

第 5 章

FAT32 文件系统

5.1 FAT 文件系统概述

FAT(File Allocation Table)文件系统是微软公司最初为 DOS 操作系统设计的一种磁盘格式。FAT 文件系统用“簇”作为文件或文件夹(即子目录)的分配单元,1 个“簇”等于 2^n 个连续的扇区(注: $0 \leq n \leq 10$),1 簇等于多少个扇区在逻辑盘的分区引导扇区中有定义。

目前,FAT 文件系统主要有 FAT12、FAT16、FAT32 和 exFAT 文件系统,最早的 FAT 文件系统是指 FAT12,FAT12 是用 12 位(即 1.5 字节)来对簇进行编号;可以管理的最大容量是 8MB,在当时的情况下,FAT12 管理能力是非常大的。但随着其他外存储器(如: U 盘、手机卡、存储卡等)的出现以及硬盘容量的不断增大,微软公司先后又推出了 FAT16、FAT32 和 exFAT 文件系统。

FAT16 是用 16 位(即 2 字节)来对簇进行编号;FAT32 则是用 32 位(即 4 字节)来对簇进行编号。对于 FAT12 和 FAT16 文件系统而言,如果逻辑盘被(快速)格式化,在该逻辑盘上存储的第一个文件或者子目录,它的开始簇号是 2;而对于 FAT32 文件系统而言,逻辑盘的根目录开始簇号是 2。

FAT12 文件系统主要用于软磁盘,而 FAT16 文件系统主要用于小容量的硬盘、U 盘、存储卡等;FAT32 文件系统主要用于大容量的硬盘、U 盘、存储卡等;而 exFAT 是微软公司在 Windows Embedded 6.0 中引入的一种适合于闪存的文件系统。

5.2 FAT16 文件系统

5.2.1 FAT16 文件系统介绍

DOS2.0 以前的操作系统使用的文件系统是 FAT12;在 DOS2.0 以后,微软公司推出了新的文件系统 FAT16,FAT16 除了采用 16 位(即 2 字节)来对一个簇进行编号外,在其他方面与 FAT12 没有区别。因此,FAT16 可以管理的总簇数增加到了 65534 个(即 0X02 ~

0xFFFF)。当簇的总数小于 4096 时,文件系统还是使用 FAT12; 当簇的总数大于 4096 时,文件系统使用的就是 FAT16。刚推出的 FAT16 文件系统管理磁盘的能力实际上是 32M,这在当时看来是足够大的。

1987 年以后,硬盘技术的发展推动了文件系统的发展。在 DOS4.0 以后,FAT16 可以管理到 128M 的磁盘,然而这个数字还在不断增加,一直到 2GB。需要指出的是,在 Windows 95 系统中,采用了一种比较独特的技术,叫作 VFAT 来解决长文件名的问题。然而 FAT16 分区格式存在严重的缺点,即大容量磁盘利用率非常低; 因为磁盘文件的分配以簇为单位,当一个簇比较大时,存储的小文件越多,浪费的磁盘空间也就越多,也就是说,即使一个很小的文件也要占用一个簇,这个簇剩余的空间便会全部闲置,这样就造成了磁盘空间的浪费。

5.2.2 FAT16 文件系统组成

FAT16 文件系统由保留扇区、FAT 表(注: FAT 表由 FAT1 表和 FAT2 表组成)、根目录区域和数据区 4 大部分组成,总体布局大致如图 5.1 所示(其中: i 和 j 均为正整数,且 $i < j$)。在 FAT16 文件系统中,根目录占用区域是固定的,位于 FAT 表和数据区之间,因此,根目录下所存放的文件(夹)数量也是有限的。

保留扇区	FAT表		根目录 区域	数据区
	FAT1表	FAT2表		
0号扇区, 1号扇区,	i 号扇区,	j 号扇区, $j+1$ 号扇区,, 2号簇, 3号簇,

图 5.1 FAT16 文件系统总体布局

5.2.3 FAT16 逻辑扇区号与簇号

当用户使用 FAT16 文件系统对一个逻辑盘进行(快速)格式化后,查看到的逻辑盘总容量是指数据区的容量,而不是指整个逻辑盘总扇区数的容量。对于逻辑盘而言,逻辑扇区是从 0 开始连续编号,即该分区引导扇区的逻辑扇区号为 0; 而簇则是从数据区开始连续编号,一般情况下,簇的开始号为 2。

计算逻辑盘总扇区数如式(5.1):

$$\text{逻辑盘总扇区数} = \text{保留扇区数} + \text{每个 FAT 表占用扇区数} \times 2 + (\text{根目录区域存放的目录项数} \times 32) \div 512 + \text{数据区占用扇区数} \quad (5.1)$$

簇号与逻辑扇区号转换公式如式(5.2):

$$\begin{aligned} \text{逻辑扇区号} = & \text{保留扇区数} + \text{每个 FAT 表占用扇区数} \times 2 + \\ & (\text{根目录区域存放目录项数} \times 32) \div \\ & 512 + (\text{簇号} - 2) \times \text{扇区数} / \text{簇} + N - 1 \end{aligned} \quad (5.2)$$

其中: 扇区数/簇、保留扇区数、每个 FAT 表占用扇区数、根目录区域存放的目录项数这 4 个参数直接从 FAT16 文件系统分区引导扇区的 BPB 参数中获得; N 为小于或者等于扇区数/簇的正整数集合。

5.3 FAT32 文件系统

5.3.1 FAT32 文件系统介绍

FAT32 文件系统目前主要用于 U 盘、照相机(或者摄像机)的 SD 卡、手机(或看戏机)的 TF 卡、移动硬盘以及硬盘等外部存储器。

FAT32 文件系统是 FAT 系列文件系统中的一个产品。它采用 32 位的文件分配表，突破了 FAT16 文件系统 2GB 的分区容量限制，增强了磁盘的管理能力。

FAT32 推出时，主流硬盘空间并不大，所以微软公司将其设计在一个不超过 8GB 的分区中，每个簇的大小固定为 4KB，与 FAT16 相比大大减少了磁盘空间的浪费，同时也提高了磁盘空间的利用率。一般情况下，FAT32 文件系统中逻辑盘容量与簇大小的关系大致见表 5.1 所列。

表 5.1 FAT32 文件系统逻辑盘容量与簇大小的关系

逻辑盘的容量	簇的大小
逻辑盘容量≤8GB	4KB(即 8 个扇区)
8GB<逻辑盘容量≤16GB	8KB(即 16 个扇区)
16GB<逻辑盘容量≤32GB	16KB(即 32 个扇区)
逻辑盘容量>32GB	32KB(即 64 个扇区)

支持这种格式的操作系统有 Windows 95、Windows 98、Windows 98 SE、Windows Me、Windows 2000、Windows XP 和 Windows 7，Linux Redhat 部分版本也对 FAT32 提供有限支持。

5.3.2 FAT32 文件系统总体布局

FAT32 文件系统由保留扇区、FAT 表和数据区 3 大部分组成。这 3 个区域是在逻辑盘被(快速)格式化完成后创建的。注：保留扇区由分区引导扇区即 FAT32_DBR、FSINFO、FAT32_DBR 备份和 FSINFO 备份等组成；FAT 表则由 FAT1 表和 FAT2 表组成；FAT32 文件系统的总体布局大致如图 5.2 所示(注：在图 5.2 中， i 和 j 均为正整数， $i=$ 保留扇区数， $j=i+$ 每个 FAT 表占用扇区数 $\times 2$)。



图 5.2 FAT32 文件系统总体布局

FAT32 文件系统与 FAT16 文件系统最大的区别在于：FAT32 文件系统将根目录作为数据区来处理，而 FAT16 文件系统则将根目录作为一块固定的区域来处理。因此，FAT32 文

件系统对根目录的管理与对子目录(文件夹)的管理是一样的。簇的编号也是从数据区开始,开始簇号也是2,也就是根目录的开始簇号。从分区引导扇区到2号簇之间的扇区只能作为逻辑扇区来进行管理。

例 5.1 某硬盘总扇区数为419841(从0至419840号扇区)。在该硬盘0号扇区存储有一个MBR分区表;MBR分区的相对扇区为128,总扇区数为413696,该分区表对应的盘符为J盘。FAT32_DBR位于J盘的0号扇区,从FAT32_DBR可知,保留扇区数为5022,每个簇的扇区数为2,每个FAT表占1585个扇区,J盘总扇区数为413696(扇区编号范围0~413695),文件系统信息(即FSINFO所在扇区)在J盘的1号扇区,FAT32_DBR备份在J盘的6号扇区;整个硬盘扇区号、J盘扇区号与簇号的对应关系见表5.2所列(注:在本章中如果没有作特别说明,J盘是指由素材文件abcd5.vhd通过计算机管理附加后所产生的虚拟硬盘)。

表5.2 整个硬盘扇区号、J盘扇区号与簇号的对应关系表

扇区用途	整个硬盘 扇区号	J盘扇区号	簇号	描述	占用 扇区数	备注
相对扇区	0~127			从分区表至J盘FAT32_DBR的扇区数	128	
保留 扇区	128	0		FAT32_DBR	1	J 盘 占 用 扇 区
	129	1		FSINFO	1	
	130~133	2~5		未用	4	
	134	6		FAT32_DBR备份	1	
	135	7		FSINFO备份	1	
	136~5149	8~5021		未用	5014	
FAT	FAT1表	5150~6734	5022~6606	FAT1表	1585	
	FAT2表	6735~8319	6607~8191	FAT2表	1585	
数据区	8320~8321	8192~8193	2	根目录开始簇号	2	
	8322~8323	8194~8195	3	子目录、根目录的下一个簇号、文件和数据存放位置	2	
	8324~413823	8196~413695	4~202753		405500	
未分区的扇区号	413824~419840				6017	

从表5.2可知,如果FAT32_DBR被破坏,而FAT32_DBR备份完好无损,可以通过菜单栏“工具→打开磁盘→选择Physical Media下的物理盘”来打开整个硬盘,然后再根据整个硬盘扇区号与J盘扇区号的对应关系,将FAT32_DBR备份复制到FAT32_DBR处即可解决。

注:如果FAT32_DBR被破坏,该逻辑盘的文件系统为RAW。在资源管理器中,单击该盘符时,会出现“磁盘未格式化”提示。

切记!此时“千万不要对逻辑盘进行格式化”操作。

5.3.3 FAT32分区引导扇区

FAT32分区引导扇区(简称FAT32_DBR)位于逻辑盘的0号扇区,共计512字节。由跳转指令、空操作指令、OEM ID、BPB(BIOS Parameter Block)参数、扩展BPB参数、分区引导记录、结束标志7部分组成。分区引导扇区各部分划分情况见表5.3所列。

表 5.3 FAT32_DBR 中各部分的位置划分

字节偏移	字段长度(单位:字节)	字段含义
0X00	2	跳转指令
0X02	1	空操作指令
0X03	8	OEM ID
0X0B	53	BPB 参数
0X40	26	扩展 BPB 参数
0X5A	420	分区引导记录
0X01FE	2	结束标志

FAT32_DBR 前 3 字节必须是合法的、可执行的、基于 x86 的 CPU 指令,一般为跳转指令和空操作指令,跳转指令负责跳转到操作系统引导记录代码的开始位置。

空操作指令后是 8 字节长的 OEM ID,它是一组字符串,OEM ID 一般标识了格式化该分区操作系统的名称和版本号或者是厂商标志。为了保留与 MS-DOS 的兼容性,通常在被 Windows 95 格式化的逻辑盘上,OEM ID 字段出现“MSWIN4.0”;在被 Windows 95 OSR2 和 Windows 98 格式化的逻辑盘上,OEM ID 字段出现“MSWIN4.1”;在被 Windows 2000 以上版本格式化的逻辑盘上,OEM ID 字段出现“MSDOS 5.0”;在被厂商格式化的逻辑盘上会标识为厂商标志。

扇区偏移 0X0B~0X59 描述的是能够被可执行引导记录找到相关参数的信息。通常称为 BPB 参数,BPB 一般开始于相同的字节偏移量,标准的参数处于一个已知的位置。BPB 参数包含了该逻辑盘的一些基本参数,主要包括:每扇区的字节数、每簇的扇区数、FAT 表的个数、每个 FAT 表占用扇区数等。由于引导扇区的第一部分是一个 x86 跳转指令,因此将来通过在 BPB 末端附加新的信息,可以对 BPB 进行扩展。只需要对该跳转指令作一个小的调整就可以适应 BPB 的扩展。表 5.4 和表 5.5 分别给出了 FAT32_DBR 的 BPB 参数和扩展 BPB 参数含义。

表 5.4 FAT32_DBR 的 BPB 字段

字节偏移	字节数	含 义
0X0B	2	字节数/扇区
0X0D	1	扇区数/簇
0X0E	2	保留扇区数,即 FAT32_DBR 至 FAT1 表之间的扇区
0X10	1	FAT 表的个数,一般为 2,即有 2 个 FAT 表
0X11	2	对于 FAT32 分区为 00,即未用
0X13	2	对于 FAT32 分区为 00,即未用
0X15	1	媒体描述符,0XF8 表示硬盘、U 盘、SD 卡等
0X16	2	对于 FAT32 分区为 00,即未用
0X18	2	扇区数/磁道
0X1A	2	磁头数
0X1C	4	隐藏扇区数,即分区表至 FAT32_DBR 之间的扇区,注:该值的正确性系统一般不做检验
0X20	4	FAT32 文件系统的总扇区数,该值等于 MBR 分区表中的总扇区数
0X24	4	每个 FAT 表占用扇区数
0X28	2	扩展标志
0X2A	2	文件系统版本
0X2C	4	根目录的开始簇号,一般为 2
0X30	2	文件系统信息扇区号,位于 FAT32_DBR 之后,一般为 1
0X32	2	FAT32_DBR 备份所在扇区号,一般为 6
0X34	12	保留,供以后扩充使用的保留空间

表 5.5 FAT32_DBR 的扩展 BPB 字段

字节偏移	字节数	含义
0X40	1	物理驱动器号,与 BIOS 物理驱动器号有关,软盘驱动器被标识为 0X00,物理硬盘被标识为 0X80,而与物理磁盘驱动器无关;一般地,在发出一个 INT13H BIOS 调用之前设置该值,具体指定所访问的设备;只有当该设备是一个引导设备时,这个值才有意义
0X41	1	保留,FAT32 分区总是将本字段的值设置为 00
0X42	1	扩展引导标签,本字段必须要有能被 Windows 2000 所识别的值 0X28 或 0X29
0X43	4	卷标序号,在格式化磁盘后所产生的一个随机序号
0X47	11	FAT12 的卷标,现在的卷标保存在根目录中,一般为“NO NAME”
0X52	8	系统 ID,FAT32 文件系统中一般取为“FAT32”

分区引导记录主要功能是:将逻辑盘中存储的操作系统主要文件调入到内存中;而有效结束标志则表示该扇区为有效扇区,固定值为“55 AA”(存储形式)。

例 5.2 J 盘 FAT32 文件系统分区引导扇区,即 FAT32_DBR,如图 5.3 所示。



图 5.3 J 盘的 FAT32_DBR

J 盘 FAT32_DBR 的 BPB 参数见表 5.6 所列, 扩展 BPB 参数见表 5.7 所列。

表 5.6 J 盘 FAT32_DBR 中 BPB 字段

字节偏移	字节数	值			含 义
		十进制	十六进制	存储形式	
0X00	2			EB 58	跳转指令
0X02	1			90	空操作指令
0X03	8			4D 53 44 4F 53 35 2E 30	OEM ID, 即厂商标志或者操作系统版本号
0X0B	2	512	0200	00 02	字节数/扇区
0X0D	1	2	2	02	扇区数/簇
0X0E	2	5022	139E	9E 13	保留扇区数
0X10	1	2	2	02	FAT 表的个数
0X11	2	00	00	00 00	对于 FAT32 分区, 本字段必须为 00
0X13	2	00	00	00 00	对于 FAT32 分区, 本字段必须为 00
0X15	1			F8	0XF8 表示硬盘、U 盘、SD 卡等
0X16	2	00	00	00 00	对于 FAT32 分区, 本字段必须为 00
0X18	2	63	3F	00 3F	扇区数/磁道
0X1A	2	255	FF	FF 00	磁头数
0X1C	4	128	80	80 00 00 00	隐藏扇区数
0X20	4	413696	65000	00 50 06 00	总扇区数
0X24	4	1585	631	31 06 00 00	每个 FAT 表占用扇区数
0X28	2	0	0	00 00	扩展标志
0X2A	2	0	0	00 00	文件系统版本
0X2C	4	0	2	02 00 00 00	根目录的开始簇号, 一般为 2
0X30	2	1	1	01 00	文件系统信息所在扇区号, 一般为 1
0X32	2	6	6	06 00	FAT32_DBR 备份所在扇区号, 一般为 6
0X34	12				保留, 供以后扩充使用的保留空间

表 5.7 J 盘 FAT32_DBR 中扩展 BPB 字段

字节偏移	字节数	值			含 义
		十进制	十六进制	存储形式	
0X40	1	128	80	80	物理驱动器号
0X41	1			0	保留, 总为 00
0X42	1		29	29	扩展引导标签
0X43	4	914769280	36864580	80 45 86 36	卷标序号
0X47	11				一般为“NO NAME”
0X52	8				一般为“FAT32”

使用 WinHex 软件的模块管理器读取 J 盘 FAT32_DBR 参数, 结果如图 5.4 所示。

Boot Sector FAT32(Chinese), Base Offset: 0		
Offset	Title	Value
0	JMP 指令	EB 58
2	空操作指令	90
3	OEM	MSDOS5.0
BIOS 参数		
B	每个扇区字节数	512
D	每个簇的扇区数	2
E	保留扇区数	5022
10	FAT 的个数(一般为2)	2
11	(未使用)	0
13	(未使用)	0
15	介质描述(hex)	F8
16	未用	0
18	每个磁道的扇区数	63
1A	磁头数	255
1C	隐藏扇区数	128
20	总扇区数	413696
FAT32 Section		
24	每个FAT表所占扇区数	1585
28	扩展标志	0
2A	标志(通常为0)	0
2C	根目录的簇号	2
30	FSInfo所在扇区号	1
32	DBR备份所在扇区号	6
34	(保留)	00 00 00 00
40	BIOS 驱动器(hex, HD=8x)	80
41	(未用)	0
42	扩展引导标记	29
43	卷的系列号(十进制)	914769280
43	卷的系列号(十六进制)	80 45 86 36
47	卷标	NO NAME
52	文件系统	FAT32
1FE	青妙结束标志	55 AA
模块制作:陈培德(云南大学)		

图 5.4 FAT32_DBR 的 BPB 参数

5.3.4 计算 FAT32 有关参数

在使用 FAT32 文件系统的逻辑盘时,常常会碰到计算逻辑盘的一些参数,其公式如式(5.3)~式(5.6):

$$\text{数据区占扇区数} = \text{逻辑盘总扇区数} - (\text{保留扇区数} + \text{每个FAT表占用扇区数} \times 2) \quad (5.3)$$

$$\text{逻辑盘总容量} = \text{逻辑盘总扇区数} \times 512 \text{ 字节} \quad (5.4)$$

$$\text{剩余扇区数} = \text{mod}(\text{数据区占扇区数}, \text{每个簇的扇区数}) \quad (5.5)$$

$$\text{格式化后逻辑盘容量} = (\text{数据区占扇区数} - \text{剩余扇区数}) \times 512 \text{ 字节} \quad (5.6)$$

注: 在式(5.3)至式(5.6)中,每个扇区的字节数为 512。

其中: 逻辑盘总扇区数、保留扇区数和每个FAT表占用扇区数,可以从逻辑盘FAT32_DBR 的BPB参数中获得。逻辑盘总扇区数也就是该逻辑盘所对应分区的总扇区数,mod 表示取余数。

