇区	哈希集
分区 1		3.7 GB					8192	
剩余扇区		0.6 MB					7743330	
非分区空间		4.0 MB					0	

可移动磁?11	Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
样式: Mass Storage Device	00000000	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
固件版本: r	00000010	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Bus: USB	00000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
默认的编辑模式	00000030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
状态: 原始	00000040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
撤销级别: 0	00000050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
撤销相反: n/a	00000060	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	00000070	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
总计容量: 3.7 GB	00000080	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
3,965,190,144 字节	00000090	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
柱面数: 482	000000A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
磁头数: 255	000000B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
扇区/磁道: 63	000000C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
字节/扇区: 512	000000D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
剩余扇区在结尾: 1182	000000E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	000000F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
柱面编号: 0	00000100	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
磁头编号: 0	00000110	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
扇区编号: 1	00000120	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
分区: <1	00000130	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
相对扇区编号: n/a	00000140	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	00000150	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
模式: 16 进制	00000160	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
字符集: ANSI ASCII	00000170	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
偏移量: 16 进制	00000180	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
字节/页面: 33x16=528	00000190	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
窗口 #: 1	000001A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
窗口编号: 2	000001B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
剪贴板: 空	000001C0	03	01	0B	30	F0	C0	00	20	00	00	00	0C	76	00	00	00	...0 咕.v...
临时文件夹: 5.9 GB 空闲	000001D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
C:\DOCUMENT~1\ADMINI~1\LOCALS~1\Temp	000001E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	000001F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	55	AAU
	00000200	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

图 5

根据表 1 知道：

红线区域（00002000）为下个分区的扇区地址，即第 8192 扇区，见图 3 分区 1 的起始扇区。

蓝线区域（00760C00）为 SD 卡总的扇区个数，我们可以计算一下：

$0x760C00 * 512$ （每扇区字节）= 3960995840（字节），与实际大小基本一样。

② **分区 1（蓝线所示）**：起始扇区 8192。

③ **剩余扇区**：才疏学浅，我不知道干嘛的，呵呵

1.3 SD 卡存储结构

由此可知 SD 卡文件系统并不是处在整个 SD 卡最开始的地方，它处在 MBR 所处的保留区之后，于是我们可以对使用 FAT32 文件系统的 SD 卡整体布局给出如下图示。

	A	B	C	D	E	F	G	H	I	J	K	L
1	名称	保留区		保留区			文件分配表	文件分配表		数据区		
2	结构图	MBR, 512字节		DBR, 512字节	FSINFO信息扇区, 512字节		FAT1	FAT2	根目录			
3	物理地址范围 和特殊说明	0x0~0xN-1扇区; (1) N由 MBR表中0x1C6开始的4个字节决定; (2) 0x1CA开始的4个字节为总扇区数。		N + (0~M-1) 扇区; (1) M由偏移地址 0x0e-0x0f保留扇区数决定; (2) 每个 FAT表所占用的扇区数 K 等于偏移地址 0x24-0x77共4个字节			N + (M~K-1)	N + (M+K~M+K+K-1)	数据区起始地址: N+M+K+K扇区			
4		存储器分区结构图										

图 6

第二章 FAT32 文件系统介绍

2.1 FAT 文件系统简介

FAT (File Allocation Table, 文件分配表) 文件系统是 windows 操作系统所使用的一种文件系统, 它的发展过程经历了 FAT12、FAT16、FAT32 三个阶段。FAT 文件系统用“簇”作为数据单元。一个“簇”由一组连续的扇区组成, 簇所含的扇区数必须是 2 的整数次幂。簇的最大值为 64 个扇区, 即 32KB。所有簇从 2 开始进行编号, 每个簇都有一个自己的地址编号。用户文件和目录都存储在簇中。 本文每簇 4KB 大小。

FAT 文件系统的结构中有两个重要的结构: 文件分配表和目录项:

文件分配表: 文件和文件夹内容储存在簇中, 如果一个文件或文件夹需要多余一个簇的空间, 则用 FAT 表来描述, 如何找到另外的簇。FAT 结构用于指出文件的下一个簇, 同时也说明了簇的分配状态。FAT12、FAT16、FAT32 这三种文件系统之间的主要区别在与 FAT 项的大小不同。

目录项: FAT 文件系统的每一个文件和文件夹都被分配到一个目录项, 目录项中记录着文件名、大小、文件内容起始地址以及其他一些元数据。

在 FAT 文件系统中, 文件系统的数据记录在“引导扇区中 (DBR)”中。引导扇区位于整个文件系统的 0 号扇区, 是文件系统隐藏区域 (也称为保留区) 的一部分, 我们称其为 DBR (DOS Boot Recorder——DOS 引导记录) 扇区, DBR 中记录着文件系统的起始位置、大小、FAT 表个数及大小等相关信息。在 FAT 文件系统中, 同时使用“扇区地址”和“簇地址”两种地址管理方式。这是因为只有存储用户