用户使用FAT32文件系统对逻辑盘进行格式化后,报告的总容量是指数据区减去剩余扇区后的容量,而不是指整个逻辑盘的容量。

簇号转换成逻辑扇区号公式如式(5.7):

$$\text{逻辑扇区号} = \text{保留扇区数} + \text{每个FAT表占用扇区数} \times 2 + (\text{簇号} - 2) \times \text{每个簇的扇区数} + N - 1 \quad (5.7)$$

在式(5.7)中: 保留扇区数、每个FAT表占用扇区数、每个簇的扇区数直接从FAT32_DBR 的BPB参数中获得; N 为小于或者等于每个簇的扇区数的正整数集合; 簇号 ≥ 2 。

逻辑扇区号转换成簇号公式如式(5.8):

$$\text{簇号} = 2 + \text{INT}[(\text{逻辑扇区号} - \text{保留扇区数} - \text{每个FAT表占用扇区数} \times 2) \div \text{每个簇的扇区数}] \quad (5.8)$$

在式(5.8)中: INT 表示取整,逻辑扇区号 \geq 保留扇区数+每个FAT表占用扇区数 $\times 2$

计算FAT1表、FAT2表在逻辑盘中的位置以及根目录开始扇区号(即2号簇开始扇区号)在逻辑盘中位置公式如式(5.9)~式(5.15):

$$\text{FAT32_DBR 所在扇区号} = 0 \quad (5.9)$$

$$\text{保留扇区数} = \text{FAT1 表开始扇区号} - \text{FAT32_DBR 所在扇区号} \quad (5.10)$$

$$\text{FAT1 表结束扇区号} = \text{保留扇区数} + \text{每个FAT表占用扇区数} - 1 \quad (5.11)$$

$$\text{FAT2 表开始扇区号} = \text{保留扇区数} + \text{每个 FAT 表占用扇区数} \quad (5.12)$$

$$\text{FAT2 表结束扇区号} = \text{保留扇区数} + \text{每个 FAT 表占用扇区数} \times 2 - 1 \quad (5.13)$$

$$\begin{aligned} \text{每个 FAT 表占用扇区数} &= \text{FAT2 表开始扇区号} - \text{FAT1 表开始扇区号} \\ &= \text{根目录开始扇区号} - \text{FAT2 表开始扇区号} \end{aligned} \quad (5.14)$$

$$\text{根目录开始扇区号} = \text{保留扇区数} + \text{每个 FAT 表占用扇区数} \times 2 \quad (5.15)$$

说明：在式(5.7)~式(5.15)中，扇区号均为单独打开逻辑盘后的扇区号。根目录开始扇区号也就是2号簇的开始扇区号。

例 5.3 在例 5.2 中，从 FAT32_DBR 可知，保留扇区数为 5022，每个簇的扇区数等于 2，每个 FAT 表占用 1585 个扇区，逻辑盘总扇区数为 413696。请回答下列问题：

- (1) 计算 FAT1 表和 FAT2 表在 J 盘中的开始扇区号和结束扇区号。
- (2) 计算 J 盘格式化后的容量，即在资源管理器通过查看 J 盘属性的总容量。
- (3) 计算 2 号簇(即 J 盘根目录开始簇号)所对应 J 盘的逻辑扇区号。
- (4) 计算 8416 号扇区所对应 J 盘中的簇号。

(5) 由于 J 盘每个簇的扇区数为 2，通过每个 FAT 表占用扇区数，计算 J 盘可以表示的最大簇号和最小簇号。

- (6) 计算 J 盘可以表示的最大磁盘空间和最小磁盘空间。

解：

- (1) 由于保留扇区数为 5022，而每个 FAT 表占用 1585 个扇区

由式(5.10)和式(5.11)可知：

$$\text{FAT1 表开始扇区号} = \text{保留扇区数} = 5022$$

$$\begin{aligned} \text{FAT1 表结束扇区号} &= \text{保留扇区数} + \text{每个 FAT 表占用扇区数} - 1 \\ &= 5022 + 1585 - 1 = 6606 \end{aligned}$$

由式(5.12)和式(5.13)可知：

$$\begin{aligned} \text{FAT2 表开始扇区号} &= \text{保留扇区数} + \text{每个 FAT 表占用扇区数} \\ &= 5022 + 1585 = 6607 \end{aligned}$$

$$\begin{aligned} \text{FAT2 表结束扇区号} &= \text{保留扇区数} + \text{每个 FAT 表占用扇区数} \times 2 - 1 \\ &= 5022 + 1585 \times 2 - 1 = 8191 \end{aligned}$$

所以，FAT1 表在 J 盘中的位置为 5022~6606 号扇区；而 FAT2 表在 J 盘中的位置为 6607~8191 号扇区。

- (2) 由式(5.3)可知：

$$\begin{aligned} \text{J 盘数据区占扇区数} &= \text{J 盘总扇区数} - (\text{保留扇区数} + \text{每个 FAT 占用扇区数} \times 2) \\ &= 413696 - (5022 + 1585 \times 2) \\ &= 405504 \end{aligned}$$

由式(5.5)可知：

$$\begin{aligned} \text{剩余扇区数} &= \text{mod}(\text{J 盘数据区占扇区数}, \text{每个簇的扇区数}) \\ &= \text{mod}(405504, 2) = 0 \end{aligned}$$

由式(5.6)可知：

$$\begin{aligned} \text{J 盘格式化后的容量} &= (\text{J 盘数据区占扇区数} - \text{剩余扇区数}) \times 512 \text{ 字节} \\ &= (405504 - 0) \times 512 \text{ 字节} \\ &= 207\ 618\ 048 \text{ 字节} \\ &= 198\text{MB} \end{aligned}$$

从 J 盘的属性可知, J 盘格式化后的容量为 198MB, 如图 5.5 所示。

(3) 由于扇区数/簇 = 2, 所以, $N = \{1, 2\}$ 。

当 $N=1$ 时, 由式(5.7)可知:

$$\begin{aligned}\text{逻辑扇区号} &= \text{保留扇区数} + \text{每个 FAT 表占用扇区数} \times 2 + \\&\quad (\text{簇号} - 2) \times \text{扇区数 / 簇} + N - 1 \\&= 5022 + 1585 \times 2 + (2 - 2) \times 2 + 1 - 1 \\&= 5022 + 3170 + 0 + 0 \\&= 8192\end{aligned}$$



图 5.5 逻辑盘 J 的属性

当 $N=2$ 时, 由式(5.7)可知, 逻辑扇区号 = 8193;

所以, 2 号簇所对应的扇区号为 8192 和 8193, 共计 2 个扇区。

(4) 由式(5.8)可知:

$$\begin{aligned}\text{簇号} &= \text{INT}[(\text{逻辑扇区号} - \text{保留扇区数} - \text{每个 FAT 表占用扇区数} \times 2) \div \text{扇区数 / 簇}] + 2 \\&= \text{INT}[(8416 - 5022 - 1585 \times 2) \div 2] + 2 \\&= \text{INT}(224 \div 2) + 2 = 114\end{aligned}$$

即 8416 号扇区所对应的簇号为 114。

同理可以验证, 8417 号扇区所对应的簇号也为 114。

(5) 由于 J 盘每个 FAT 表占 1585 个扇区, 而每个扇区可以存储 128 个簇号项。

J 盘的 FAT 表可以存储的最多簇号数 = $1585 \times 128 = 202880$, 因此最大簇号为 202879;

由于每个 FAT 表占 1585 个扇区, 而每个扇区可以存储 128 个簇号项。J 盘的 FAT 表可以表示的最小簇号为最后一个扇区只使用了一个簇号项(即 4 字节), 所以, J 盘可以存储的最少簇号数 = $(1585 - 1) \times 128 + 1 = 202753$, 因此最小簇号为 202752。

(6) 由于 J 盘的最大簇号为 202879, 而 FAT32 文件系统中 0 号簇和 1 号簇均未被数据区使用, 所以 J 盘簇号范围为 2~202879, 共计 202878 个簇。

$$\text{最大磁盘空间} = 202878 \text{ 簇} \times 2 \text{ 扇区数 / 簇} \times 512 \text{ 字节 / 扇区} = 198.123 \text{ MB}$$

由于 J 盘的最小簇号为 202752, 而 FAT32 文件系统中 0 号簇和 1 号簇均未被数据区使用, 所以 J 盘簇号范围为 2~202752, 共计 202751 个簇。

$$\text{最小磁盘空间} = 202751 \text{ 簇} \times 2 \text{ 扇区数 / 簇} \times 512 \text{ 字节 / 扇区} = 197.999 \text{ MB}$$

因此, J 盘可以表示的最大磁盘空间为 198.123MB, 最小磁盘空间为 197.999MB。

5.3.5 FAT32 FSINFO 信息扇区

FAT32 在保留扇区中增加了一个 FSINFO 扇区, 存储于逻辑盘的 1 号扇区, 用以记录文件系统中空闲簇的数量以及下一个可用簇等信息, 供操作系统参考, FSINFO 扇区结构见表 5.8 所列。

表 5.8 FSINFO 信息扇区的结构

字节偏移	字节数	含义
0X0000	4	扩展引导标志“RRaA”
0X0004	480	未用, 全 00
0X01E4	4	FSINFO 签名标志“rrAa”

续表

字节偏移	字节数	含义
0X01E8	4	空闲簇,用以计算该逻辑盘可用空间
0X01EC	4	下一个可用簇号(注:该值不一定正确)
0X01F0	14	未使用
0X01FE	4	标志“55 AA”

例 5.4 J 盘的 FSINFO 扇区信息如图 5.6 所示。

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0AFF0000	52	52	61	41	00	00	00	00-00	00	00	00	00	00	00	00	RRaA.....
	... (全 00)															
0AFF01E0	00	00	00	00	72	72	41	61- 8A 46 02 00	02	00	00	00rrAa..E!			
0AFF01F0	00	00	00	00	00	00	00	00-00	00	00	00	00	00	55	AA	U.

图 5.6 逻辑盘的 FSINFO 属性

从图 5.6 显示的信息可知,J 盘的下一个可用簇为 0X00000002,即 2; 空闲簇为 0X0002468A,即 149130,由此可以计算出 J 盘的可用空间=149130×512 字节=152709120B=145MB; 从 J 盘的属性可知,J 盘的可用空间与通过 FSINFO 扇区信息计算的可用空间相等。注:有时从该扇区所得到的信息不一定准确,但并不影响逻辑盘的正常使用,该扇区中的数据只能作为一种参考。

通常情况下,在逻辑盘的 7 号扇区有一个 FSINFO 信息备份,但该备份与 1 号扇区没有进行同时操作。因此,该扇区的信息不一定总是正确的,但这并不影响 FAT32 文件系统的正常使用。

5.3.6 FAT 表

在 FAT32 文件系统中,文件(夹)内容存储在数据区,并以簇为单位进行分配;而 FAT32 文件系统的文件分配表(简称 FAT 表)则以簇号的形式记录着 FAT32 文件系统数据区的使用状况。

FAT 表通常有两个(一般称为 FAT1 表和 FAT2 表),这两个 FAT 表的内容和长度完全相同。每个 FAT 表占用多少个扇区取决于逻辑盘的大小和每个簇的扇区数。在 FAT 表中,不同的值有着不同的含义,表 5.9 给出了 FAT 表中 XX 号簇项取值的含义。

表 5.9 FAT 表中 XX 号簇项取值含义(十六进制)

XX 号簇项		对应簇号项的含义
取值	取值在 FAT 中的存储形式	
0FFFFF8	F8 FF FF 0F	FAT 表的开始标志
00000000	00 00 00 00	未分配的簇
00000003~0FFFFFFE	03 00 00 00~FE FF FF 0F	已分配的簇,其值为存储文件后续内容下一个簇的簇号
0FFFFFF0~0FFFFFF6	F0 FF FF 0F~F6 FF FF 0F	系统保留
0FFFFFF7	F7 FF FF 0F	坏簇
0FFFFFF8~0FFFFFFF	F8 FF FF 0F~FF FF FF 0F	文件或目录结束簇
0FFFFFFF	FF FF FF 0F	文件或目录结束簇一般用该值

FAT 表具有下列特性：

(1) 数据区对应簇号的状态保存在 FAT 表的相应位置；数据区每个簇号在 FAT 表中的位置，称之为 XX 号簇项，XX 号簇项所存储的内容称之为 XX 号簇项值；XX 号簇项值占用 4 字节(即 32 位)，存储形式采用小头位序。

(2) 0 号簇项和 1 号簇项系统保留，注：实际上是不存在的簇号。

(3) FAT 表以 0FFFFFF8(存储形式为 F8 FF FF 0F)开头，即 0 号簇项值。

(4) 1 号簇项值一般为 FFFFFFFF 或者 0FFFFFFF 或者 7FFFFFFF，对应的存储形式分别为 FF FF FF FF 或者 FF FF FF 0F 或者 FF FF FF 7F。

(5) 由于 FAT32 文件系统数据区是从 2 号簇开始，2 号簇被根目录占据；2 号簇项值如果是 0FFFFFFF，说明根目录只占据 1 个簇的位置，即 2 号簇；2 号簇项值如果不是 0FFFFFFF，则说明根目录占据了 2 个以上的簇号。

(6) 如果 1 个文件(夹)占据了 2 个以上的簇号，在 FAT 表中，该文件(夹)所占据的簇号将使用链表的形式链接。

(7) 如果数据区 XX 号簇未被分配，则 FAT 表中 XX 号簇项值为 0(存储形式为 00 00 00 00)。

(8) 如果数据区 XX 号簇已被分配，即 XX 号簇项值不再是 0，如果对应 XX 号簇项的值是 0FFFFFFF(存储形式为 FF FF FF 0F)，表示 XX 号簇为某文件(夹)的最后一个簇号。

(9) 如果数据区 XX 号簇已被分配，即 XX 号簇项值不再是 0，也不是 0FFFFFFF，那么对应 XX 号簇项值为某文件(夹)的下一个簇号。

(10) 如果对逻辑盘进行格式化时，发现有坏扇区，则该扇区所对应的簇号在 FAT 表中内容填为 F7FFFFFF(存储形式为 FF FF FF F7)。

例 5.5 J 盘的 FAT 表前 96 字节如图 5.7 所示(注：XX 号簇项用十六进制表示)。

	00 号簇项	01 号簇项	02 号簇项	03 号簇项
00273C00	F8 FF FF OF	FF FF FF FF	B7 D4 00 00	FF FF FF OF
	04 号簇项	05 号簇项	06 号簇项	07 号簇项
00273C10	FF FF FF OF	FF FF FF OF	07 00 00 00	FF FF FF OF
	08 号簇项	09 号簇项	0A 号簇项	0B 号簇项
00273C20	FF FF FF OF	0A 00 00 00	0B 00 00 00	0C 00 00 00
	0C 号簇项	0D 号簇项	0E 号簇项	0F 号簇项
00273C30	0D 00 00 00	0E 00 00 00	0F 00 00 00	10 00 00 00
	10 号簇项	11 号簇项	12 号簇项	13 号簇项
00273C40	11 00 00 00	12 00 00 00	13 00 00 00	14 00 00 00
	14 号簇项	15 号簇项	16 号簇项	17 号簇项
00273C50	FF FF FF OF	16 00 00 00	17 00 00 00	18 00 00 00

图 5.7 J 盘的 FAT 表前 96 字节(注：图中的数据均为十六进制)

J 盘 FAT 表前 96 字节说明如下。

(1) 0 号簇项值为 0X0FFFFFF8(存储形式为 F8 FF FF 0F)，表示 FAT 表的开始。

(2) 1 号簇项值为 0xFFFFFFFF(存储形式为 FF FF FF FF)。

(3) 2 号簇项值为 0X0000D4B7(存储形式为 B7 D4 00 00)，即根目录的下一个簇号为 0X0000D4B7。

(4) 3 号簇项、4 号簇项、5 号簇项和 8 号簇项值均为 0X0FFFFFFF(存储形式为 FF FF FF 0F)；表示这 4 个簇分别被小于或者等于 1 个簇的 4 个文件(夹)占用。

(5) 6号簇项值为0X07(存储形式为07 00 00 00);表示它的下一个簇号为0X07,7号簇项值为0X0FFFFFFF(存储形式为FF FF FF 0F),表示某文件(夹)已结束,即某文件(夹)占用2个簇,即6号簇和7号簇,6号簇和7号簇在FAT表中以线性链表的形式链接。

(6) 0X09~0X14号簇项为某一文件(夹)的分配表,即该文件(夹)共占用了0X0C个簇(即12个簇);该文件(夹)的开始簇号为0X09(即9号簇),而结束簇号为0X14(即20号簇),0X09号簇项值是0000000A,即下一个簇是0X0A号簇(即10号簇);而在0X0A号簇(即10号簇)值是0000000B,即下一个簇是0X0B号簇(即11号簇),以此类推,直到0X14号簇项(即20号簇)值是0X0FFFFFFF(存储形式为FF FF FF 0F),表示该文件(夹)已经结束,分配链表如图5.8所示(注:图5.8中的数据均为十六进制)。如果该文件(夹)被彻底删除,该链表将被填充为00000000,表示这12个簇号未被分配。

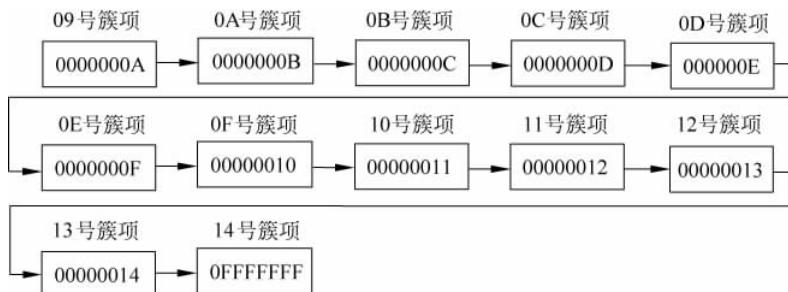


图5.8 某文件或子目录(文件夹)的文件分配链表(图中数据均为十六进制)

计算簇号项在FAT表中的扇区号如式(5.16):

$$\text{簇号项在FAT表中的扇区号} = \text{INT}(\text{簇号项}/128) + 1 \quad (5.16)$$

计算簇号项在FAT表中的扇区偏移和扇区号如式(5.17)~式(5.19):

$$\text{簇号项在FAT表中的扇区偏移} = 4 \times \text{MOD}(\text{簇号项}, 128) + N - 1 \quad (5.17)$$

$$\text{簇号项在逻辑盘中的扇区号(针对FAT1表)} = \text{保留扇区数} + \text{INT}(\text{簇号项}/128) \quad (5.18)$$

$$\text{簇号项在逻辑盘中的扇区号(针对FAT2表)} = \text{保留扇区数} + \text{每个FAT占扇区数} + \text{INT}(\text{簇号项}/128) \quad (5.19)$$

注:在式(5.16)~式(5.19)中,INT表示取整,MOD表示取余数,N为小于或者等于4的正整数集合,即N={1,2,3,4}。

由于FAT32文件系统有两个FAT表,式(5.18)给出了簇号项在逻辑盘中的扇区号(针对FAT1表),式(5.19)给出了簇号项在逻辑盘中的扇区号(针对FAT2表)。

例5.6 在例5.4中,计算518号簇项分别在J盘FAT1表和FAT2表中的位置。

解:

由式(5.16)可知:

$$\text{簇号项在FAT表的扇区号} = \text{INT}(\text{簇号项}/128) + 1 = \text{INT}(518/128) + 1 = 5$$

由式(5.17)可知:

$$\text{簇号项在FAT表中的扇区偏移} = 4 \times \text{MOD}(\text{簇号项}, 128) + N - 1 = \{24, 25, 26, 27\}$$

即518号簇项在FAT表的第5扇区,扇区偏移为24~27(即0X18~0X1B)。将光标移动到FAT表的第5扇区偏移为24~27(即0X18~0X1B)处,即可找到第518簇号项的位置,注:FAT表是指FAT1表和FAT2表。

从 J 盘的 FAT32_DBR 可知,保留扇区数为 5022,每个 FAT 表占用扇区数为 1585;由式(5.18)可知:

$$\begin{aligned} 518 \text{ 号簇在 J 盘的逻辑扇区号(针对 FAT1 表)} &= \text{保留扇区数} + \text{INT(簇号项 / 128)} \\ &= 5022 + \text{INT}(518/128) = 5026 \end{aligned}$$

由式(5.19)可知:

$$\begin{aligned} 518 \text{ 号簇在 J 盘的逻辑扇区号(针对 FAT2 表)} &= \text{保留扇区数} + \text{FAT 表所占扇区数} + \\ &\quad \text{INT(FAT 簇号 / 128)} \\ &= 5022 + 1585 + \text{INT}(518/128) = 6611 \end{aligned}$$

所以,518 号簇项在 J 盘的逻辑扇区号为 5026(针对 FAT1 表)和 6611(针对 FAT2 表),而扇区偏移地址均为 24~27(即 0X18~0X1B)处。

注: 在 WinHex 中,可以通过“菜单栏→位置→Go To FAT Entry……”,在弹出的窗口中输入 FAT 表项的值(即簇号值),即可将光标移动到 FAT 表项处,但只能将光标移动到 FAT1 表项处,如果要对 FAT 表进行操作,还必须对 FAT2 表进行操作,这一点要特别注意。

5.3.7 FAT32 目录项

FAT32 文件系统目录项按用途可以划分为 5 种,即卷标(注:有的逻辑盘没有卷标)、短文件(夹)名、长文件(夹)名、“.”目录项和“..”目录项。目录项以 32 字节为分配单位,卷标、短文件(夹)名、“.”目录项和“..”目录项长度是 32 字节,而长文件(夹)名的长度则是 32 字节的倍数。

1. 卷标

卷标是逻辑盘的一个标志,是用户在逻辑盘格式化前直接输入;或者在资源管理器中通过重命名的方式添加。卷标存储在逻辑盘根目录中,其属性为“08”,卷标在“我的电脑”或“资源管理器”逻辑盘的盘符前会显示出来。如果本地逻辑盘没有卷标,在“我的电脑”或“资源管理器”逻辑盘的盘符前会显示“本地磁盘”;U 盘或移动硬盘没有卷标,在“我的电脑”或“资源管理器”U 盘或移动硬盘的盘符前会显示“可移动磁盘”。

2. 短文件(夹)名

短文件(夹)名是指文件名的命名规则是 8.3 格式,即短文件(夹)名由主文件名和扩展名两部分组成,主文件名的长度为 1~8 个字符,而扩展名的长度为 0~3 个字符。主文件名和扩展名可以使用的字符为 52 个大小英文字母,0~9 这十个数字,除“\ : * ? < > |”外的其他专用符号。每个目录项的长度为 32 字节。主文件名不足 8 个字符、扩展名不足 3 个字符时用空格填充。短文件(夹)名具体的定义见表 5.10 所列。

说明:

(1) 主文件名与扩展名:偏移地址为 0X00~0X0A,对于 8.3 格式的短文件名,将文件名分为两部分存储,即主文件名和扩展名。偏移地址 0X00~0X07 字节存储主文件名;偏移地址 0X08~0X0A 存储扩展名;不存储主文件名与扩展名之间的“.”。主文件名不足 8 个字符、扩展名不足 3 个字符则以空格(ASCII 码值为 0X20)填充。

偏移地址 0X00 处的取值若为 0X00,表示该目录项为空,在资源管理器中显示的目录到

此结束；若为 0XE5，表示目录项曾经被使用，但对应的文件(夹)已被删除。

表 5.10 FAT32 短文件目录项 32 字节的表示定义

字节偏移(十六进制)	字节数	定 义
00~07	8	主文件(夹)名，以大写字母的形式存储
08~0A	3	扩展名，以大写字母的形式存储
0B	1	属性，见说明(2)
0C	1	文件(夹)名大/小字母显示标志，见说明(3)
0D	1	创建时间，精确到 10ms 的值
0E~0F	2	创建时间，包括时、分、秒
10~11	2	创建日期，包括时、分、秒
12~13	2	最后访问日期，包括年、月、日
14~15	2	文件(夹)开始簇号的高 16 位
16~17	2	最近修改时间，包括时、分、秒
18~19	2	最近修改日期，包括年、月、日
1A~1B	2	文件(夹)开始簇号的低 16 位
1C~1F	4	文件实际大小，单位：字节

(2) 属性：偏移地址为 0X0B，该目录项描述文件或文件夹的属性见表 5.11 所列。

表 5.11 文件或文件夹属性描述

偏移地址	0X0B							
二进制位	bit ₇	bit ₆	bit ₅	bit ₄	bit ₃	bit ₂	bit ₁	bit ₀
二进制值	0	0	1	1	1	1	1	1
含义	保留	保留	存档	子目录	卷标	系统	隐藏	只读

从表 5.11 可知，该字节的最高两位(即 bit₇ 和 bit₆)保留、未用，bit₅~bit₀ 分别是存档位、子目录位、卷标位、系统位、隐藏位和只读位。对应位为“1”时表示真，为“0”时表示假；属性值也可以进行组合。

例如：如果某文件的属性值为 0X27(对应二进制数为 0010 0111)，则表示该文件属性为存档、系统、隐藏和只读。

(3) 文件(夹)名大/小字母显示标志：偏移地址为 0X0C，该目录项描述文件(夹)名在资源管理器中是以大写字母还是以小写字母显示。在该字节中只使用了两位，即 bit₄ 和 bit₃，对应位为“1”时表示真(即以小写字母显示)，为“0”时表示假(即以大写字母显示)；其他位的值可以是“0”，也可以是“1”，详见表 5.12 所列。

表 5.12 文件(夹)名大/小写显示标志

偏移地址	0X0C							
二进制位	bit ₇	bit ₆	bit ₅	bit ₄	bit ₃	bit ₂	bit ₁	bit ₀
二进制值	0	0	0	1	1	0	0	0
含义	保留	保留	保留	扩展名 以小写字母显示	主文件(夹)名 以小写字母显示	保留	保留	保留

说明：

□ bit₄=0 且 bit₃=1：主文件名以小写字母显示，而扩展名则以大写字母显示；

□ bit₄=0 且 bit₃=0：主文件名和扩展名均以大写字母显示；

□ bit₄=1 且 bit₃=0：主文件名以大写字母显示，而扩展名则以小写字母显示；

□ bit₄=1 且 bit₃=1：主文件名和扩展名均以小写字母显示。

(4) 创建时间(10 毫秒)：偏移地址为 0X0D，以 10 毫秒为单位。

(5) 创建时间(时：分：秒)：偏移地址为 0X0E~0X0F，这两个字节共 16 位被划分成 3 个部分，其含义见表 5.13 所列。

表 5.13 文件创建时间

偏移地址	0X0F								0X0E							
	bit ₁₅	bit ₁₄	bit ₁₃	bit ₁₂	bit ₁₁	bit ₁₀	bit ₉	bit ₈	bit ₇	bit ₆	bit ₅	bit ₄	bit ₃	bit ₂	bit ₁	bit ₀
二进制位	时								分							
含义																

说明：

bit₄~bit₀ 表示秒，以 2 秒为单位，有效值为 0~29，可以表示 0~58 秒；

bit₁₀~bit₅ 表示分，有效值为 0~59，可以表示 0~59 分；

bit₁₅~bit₁₁ 表示时，有效值为 0~23，可以表示 0~23 时；

(6) 创建日期：偏移地址为 0X10~0X11，这两个字节共 16 位被划分成 3 个部分，其含义见表 5.14 所列。

表 5.14 文件创建日期

偏移地址	0X11								0X10							
	bit ₁₅	bit ₁₄	bit ₁₃	bit ₁₂	bit ₁₁	bit ₁₀	bit ₉	bit ₈	bit ₇	bit ₆	bit ₅	bit ₄	bit ₃	bit ₂	bit ₁	bit ₀
二进制位	年								月							
含义																

说明：

bit₄~bit₀ 表示日，有效值与对应的月份有关，有效值分别为：2 月份为 1~28(不闰月)或 1~29(闰月)，1、3、5、7、8、10 和 12 月份为 1~31，4、6、9 和 11 月份为 1~30；

bit₈~bit₅ 表示月，有效值为 1~12；

bit₁₅~bit₉ 表示年，有效值为 0~127，真正的年份为该值再加上 1980，即表示年份为 1980—2107 年。

(7) 最后访问日期：偏移地址为 0X12~0X13，其描述与创建日期相同，见表 5.14 所列。

(8) 文件(子目录)开始簇号：偏移地址为 0X14~0X15 存放文件(目录)开始簇号高 16 位，偏移地址为 0X1A~0X1B 存放文件(目录)开始簇号低 16 位。

开始簇号=[0X15][0X14][0X1B][0X1A]

(9) 最后修改时间：偏移地址为 0X16~0X17，其描述与创建时间相同，见表 5.13 所列。

(10) 最后修改日期：偏移地址为 0X18~0X19，其描述与创建日期相同，见表 5.14 所列。

(11) 文件实际大小：偏移地址为 0X1C~0X1F，存储形式为小头位序，单位：字节；即文件实际大小=[0X1F][0X1E][0X1D][0X1C]字节。注：子目录(文件夹)、“.”“..”和卷标该值为 0。

计算文件所占簇数、文件结束簇号和占用空间如式(5.20)~式(5.22):

$$\text{文件所占簇数} = \text{ROUNDUP}(\text{文件实际大小} / (\text{每个簇的扇区数} \times 512), 0) \quad (5.20)$$

文件结束簇号 = 文件开始簇号 + 文件所占簇数 - 1 (注: 假设文件内容是连续存储的) (5.21)

$$\text{文件占用空间} = \text{文件所占簇数} \times \text{每个簇的扇区数} \times 512 \quad (5.22)$$

注：在式(5.20)中，ROUNDUP 为向上舍入函数，每个扇区等于 512 字节，每个簇的扇区数可以从 FAT32_DBR 中的 BPB 参数中获得。

例 5.7 J 盘根目录下某文件目录项的存储形式如图 5.9 所示(注: 图 5.9 方框中的部分), 请回答下列问题:

- (1) 写出该文件名在 J 盘根目录中的存储形式, 在资源管理器中查看到的文件名。
 - (2) 该文件目录项偏移 0X0B 的值是 0X20, 该值描述的是什么类型的文件?
 - (3) 写出该文件创建的时间和日期。
 - (4) 写出该文件修改的时间和日期。
 - (5) 写出该文件内容的开始簇号和实际大小。
 - (6) 计算该文件所占簇数、结束簇号和占用空间(注: 该文件内容是连续存储)。
 - (7) 画出该文件 FAT 表的链表结构图。
 - (8) 计算出该文件开始簇号项和结束簇号项分别在 FAT1 表和 FAT2 表中的位置。
 - (9) 如果将该文件删除并清空回收站, 该文件目录项、FAT1 表和 FAT2 表有何变化?

图 5.9 J 盘根目录下某文件名项的存储形式

解：

(1) 该文件名在 J 盘根目录中的存储形式为“41 30 31 20 20 20 20 20 44 4F 43”，所以，该文件的名称为“A01.DOC”；文件目录项偏移 0X0C 的值为 0X18，在资源管理器中主文件名和扩展名均以小写字母显示，即用户查看到的文件名称为“a01.doc”。

(2) 由于文件目录项偏移 0X0B 的值是 0X20, 该值描述的是存档文件。

(3) 文件目录项偏移 0X0D、0X0E 和 0X0F 的值分别为 0X69、0X22 和 X04C。

由于偏移地址 0X0D 的值为 0X69, 等于十进制数的 105, 表示 1050 毫秒, 即 1.050 秒。

偏移地址[0XOE]=0X22,[0XOF]=0X4C,共16位被划分成3个部分,见表5.15所列。

表 5.15 文件创建时间

偏移地址	0X0F								0X0E							
十六进制	4C								22							
二进制位	bit ₁₅	bit ₁₄	bit ₁₃	bit ₁₂	bit ₁₁	bit ₁₀	bit ₉	bit ₈	bit ₇	bit ₆	bit ₅	bit ₄	bit ₃	bit ₂	bit ₁	bit ₀
二进制	0	1	0	0	1	1	0	0	0	0	1	0	0	0	1	0
十进制	9				33								2			
含义	时				分								秒			
表示的时间	9				33								4			

注：由于偏移地址 0X0E 值的 Bit₄～Bit₀ 表示的是秒，以 2 秒为单位；而偏移地址 0X0D 值也是表示秒，单位为 10 毫秒，值 0X69 表示 1050 毫秒(即 1.050 秒)。所以，a01.doc 文件的创建时间为 9:33:05 秒。

偏移地址[0X10]=0X73,[0X11]=0X46,这两个字节共 16 位被划分成 3 个部分,见表 5.16 所列。所以,a01.doc 文件的创建日期为 2015 年 3 月 19 日。

表 5.16 文件创建日期

偏移地址	0X11							0X10								
十六进制	46							73								
二进制位	bit ₁₅	bit ₁₄	bit ₁₃	bit ₁₂	bit ₁₁	bit ₁₀	bit ₉	bit ₈	bit ₇	bit ₆	bit ₅	bit ₄	bit ₃	bit ₂	bit ₁	bit ₀
二进制	0	1	0	0	0	1	1	0	0	1	1	1	0	0	1	1
十进制	35							3			19					
含义	年							月			日					
表示的日期	1980 + 35 = 2015							3			19					

(4) 偏移地址[0X16]=0X88,[0X17]=0X52,这两个字节共 16 位被划分成 3 个部分,见表 5.17 所列。所以,a01.doc 文件的最近修改时间为 10 : 20 : 16。

表 5.17 文件最近修改时间

偏移地址	0X17								0X16							
十六进制	52								88							
二进制位	bit ₁₅	bit ₁₄	bit ₁₃	bit ₁₂	bit ₁₁	bit ₁₀	bit ₉	bit ₈	bit ₇	bit ₆	bit ₅	bit ₄	bit ₃	bit ₂	bit ₁	bit ₀
二进制	0	1	0	1	0	0	1	0	1	0	0	0	1	0	0	0
十进制	10				20								8			
含义	时				分								秒			
表示的时间	10				20								16			

偏移地址[0X18]=0X5E,[0X19]=0X43,这两个字节共16位被划分成3个部分,见表5.18所列。所以,a01.doc文件最近修改日期为2013年10月30日。

表 5.18 文件最近修改日期

偏移地址	0X19								0X18							
十六进制	43								5E							
二进制位	bit ₁₅	bit ₁₄	bit ₁₃	bit ₁₂	bit ₁₁	bit ₁₀	bit ₉	bit ₈	bit ₇	bit ₆	bit ₅	bit ₄	bit ₃	bit ₂	bit ₁	bit ₀
二进制	0	1	0	0	0	0	1	1	0	1	0	1	1	1	1	0
十进制	33								10				30			
含义	年								月				日			
表示的日期	1980+33=2013								10				30			

(5) 由于文件目录项偏移 $[0X14]=0X00$ 、 $[0X15]=0X00$ 、 $[0X1A]=0X87$ 、 $[0X1B]=0XD3$,所以, a01.doc 文件内容开始簇号= $[0X15][0X14][0X1B][0X1A]=0X0000D387$ (即 54151)。

文件目录项偏移 $0X1C\sim0X1F$ 的值分别为 $0X00$ 、 $0X40$ 、 $0X04$ 和 $0X00$,所以 a01.doc 文件实际大小= $[0X1F][0X1E][0X1D][0X1C]=0X00044000$,即 278528 字节。

(6) 由于 a01.doc 文件内容在 J 盘是连续存储,而 J 盘每个簇的扇区数为 2,每个扇区等于 512 字节,由式(5.20)可知:

$$\begin{aligned} \text{a01.doc 文件所占簇数} &= \text{ROUNDUP}(\text{文件实际大小} / (\text{每个簇的扇区数} \times 512), 0) \\ &= \text{ROUNDUP}(278528 / (2 \times 512)) = 272 \end{aligned}$$

由式(5.21)可知:

$$\text{a01.doc 文件结束簇号} = \text{文件开始簇号} + \text{文件所占簇数} - 1 = 54151 + 272 - 1 = 54422$$

由式(5.22)可知:

$$\begin{aligned} \text{a01.doc 文件占用空间} &= \text{文件所占簇数} \times \text{每个簇的扇区数} \times 512 \\ &= 272 \times 2 \times 512 = 278528 \text{ 字节} \end{aligned}$$

(7) 由于 a01.doc 文件内容是连续存储,开始簇号为 54151(即 0XD387),结束簇号为 54422(即 0XD496),所以,a01.doc 文件分配表的链表结构图如图 5.10 所示(注: 图 5.10 中的数据均为十六进制)。

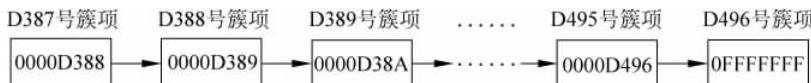


图 5.10 a01.doc 文件的 FAT 表

(8) 从 J 盘 FAT32_DBR 可知,保留扇区数等于 5022,每个 FAT 表占用扇区数为 1585;

由于 a01.doc 文件内容开始簇号为 54151,由式(5.18)和式(5.19)可知:

$$\begin{aligned} 54151 \text{ 号簇项在 J 盘的逻辑扇区号(针对 FAT1 表)} &= \text{保留扇区数} + \text{INT}(\text{簇号项} / 128) \\ &= 5022 + \text{INT}(54151 / 128) = 5445 \end{aligned}$$

$$\begin{aligned} 54151 \text{ 号簇项在 J 盘的逻辑扇区号(针对 FAT2 表)} &= \text{保留扇区数} + \text{每个 FAT 表占扇区数} + \\ &\quad \text{INT}(\text{簇号项} / 128) \\ &= 5022 + 1585 + \text{INT}(54151 / 128) \\ &= 7030 \end{aligned}$$

由式(5.17)可知:

$$\begin{aligned} 54151 \text{ 号簇项在 FAT 表的扇区偏移} &= 4 \times \text{MOD}(\text{簇号项}, 128) + N - 1 \\ &= 4 \times \text{MOD}(54151, 128) + N - 1 = \{28, 29, 30, 31\} \end{aligned}$$

所以,54151 号簇项在 FAT1 表的位置为 5445 号扇区,扇区偏移为 28~31(即 0X1C~0X1F); 54151 号簇项在 FAT2 表的位置为 7030 号扇区,扇区偏移为 28~31(即 0X1C~0X1F)。

同理,可以计算 54422 号簇项在 FAT1 表的位置为 5447 号扇区,扇区偏移为 88~91(即 0X58~0X5B); 54422 号簇项在 FAT2 表的位置为 7032 号扇区,扇区偏移为 88~91(即 0X58~0X5B)。

(9) 如果将 a01.doc 文件删除并清空回收站,目录项如图 5.11 所示,即 a01.doc 目录项首字节的 ASCII 码由 0X41 变为 0XE5。

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00400380	EF	30	31	20	20	20	20	44	4F	43	20	18	69	22	4C	21	DOC.i" L
00400390	73	46	74	46	00	00	88	52	5E	43	87	D3	00	40	04	00	sFtF..压`C囉@.