数据的数据区使用簇进行管理 (FAT12 和 FAT16 的根目录除外), 所有簇都位于数据区。其他文件系统管理数据区域是不以簇进行管理的, 这部分区域使用扇区地址进行管理。文件系统的起始扇区为 0 号扇区。

2.2 FAT32 文件系统结构

FAT 文件系统整体分布如上图 存储器文件结构图 所示, 有:

- 【1】0x00~0x02: 3 字节, 跳转指令。
- 【2】0x03~0x0A: 8 字节, 文件系统标志和版本号, 这里为 MSDOC5.0。
- 【3】0x0B~0x0C: 2 字节, 每扇区字节数, 512 (0x02 00)。
- 【4】0x0D~0x0D: 1 字节, 每簇扇区数, 8 (0x08)。
- 【5】0x0E~0x0F: 2 字节, 保留扇区数, 38 (0x00 26), 符合 FAT1 起始地址为 38 扇区。
- 【6】0x10~0x10: 1 字节, FAT 表个数, 2。
- 【7】0x11~0x12: 2 字节, FAT32 必须等于 0, FAT12/FAT16 为根目录中目录的个数;
- 【8】0x13~0x14: 2 字节, FAT32 必须等于 0, FAT12/FAT16 为扇区总数。
- 【9】0x15~0x15: 1 字节, 哪种存储介质, 0xF8 标准值, 可移动存储介质, 常用的 0xF0。
- 【10】0x16~0x17: 2 字节, FAT32 必须为 0, FAT12/FAT16 为一个 FAT 表所占的扇区数。
- 【11】0x18~0x19: 2 字节, 每磁道扇区数, 只对于有“特殊形状”(由磁头和柱面每分割为若干磁道)的存储介质有效, 63 (0x00 3F)。
- 【12】0x1A~0x1B: 2 字节, 磁头数, 只对特殊的介质才有效, 255 (0x00 FF)。
- 【13】0x1C~0x1F: 4 字节, EBR 分区之前所隐藏的扇区数, 8192 (0x00 00 20 00), 与 MBR 中地址 0x1C6 开始的 4 个字节数值相等。
- 【14】0x20~0x23: 4 字节, 文件系统总扇区数, 7736320 (0x 00 76 0C 00), $7736320 * 512 = 3960995840 \approx 3.67\text{GB}$ 。
- 【15】0x24~0x27: 4 字节, 每个 FAT 表占用扇区数, 7541 (0x 00 00 1D 75)。
- 【16】0x28~0x29: 2 字节, 标记, 此域 FAT32 特有。
- 【17】0x2A~0x2B: 2 字节, FAT32 版本号 0.0, FAT32 特有。
- 【18】0x2C~0x2F: 4 字节, 根目录所在第一个簇的簇号, 2。(虽然在 FAT32 文件系统下, 根目录可以存放在数据区的任何位置, 但是通常情况下还是起始于 2 号簇)
- 【19】0x30~0x31: 2 字节, FSINFO (文件系统信息扇区) 扇区号 1, 该扇区为操作系统提供关于空簇总数及下一可用簇的信息。
- 【20】0x32~0x33: 2 字节, 备份引导扇区的位置。备份引导扇区总是位于文件系统的 6 号扇区。
- 【21】0x34~0x3F: 12 字节, 用于以后 FAT 扩展使用。
- 【22】0x40~0x40: 1 字节, 与 FAT12/16 的定义相同, 只不过两者位于启动扇区不同的位置而已。
- 【23】0x41~0x41: 1 字节, 与 FAT12/16 的定义相同, 只不过两者位于启动扇区不同的位置而已。
- 【24】0x42~0x42: 1 字节, 扩展引导标志, 0x29。与 FAT12/16 的定义相同, 只不过两者位于启动扇区不同的位置而已。

【25】 0x43~0x46: 4 字节, 卷序列号。通常为一个随机值。

【26】0x47~0x51：11 字节，卷标（ASCII 码），如果建立文件系统的时候指定了卷标，会保存在此。

【27】0x52~0x59: 8 字节，文件系统格式的 ASCII 码，FAT32。

★【28】0x5A~0x1FD: 90~509 共 410 字节, 未使用。该部分没有明确的用途。

【29】 0x1FE~0x1FF: 签名标志“55AA”。

★说明：引导代码

FAT 文件系统将引导代码与文件形同数据结构融合在一起, FAT32 文件系统引导扇区的 512 字节中, 90~509 字节为引导代码, 而 FAT12/16 则是 62~509 字节为引导代码。同时, FAT32 还可以利用引导扇区后的扇区空间存放附加的引导代码。一个 FAT 卷即使不是可引导文件系统, 也会存在引导代码。

2.2.1.2 FSInfo 信息分区

FAT32 在保留区中增加了一个 FSINFO 扇区,用以记录文件系统中空闲簇的数量以及下一可用簇的簇号等信息,以供操作系统作为参考。FSINFO 信息扇区一般位于文件系统的 1 号扇区,结构非常简单。FSINFO 信息扇区结构。

SYSTEM		4.0 KB	2013-05-08 14...	2013-05-08 14...	2013-05-08													15128				
(根目录)		4.0 KB													15120							
剩余空间:		3.7 GB																				
可移动磁盘 1, 分区 1		100% 空闲	Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F			
文件系统:		FAT32	00000200	52	52	61	41	00	00	00	00	00	00	00	00	00	00	00	00	RRaA.....		
卷标:		KINGSTON	00000210	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
			00000220	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
默认的编辑模式			00000230	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
状态:		原始	00000240	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
撤销级别:		0	00000250	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
撤销相反:		n/a	00000260	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
分配可见的驱动器空间。			00000270	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
簇编号:		n/a	00000280	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
保留扇区			00000290	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
			000002A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
快照获得		3 小时以前	000002B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
物理扇区编号:		8193	000002C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
逻辑扇区编号:		1	000002D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	RRaA.....		
			000002E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
已使用空间:		2.7 MB	000002F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
		2,801,664 字节	00000300	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
剩余空间:		3.7 GB	00000310	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
		3,950,452,736 字节	00000320	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
总计容量:		3.7 GB	00000330	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
		3,960,995,840 字节	00000340	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
			00000350	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
字节/簇:		4,096	00000360	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
剩余簇:		964,466	00000370	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
总计簇:		965,150	00000380	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
字节/扇区:		512	00000390	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
可用扇区:		7,721,200	000003A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
第一数据扇区:		15120	000003B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
			000003C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
物理磁盘:		1	000003D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
显示时区:		原始	000003E0	00	00	00	00	72	72	41	61	72	B7	0E	00	15	00	00	00	rrAar....		
模式:		16 进制	000003F0	00	00	00	00	00	00	00	00	00	00	00	00	00	55	AA	U		
字符集:		ANSI ASCII	00000400	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
偏移量:		16 进制																				
字节/页面:		33x16=528																				
扇区1 / 7736320			偏移:												3ED				= 0 选择			

图 8

- 【1】 0x00~0x03: 4 个字节，扩展引导标志“0x52526141”。
- 【2】 0x04~0x1E3: 480 个字节，未使用，全部置 0。
- 【3】 0x1E4~0x1E7: 4 个字节，FSINFO 签名“0x72724161”。
- 【4】 0x1E8~0x1EB: 4 个字节，文件系统的空簇数，964466 (0x00 0E B7 72)。
- 【5】 0x1EC~0x1EF: 4 个字节，下一可用簇号 (0x 00 00 00 15)。
- 【6】 0x1F0~0x1FD: 14 个字节，未使用。
- 【7】 0x1FE~0x1FF: 2 个字节，“55 AA”标志。

温馨提示：通常情况下，文件系统的 2 号扇区结尾也会被设置“55 AA”标志。6 号扇区也会有一个引导扇区的备份，相应的，7 号扇区应该是一个备份 FSINFO 信息扇区，8 号扇区可以看做是 2 号扇区的备份。

2.2.2 文件分区 FAT 表（黄色区域）

紧跟在保留分区后面的是 FAT 区，其由两个完全相同的 FAT（File Allocation Table，文件分配表）表组成，FAT 文件系统的名字也是因此而来。**FAT 表（File Allocation Table）是一组与数据簇号对应的列表。**FAT2 紧跟在 FAT1 之后，它的位置可以通过 FAT1 的位置加上 FAT 表的大小扇区数计算出来。