图 5.11 删除 J 盘根目录下 a1.doc 文件后文件名目录项的存储形式

而在 FAT1 表和 FAT2 表从 54151 号簇项至 54422 号簇项的链表值被填充为“000000”，即在 J 盘位置为 5445 号扇区（即在 FAT1 表的位置）偏移 28（即 0X1C）至 5447 号扇区（即在 FAT1 表的位置）偏移 91（即 0X5B）的值被填充为“00”；而 J 盘位置为 7030 号扇区（即在 FAT2 表的位置）偏移 28（即 0X1C）至 7032 扇区（即在 FAT2 表的位置）偏移 91（即 0X5B）的值也被填充为“00”；文件内容即完好无损地保存着。

3. 长文件(夹)名

FAT32 一个重要的特点是支持长文件名。为了兼容低版本的 DOS 系统或程序能正确读取长文件名文件，系统会自动为所有长文件名文件创建一个对应的短文件名，使文件内容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 DOS 系统或程序会将长文件名忽略，认为是不合法的文件名字段，而支持长文件名的操作系统或程序则以长文件名来对文件内容进行存取、编辑等操作，并隐藏其短文件名。当用户对文件(夹)进行命名时，如果出现下列情况之一或者是 3 种情况之间的相互组合，在目录中将会存储两个文件名，一个以长文件名的形式存储，而另一个则以短文件名的形式存储。

【情况 1】 在文件(夹)名中同时包括大、小写英文字母；

【情况 2】 以汉字作为文件(夹)名；

【情况 3】 主文件(夹)名的长度超过 8 个字符或扩展名的长度超过 3 个字符。

当创建一个长文件名的文件后，系统会在长文件之后自动生成一个相应的短文件名目录项，短文件命名规则一般如下：

(1) 取长文件名的前 6 个字符加上“~1”作为对应的短文件名，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 4 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) 如果存在老操作系统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 0X0B 的属性字节，当该字节的属性值为 0X0F（即只读、隐藏、系统、卷标属性的组合， $0X0F = 0X01 + 0X02 + X04 + X08$ ），DOS 或 WIN32 会认为其不合法而忽略其存在。Windows 9X 或 Windows 2000、Windows 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。所以可能一个长文件名需要多个目录项，这时长文件名的各个目录项按倒序排列在目录中，以防与其他文件名混淆。

长文件名中的字符采用 Unicode 码形式编码，每个字符占据 2 字节的空间，其目录项定义见表 5.19 所列。

表 5.19 FAT32 长文件目录项 32 字节表示的含义

字节偏移	字节数	含 义
0X00	1	长文件名序列号,一个文件的第一个长文件名序列号为 01,然后依次递增。如果是该文件的最后一个长文件名目录项,则将该目录项序号与 0X40 进行“或运算”的结果写入该位置。如果长文件名目录项对应的文件或子目录被删除,则将该字节设置成删除标志 0XE5。
0X01~0X0A	10	长文件名的第 1~5 个字符(Unicode 码),未使用的部分填充两个字节“00”,然后用 0xFF 填充。
0X0B	1	长文件名目录项标志,取值 0X0F。
0X0C	1	系统保留
0X0D	1	校验值和,如果一个文件的长文件名需要几个长文件名目录项进行存储,则这些长文件名目录项具有相同的校验和。
0X0E~0X19	12	长文件名的第 6~11 个字符(Unicode 码),未使用的部分填充两个字节“00”,然后用 0xFF 填充。
0X1A~0X1B	2	保留
0X1C~0X1F	4	长文件名的第 12~13 个字符(Unicode 码),未使用的部分填充两个字节“00”,然后用 0xFF 填充。

FAT32 文件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 5.19 可知,长文件名中并不存储对应文件的开始簇号、文件大小、各种时间和日期属性,文件的这些属性存储在对应短文件名目录项中,一个长文件名总是与其相应的短文件名一一对应,短文件名没有对应的长文件名仍然可以进行正常的读/写;但长文件名如果没有对应的短文件名,不管什么系统都将其忽略。所以短文件名对于长文件名是非常重要的。在不支持长文件名的操作系统中对短文件名字段进行改动,都会使长文件名形同虚设。长文件名的字节偏移 0X0D“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;如果通过短文件名计算出来的“校验和”与长文件名中的字节偏移 0X0D 处数据不匹配,则系统不会将它们配对;如果匹配,则依据长文件名和短文件名对目录项的定义,以短文件名的开始簇号和 FAT 表的相应链表,就可以对该文件进行访问。

例 5.8 J 盘根目录下某文件目录项存储形式如图 5.12 所示(注:图 5.12 中方框部分)。

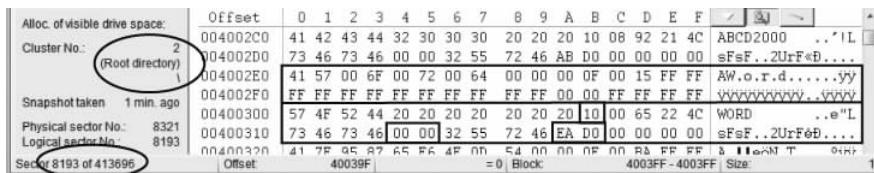


图 5.12 J 盘根目录下 Word 文件夹的存储形式

说明:

(1) 用户对文件夹名的命名规则属于【情况 1】，该目录项描述的是一个文件夹，文件夹名称存储在根目录下，用户在资源管理器中查看到的文件夹名为 Word。

(2) 在目录项中，存在有两个文件夹名，一个描述的是长文件夹名，另一个描述的是短文件夹名。

(3) 短文件夹名称为 WORD, 短文件夹名以 ASCII 码的形式存储, 属性代码为 0X10, 即文件夹, 开始簇号为 0X0000D0EA。

(4) 长文件夹名称为 Word, 长文件夹名以 Unicode 码存储, 即“W”“o”“r”和“d”的 Unicode 码分别为 0057、006F、0072 和 0064(注: 存储形式为小头位序)。即用户在资源管理器中查看到的文件夹名称为 Word, 由于长文件名的存在, 用户不能在资源管理器中查看到对应的短文件夹名。

例 5.9 J 盘根目录下某目录项的存储形式如图 5.13 所示(注: 在 J 盘的 8193 号扇区)。

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00400320	41	7F	95	87	65	F6	4E	0D	54	00	00	0F	00	BA	FF	FF
00400330	FF	00	00	FF	FF	FF	FF	..								
00400340	B3	A4	CE	C4	BC	FE	C3	FB	20	20	20	10	00	67	22	4C
00400350	73	46	73	46	00	00	32	55	72	46	53	D3	00	00	00	00

图 5.13 J 盘根目录下长文件名文件夹目录项的存储形式

说明:

(1) 用户对文件夹名的命名规则属于【情况 2】，该目录项描述的是一个文件夹，文件夹存储在根目录下，用户在资源管理器中查看到的文件夹名为“长文件名”。

(2) 在该目录项中，存在两个文件夹名，一个描述的是长文件夹名，另一个描述的是短文件夹名。

(3) 短文件夹名称为“长文件名”，短文件夹名以 ASCII 码存储，即“长”“文”“件”和“名”的 ASCII 码分别为 A4B3、C4CE、FEB3 和 FBC3(注：存储形式为小头位序)。属性代码为 0X10，即文件夹，开始簇号为 0X0000D353。

(4) 长文件夹名称为“长文件名”，长文件夹名称以 Unicode 码存储，即“长”“文”“件”和“名”的 Unicode 码分别为 957F、6587、4EF6 和 540D(注：存储形式为小头位序)。

例 5.10 在 J 盘“长文件名”文件夹下存储的文件名为 abcdefghijklmnopqrstuvwxyz0123456789abcdefghijklmnopqrstuvwxyz01234567890.doc 的目录项，存储形式如图 5.14 所示(注：存储在 J 盘的 116386 号扇区)。

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
038D4440	46	33	00	34	00	35	00	36	00	37	00	0F	00	8E	38	00
038D4450	39	00	30	00	2E	00	64	00	6F	00	00	00	63	00	00	00
038D4460	05	71	00	72	00	73	00	74	00	75	00	0F	00	8E	76	00
038D4470	77	00	78	00	79	00	7A	00	30	00	00	00	31	00	32	00
038D4480	04	64	00	65	00	66	00	67	00	68	00	0F	00	8E	69	00
038D4490	6A	00	6B	00	6C	00	6D	00	6E	00	00	00	6F	00	70	00
038D44A0	03	30	00	31	00	32	00	33	00	34	00	0F	00	8E	35	00
038D44B0	36	00	37	00	38	00	39	00	61	00	00	00	62	00	63	00
038D44C0	02	6E	00	6F	00	70	00	71	00	72	00	0F	00	8E	73	00
038D44D0	74	00	75	00	76	00	77	00	78	00	00	00	79	00	7A	00
038D44E0	01	61	00	62	00	63	00	64	00	65	00	0F	00	8E	66	00
038D44F0	67	00	68	00	69	00	6A	00	6B	00	00	00	6C	00	6D	00
038D4500	41	42	43	44	45	46	7E	31	44	4F	43	20	00	68	22	4C
038D4510	73	46	73	46	00	00	AD	88	29	46	54	D3	00	2C	00	00

图 5.14 存储 abcdefghijklmnopqrstuvwxyz0123456789abcdefghijklmnopqrstuvwxyz01234567890.doc 文件的目录项

说明：

(1) 用户对文件名的命名规则属于【情况 3】，从文件名目录项的存储形式可知，在长文件名后生成一个短文件名，短文件名以 ASCII 码的形式存储，短文件名称为“ABCDEF~1.DOC”，该文件名项记录了文件的一些基本属性(如：文件创建日期、时间，文件最近修改的日期、时间，文件内容开始簇号，文件大小等)，如图 5.14 中的阴影部分；短文件名在资源管理器中并不显示，所以用户不能在资源管理器中查看到短文件名。

文件开始簇号 = [0X15][0X14][0X1B][0X1A] = 0X0000D354(即 54100)

文件实际大小 = [0X1F][0X1E][0X1D][0X1C] = 0X00002C00(即 11264) 字节

(2) 该文件的长文件名以 Unicode 码的方式存储，以倒序的方式占用 6 个文件目录项，即 192 字节(注：每个文件目录项的长度为 32 字节)。

第 1 个文件目录项以 0X01 开头，存储的名称为 abcdefghijklm；

第 2 个文件目录项以 0X02 开头，存储的名称为 nopqrstuvwxyz；

第 3 个文件目录项以 0X03 开头，存储的名称为 0123456789abc；

第 4 个文件目录项以 0X04 开头，存储的名称为 defghijklmnop；

第 5 个文件目录项以 0X05 开头，存储的名称为qrstuvwxyz012；

第 6 个文件目录项以 0X46 开头，是 0X40 和 0X06 进行“或运算”的结果，是最后一个文件名项，存储的名称为 34567890.doc，后面未使用的文件名部分填充 1 字节的“00”；

(3) 这 6 个长文件名属性标志均为 0XF，校验值为 0XE。

(4) 该文件的长文件名=第 1 个文件名项+第 2 个文件名项+……+第 6 个文件名项，即长文件名为 abcdefghijklmnopqrstuvwxyz0123456789abcdefgijklmnopqrstuvwxyz01234567890.doc，也就是用户在资源管理器中查看到的文件名。

4. “.”目录项与“..”目录项

“.”目录项与“..”目录项是子目录中非常重要的两个目录项，在子目录中总是成对出现，分别位于每个子目录的第 1 个和第 2 个目录项。

“.”目录项的开始簇号就是该子目录在逻辑盘中的开始簇号；“..”目录项的开始簇号则是该子目录上一级目录(即父目录)的开始簇号；如果该目录项的开始簇号是 00，则说明该子目录的上一级目录为根目录。“.”目录项与“..”目录项在 5.3.8 节中还将以实例的形式进一步说明。

注：一般情况下，FAT32 文件系统根目录在数据区的开始簇是 2，但是由于在 FAT12/FAT16 文件系统中，默认为 0 号簇为根目录所在簇号，为了与 FAT12/FAT16 文件系统兼容，所以在 FAT32 文件系统中也就默认为 0 号簇就是根目录所在簇号，特别是在“..”表示上一级目录时，以 00 为根目录开始簇号，而不是以 2 作为根目录的开始簇号。

5.3.8 根目录、子目录与回收站

1. 根目录

用户使用 FAT32 文件系统对逻辑盘格式化后，根目录的开始簇号位于 FAT2 表之后，占一个簇的位置。当根目录下存储的文件(夹)数量不断增加，一个簇不够存放时，FAT32 将会

再申请一个簇，并以线性链表的形式将这两个簇连接起来；当两个簇仍然不够存放时，则再申请一个簇，并以线性链表的形式将这3个簇连接起来，以此类推。根目录的开始簇号可以从FAT32_DBR的BPB参数中获得；一般情况下，根目录的开始簇号为2。如果FAT表第一个扇区偏移0X08~0X0B处（即2号簇项）存储的值是“FF FF FF 0F（存储形式）”，说明根目录只占用一个簇，即2号簇；否则根目录占用两个以上的簇号，从该值可以找到根目录的下一个簇号。

例5.11 J盘FAT表第一扇区前80字节如图5.15所示。

Alloc of visible drive space:	Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F			
Cluster No.: n/a	00273C00	F8	FF	FF	OF	FF	FF	FF	FF	B7	D4	00	00	FF	FF	FF	OF	n/a	FFFF	0
FAT 1 Cluster 2 -> 54455 (Root directory)	00273C10	FF	FF	FF	OF	FF	FF	FF	OF	07	00	00	00	FF	FF	FF	OF	FFFF	FFFF	FFFF
Snapshot taken 16 min. ago	00273C20	FF	FF	FF	OF	0A	00	00	00	0B	00	00	00	0C	00	00	00	00	FFFF	FFFF
Sector 5022 of 413696	00273C30	0D	00	00	00	OE	00	00	00	0F	00	00	00	10	00	00	00	00	FFFF	FFFF
	00273C40	11	00	00	00	12	00	00	00	13	00	00	00	14	00	00	00	00	FFFF	FFFF
		Offset:	273C0A		= 0	Block:												n/a	Size:	n/a

图5.15 J盘的FAT表第1个扇区的前80字节

由于根目录的开始簇号为0X02，而FAT表的1号扇区偏移0X08~0X0B（即2号簇项）存储的值为0X0000D4B7，也就是说，J盘根目录的下一个簇号是0X0000D4B7（十进制值为54455），由式(5.16)可知：

$$\begin{aligned} \text{54455号簇项在FAT表的扇区号} &= \text{INT}(\text{簇号项}/128) + 1 \\ &= \text{INT}(54455/128) + 1 = 426 \end{aligned}$$

由式(5.17)可知：

$$\begin{aligned} \text{54455号簇项在FAT表中的扇区偏移} \\ = 4 \times \text{MOD}(\text{簇号项}, 128) = 4 \times \text{MOD}(54455, 128) + N - 1 = \{220, 221, 222, 223\} \end{aligned}$$

54455号簇项在FAT表的426号扇区偏移220~223（即0X0DC~0X0DF）处；从J盘FAT32_DBR可知，保留扇区数为5022，每个FAT表占用扇区数为1585。

54455号簇项针对FAT1表在J盘的5447号扇区偏移220~223（0X0DC~0X0DF）处，针对FAT2表在J盘的7032号扇区偏移220~223（0X0DC~0X0DF）处。

将光标移动到5447号扇区（或7032号扇区）偏移220~223（即0X0DC~0X0DF）处；其内容如图5.16所示。从图5.16可知，J盘5447号扇区偏移220~223处，其内容为0X0FFFFFFF（存储形式为FF FF FF 0F），表示该链表已经结束。

Cluster No.: n/a	Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F			
FAT 1 Cluster 54455: end (Root directory)	002A8ED0	AD	D4	00	00	AE	D4	00	00	AF	D4	00	00	B0	D4	00	00	-0..	00..	0..*
Snapshot taken 27 min. ago	002A8EC0	B1	D4	00	00	B2	D4	00	00	B3	D4	00	00	B4	D4	00	00	+0..	20..	0..0..
Physical sector No.: 5575	002A8ED0	B5	D4	00	00	B6	D4	00	00	FF	0F	00..	00..	FFFF						
Logical Sector No.: 5447	002A8EE0	B9	D4	00	00	BA	D4	00	00	BB	D4	00	00	BC	D4	00	00	+0..	00..	00..
Sector 5447 of 413696	002A8EF0	BD	D4	00	00	BE	D4	00	00	BF	D4	00	00	CO	D4	00	00	+0..	30..	00..A0..
		Offset:	2A8EDC		= 255	Block:												n/a	Size:	n/a

图5.16 J盘5447扇区偏移地址220~223处

所以，J盘根目录共占用2个簇，即2号簇和54455（即0XD4B7）号簇，J盘根目录的链表结构如图5.17所示。



图5.17 J盘根目录的链表结构图(注：图中的数据均为十六进制)

例 5.12 使用资源管理器查看 J 盘根目录中所存储的文件和文件夹如图 5.18 所示,而使用 WinHex 软件查看 J 盘根目录的存储形式如图 5.19 和图 5.20 所示(注: J 盘根目录存储在 2 号簇和 54455 号簇)。

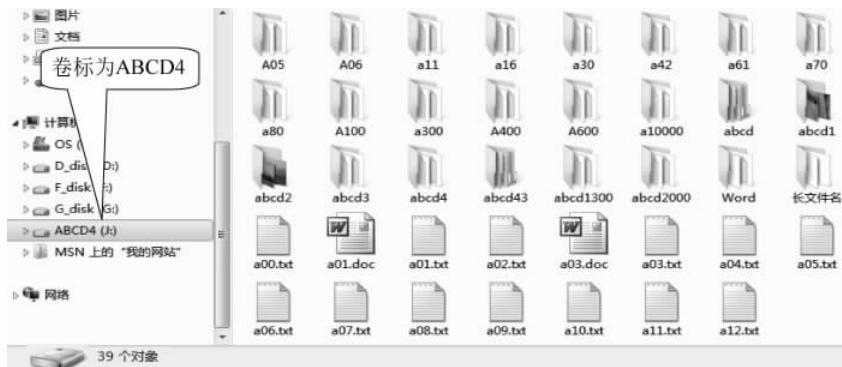


图 5.18 J 盘根目录中所存储的文件和文件夹

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00400000	41	42	43	44	34	20	20	20	20	20	08	00	00	00	00	ABCD4
00400010	00	00	00	00	00	00	07	4C	73	46	00	00	00	00	00	00LsF....
00400020	41	30	35	20	20	20	20	20	20	20	10	00	B7	17	4C	A05	..?L
00400030	73	46	73	46	00	00	80	7D	72	46	03	00	00	00	00	00	sFsF..€}rF....
00400040	41	30	36	20	20	20	20	20	20	20	10	00	BA	17	4C	A06	..?L
00400050	73	46	73	46	00	00	29	55	72	46	04	00	00	00	00	00	sFsF..)UrF.....
00400060	41	31	31	20	20	20	20	20	20	20	10	08	BB	17	4C	A11	..?L
00400070	73	46	73	46	00	00	29	55	72	46	05	00	00	00	00	00	sFsF..)UrF.....
00400080	41	31	36	20	20	20	20	20	20	20	10	08	C1	17	4C	A16	..?L
00400090	73	46	73	46	00	00	29	55	72	46	1E	00	00	00	00	00	sFsF..)UrF.....
004000A0	41	33	30	20	20	20	20	20	20	20	10	08	C2	17	4C	A30	..?L
004000B0	73	46	73	46	00	00	29	55	72	46	20	00	00	00	00	00	sFsF..)UrF.....
004000C0	41	34	32	20	20	20	20	20	20	20	10	08	C4	17	4C	A42	..?L
004000D0	73	46	73	46	00	00	29	55	72	46	21	00	00	00	00	00	sFsF..)UrF!...
004000E0	41	36	31	20	20	20	20	20	20	20	10	08	C7	17	4C	A61	..?L
004000F0	73	46	73	46	00	00	29	55	72	46	23	00	00	00	00	00	sFsF..)UrF#.....
00400100	41	37	30	20	20	20	20	20	20	20	10	08	04	18	4C	A70L
00400110	73	46	73	46	00	00	29	55	72	46	25	00	00	00	00	00	sFsF..)UrF%....
00400120	41	38	30	20	20	20	20	20	20	20	10	08	09	18	4C	A80L
00400130	73	46	73	46	00	00	2A	55	72	46	28	00	00	00	00	00	sFsF..*UrF(.....
00400140	41	31	30	20	20	20	20	20	20	20	10	00	17	18	4C	A100L
00400150	73	46	73	46	00	00	2A	55	72	46	2F	00	00	00	00	00	sFsF..*UrF/.....
00400160	41	33	30	30	20	20	20	20	20	20	10	08	1E	18	4C	A300L
00400170	73	46	73	46	00	00	2A	55	72	46	33	00	00	00	00	00	sFsF..*UrF3.....
00400180	41	34	30	30	20	20	20	20	20	20	10	00	33	18	4C	A400	..3.L
00400190	73	46	73	46	00	00	2A	55	72	46	3D	00	00	00	00	00	sFsF..*UrF=.....
004001A0	41	36	30	30	20	20	20	20	20	20	10	00	4E	18	4C	A600	..N.L
004001B0	73	46	73	46	00	00	2A	55	72	46	4A	00	00	00	00	00	sFsF..*UrFJ.....
004001C0	41	31	30	30	30	20	20	20	20	20	10	08	7F	18	4C	A10000L
004001D0	73	46	73	46	00	00	30	55	72	46	61	00	00	00	00	00	sFsF..0UrFa.....