2.2.2.1 文件系统概述

文件系统分配磁盘空间按簇来分配。因此，文件占有磁盘空间时，基本单位不是字节而是簇，即使某个文件只有一个字节，操作系统也会给它分配一个最小单元：即一个簇。对于大文件，需要分配多个簇。同一个文件的数据并不一定完整地存放在磁盘中一个连续地区域内，而往往会分若干段，像链子一样存放。这种存储方式称为文件的链式存储。为了实现文件的链式存储，文件系统必须准确地记录哪些簇已经被文件占用，还必须为每个已经占用的簇指明存储后继的下一个簇的簇号，对于文件的最后一簇，则要指明本簇无后继簇。这些都是由 FAT 表来保存的，FAT 表对应表项中记录着它所代表的簇的有关信息：诸如是空，是不是坏簇，是否是已经是某个文件的尾簇等。

❖ 对于文件系统来说，FAT 表有两个重要作用：描述簇的分配状态以及标明文件或目录的下一簇的簇号。

❖ 通常情况下，一个 FAT 把文件系统会有两个 FAT 表，但有时也允许只有一个 FAT 表，FAT 表的具体个数记录在引导扇区的偏移 0x10 字节处。

❖ 由于 FAT 区紧跟在文件系统保留区后，所以 FAT1 在文件系统的位置可以通过引导记录中偏移 0x0E~0x0F 字节处的“保留扇区数”得到，如存储器结构体图中 M 值。

2.2.2.2 FAT 表分析说明

FAT32 中每个簇的簇地址是有 32bit (4 个字节), FAT 表中的所有字节位置以 4 字节为单位进行划分, 并对所有划分后的位置由 0 进行地址编号。0 号地址与 1 号地址被系统保留并存储特殊标志内容。从 2 号地址开始, 每个地址对应于数据区的簇号, FAT 表中的地址编号与数据区中的簇号相同。我们称 FAT 表中的这些地址为 FAT 表项, FAT 表项中记录的值为 FAT 表项值。

当文件系统被创建, 也就是进行格式化操作时, 分配给 FAT 区域的空间将会被清空, 在 FAT1 与 FAT2 的 0 号表项与 1 号表项写入特定值。由于创建文件系统的同时也会创建根目录, 也就是为根目录分配了一个簇空间, 通常为 2 号簇, 与之对应的 2 号 FAT 表项记录为 2 号簇, 被写入一个结束标记。

几点说明:

- 由于簇号起始于 2 号, 所以 FAT 表项的 0 号表项与 1 号表项不与任何簇对应。FAT32 的 0 号表项值总是 “F8FFFF0F”。
- 1 号表项可能被用于记录脏标志, 以说明文件系统没有被正常卸载或者磁盘表面存在错误。不过这个值并不重要。正常情况下 1 号表项的值为 “FFFFFFFF” 或 “FFFFFF0F”。
- 如果某个簇未被分配使用, 它对应的 FAT 表项内容为 0;
- 当某个簇已被分配使用, 则它对应的 FAT 表项内的 FAT 表项值也就是该文件的下一个存储位置的簇号。如果该文件结束于该簇, 则在它的 FAT 表项中记录的是一个文件结束标记, 对于 FAT32 而言, 代表文件结束的 FAT 表项值为 0x0FFFFFFF。
- 如果某个簇存在坏扇区, 则整个簇会用 0xFFFFFFFF7 标记为坏簇, 这个坏簇标记就记录在它对应的 FAT 表项中。
- 在文件系统中新建文件时, 如果新建的文件只占用一个簇, 为其分配的簇对应的 FAT 表项将会写入结束标记。如果新建的文件不只占用一个簇, 则在其所占用的每个簇对应的 FAT 表项中写入为其分配的下一簇的簇号, 在最后一个簇对应的 FAT 表项中写入结束标记。
- 新建目录时, 只为其分配一个簇的空间, 对应的 FAT 表项中写入结束标记。当目录增大超出一个簇的大小时, 将会在空闲空间中继续为其分配一个簇, 并在 FAT 表中为其建立 FAT 表链以描述它所占用的簇情况。

2.2.2.3 FAT 表示例

【0 号表项】: 0x0FFFFFFF8;FAT 表起始固定标识

【1 号表项】: 0xFFFFFFFF;不是用, 默认值

【2 号表项】: 0x0FFFFFFF;根目录所在簇,

紫色的为 3 号表项, 绿色的为 4 号表项, 以此类推。注意: 0 和 1 号表项均不与实际的物理地址对应, 2 号表项开始才与物理地址对应。2 号表项物理地址为 FAT2 表后紧跟着的那个簇! 3 号在 2 号表项紧跟着的一个簇。

计算: FAT 所占扇区数 7541 (0x1D75); FAT2 起始扇区为 38+7541=7579; 根目录起

始扇区在 $7579 + 7541 = 15120$ 。记得本文 8 个扇区为一个簇，即 **4K (0x1000)**，簇是系统分配内存的最小单元。

如图表中的起始地址对比，发现根目录起始地址刚好在 15120，所以 FAT 表中 0 和 1 号表项没有对应物理地址！

文件名	扩展名	大小	创建时间	修改时间	访问时间	属性	第1扇区	哈希集
SYSTEM		4.0 KB	2013-05-07 11...	2013-05-07 11...	2013-05-07		15128	
(根目录)		4.0 KB					15120	
剩余空间:		3.7 GB						
启动扇区		19.0 KB					0	
空闲空间								
FAT 2		3.7 MB					7579	
FAT 1		3.7 MB					38	
撤销相反:	n/a	Offset	0号表项 1号表项 2号表项					
分配可见的驱动器空间。		003B3600	FF FF FF 0F	FF FF FF FF	FF FF FF 0F	FF FF FF 0F	FF FF FF 0F	?
簇编号:	n/a	003B3610	FF FF FF 0F	06 00 00 00	07 00 00 00	08 00 00 00	09 00 00 00
FAT 2	保留	003B3620	09 00 00 00	0A 00 00 00	0B 00 00 00	0C 00 00 00	0D 00 00 00
		003B3630	0D 00 00 00	0E 00 00 00	0F 00 00 00	10 00 00 00	11 00 00 00

图 9

我将 SD 格式化，新建了一个 test.txt 的文本文件，大小为 8.2kB。如下图：

文件名	扩展名	大小	创建时间	修改时间	访问时间	属性	第1扇区	哈希集
(根目录)		4.0 KB					15120	
test.txt	txt	8.2 KB	2013-05-09 09...	2013-05-09 09...	2013-05-09	A	15128	
剩余空间:		3.7 GB						
启动扇区		19.0 KB					0	
空闲空间								
FAT 2		3.7 MB					7579	
FAT 1		3.7 MB					38	

可移动磁盘 1, 分区 1 文件系统: 卷标: 默认的编辑模式 状态:	100% 空闲 FAT32 KINGSTON 原始	Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
		00004C00	F8	FF	FF	0F	FF	FF	FF	FF	FF	FF	FF	FF	0F	04	00	00	00
		00004C10	05	00	00	00	FF	FF	FF	FF	0F	00	00	00	00	00	00	00	00
		00004C20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
		00004C30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
		00004C40	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	

图 10

- 我们分析上图：
- 在图中可以看出，**test.txt** 文件起始簇为 **15128**，这个地址是我们根目录（2 号簇）后的一个簇，所以 **test.txt** 文件起始簇是 **3 号簇**，也就是 **3 号表项**（FAT 表中表项值与簇号对应）。
- 【1】：2 号表项为根目录，即 2 号簇。
 - 【2】：3 号表项为-0x00 00 00 04，test 文件的下一簇号在 4 号表项，查看 4 号表项。
 - 【3】：4 号表项为-0x00 00 00 05，test 文件下一簇号在 5 号表项，查看 5 号表项。
 - 【4】：5 号表项为-0x0F FF FF FF，结束符号。说明文件在 5 号簇时就存储完毕。

2.2.3 数据区（灰色区域）

数据区时真正用于存放用户数据的区域。数据区紧跟在 FAT2 之后，被划分成一个个的簇。所有的簇从 2 开始进行编号，也就是说，2 号簇的起始位置就是数据区的起始位置。

2.2.3.1 根目录

FAT 表示例中，根目录截图：

文件名	扩展	大小	创建时间	修改时间	访问时间	属性	第1扇区	哈希集
(根目录)		4.0 KB					15120	
test.txt	txt	8.2 KB	2013-05-09 09...	2013-05-09 09...	2013-05-09	A	15128	
剩余空间:		3.7 GB						
启动扇区		19.0 KB					0	
空闲空间								
FAT 2		3.7 MB					7579	
FAT 1		3.7 MB					38	

可移动磁盘 1, 分区 1	100% 空闲	Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
文件系统: FAT32		00762000	4E	49	4E	47	53	54	4F	4E	20	20	20	08	00	00	00	00	KINGSTON
卷标: KINGSTON		00762010	00	00	00	00	00	00	4A	4D	A9	42	00	08	00	00	00	00JM III.....
默认的编辑模式		00762020	54	45	53	54	20	20	20	20	54	58	54	20	18	84	80	4D	TEST TXT . eM
状态: 原始		00762030	A9	42	A9	42	00	00	78	4D	A9	42	03	00	EE	20	00	00	III III.. xM III.. ?...
撤销级别: 0		00762040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
撤销相反: n/a		00762050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
		00762060	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