图 5.19 2 号簇中文件和文件夹的存储形式

004001E0	41 42 43 44 20 20 20 20 20 20 10 08 9D 1D 4C	ABCD ..L	目录项 16
004001F0	73 46 73 46 00 00 30 55 72 46 9A 01 00 00 00 00	sFsF..0UrF?....	
00400200	41 42 43 44 31 20 20 20 20 20 10 08 A1 1D 4C	ABCD1 ..?L	目录项 17
00400210	73 46 73 46 00 00 30 55 72 46 B2 01 00 00 00 00	sFsF..0UrF?....	
00400220	41 42 43 44 32 20 20 20 20 20 10 08 A7 1D 4C	ABCD2 ..?L	目录项 18
00400230	73 46 73 46 00 00 30 55 72 46 F2 0D 00 00 00 00	sFsF..0UrF?....	
00400240	41 42 43 44 33 20 20 20 20 20 10 08 B0 1D 4C	ABCD3 ..?L	目录项 19
00400250	73 46 73 46 00 00 66 55 72 46 D1 1F 00 00 00 00	sFsF..fUrF?....	
00400260	41 42 43 44 34 20 20 20 20 20 10 08 B8 1D 4C	ABCD4 ..?L	目录项 20
00400270	73 46 73 46 00 00 30 55 72 46 FF 2E 00 00 00 00	sFsF..0UrF ..	
00400280	41 42 43 44 34 33 20 20 20 20 20 10 08 BF 20 4C	ABCD43 ..?L	目录项 21
00400290	73 46 73 46 00 00 82 81 29 46 75 D0 00 00 00 00	sFsF..?)Fu?... .	
004002A0	41 42 43 44 31 33 30 30 20 20 20 10 08 09 21 4C	ABCD1300 ...!L	目录项 22
004002B0	73 46 73 46 00 00 31 55 72 46 82 D0 00 00 00 00	sFsF..1UrF 億...	
004002C0	41 42 43 44 32 30 30 30 20 20 20 10 08 92 21 4C	ABCD2000 ..?L	目录项 23
004002D0	73 46 73 46 00 00 32 55 72 46 AB D0 00 00 00 00	sFsF..2UrF 吱..	
004002E0	41 57 00 6F 00 72 00 64 00 00 00 OF 00 15 FF FF	AW.o.r.d.....	
004002F0	FF FF FF FF FF FF FF FF FF 00 00 FF FF FF FF	..	目录项 24
00400300	57 4F 52 44 20 20 20 20 20 20 10 00 65 22 4C	WORD ..e" L	
00400310	73 46 77 46 00 00 32 55 72 46 EA D0 00 00 00 00	sFwF..2UrF 夏....	
00400320	41 7F 95 87 65 F6 4E 0D 54 00 00 OF 00 BA FF FF	A.断e 魟 T....?	
00400330	FF FF FF FF FF FF FF FF FF 00 00 FF FF FF FF	..	目录项 25
00400340	B3 A4 CE C4 BC FE C3 FB 20 20 20 10 00 67 22 4C	长文件名 ..g" L	
00400350	73 46 73 46 00 00 32 55 72 46 53 D3 00 00 00 00	sFsF..2UrFS?...	
00400360	41 30 30 20 20 20 20 20 54 58 54 20 18 68 22 4C	A00 TXT .h" L	目录项 26
00400370	73 46 73 46 00 00 F6 7C 72 46 80 D3 76 18 00 00	sFsF..鮑rF€ 署.	
00400380	41 30 31 20 20 20 20 20 44 4F 43 20 18 69 22 4C	A1 DOC .i" L	目录项 27
00400390	73 46 74 46 00 00 88 52 5E 43 87 D3 00 40 04 00	sFtF..压`C囉@.	
004003A0	41 30 31 20 20 20 20 20 54 58 54 20 18 69 22 4C	A01 TXT .i" L	目录项 28
004003B0	73 46 73 46 00 00 52 50 99 43 97 D4 52 00 00 00	sFsF..RP 機械R.	
004003C0	41 30 32 20 20 20 20 20 54 58 54 20 18 6A 22 4C	A02 TXT .j" L	目录项 29
004003D0	73 46 73 46 00 00 2E 78 FA 3E 00 00 00 00 00 00	sFsF..x?.....	
004003E0	41 30 33 20 20 20 20 44 4F 43 20 18 6A 22 4C	A03 DOC .j" L	目录项 30
004003F0	73 46 77 46 00 00 0C 49 73 42 98 D4 00 7A 00 00	sFwF...IsB 條 z..	

图 5.19 (续)

从图 5.19 和图 5.20 可知,在 J 盘的根目录下共存储 42 个目录项,说明如下:

(1) 目录项 1 名称为 ABCD4,偏移地址 0X0B 的值为 0X08,即 J 盘的卷标为 ABCD4。

(2) 目录项 2 至目录项 23,每个目录项的长度均为 32 字节,目录项偏移 0X0B 的值均为 0X10,即这 22 个目录项均为子目录(即文件夹)。

(3) 目录项 24 和目录项 25,每个目录项的长度均为 64 字节;每个目录项由长文件夹名描述和短文件夹名描述两部分组成;长文件夹名项偏移 0X0B 处的值为 0X0F,而对应短文件夹名项偏移 0X0B 处的值为 0X10,这两个目录均为子目录(即文件夹)。

(4) 目录项 26 至目录项 40,每个目录项的长度均为 32 字节,目录项偏移地址 0X0B 的值均为 0X20,即这 15 个目录项均为存档文件;目录项偏移地址 0X0C 的值均为 0X18,即主文件名和扩展名在资源管理器中均以小写字母显示。

(5) 目录项 41,长度为 32 字节,名称为 \$ RECYCLEBIN,目录项偏移地址 0X0B 的值均

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F		
0392D400	41	30	33	20	20	20	20	54	58	54	20	18	6A	22	4C	A03	TXT .j" L	目录项31
0392D410	73	46	73	46	00	00	2E	78	FA	3E	00	00	00	00	00	sFsF...x?.....		
0392D420	41	30	34	20	20	20	20	54	58	54	20	18	6A	22	4C	A04	TXT .j" L	目录项32
0392D430	73	46	73	46	00	00	2E	78	FA	3E	00	00	00	00	00	sFsF...x?.....		
0392D440	41	30	35	20	20	20	20	54	58	54	20	18	6A	22	4C	A05	TXT .j" L	目录项33
0392D450	73	46	73	46	00	00	2E	78	FA	3E	00	00	00	00	00	sFsF...x?.....		
0392D460	41	30	36	20	20	20	20	54	58	54	20	18	6A	22	4C	A06	TXT .j" L	目录项34
0392D470	73	46	73	46	00	00	2E	78	FA	3E	00	00	00	00	00	sFsF...x?.....		
0392D480	41	30	37	20	20	20	20	54	58	54	20	18	6A	22	4C	A07	TXT .j" L	目录项35
0392D490	73	46	73	46	00	00	2E	78	FA	3E	00	00	00	00	00	sFsF...x?.....		
0392D4A0	41	30	38	20	20	20	20	54	58	54	20	18	6A	22	4C	A08	TXT .j" L	目录项36
0392D4B0	73	46	73	46	00	00	2E	78	FA	3E	00	00	00	00	00	sFsF...x?.....		
0392D4C0	41	30	39	20	20	20	20	54	58	54	20	18	6A	22	4C	A09	TXT .j" L	目录项37
0392D4D0	73	46	73	46	00	00	2E	78	FA	3E	00	00	00	00	00	sFsF...x?.....		
0392D4E0	41	31	30	20	20	20	20	54	58	54	20	18	6A	22	4C	A10	TXT .j" L	目录项38
0392D4F0	73	46	73	46	00	00	2E	78	FA	3E	00	00	00	00	00	sFsF...x?.....		
0392D500	41	31	31	20	20	20	20	54	58	54	20	18	6A	22	4C	A11	TXT .j" L	目录项39
0392D510	73	46	77	46	00	00	D3	5D	45	46	B8	D4	C8	2C	00	00	sFwF..覩EF冈?..	
0392D520	41	31	32	20	20	20	20	54	58	54	20	18	6B	22	4C	A12	TXT .k" L	目录项40
0392D530	73	46	73	46	00	00	0C	5E	45	46	C4	D4	5B	18	00	00	sFsF..^EF脑[...]	
0392D540	24	52	45	43	59	43	4C	45	42	49	4E	16	00	78	42	4C	\$RECYCLEBIN..xBL	目录项41
0392D550	73	46	文件夹、系统、隐藏				CB	D4	00	00	00	00	sFsF..CLsF嘅...					
0392D560	E5	36	23	20	20	20	10	00	8B	2F	4F	2638377	..?	0	0			目录项42
0392D570	73	46	73	删除标志				73	46	CD	D4	00	00	00	00	sFsF..00sF駄...		

图 5.20 54455 号簇中文件和文件夹的存储形式(注: 无用数据已删除)

为 0X16, 即属性为文件夹、隐藏、系统, 在资源管理器中用户不能查看到该文件夹。

(6) 目录项 42, 长度为 32 字节, 名称第 1 字节的 ASCII 码为“E5”, 即表示该目录项已被删除, 用户在资源管理器中不能查看到该文件夹。

(7) 用户在资源管理器中, 只能看到 24 个文件夹、15 个文件和一个卷标。

(8) 根目录下存储的 42 个目录项说明见表 5.20 所列。

表 5.20 J 盘根目录存储文件名和文件夹名情况表

序号	存储名称	名称在 J 盘根目录中的显示方式	类型	属性值	名称显示方式代码	开始簇号	文件所占字节
1	ABCD4	ABCD4	卷标	0X08	0X00		
2	A05	A05	文件夹	0X10	0X00	3	
3	A06	A06	文件夹	0X10	0X00	4	
4	A11	a11	文件夹	0X10	0X08	5	
5	A16	a16	文件夹	0X10	0X08	30	
6	A30	a30	文件夹	0X10	0X08	32	
7	A42	a42	文件夹	0X10	0X08	33	
8	A61	a61	文件夹	0X10	0X08	35	
9	A70	a70	文件夹	0X10	0X08	37	
10	A80	a80	文件夹	0X10	0X08	40	
11	A100	A100	文件夹	0X10	0X00	47	
12	A300	a300	文件夹	0X10	0X08	51	
13	A400	A400	文件夹	0X10	0X00	61	

续表

序号	存储名称	名称在J盘根目录中的显示方式	类型	属性值	名称显示方式代码	开始簇号	文件所占字节
14	A600	A600	文件夹	0X10	0X00	74	
15	A10000	a10000	文件夹	0X10	0X08	97	
16	ABCD	abcd	文件夹	0X10	0X08	410	
17	ABCD1	abcd1	文件夹	0X10	0X08	434	
18	ABCD2	abcd2	文件夹	0X10	0X08	3570	
19	ABCD3	abcd3	文件夹	0X10	0X08	8145	
20	ABCD4	abcd4	文件夹	0X10	0X08	12031	
21	ABCD43	abcd43	文件夹	0X10	0X08	53365	
22	ABCD1300	abcd1300	文件夹	0X10	0X08	53378	
23	ABCD2000	abcd2000	文件夹	0X10	0X08	53419	
24	WORD	Word	文件夹	0X10	0X00	53482	
25	长文件名	长文件名	文件夹	0X10	0X00	54099	
26	A00 TXT	a00.txt	文件	0X20	0X18	54144	6262
27	A01 DOC	a01.doc	文件	0X20	0X18	54151	278528
28	A01 TXT	a01.txt	文件	0X20	0X18	54423	82
29	A02 TXT	a02.txt	文件	0X20	0X18		0
30	A03 DOC	a03.doc	文件	0X20	0X18	54424	31232
31	A03 TXT	a03.txt	文件	0X20	0X18		0
32	A04 TXT	a04.txt	文件	0X20	0X18		0
33	A05 TXT	a05.txt	文件	0X20	0X18		0
34	A06 TXT	a06.txt	文件	0X20	0X18		0
35	A07 TXT	a07.txt	文件	0X20	0X18		0
36	A08 TXT	a08.txt	文件	0X20	0X18		0
37	A09 TXT	a09.txt	文件	0X20	0X18		0
38	A10 TXT	a10.txt	文件	0X20	0X18		0
39	A11 TXT	a11.txt	文件	0X20	0X18	54456	11464
40	A12 TXT	a12.txt	文件	0X20	0X18	54468	6235
41	\$ RECYCLEBIN	\$ RECYCLEBIN	文件夹	0X16	0X00	54475	
42	? 638377_		文件夹	0X10	0X00	54477	

2. 子目录

在逻辑盘的根目录下常常需要创建一些子目录(文件夹)。当在根目录下创建一个子目录后,在根目录空白目录项处就创建了一个子目录项,该子目录项占0字节空间。每个子目录开始簇号前两个目录项的名称分别为“.”和“..”。

“.”目录项位于子目录开始簇号的第1个目录项位置,它的开始簇号也就是该子目录的开始簇号,其他的定义与短文件名定义相同;通过两个子目录的开始簇号和开始扇区号,可以计算逻辑盘每个簇的扇区数,计算公式如式(5.23):

$$\begin{aligned}
 \text{每个簇的扇区数} &= (\text{第2个子目录开始扇区号} - \text{第1个子目录开始扇区号}) \div \\
 &\quad (\text{第2个子目录开始簇号} - \text{第1个子目录开始簇号}) \\
 &= (\text{第1个子目录开始扇区号} - \text{根目录开始扇区号}) \div \\
 &\quad (\text{第1个子目录开始簇号} - \text{根目录开始簇号})
 \end{aligned} \tag{5.23}$$

注:根目录开始簇号为2。

“..”目录项位于子目录开始簇号的第2个目录项位置,用于描述该子目录的父目录的相

关信息。它的开始簇号也就是该子目录的父目录开始簇号,其他的定义与短文件名定义相同。

在子目录中还可以再建立子目录,子目录通过各自的目录项、“.”和“..”目录项的开始簇号完成对目录树型结构的管理。

例 5.13 在 J 盘 abcd 文件夹下“.”与“..”目录项存储形式如图 5.21 所示。

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00466000	2E	20	20	20	20	20	20	20	20	10	00	9D	1D	4CL
00466010	73	46	73	46	00	00	20	4C	73	46	9A	01	00	00	00	sFsF..LsF?....
00466020	2E	2E	20	20	20	20	20	20	20	10	00	9D	1D	4CL
00466030	73	46	73	46	00	00	20	4C	73	46	00	00	00	00	00	sFsF..LsF.....

图 5.21 J 盘 abcd 文件夹存储的“.”与“..”目录项存储形式

说明:

(1) 从图 5.21 可知,两个目录项的属性代码均为 0X10,即属性为文件夹。

(2) 第 1 个目录项的名称为“.”,对应的 ASCII 码为 0X2E; 该目录项的开始簇号为 0X00000019A; 即 abcd 文件夹在 J 盘的开始簇号为 0X00000019A。

(3) 第 2 个目录项的名称为“..”,对应的 ASCII 码为 0X2E、0X2E; 该目录项的开始簇号为 0X000000000,即 abcd 文件夹的上一级文件夹(即根目录)在 J 盘的开始簇号为 0X00; 也就是说 abcd 文件夹是 J 盘根目录下的一个文件夹。

注: J 盘 abcd 文件夹目录项,请读者查看图 5.19 中的目录项 16。

例 5.14 在 J:\abcd\abcd1 文件夹下所存储的“.”与“..”目录项存储形式如图 5.22 所示。

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00466400	2E	20	20	20	20	20	20	20	20	10	00	9E	1D	4C	.	..?L
00466410	73	46	73	46	00	00	20	4C	73	46	9B	01	00	00	00	sFsF.. LsF?....
00466420	2E	2E	20	20	20	20	20	20	20	10	00	9E	1D	4C?L
00466430	73	46	73	46	00	00	20	4C	73	46	9A	01	00	00	00	sFsF.. LsF?....