图 11

虽然原则上 FAT32 允许根目录位于数据区的任何位置，但通常情况下它都位于数据区起始扇区，2 号簇，可以在 **DBR 偏移地址 0x2C~0x2F 查看**。在 FAT 文件系统中，先要寻找数据区的第一簇（即 2 号簇）的位置，它不是位于文件系统开始处，而是位于数据区。从前面的学习知道，在数据区前面是保留区域和 **FAT 区域**，在前面还有 **MBR 区域**，这些区域都不使用 **FAT 表** 进行管理。因此，数据区以前的区域只能使用扇区地址，而无法使用簇地址。

其实在 2.2.2.3 节，FAT 表示例中我们就已经计算过其实地址（**15128**），**注意这个地址不是物理地址哦，只是在 FAT 文件区域中一个相对地址！因为在之前还有 MBR 保留区域！**

【保留区域大小（绿色部分）】：DBR 偏移地址 **0x0E~0x0F**，**38（0x26）**，

【FAT 表个数】：DBR 偏移地址 **0x10** 开始 **2 字节**，**2 个**，

【每个 FAT 表扇区数】：DBR 偏移地址 **0x24~0x27**，**7541（0x 00 00 1D 75）**，

说明：以上值均参见 2.2.1.1 节。

【计算公式】：

数据区起始扇区号 = 保留扇区数 + 每个 FAT 表大小扇区数 × FAT 表个数

【示例】：数据区起始扇区号 = 38 + 7541*2 = 15120

为了避免根目录被更改，也可以用下面的计算公式计算出根目录扇区：

根目录起始扇区=保留扇区数+FAT×2+(根目录起始簇-2)× 每簇的扇区数。

说明：如果要得到物理地址，需要加上 MBR 保留区域大小。

2.2.3.2 根目录的短文件目录项定义

目录所在的扇区，都是以 32 Bytes 划分为一个单位，每个单位称为一个目录项(Directory Entry)，即每个目录项的长度都是 32 Bytes。根目录由若干个目录项组成，一个目录项占用 32 个字节，可以是长文件名目录项、文件目录项、子目录项等。32 字节的具体定义如下图：

表1 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,如果设值为0FH,目录段为长文件名目录段

图 12

示例:

文件名	扩展名	大小	创建时间	修改时间	访问时间	属性	第1扇区	哈希集
(根目录)		4.0 KB					15120	
test.txt	txt	8.2 KB	2013-05-09 09...	2013-05-09 09...	2013-05-09	A	15128	
剩余空间:		3.7 GB						
启动扇区		19.0 KB					0	
空闲空间								
FAT 2		3.7 MB					7579	
FAT 1		3.7 MB					38	

可移动磁盘 1, 分区 1	100% 空闲	Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
文件系统: FAT32		00762000	4B	49	4E	47	53	54	4F	4E	20	20	20	08	00	00	00	00	KINGSTON
卷标: KINGSTON		00762010	00	00	00	00	00	00	4A	4D	A9	42	00	00	00	00	00	00JMIII.....
默认的编辑模式		00762020	54	45	53	54	20	20	20	20	54	58	54	20	18	84	80	4D	TEST TXT .em
状态: 原始		00762030	A9	42	A9	42	00	00	78	4D	A9	42	03	00	EE	20	00	00	III III..xMIII..?..
		00762040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

图 13

特别关注的参数说明:

【1】: 文件或者文件夹存储的起始簇号, 上图中紫色区域, 偏移地址: 0x14-0x15 (高 16 为) 和 0x1A-0x1B (低 16 位), 0x 00 00 00 03, 表示这个文件存储在 3 号簇的位置, 在 FAT 表中为 3 号表项。

【2】: 文件大小: 偏移地址 0x1C-0x1F, 0x 00 00 20 EE (8430 字节);

【3】文件属性: 偏移地址 0x0B-0x0B, 0x20, 归档。

其他说明：

【1】文件名的第一个字节，为 0xE5，表示该项已被删除。

【2】名字为 0x2E(“.”)，表示当前目录。

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
007D0000	2E	20	20	20	20	20	20	20	20	20	20	10	00	1C	37	73
007D0010	A8	42	A8	42	00	00	38	73	A8	42	70	00	00	00	00	00
007D0020	2E	2E	20	20	20	20	20	20	20	20	20	10	00	1C	37	73
007D0030	A8	42	A8	42	00	00	38	73	A8	42	03	00	00	00	00	00

图 14

【3】名字为 0x2E 0x2E(“..”)，表示上一级目录。

2.2.3.3 长文件目录定义

表2 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 3 2 顺序号数值 1 0
0x1~0xA	10	长文件名unicode码
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
0x1A~0x1B	2	文件起始簇号(目前常置 0)
0x1C~0x1F	4	长文件名unicode码