图 5.22 J:\abcd\abcd1 文件夹存储的“.”与“..”目录项存储形式

说明:

(1) 从图 5.22 可知,两个目录项的属性代码均为 0X10,即属性为文件夹。

(2) 第 1 个目录项的名称为“.”,对应的 ASCII 码为 0X2E; 目录项的开始簇号为 0X00000019B,即 J:\abcd\abcd1 文件夹在 J 盘的开始簇号为 0X00000019B。

(3) 第 2 个目录项的名称为“..”,对应的 ASCII 码为 0X2E、0X2E; 目录项的开始簇号为 0X00000019A; 即 J:\abcd\abcd1 文件夹的上一级文件夹(即 J 盘 abcd 文件夹)的开始簇号为 0X00000019A; 也就是说 J:\abcd\abcd1 文件夹是 J:\abcd 文件夹下的一个文件夹。

3. 回收站

不同的操作系统对回收站的管理不同,这里只介绍 Windows XP 和 Windows 7; 其他操作系统对回收站的管理,请读者查阅有关资料。

在 Windows XP 操作系统下,FAT32 文件系统逻辑盘的回收站是一个特殊的子目录,它位于逻辑盘的根目录下,名称为 Recycled,它的属性值是 0X16,即子目录、系统、隐藏。在回收站中,除了有“.”和“..”目录项外,还有一个名为 INFO2 的文件,该文件属性值为 0X22,即隐

藏、归档；该文件记录了被删除文件的原始信息，即被删除的盘符、路径、长文件(夹)名。如果某目录下的文件被删除，也就是说将要删除的文件放入回收站后，系统将该目录下要删除文件目录第一个字符的 ASCII 码改为 0XE5，并将该文件原来的盘符、路径、文件名以 Unicode 码的形式存储到 INFO2 文件中，并对该文件进行顺序编号；在回收站中创建一个文件，文件名的命名规则为“D”+逻辑盘符+顺序号，扩展名不变，被删除文件在 FAT 表中对应的链表仍然完好。其中：逻辑盘符为被删除文件所在盘符，顺序号为文件在 INFO2 文件中的顺序号。

Windows 7 操作系统对回收站的管理与 Windows XP 不同；在 Windows 7 操作系统下，回收站也是一种特殊的子目录，位于逻辑盘的根目录下，名称为 \$RECYCLE.BIN，它的属性值也是 0X16，即子目录、隐藏、系统。在回收站目录中，除了有“.”和“..”目录项，还有一个名为 desktop.ini 的文件，它的属性值是 0X26，即归档、隐藏、系统。

如果用户将文件删除（即放入回收站）后，在 \$RECYCLE.BIN 文件夹中会创建 2 个文件，两个文件名的命名规则分别为：一个以“\$I+6 个随机字符”为主文件名；而另一个则以“\$R+6 个随机字符”为主文件名，扩展名不变。两个文件的“6 个随机字符”是一样的。“\$I+6 个随机字符”文件的内容存储着被删除文件的日期、时间、盘符、路径和文件名；而“\$R+6 个随机字符”则存储着被删除文件的内容。

5.3.9 删除文件对目录项、回收站等的影响

删除文件是一种常见的操作，删除文件一般有两种方式。

【方式一】 将文件直接删除，即文件不放入回收站；不同磁盘之间的剪切文件也属于这种方式。过程如下：

(1) 将要删除文件目录第一个字符的 ASCII 码改为 0XE5，保留文件开始簇号的低 16 位、文件建立的日期、时间和文件的大小等信息，将文件开始簇号的高 16 位填充为 0000。

(2) 将要删除的文件在 FAT 表所占据的文件分配链表填充为 00000000，而文件的内容仍然保留。

注：由于文件开始簇号的高 16 位被填充为 0000 后，这也正是使用许多数据恢复软件无法恢复被删除文件的原因所在，对于这种情况建议读者使用 DiskGenius 软件来恢复被删除的文件。

【方式二】 将文件放入回收站，然后再将回收站清空；或者在回收站中选择文件再将其删除。由于不同的操作系统对回收站的管理不同，所以，这种方式在不同的操作系统下对回收站的影响也不尽相同。本节重点讨论 Windows XP 和 Windows 7 操作系统。

在 Windows XP 操作系统下，过程如下：

(1) 将要删除文件目录第一个字符的 ASCII 码改为 0XE5。

(2) 将要删除的文件放入回收站后，在回收站文件夹（即 Recycled 文件夹）中创建一个文件，文件名的命名规则为“D+盘符+顺序号+. 扩展名”；而文件建立的日期、时间、开始簇号、文件大小等信息与被删除文件相同。

(3) 在 INFO2 文件中追加两条记录，一条记录是以 ASCII 码的形式存储被删除文件的逻辑盘符、路径、文件名；而另一条记录则是以 Unicode 码的形式存储被删除文件的逻辑盘符、路径、文件名、顺序号、文件删除日期、时间等信息。

(4) 如果将回收站清空，或者在回收站中选择文件再将其删除，则将回收站中以“D+盘符

十顺序号十. 扩展名”为文件名第一个字符的 ASCII 码改为 0XE5；被删除文件在 FAT 表中占据的文件分配链表填充为 00000000，而文件内容仍然保留；将 INFO2 中的两条记录删除。

在 Windows 7 操作系统下，过程如下：

(1) 将要删除文件目录项的第一个字符的 ASCII 码改为 0XE5。

(2) 将要删除的文件放入回收站后，在回收站文件夹中创建 2 个文件，这 2 个文件的命名规则是：一个以“\$ I+6 个随机字符”为文件名；另一个则以“\$ R+6 个随机字符”为文件名，扩展名不变。两个文件的“6 个随机字符”是一样的。其中：“\$ I+6 个随机字符”文件大小为 544 字节，文件内容存储着被删除文件的日期、时间、盘符、路径和文件名；而“\$ R+6 个随机字符”文件的开始簇号、文件大小等信息与被删除文件相同。

(3) 如果将回收站清空，或者在回收站中选择文件再将其删除，则将回收站中的“\$ R+6 个随机字符. 扩展名”和“\$ R+6 个随机字符. 扩展名”文件的第一个字符的 ASCII 码改为 0XE5；将这两个文件所占据的文件分配链表填充为 00000000，而文件的内容仍然保存。

例 5.15 在 Windows 7 操作系统下，假设回收站已空，将 J 盘根目录下的 a03. doc 文件放入回收站，并清空回收站(注：a03. doc 文件内容在 J 盘上是连续存储的)。

1) 对被删除文件目录项的影响

(1) 将 J 盘根目录下的 a03. doc 文件放入回收站前

使用 WinHex 软件查看 J 盘根目录下的 a03. doc 文件的目录项如图 5.23 所示。从图 5.23 可知，a03. doc 文件开始簇号 = 0X0000D498(即 54424)，实际大小为 0X00007A00(即 31232) 字节，由于 FAT32 文件系统是以簇为单位对数据区进行管理，所以 a03. doc 文件内容实际占据了 J 盘的 31744 字节空间，即 31 个簇(注：J 盘每个簇的扇区数=2)。

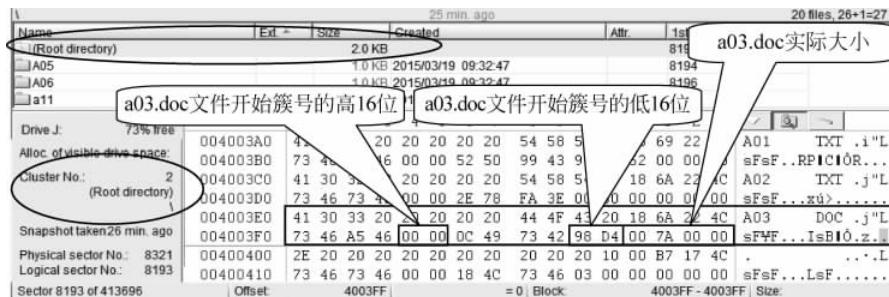


图 5.23 将 a03. doc 文件放入回收站前，J 盘根目录下 a03. doc 文件的目录项存储形式

(2) 将 J 盘根目录下的 a03. doc 文件放入回收站后

使用 WinHex 软件查看 J 盘根目录下 a03. doc 文件的目录项如图 5.24 所示；与图 5.23 比较，J 盘根目录下 a03. doc 文件目录项第 1 个字符的 ASCII 码由 0X41 变为 0XE5，而 a03. doc 目录项的其他内容没有发生变化。由于该文件目录项第 1 个字符的 ASCII 码已经被修改为 0XE5，所以在资源管理器中查看 J 盘的根目录时将不显示该文件名。

(3) 将回收站清空后

J 盘根目录下 a03. doc 文件的目录项与清空回收站前相同，未发生变化。

2) 对 J 盘 \$ RECYCLE. BIN 文件夹的影响

(1) 将 J 盘根目录下 a03. doc 文件放入回收站前

使用 WinHex 软件查看 \$ RECYCLE. BIN 文件夹，所存储的文件如图 5.25 所示。从

Name	Ext.	Size	Created	Attr.	1st sector
(Root directory)		2.0 KB			8192
A05		1.0 KB	2015/03/19 09:32:47		8194
A06		1.0 KB	2015/03/19 09:32:47		8196
a11		1.0 KB	2015/03/19 09:32:47		8198
Drive J:	73% free	Offset	0 1 2 3 4 5 6 7 8 9 A B C D E F		
Alloc of visible drive space:		004003A0	41 30 31 20 20 20 .. 58 54 20 18 69 22 4C	AU1	TXT .i" L
Cluster No.:	2	004003B0	73 46 73 46 00 .. 50 99 43 97 D4 52 00 00 00	sFsF..RPIC1ÖR...	
(Root directory)	1	004003C0	41 30 32 .. 20 20 20 54 58 54 20 18 6A 22 4C	A02	TXT .j" L
		004003D0	73 46 73 46 00 00 00 2E 78 FA 3E 00 00 00 00 00	sFsF...xú).....	
Snapshot taken 26 min. ago		004003E0	E5 30 33 20 20 20 20 44 4F 43 20 18 6A 22 4C	à03	DOC .j" L
Physical sector No.: 8321		004003F0	73 46 45 46 00 00 0C 49 73 42 98 D4 00 7A 00 00	eFVF...IsB1ö.z...	
Logical sector No.: 8193		00400400	2E 20 20 20 20 20 20 20 20 10 00 B7 17 4CL	
Sector 8193 of 413696		00400410	73 46 73 46 00 00 18 4C 73 46 03 00 00 00 00 00	sFsF...LsF.....	
		Offset	4003FF	= 0 Block:	4003FF-4003FF Size:

图 5.24 将 a03.doc 文件放入回收站后, J 盘根目录下 a03.doc 文件的目录项存储形式

图 5.25 可知, 在 J 盘 \$ RECYCLE. BIN 文件夹中, 除了“.”和“..”目录项外, 还存储着 desktop.ini 文件。

Name	Ext.	Size	Created	Attr.	1st sector
长文件名		1.0 KB	2015/03/19 09:33:05		116386
1\$639377		1.0 KB	2015/03/19 09:57:31		117142
\$RECYCLE.BIN	BIN	1.0 KB	2015/03/19 09:34:05	SH	117138
a01.doc	doc	272 KB	2015/03/19 09:33:05	A	116490
Drive J:	73% free	Offset	0 1 2 3 4 5 6 7 8 9 A B C D E F		
Alloc of visible drive space:		03932400	2E 20 20 20 20 20 20 20 20 20 10 00 78 42 4C	.	..xBL
Cluster No.:	54475	03932410	73 46 73 46 00 .. 43 4C 73 46 CB D4 00 00 00	sFsF..CLsFEÖ....	
\$RECYCLE.BIN		03932420	2E 2E 20 20 20 20 20 20 20 20 10 00 78 42 4CxBL
		03932430	73 46 73 46 00 .. 43 4C 73 46 00 00 00 00 00	sFsF..CLsF.....	
Snapshot taken 12 min. ago		03932440	44 45 53 4B 54 4F 50 20 49 4E 49 26 18 7A 42 4C	DESKTOP INI&.zBL	
Physical sector No.: 117266		03932450	73 46 73 46 00 .. 43 4C 73 46 CC D4 B1 00 00 00	sFsF..CLsFiö....	
Logical sector No.: 117138		03932460	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
Sector 117138 of 413696		03932470	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	n/a Size:	
		Offset	39324CF	= 0 Block:	n/a

图 5.25 a03.doc 文件放入回收站前, J 盘 \$ RECYCLE. BIN 文件夹所存储的文件

(2) 将 J 盘根目录下的 a03.doc 文件放入回收站后

使用 WinHex 软件查看 \$ RECYCLE. BIN 文件夹, 所存储的文件如图 5.26 所示。与图 5.25 比较, 在 \$ RECYCLE. BIN 文件夹下添加了两文件, 一个文件名为 \$ I6BZFCT. doc, 另一文件名为 \$ R6BZFCT. doc; \$ I6BZFCT. doc 开始簇号 = 0X10002, 实际大小为 0X220 字节; \$ R6BZFCT. doc 开始簇号 = 0XD498, 实际大小为 0X7A00 字节。也就是说, 文件 \$ R6BZFCT. doc 的开始簇号、实际大小与删除 a03.doc 前的开始簇号、实际大小完全相同。

Name	Ext.	Size	Created	Attr.	1st sector
\$I6bzfcf.doc	doc	0.5 KB	2015/05/05 16:19:20	A	139254
\$R6bzfcf.doc	doc	30.5 KB	2015/03/19 09:33:05	A	117036
desktop.ini	ini	129 B	2015/03/19 09:34:05	SHA	117140
Drive J:	73% free	Offset	0 1 2 3 4 5 6 7 8 9 A B C D E F		
File system:	FAT32	03932400	2E 20 20 20 20 20 20 20 20 20 10 00 78 42 4C	.	..xBL
Volume label:	ABCD4	03932410	73 46 73 46 00 .. 43 4C 73 46 CB D4 00 00 00	sFsF..CLsFEÖ....	
Alloc of visible drive space:		03932420	2E 20 20 20 20 20 20 20 20 20 10 00 78 42 4CxBL
Cluster No.:	54475	03932430	73 46 73 46 00 .. 43 4C 73 46 00 00 00 00 00	sFsF..CLsF.....	
\$RECYCLE.BIN		03932440	44 45 53 4B 54 4F 50 20 49 4E 49 26 18 7A 42 4C	DESKTOP INI&.zBL	
		03932450	73 46 73 46 00 .. 43 4C 73 46 CC D4 B1 00 00 00	sFsF..CLsFiö....	
Snapshot taken 1 min. ago		03932460	24 49 36 42 5A 46 43 54 44 4F 43 20 10 57 6A 82	\$I6BZFCTDOC .Wj1	
		03932470	A5 46 A5 46 01 00 6B 82 A5 46 02 00 20 02 00 00	WFVF..k1WF ..	
Physical sector No.: 117266		03932480	24 52 36 42 5A 46 43 54 44 4F 43 20 10 6A 22 4C	\$R6BZFCTDOC .j" L	
Logical sector No.: 117138		03932490	73 46 A5 46 00 00 OC 49 73 42 98 D4 00 7A 00 00	sFVF...IsB1ö.z...	
Sector 117138 of 413696		039324A0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
		Offset	39324CF	= 0 Block:	n/a

图 5.26 a03.doc 文件放入回收站后, J 盘 \$ RECYCLE. BIN 文件夹所存储的文件

使用 WinHex 软件打开 J 盘 \$ RECYCLE. BIN 文件夹下的 \$ I6BZFCT. doc 文件, 其内容如图 5.27 所示。从图 5.27 可知, 在 \$ I6BZFCT. doc 文件中存储着被删除文件大小, 被删除

文件的日期和时间,被删除文件的盘符、路径和文件名的 Unicode 码等信息。

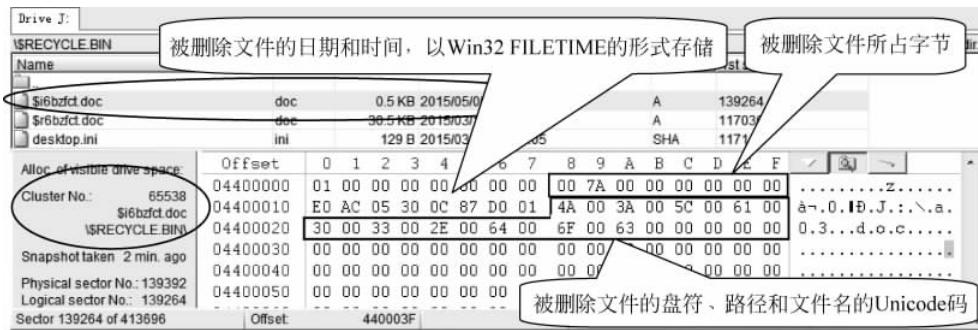


图 5.27 \$I6BZFC.doc 文件的内容

在资源管理器中查看回收站文件夹,其内容如图 5.28 所示,从回收站中所看到的盘符、路径、文件名、删除日期和文件大小均是 \$I6BZFC.doc 文件中所存储的内容。



图 5.28 在回收站文件夹中所查看到被删除的文件 a03.doc

(3) 将回收站清空后

J 盘 \$ RECYCLE. BIN 文件夹所存储的文件如图 5.29 所示。从图 5.29 可知,J 盘 \$ RECYCLE. BIN 文件夹中的 \$I6BZFC.doc 和 \$R6BZFC.doc 文件名项首字母的 ASCII 码已经变为 0XE5,表示这两个文件已从 J 盘 \$ RECYCLE. BIN 中删除。

Name	Ext.	Size	Created	Attr.	1st sector
..					
\$I6bzfc.doc	doc	0.5 KB	2015/05/05 16:19:20	A	8192
\$R6bzfc.doc	doc	30.5 KB	2015/03/19 09:33:05	A	117036
desktop.ini	ini	129 B	2015/03/19 09:34:05	SHA	11740
Drive J:	73% free				
File system:	FAT32				
Volume label:	ABC04				
Alloc. of visible drive space:					
Cluster No.:	54475				
\$RECYCLE.BIN					
Snapshot taken:	1 min. ago				
Physical sector No.:	117266				
Logical sector No.:	117138				
Sector 117138 of 413696					
Offset:	0 1 2 3 4 5 6 7 8 9 A B C D E F				
03932400	2E 20 20 20 20 20 20 20 20 10 00 78 42 4C				
03932410	73 46 73 46 00 00 43 4C 73 46 CB D4 00 00 00 00				
03932420	2E 2E 20 20 20 20 20 20 20 20 10 00 78 42 4C				
03932430	73 46 73 46 00 00 43 4C 73 46 00 00 00 00 00 00				
03932440	44 45 53 4B 54 4F 50 20 49 4E 49 26 18 7A 42 4C				
03932450	73 46 AB 46 00 00 43 4C 73 46 CC D4 81 00 00 00				
03932460	E5 49 36 42 5A 46 43 54 44 4F 43 20 10 57 6A 82				
03932470	A5 46 A7 46 00 00 6B 82 A5 46 02 00 20 02 00 00				
03932480	E5 52 36 42 5A 46 43 54 44 4F 43 20 10 6A 22 4C				
03932490	73 46 A5 46 00 00 0C 49 73 42 98 D4 00 7A 00 00				
039324A0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00				
Offset:	39324AF	= 0, Block:	39324AF - 39324AF	Size:	1

图 5.29 清空回收站后,J 盘 \$ RECYCLE. BIN 文件夹中所存储的文件

3) 对 FAT1 表和 FAT2 表的影响

(1) 将 J 盘根目录下的 a03.doc 文件放入回收站前

通过 a03.doc 文件名目录项可知,a03.doc 文件内容的开始簇号为 54424,占 31232 字节空间。由式(5.16)可知:

$$\begin{aligned}
 \text{54424号簇项在FAT表的扇区号} &= \text{INT}(簇号项 / 128) + 1 \\
 &= \text{INT}(54424 / 128) + 1 \\
 &= 425
 \end{aligned}$$

由式(5.17)可知：

$$\begin{aligned}
 \text{54424号簇项在FAT表中的扇区偏移地址} &= 4 \times \text{MOD}(簇号项, 128) + N - 1 \\
 &= 4 \times \text{MOD}(54424, 128) + N - 1 \\
 &= \{96, 97, 98, 99\}
 \end{aligned}$$

54424号簇项在FAT表的第425号扇区，扇区偏移为96~99(即0X60~0X63)处；从J盘的DBR可知，保留扇区数为5022，每个FAT表占用扇区数为1585。所以54424号簇项在J盘的5447号扇区(针对FAT1表)偏移为96~99(即0X60~0X63)处。a03.doc文件内容在FAT1表中的文件分配表存储形式如图5.30所示。从图5.30可知，a03.doc文件在J盘中的存储是连续的。根据图5.30可以画出a03.doc文件的文件分配链表如图5.31所示。

图5.30 a03.doc文件所占簇号项在FAT2表中文件分配表的存储形式



图5.31 a03.doc文件的文件分配链表(注：图中的数据均为十六进制)

由式(5.20)可以计算出a03.doc文件所占簇数。

$$\begin{aligned}
 \text{a03.doc文件所占簇数} &= \text{ROUNDUP}(\text{a03.doc文件所占字节数} / (\text{每个簇的扇区数} \times 512), 0) \\
 &= \text{ROUNDUP}(31232 / (2 \times 512), 0) = 31
 \end{aligned}$$

a03.doc文件内容开始簇号为54424(即0XD498)，由于a03.doc文件在J盘上的存储是连续的，所以a03.doc文件内容结束簇号为54454(即0XD4B6)。

由式(5.16)可知：

$$\begin{aligned}
 \text{54454号簇项在FAT表的扇区号} &= \text{INT}(簇号项 / 128) + 1 = \text{INT}(54454 / 128) + 1 \\
 &= 425
 \end{aligned}$$