图 15

附录

DBR 扇区中包含有一个重要数据结构 BPB(BIOS Parameter Block)。

NOTE: 在以下的叙述中, 名字以 BPB_开头的属于 BPB 部分, 以 BS 开头的属于启动扇区 (Boot Sector) 部分, 实际上并不属于 BPB。

	offset (byte)	长度 (byte)	描述
BS_jumpBoot	0x00	3	跳转指令, 指向启动代码
BS_OEMName	0x03	8	建议值为“MSWIN4.1”。有些厂商的 FAT 驱动可能会检测此项, 所以设为“MSWIN4.1”可以尽量避免兼容性的问题
BPB_BytsPerSec	0x0b	2	每扇区的字节数, 取值只能是以下几种: 512, 1024, 2048 或是 4096。设为 512 会取得最好的兼容性, 目前有很多 FAT 代码都是硬性规定每扇区的字节数为 512, 而不是实际的检测此值。但微软的操作系统能够很好支持 1024, 2048 或是 4096
BPB_SecPerClus	0x0d	1	每簇的扇区数, 其值必须中 2 的整数次方 (该整数必须 ≥ 0), 同时还要保证每簇的字节数不能超过 32K, 也就

			是 1024*32 字节
BPB_RsvdSecCnt	0x0e	2	保留扇区的数目, 此域不能为 0, FAT12/FAT16 必须为 1, FAT32 的典型值取为 32, , 微软的系统支持任何非 0 值
BPB_BumFATs	0x10	1	分区中 FAT 表的份数, , 任何 FAT 格式都建议为 2
BPB_RootEntCnt	0x11	2	对于 FAT12 和 FAT16 此域包含根目录中目录的个数 (每项长度为 32bytes), 对于 FAT32, 此项必须为 0。对于 FAT12 和 FAT16, 此数乘以 32 必为 BPB_BytesPerSec 的偶数倍, 为了达到更好的兼容性, FAT12 和 FAT16 都应该取值为 512
BPB_ToSec16	0x13	2	早期版本中 16bit 的总扇区, 这里总扇区数包括 FAT 卷上四个基本分区的全部扇区, 此域可以为 0, 若此域为 0, 那么 BPB_ToSec32 必须为 0, 对于 FAT32, 此域必为 0。对于 FAT12/FAT16, 此域填写总扇区数, 如果该值小于 0x10000 的话, BPB_ToSec32 必须为 0

BPB_Media	0x15	1	对于“固定”（不可移动）存储介质而言，0xF8 是标准值，对于可移动存储介质，经常使用的数值是 0xF0，此域合法的取值可以取 0xF0, 0xF8, 0xF9, 0xFA, 0xFC, 0xFD, 0xFE, 0xFF。另外要提醒的是，无论此域写入什么数值，同时也必须在 FAT[0] 的低字节写入相同的值，这是因为早期的 MSDOS 1.x 使用该字节来判定是何种存储介质
BPB_FATSz16	0x16	2	FAT12/FAT16 一个 FAT 表所占的扇区数，对于 FAT32 来说此域必须为 0，在 BPB_FATZ32 中有指定 FAT 表的大小
BPB_SecPerTrk	0x18	2	每磁道的扇区数，用于 BIOS 中断 0x13，此域只对于有“特殊形状”（由磁头和柱面每分割为若干磁道）的存储介质有效，同时必须可以调用 BIOS 的 0x13 中断得到此数值
BPB_NumHeads	0x1A	2	磁头数，用于 BIOS 的 0x13 中断，类似于上面的 BPB_SecPerTrk，只对特殊的介质才有效，此域包含一个至少为 1 的数值，比如 1,4M 的软盘此域为 2
BPB_HidSec	0x1C	4	在此 FAT 分区之前所隐藏的扇区数，必须使得调用 BIOS 的 0x13 中断可以得到此数值，对于那些没有分区的存储介质，此域必须为 0，具体使用什么值由操作系统决定
BPB_ToSec32	0x20	4	该卷总扇区数（32bit），这里的扇区总数包括 FAT 卷四个基本分的全部扇区，此域可以为 0，若此域为 0，BPB_ToSec16 必须为非 0，对 FAT32，此域必须是非 0。对于 FAT12/FAT16 如果总扇区数大于或等于 0x10000 的话，此域就是扇区总数，同时 BPB_ToSec16 的值为 0。

FAT32 的 BPB 的内容和 FAT12/16 的内容在地址 0x36 以前是完全一样的，从偏移量 0x36 开始，他们的内容有所区别，具体的内容要看 FAT 类型为 FAT12/16 还是 FAT32，这点保证了在启动扇区中包含一个完整的 FAT12/16 或 FAT32 的 BPB 的内容，这么做是为了达到最好的兼

从 offset 36 开始 FAT12/FAT16 的内容开始区别于 FAT32，下面分两个表格列出，下表为 FAT12/FAT16 的内容

名称	offset (byte)	长度 (byte)	描述
BS_drvNum	0x24	1	用于 BIOS 中断 0x13 得到磁盘驱动器参数，（0x00 为软盘，0x80 为硬盘）。此域实际上由操作系统决定
BS_Reseved1	0x25	1	保留（供 NT 使用），格式化 FAT 卷时必须设为 0
BS_VolID	0x26	1	扩展引导标记（0x29）用于指明此后的 3 个域可用
BS_BootSig	0x27	4	卷标序列号，此域以 BS_VolLab 一起可以用来检测磁盘是否正确，FAT 文件系统可以用此判断连接的可移动磁盘是否正确，引域往往是由时间和日期组成的一个 32 位的值
BS_VolLab	0x2B	11	磁盘卷标，此域必须与根目录中 11 字节长的卷标一致。FAT 文件系统必须保证在根目录的卷标文件列改或是创建的同时，此域的内容能得到时的更新，当 FAT 卷没有卷标时，此域的内容为“NO NAME”

BS_FilSysType	0x36	8	以下的几种之一：“FAT12”，“FAT16”，“FAT32” 不少人错误的认为 FAT 文件系统的类型由此域来确认，他细点你就能发现此域并不是 BPB 的一部分，只是一个字符串而已，微软的操作系统并不使用此域来确定 FAT 文件的类型，；因为它常常被写错或是根本就不存在。
---------------	------	---	--

下表为 FAT32 的内容

名称	offset (byte)	长度 (byte)	描述
BPB_FATSz32	0x24	4	一个 FAT 表所占的扇区数，此域为 FAT32 特有，同时 BPB_FATSz16 必须为 0

BPB_Flags	0x28	2	此域 FAT32 特有。 Bits0-3:不小于 0 的 FAT (active FAT) 数目，只有在镜像 (mirrorig) 禁止时才有效。 Bits 4-6: 保留 Bits 7: 0 表示 FAT 实时镜像到所有的 FAT 表中 1 表示只有一个活动的 FAT 表。这个表就是 Bits0-3 所指定的那个 Bits8-15: 保留
BPB_FSVer	0x2A	2	此域为 FAT32 特有， 高位为 FAT32 的主版本号，低位为次版本号，这个版本号是为了以后更高级的 FAT 版本考虑，假设当前的操作系统只能支持的 FAT32 版本号为 0.0。那么该操作系统检测到此域不为 0 时，它便会忽略 FAT 卷，因为它的版本号比系统能支持的版本号要高
BPB_RootClus	0x2C	4	根目录所在第一个簇的簇号，通常该数值为 2，但不是必

			须为 2 磁盘工具在改变根目录位置时，必须想办法让磁盘上第一个非坏簇作为根目录的第一个簇（比如第 2 簇，除非它已经被标记为坏簇），这样的话，如果此域正好为 0 的话磁盘检测工具也能轻松的找到根目录所在簇的位置
BPB_FSInfo	0x30	2	保留区中 FAT32 卷 FSINFO 结构所占的扇区数，通常为 1 在 Backup Boot 中会有一个 FSINFO 的备份，但该备份只是更新其中的指针，也就是说无论是主引导记录还是备份引导记录都是指向同一个 FSINFO 结构
BPB__BkBootSec	0x32	2	如果不为 0，表示在保留区中引导记录的备数据所占的扇区数，通常为 6。同时不建议使用 6 以外的其他数值
BPB_Reserved	0x34	12	用于以后 FAT 扩展使用，对 FAT32。此域用 0 填充
BS_DrvNum	0x40	1	与 FAT12/16 的定义相同，只不过两者位于启动扇区不同的位置而已

BS_Reserved1	0x41	1	与 FAT12/16 的定义相同，只不过两者位于启动扇区不同的位置而已
BS_BootSig	0x42	1	与 FAT12/16 的定义相同，只不过两者位于启动扇区不同的位置而已
BS_VolID	0x43	4	与 FAT12/16 的定义相同，只不过两者位于启动扇区不同的位置而已
BS_FilSysType	0x47	11	与 FAT12/16 的定义相同，只不过两者位于启动扇区不同的位置而已
BS_FilSysType	0x52	8	通常设置为“FAT32”，请参照 FAT12/16 此部分的陈述。