由式(5.17)可知：

$$\begin{aligned}
 \text{54454号簇项在FAT表中的扇区偏移地址} &= 4 \times \text{MOD}(簇号项, 128) + N - 1 \\
 &= 4 \times \text{MOD}(54454, 128) + N - 1 \\
 &= \{216, 217, 218, 219\}
 \end{aligned}$$

所以,a03.doc 文件的分配链表在 FAT 表(即 FAT1 表和 FAT2 表)中的位置为 425 号扇区,开始簇号在 FAT 表中的扇区偏移为 0X60~0X63(即 98~99),而结束簇号在 FAT 表中的扇区偏移为 0XD8~0XDB(即 216~219)。

由于 FAT1 表在 J 盘的开始扇区号为 5022,每个簇的扇区数为 1585,FAT2 表在 J 盘的开始扇区号为 6607。

因此,a03.doc 文件的分配链表在 J 盘的 5447 号扇区(针对 FAT1 表)和 7032 号扇区(针对 FAT2 表),a03.doc 文件内容开始簇号的扇区偏移为 0X60~0X63,而结束簇号的扇区偏移为 0XD8~0XDB。

将光标移动到 J 盘 7032 号扇区,在左侧 Cluster No. 的下方显示“FAT2”,在窗口左下角显示“Sector 7032 of 413696”,如图 5.32 所示,对比图 5.30 和图 5.32 可知,这两个图所显示的内容完全一样。

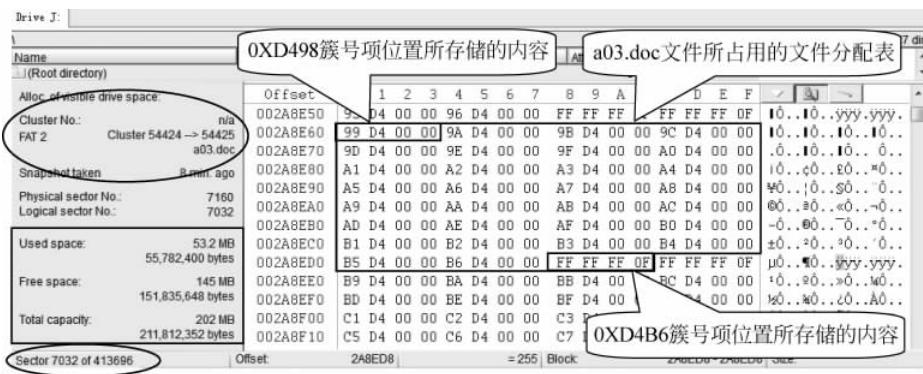


图 5.32 a03.doc 文件所占簇号项在 FAT2 表中的存储形式

经计算,65538 号簇项在 FAT 表的位置为第 512 号扇区,即 J 盘的 5534 号扇区(针对 FAT1 表)和 7119 号扇区(针对 FAT2 表),扇区偏移为 0X08~0X0B,所存储的内容为 00000000,如图 5.33 所示。

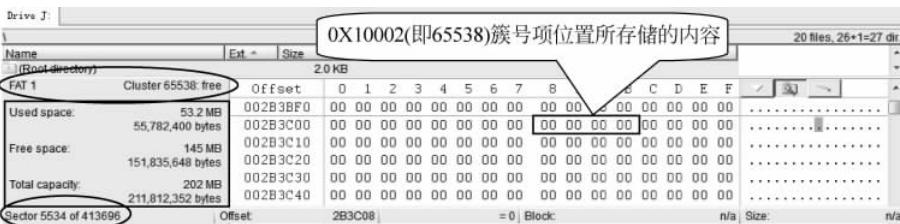


图 5.33 a03.doc 文件放入回收站前,65538 号簇项在 FAT1 表中的位置

(2) 将 J 盘根目录下的 a03.doc 文件放入回收站后

将 J 盘根目录下的 a03.doc 文件放入回收站后,对 a03.doc 文件所占的文件分配链表没有影响。由于在回收站中新建立了两个文件,文件名分别为 \$I6BZFC.T. doc 和 \$R6 BZFCT . doc,文件 \$R6BZFC.T. doc 的开始簇号和实际大小也就是 a03.doc 的开始簇号和实际大小,所以新建立的 \$R6BZFC.T. doc 文件对 FAT 表没有影响。

而新建立的 \$I6BZFC.T. doc 文件开始簇号为 65538,占 544 字节空间,FAT32 文件系统

将分配一个簇的空间给 \$I6BZFCT.doc 文件。65538 号簇项在 FAT 表的位置为第 512 号扇区,即 J 盘的 5534 号扇区(针对 FAT1 表)和 7119 号扇区(针对 FAT2 表),扇区偏移为 0X08~0X0B,所存储的内容为 0X0FFFFFFF(存储形式为 FF FF FF 0F),如图 5.34 所示。

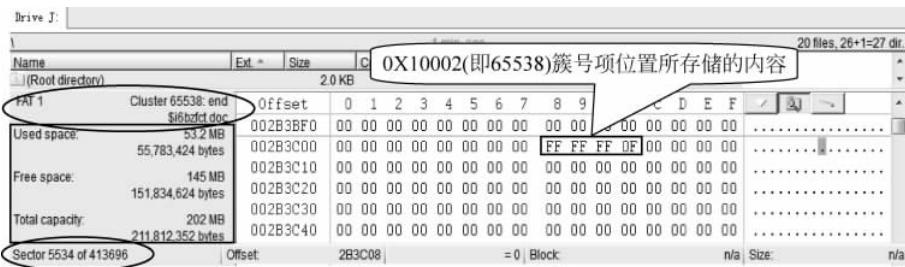


图 5.34 a03.doc 文件放入回收站后,J 盘 FAT1 的 65538 号簇项所在位置内容

(3) 将回收站清空

回收站清空后,\$R6BZFCT.doc 文件所占据的文件分配表(即 a03.doc 放入回收站前所占据)已被置为 00000000,如图 5.35 所示,即从 0XD498 号簇至 0XD4B6 号簇已经成为自由簇。

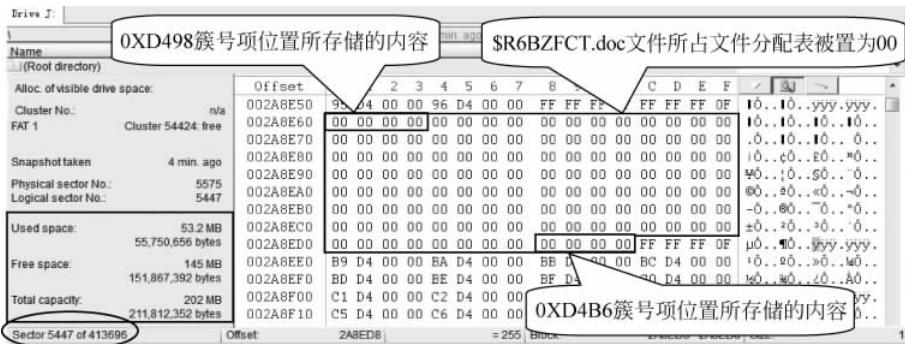


图 5.35 清空回收站后,a03.doc 所占簇号项已被置为 00

而 \$I6BZFCT.doc 文件所占据的文件分配表已经被置为 00,如图 5.36 所示,即 0X10002 号簇已成为自由簇。

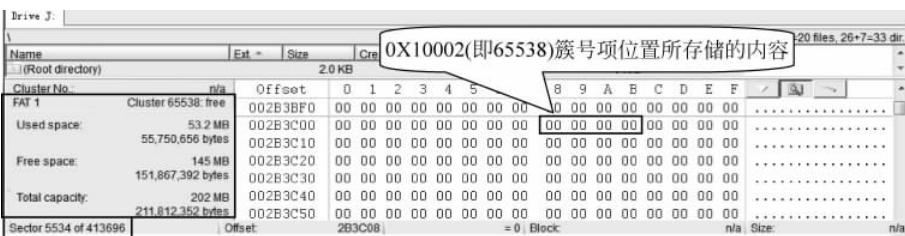


图 5.36 清空回收站后,J 盘 FAT1 表的 65538 号簇项内容为“00000000”

4) 对 J 盘已用磁盘空间和自由磁盘空间的影响

a03.doc 文件放入回收站前、回收后与清空回收站 J 盘已使用空间情况见表 5.21 所列。

表 5.21 a03.doc 文件放入回收站前、放入回收站后与清空回收站 J 盘空间情况变化表

删除文件过程 J 盘空间	a03.doc 放入回收站前	a03.doc 放入回收站后	清空回收站
J 盘已使用空间	55 782 400 字节	55 783 424 字节	55 750 656 字节
J 盘自由空间	151 835 648 字节	151 834 624 字节	151 867 392 字节

说明：

(1) 将 a03.doc 文件放入回收站前,从图 5.33 可知,J 盘已使用空间为 55 782 400 字节;而自由空间为 151 835 648 字节。

(2) 将 a03.doc 文件放入回收站后,从图 5.34 可知,J 盘已使用空间为 55 783 424 字节;而自由空间为 151 834 624 字节;由于在 J 盘回收中创建了两个文件,一个文件名为 \$R6BZFC.T. doc,另一个文件名为 \$I6BZFC.T. doc; \$R6BZFC.T. doc 文件的开始簇号也就是被删除文件 a03.doc 的开始簇号,文件大小也就是 a03.doc 文件的大小,此文件对 J 盘的空间没有影响。而创建的文件 \$I6BZFC.T. doc 大小为 544 字节,占一个簇的空间,所以 J 盘的已使用空间增加 1 个簇,自由空间减少 1 个簇即 1024 字节。

(3) 将回收站清空后,从图 5.35 可知,J 盘已使用空间为 55 750 656 字节;而自由空间为 151 867 392 字节;与 a03.doc 文件放入回收站前相比较,J 盘已使用空间减少了 31744 字节,自由空间增加了 31744 字节,即 a03.doc 文件所占用的空间已被释放。

注：删除文件夹对目录项、回收站等的影响与删除文件对目录项、回收站的影响基本相同。

思考题

(注：在本章思考题中,H 盘是指“ZY5. vhd 文件”通过计算机管理中的磁盘管理功能附加后所产生的虚拟硬盘)

5.1 FAT32 文件系统主要使用在哪几种外存储器中? FAT32 文件系统由哪几大部分组成?

5.2 FAT32 文件系统数据区的开始簇号是几号? 根目录的开始簇号又是几号?

5.3 简述 FAT32 文件系统的总体布局。

5.4 简述 FAT16 与 FAT32 文件系统的相同点和不同点。

5.5 FAT32_DBR 由哪几部分组成?

5.6 存储在 FAT32 文件系统中的一个文件由哪几部分组成? 请简述各部分之间的关系。

5.7 在 Windows 7 操作系统下,叙述删除存储在 FAT32 文件系统中的一个文件(即将要删除的文件放入回收站,并将回收站清空)的过程。

5.8 H 盘用 FAT32 文件系统格式化后,它的 FAT32_DBR 位于整个硬盘的 128 号扇区(即 H 盘的 0 号扇区),FAT32_DBR 前 112 字节如图 5.37 所示,请回答下列问题:

(1) 请将 H 盘 FAT32_DBR 中的 BPB 参数填入到表 5.22 中的下画线处。

(2) H 盘的 FAT1 表开始扇区号和结束扇区号分别是几号? 对于整个硬盘扇区号分别是几号? FAT2 表呢?

Partitioning style: MBR																3 files, 1 partitions						
Name	Ext.	Size	Created	Attr.	1st sector																	
Partition 1	FAT32	3.0 GB			128																	
[D:\ya23\l\abcd4_5.vhd]						Offset	0	1	2	3	4	5	6	7	8	9	A					
						00010000	EB	58	90	4D	53	44	4F	53	35	2E	30	00	02	08	2A	10
Default Edit Mode						00010010	02	00	00	00	00	F8	00	00	3F	00	FF	00	80	00	00	00
State:	original					00010020	00	E8	5F	00	EB	17	00	00	00	00	00	02	00	00	00	00
Undo level:	0					00010030	01	00	06	00	00	00	00	00	00	00	00	00	00	00	00	00
Undo reverses:	n/a					00010040	80	00	29	43	6F	60	98	4E	4F	20	4E	41	4D	45	20	20
Total capacity: 3.0 GB						00010050	20	20	46	41	54	33	32	20	20	20	C9	8E	D1	B0	F4	
3,221,225,984 bytes						00010060	7B	8E	C1	8E	D9	B0	00	7C	88	4E	02	8A	56	40	B4	41
Sector 128 of 6291457																						
						Offset	1002F								= 0	Block	1002F - 1002F	Size:				

图 5.37 H 盘的 FAT32_DBR 前 112 字节

表 5.22 H 盘 FAT32_DBR 的 BPB 参数

字节偏移	字节数	值			含义
		十进制	十六进制	存储形式	
0X00	2	略	略	EB 58	跳转指令
0X02	1	略	略	90	空操作指令
0X03	8	略	略	4D 53 44 4F 53 35 2E 30	OEM(即厂商标志), 长度为 8 个字符
0X0B	2	_____	_____	_____	每个扇区的字节数
0X0D	1	_____	_____	_____	每个簇的扇区数
0X0E	2	_____	_____	_____	保留扇区数
0X10	1	_____	_____	_____	FAT 表的个数, 本字段的值一般为 2
0X11	2	00	00	00 00	对于 FAT32 分区, 本字段必须设置为 00
0X13	2	00	00	00 00	对于 FAT32 分区, 本字段必须设置为 00
0X15	1	略	略	略	0XF8 表示硬盘、U 盘、SD 卡等
0X16	2	00	00	00 00	对 FAT32 分区, 本字段必须设置为 00
0X18	2	_____	_____	_____	扇区数/每磁道
0X1A	2	_____	_____	_____	磁头数
0X1C	4	_____	_____	_____	隐藏扇区数
0X20	4	_____	_____	_____	该分区总扇区数
0X24	4	_____	_____	_____	每 FAT 表占扇区数
0X28	2	0	0	00 00	扩展标志
0X2A	2	0	0	00 00	文件系统版本
0X2C	4	0	2	02 00 00 00	根目录开始簇号, 一般为 2
0X30	2	1	1	01 00	文件系统信息所在扇区号, 一般为 1
0X32	2	6	6	06 00	DBR 备份所在扇区号, 一般为 6
0X34	12	略	略	略	保留, 供以后扩充使用的保留空间

(3) H 盘的总扇区数是多少? 请计算 H 盘的总容量, H 盘的总容量等于用户通过查看 H 盘属性报告的容量吗? 为什么?

(4) 用户通过 H 盘属性查看到 H 盘的容量为 3 209 691 136 字节(即 2.98GB); 请通过 H 盘的 FAT32_DBR 参数来验证该数据的正确性。

(5) H 盘的开始簇号为 2, 即根目录的开始簇号, 请计算 H 盘的结束簇号。

(6) H 盘的开始扇区号是 0, 请计算 H 盘的结束扇区号。

(7) 计算 H 盘的结束簇号项分别在 FAT1 表和 FAT2 表中的位置(即在 H 盘的扇区号和扇区偏移, 注: 0 号扇区为 H 盘 FAT32_DBR 所在扇区号)。

(8) 根据 H 盘每个 FAT 表占用扇区数,计算该 FAT 表可以表示的最小磁盘空间和最大磁盘空间(单位:字节)。

(9) 按示例将表 5.23 中 H 盘的簇号转换成 H 盘对应的扇区号范围以及整个硬盘的扇区号范围,并将结果填入到表 5.23 对应单元格中。

表 5.23 H 盘簇号转换成对应的扇区号

H 盘簇号	3(示例)	10	500	5000	10892
对应 H 盘扇区号范围	16392~16399	~	~	~	~
对应整个硬盘扇区号范围	16520~16527	~	~	~	~

(10) 按示例将表 5.24 中 H 盘的扇区号转换成对应 H 盘的簇号,再将 H 盘的簇号转换为对应 H 盘的扇区号范围;并将结果填入到表 5.24 对应单元格中。

表 5.24 H 盘簇号转换成对应的扇区号

对应 H 盘簇号	2				
H 盘扇区号	16391(示例)	17676	494177	732071	970108
对应 H 盘扇区号范围	16384~16391	~	~	~	~

(11) H 盘对应的分区形式为 MBR,且分区表存储在整个硬盘 0 号扇区偏移 0X1BE~0X1CD 处,请写出 H 盘对应的 MBR 分区表(存储形式)。

(12) 将整个硬盘扇区号、H 盘扇区号与簇号的对应关系填入到表 5.25 下画线处。

表 5.25 整个硬盘扇区号、H 盘扇区号与簇号的对应关系表

扇区用途	整个硬盘逻辑扇区号	H 盘的逻辑扇区号	簇号	描述	所占扇区数	备注
相对扇区	0~127	不能通过逻辑盘扇区进行存取		分区表到 DBR 的扇区数	128	
保留扇区	128	0	不能通过簇的方式进行存取	FAT32_DBR	1	H 盘
	129	1		FSINFO	1	
	130~133	2~5		未用	4	
	134	6		DBR 备份	1	
	135	7		FSINFO 备份	1	
	~	~		未用	_____	
FAT1 表占扇区	~	~		FAT1 表	_____	
FAT2 表占扇区	~	~		FAT2 表	_____	
数据区所占扇区	~	~	2	根目录开始簇号	_____	
	~	~	3	根目录的下一个簇、子目录和数据存放位置	_____	
	~	~	4		~	
	~	~				
未分区的扇区	略				略	

(13) 假设该逻辑盘没有分区表,即将整个硬盘 128 号扇区的 FAT32_DBR 存放在整个硬盘的 0 号扇区,需要修改整个硬盘 0 号扇区 FAT32_DBR 的哪几个参数? 修改为何值? 才能使得该 FAT32 文件系统能够正常工作。

5.9 H 盘的文件系统是 FAT32, FAT32_DBR 如题 5.8 所述, FAT1 表前 160 字节如图 5.38 所示, 请回答下列问题。

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00205400	F8	FF	FF	0F	FF	FF	FF	FF	FF	FF	OF	FF	FF	FF	FF	OF	?...
00205410	FF	FF	FF	0F	FF	FF	FF	FF	FF	FF	OF	FF	FF	FF	FF	OF
00205420	FF	FF	FF	OF	0D	00	00	00	FF	FF	FF	OF	FF	FF	FF	OF
00205430	FF	FF	FF	OF	0E	00	00	00	OF	00	00	00	10	00	00	00
00205440	FF	FF	FF	OF	17	00	00	00	13	00	00	00	14	00	00	00
00205450	15	00	00	00	16	00	00	00	FF	FF	FF	OF	18	00	00	00
00205460	19	00	00	00	FF	FF	FF	OF	1B	00	00	00	1C	00	00	00
00205470	1D	00	00	00	1E	00	00	00	FF	FF	FF	OF	20	00	00	00
00205480	21	00	00	00	FF	FF	FF	OF	FF	FF	FF	OF	24	00	00	00	!....\$...
00205490	25	00	00	00	26	00	00	00	FF	FF	FF	OF	FF	FF	FF	OF	%...&....

图 5.38 某 H 盘的 FAT1 表前 160 字节

(1) 在 FAT1 表前 160 字节中, 共存储多少个文件(夹)的分配表? 请分别写出占据 1 个簇、2 个簇、3 个簇、4 个簇和 5 个簇的文件(夹)数量(注: 不包括根目录)。

(2) H 盘根目录占据了哪几个簇号? 对应的扇区号分别是多少?

(3) A12 文件夹开始簇号为 0X09, 从图 5.38 可知, A12 文件夹占据了哪几个簇号? 对应的扇区号分别是多少? 假设每个目录项占 32 字节, 请问在 A12 文件夹中最多只能存放多少个文件目录项? 也就是说, 当 A12 文件夹中存储的目录项超过多少个时? A12 文件夹再申请添加一个簇(注: 在 A12 文件夹中已有两个目录项, 即“.”和“..”)。

(4) 某文件开始簇号为 0X01A, 从图 5.38 可知, 该文件在 H 盘上是否连续存储? 为什么? 该文件内容共占据了哪几个簇号? 对应的扇区号分别是多少? 请画出该文件所占簇号的链表图; 请计算该文件内容实际大小的范围(即该文件内容的最小空间和最大空间), 当该文件内容小于最小空间时, H 盘为该文件减少一个簇, 当该文件内容大于最大空间时, H 盘为该文件再申请添加一个簇。

5.10 H 盘的文件系统是 FAT32, FAT32_DBR 如题 5.9 所示, 根目录原代码如图 5.39 所示, 请回答下列问题:

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F		
00800000	D0	C2	BC	D3	BE	ED	20	20	20	20	08	00	00	00	00	新加卷	
00800010	00	00	00	00	00	00	42	4C	BC	46	00	00	00	00	00BL 稊.....	
00800020	24	52	45	43	59	43	4C	45	42	49	4E	16	00	AB	48	4C	\$RECYCLEBIN..獮L
00800030	BC	46	BC	46	00	00	49	4C	BC	46	03	00	00	00	00	00	稊稊..IL 稊.....
00800040	41	30	31	20	20	20	20	20	20	20	10	08	23	73	4C	A01	..#sl
00800050	BC	46	BC	46	00	00	0E	59	5E	43	05	00	00	00	00	00	稊稊...Y^C.....
00800060	41	30	32	20	20	20	20	20	20	20	10	00	29	73	4C	A02	..)sL
00800070	BC	46	BC	46	00	00	32	59	5E	43	08	00	00	00	00	00	稊稊..2Y^C.....
00800080	41	31	32	20	20	20	20	20	20	20	10	00	2C	73	4C	A12	..,sL
00800090	BC	46	BC	46	00	00	4A	56	57	43	09	00	00	00	00	00	稊稊..JVWC.....
008000A0	42	22	7D	15	5F	EE	76	55	5F	29	00	0F	00	9E	00	00	B").._顧U_)...?.
008000B0	FF	00	00	FF	FF	FF	FF									
008000C0	01	41	00	31	00	32	00	28	00	58	5B	0F	00	9E	3E	65	.A.1.2.().X[..?e
008000D0	34	00	30	00	30	00	2A	4E	87	65	00	00	F6	4E	84	76	4.0.0.*N喂..鰥剗
008000E0	41	31	32	28	B4	E6	7E	31	20	20	20	10	00	64	73	4C	A12(存~1..dsL
008000F0	BC	46	BC	46	00	00	4A	56	57	43	11	00	00	00	00	00	稊稊..JVWC.....
00800100	41	31	33	20	20	20	20	20	20	20	10	08	8E	73	4C	A13	..曠L
00800110	BC	46	BC	46	00	00	4B	56	57	43	1F	00	00	00	00	00	稊稊..KVWC.....

图 5.39 H 盘 2 号簇(即根目录)存储的内容

00800120	42 15 5F EE 76 55 5F 29	00 00 00 OF 00 9F FF FF	B._頤U_).....?
00800130	FF FF FF FF FF FF FF	FF FF 00 00 FF FF FF FF	..
00800140	01 41 00 31 00 33 00 28	00 58 5B OF 00 9F 3E 65	.A.1.3. (.X[..?e
00800150	31 00 30 00 30 00 2A 4E	87 65 00 00 F6 4E 22 7D	1.0.0.*N嘍..鯪")
00800160	41 31 33 28 B4 E6 7E 31	20 20 20 <u>10</u> 00 AA 73 4C	A13(存~1 ..獮L
00800170	BC 46 BC 46 00 00 4B 56	57 43 22 00 00 00 00 00	糉糉..KVWC".....
00800180	41 31 34 20 20 20 20 20	20 20 20 <u>10</u> 08 B2 73 4C	A14 ..暭L
00800190	BC 46 BC 46 00 00 4B 56	57 43 27 00 00 00 00 00	糉糉..KVWC'.....
008001A0	41 31 35 20 20 20 20 20	20 20 20 <u>10</u> 08 B8 73 4C	A15 ..竑L
008001B0	BC 46 BC 46 00 00 4B 56	57 43 28 00 00 00 00 00	糉糉..KVWC(.....
008001C0	41 31 36 20 20 20 20 20	20 20 20 <u>10</u> 08 BE 73 4C	A16 ..緘L
008001D0	BC 46 BC 46 00 00 4B 56	57 43 41 00 00 00 00 00	糉糉..KVWCA.....
008001E0	41 31 37 20 20 20 20 20	20 20 20 <u>10</u> 08 2A 74 4C	A17 ..*tL
008001F0	BC 46 BC 46 00 00 4B 56	57 43 46 00 00 00 00 00	糉糉..KVWCF.....
00800200	41 38 30 20 20 20 20 20	20 20 20 <u>10</u> 08 30 74 4C	A80 ..0tL
00800210	BC 46 BC 46 00 00 4B 56	57 43 47 00 00 00 00 00	糉糉..KVWCG.....
00800220	41 61 00 31 00 30 00 30	00 7F 95 OF 00 2A 87 65	Aa.1.0.0..?.*嘍
00800230	F6 4E OD 54 00 00 FF FF	FF FF 00 00 FF FF FF FF	鯪T.....
00800240	41 31 30 30 B3 A4 7E 31	20 20 20 <u>10</u> 00 40 74 4C	A100長~1 ..@tL
00800250	BC 46 BC 46 00 00 4B 56	57 43 49 00 00 00 00 00	糉糉..KVWC1.....
00800260	42 22 7D 15 5F EE 76 55	5F 29 00 OF 00 BB 00 00	B"}._頤U_)...?.
00800270	FF FF FF FF FF FF FF	FF FF 00 00 FF FF FF FF	..
00800280	01 41 00 36 00 30 00 30	00 28 00 OF 00 BB 58 5B	.A.6.0.0.(...撻[
00800290	3E 65 36 00 30 00 30 00	2A 4E 00 00 87 65 F6 4E	>e6.0.0.*N..嘍鯪
008002A0	41 36 30 30 28 7E 31 20	20 20 20 <u>10</u> 00 45 74 4C	A600(~1 ..EtL
008002B0	BC 46 BC 46 00 00 4C 56	57 43 4A 00 00 00 00 00	糉糉..LVWCJ.....
008002C0	43 29 00 00 00 FF FF FF	FF FF FF OF 00 B0 FF FF	C).....?
008002D0	FF FF FF FF FF FF FF	FF FF 00 00 FF FF FF FF	..
008002E0	02 30 00 2A 4E 87 65 F6	4E 30 00 OF 00 B0 30 00	.0.*N嘍鯪0...?.
008002F0	30 00 30 00 7E 00 39 00	39 00 00 00 39 00 39 00	0.0.^9.9...9.9.
00800300	01 61 00 31 00 30 00 30	00 30 00 OF 00 B0 30 00	.a.1.0.0.0...?.
00800310	28 00 58 5B 3E 65 31 00	30 00 00 00 30 00 30 00	(.X[>e1.0...0.0.
00800320	41 31 30 30 30 30 7E 31	20 20 20 <u>10</u> 00 76 74 4C	A10000~1 ..vtL
00800330	BC 46 BC 46 00 00 C5 4D	89 43 54 00 00 00 00 00	糉糉..嚮望T.....
00800340	41 41 20 20 20 20 20 20	20 20 20 <u>11</u> 08 31 7A 4C	AA ..1zL
00800350	BC 46 BC 46 00 00 A4 72	67 43 A3 00 00 00 00 00	糉糉..姆gC?....
00800360	41 41 41 41 41 41 20 20	20 20 20 <u>10</u> 08 33 7A 4C	AAAAAA ..3zL
00800370	BC 46 BC 46 00 00 D8 81	5C 43 A5 00 00 00 00 00	糉糉..?\\C?....
00800380	41 61 00 62 00 63 00 64	00 28 00 OF 00 2D 31 00	Aa.b.c.d.(...-1.
00800390	30 00 30 00 30 00 2A 4E	87 65 00 00 F6 4E 29 00	0.0.0.*N嘍..鯪).
008003A0	41 42 43 44 28 31 7E 31	20 20 20 <u>10</u> 00 36 7A 4C	ABCD(1~1 ..6zL
008003B0	BC 46 BC 46 00 00 52 56	57 43 A6 00 00 00 00 00	糉糉..RVWC?....
008003C0	41 61 00 62 00 63 00 64	00 28 00 OF 00 25 32 00	Aa.b.c.d.(...%2.
008003D0	30 00 30 00 30 00 2A 4E	87 65 00 00 F6 4E 29 00	0.0.0.*N嘍..鯪).
008003E0	41 42 43 44 28 32 7E 31	20 20 20 <u>10</u> 00 A8 7A 4C	ABCD(2~1 ...L
008003F0	BC 46 BC 46 00 00 53 56	57 43 B1 00 00 00 00 00	糉糉..SWC?....
00800400	42 30 00 28 00 31 00 33	00 30 00 OF 00 0A 30 00	B0. (.1.3.0....0.
00800410	2A 4E 87 65 F6 4E 29 00	00 00 00 00 FF FF FF FF	*N嘍鯪).....
00800420	01 61 00 62 00 63 00 64	00 5F 00 OF 00 0A 61 00	.a.b.c.d._...a.
00800430	31 00 30 00 31 00 7E 00	61 00 00 00 35 00 30 00	1.0.1.^a...5.0.
00800440	41 42 43 44 5F 41 7E 31	20 20 20 <u>10</u> 00 9D 7B 4C	ABCD_A~1 0..[L
00800450	BC 46 BC 46 00 00 07 53	5E 43 C1 00 00 00 00 00	糉糉..S^C?....
00800460	42 30 00 28 00 58 5B 3E	65 31 00 OF 00 AA 33 00	B0. (.X[>e1...?.
00800470	30 00 30 00 2A 4E 87 65	F6 4E 00 00 29 00 00 00	0.0.*N嘍鯪)...).
00800480	01 61 00 62 00 63 00 64	00 5F 00 OF 00 AA 61 00	.a.b.c.d._...獮.
00800490	31 00 30 00 31 00 7E 00	61 00 00 00 35 00 30 00	1.0.1.^a...5.0.
008004A0	41 42 43 44 5F 41 7E 32	20 20 20 <u>10</u> 00 56 7C 4C	ABCD_A~2 ..V L
008004B0	BC 46 BC 46 00 00 55 56	57 43 E6 00 00 00 00 00	糉糉..UVWC?....
008004C0	42 39 00 28 00 31 00 32	00 30 00 OF 00 4B 30 00	B9. (.1.2.0...K0.
008004D0	66 00 69 00 6C 00 65 00	73 00 00 00 29 00 00 00	f. i. l. e. s...)...

图 5.39 (续)

008004E0	01 61 00 62 00 63 00 64 00 5F 00 0F 00 4B 61 00	. a. b. c. d. ... Ka.
008004F0	31 00 30 00 31 00 7E 00 62 00 00 00 32 00 39 00	1. 0. 1. ^. b. . 2. 9.
00800500	41 42 43 44 5F 41 7E 33 20 20 20 <u>10</u> 00 03 7D 4C	ABCD_A^3 ... }L
00800510	BC 46 BC 46 00 00 56 56 57 43 F2 00 00 00 00 00	糉糉.. VVWC?....
00800520	41 30 33 20 20 20 20 20 54 58 54 <u>20</u> 18 A3 BA 4C	A03 TXT .: L
00800530	BC 46 BC 46 00 00 03 81 B4 46 FC 00 D5 8B 42 07	糉糉.... 碌?譜 B.
00800540	42 30 33 20 20 20 20 20 54 58 54 <u>20</u> 18 9C 1B 4D	B03 TXT .?M
00800550	BC 46 BC 46 00 00 03 81 B4 46 25 75 D5 8B 42 07	糉糉.... 碌%u 譜 B.
00800560	41 30 31 20 20 20 20 20 54 58 54 <u>21</u> 18 31 8A 4D	A01 TXT .1 奩
00800570	BC 46 BC 46 00 00 03 81 B4 46 4E E9 D5 8B 42 07	糉糉.... 碌 N 檀嫵.
00800580	41 30 32 20 20 20 20 20 54 58 54 <u>20</u> 18 60 BC 4D	A02 TXT . 蔡
00800590	BC 46 BC 46 01 00 03 81 B4 46 77 5D D5 8B 42 07	糉糉.... 碌 w] 譜 B.
008005A0	41 42 43 44 45 20 20 20 20 20 <u>10</u> 08 8B 54 4E	ABCDE .. 嫚 N
008005B0	BC 46 BC 46 01 00 01 4E BC 46 A0 D1 00 00 00 00	糉糉...N 糉慘....
008005C0	4B 55 4E 4D 49 4E 47 20 44 4F 43 <u>20</u> 18 A5 0D 4F	KUNMING DOC .?0
008005D0	BC 46 BC 46 01 00 01 60 63 44 B1 D1 00 80 00 00	糉糉... `cD 慾. €..
008005E0	53 48 41 4E 47 48 41 49 44 4F 43 <u>20</u> 18 8D 0E 50	SHANGHAIDOC ...P
008005F0	BC 46 BC 46 0B 00 91 8D 72 44 02 00 00 6A 00 00	糉糉.. ?rD... j..

图 5.39 (续)

(1) 在 H 盘的根目录下存储有一个卷标名,请写出 H 盘的卷标名。

(2) 在根目录下存储有几个子目录(文件夹)? 如果子目录是采用长文件夹名命名方式,请写出长文件夹名所对应的短文件名。其中: 哪几个子目录的属性是只读? 哪几个子目录的属性是系统和隐藏?

(3) 在根目录下存储有几个文件?

(4) 在根目录下存储的第 5 个目录项如图 5.40 所示,该目录项描述的是一个子目录(文件夹),请写出在资源管理器所查看到的子目录名。该子目录的开始簇号是多少? 该子目录分配链表如图 5.38 所示,请画出该子目录的分配链表图。

00800080	41 31 32 20 20 20 20 20 20 20 <u>10</u> 00 2C 73 4C A12 .., sL
00800090	BC 46 BC 46 00 00 4A 56 57 43 09 00 00 00 00 00 糉糉.. JVWC.....

图 5.40 H 盘根目录存储的第 5 个目录项

(5) 在根目录下存储的倒数第 2 个目录项如图 5.41 所示,该目录项描述的是一个文件,请写出用户在资源管理器中查看到的文件名。该文件内容的实际大小是多少字节? 占 H 盘的空间是多少字节? 该文件内容在 H 盘上是连续存储的,它的开始簇号和结束簇号分别是多少? 计算该文件内容的开始簇号项和结束簇号项在 FAT1 表和 FAT2 表中的位置。请画出该文件的分配链表图。

008005C0	4B 55 4E 4D 49 4E 47 20 44 4F 43 <u>20</u> 18 A5 0D 4F KUNMING DOC .?0
008005D0	BC 46 BC 46 01 00 01 60 63 44 B1 D1 00 80 00 00 糉糉... `cD 慕. €..

图 5.41 H 盘根目录存储的倒数第 2 个目录项

(6) 在根目录下存储的倒数第 5 个目录项如图 5.42 所示,该目录项描述的是一个文件,在资源管理器中查看该文件的属性如图 5.43 所示,请将该文件的大小、占用空间、创建日期和时间、修改日期和时间填入图 5.43 对应下画线处; 该目录项偏移 0X0B 的值是多少? 该值所描述的文件属性是什么? 请将该属性值所描述的属性在图 5.43 对应的只读、隐藏或者存档方框中打上“√”。

```
00800560 41 30 31 20 20 20 20 54 58 54 21 18 31 8A 4D A01      TXT .1奄
00800570 BC 46 BC 46 00 00 03 81 B4 46 4E E9 D5 8B 42 07 糊N 橋嫗.
```

图 5.42 H 盘根目录存储的倒数第 5 个目录项



图 5.43 某文件的属性情况

5.11 在题 5.10 中,在 DOS 下查看到 H 盘的根目录如图 5.44 所示。请回答下列问题:

驱动器 H 中的卷是新加卷
 卷的序列号是 9860-6F43 H:\ 的目录
 2013/10/30 11:08 <DIR> a01
 2013/10/30 11:09 <DIR> A02
 2013/10/23 10:50 <DIR> A12
 2013/10/23 10:50 <DIR> A12(存放 400 个文件索引目录)
 2013/10/23 10:50 <DIR> a13
 2013/10/23 10:50 <DIR> A13(存放 100 个文件索引目录)
 2013/10/23 10:50 <DIR> a14
 2013/10/23 10:50 <DIR> a15
 2013/10/23 10:50 <DIR> a16
 2013/10/23 10:50 <DIR> a17
 2013/10/23 10:50 <DIR> a80
 2013/10/23 10:50 <DIR> a100 长文件名
 2013/10/23 10:50 <DIR> A600(存放 600 个文件索引目录)
 2013/12/09 09:46 <DIR> a10000(存放 10000 个文件 0000~9999)
 2013/11/07 14:21 <DIR> aa
 2013/10/28 16:14 <DIR> aaaaaa
 2013/10/23 10:50 <DIR> abcd(1000 个文件)
 2013/10/23 10:50 <DIR> abcd(2000 个文件)
 2013/10/30 10:24 <DIR> abcd_a101~a500(1300 个文件)
 2013/10/23 10:50 <DIR> abcd_a101~a500(存放 1300 个文件)
 2013/10/23 10:50 <DIR> abcd_a101~b299(1200files)
 2015/05/20 16:08 121,801,685 a03.txt
 2015/05/20 16:08 121,801,685 b03.txt
 2015/05/20 16:08 121,801,685 a01.txt
 2015/05/20 16:08 121,801,685 a02.txt
 2015/05/28 09:48 <DIR> abcde
 2014/03/03 12:00 32,768 kunming.doc
 2014/03/18 17:44 27,136 shanghai.doc
 6 个文件 487,266,644 字节
 22 个目录 2,721,292,288 可用字节

图 5.44 DOS 下查看 H 盘根目录

(1) 请结合图 5.39 将 H 盘根目录下存储的卷标、每个文件(夹)的属性代码填入到表 5.26 对应单元格中(注：属性代码使用十六进制)。

(2) 假设 H 盘根目录下每个文件的内容均是连续存储的,请将每个文件的开始簇号、结束簇号和文件大小填入到表 5.26 中对应单元格中。

表 5.26 H 盘根目录的文件(夹)属性代码、开始簇号等

序号	卷标名、文件名或文件夹名	属性代码	开始簇号	结束簇号	大小(字节)
1	新加卷				
2	\$ RECYCLE.BIN				
3	a01				
4	A02				
5	A12				
6	A12(存放 400 个文件的索引目录)				
7	a13				
8	A13(存放 100 个文件索引目录)				
9	a14				
10	a15				
11	a16				
12	a17				
13	a80				
14	a100 长文件名				
15	A600(存放 600 个文件索引目录)				
16	a10000(存放 10000 个文件 0000~9999)				
17	aa				
18	aaaaaa				
19	abcd(1000 个文件)				
20	abcd(2000 个文件)				
21	abcd_a101~a500(1300 个文件)				
22	abcd_a101~a500(存放 1300 个文件)				
23	abcd_a101~b299(1200files)				
24	a03.txt				
25	b03.txt				
26	a01.txt				
27	a02.txt				
28	abcde				
29	kunming.doc				
30	shanghai.doc				

(3) 将每个文件夹的开始簇号填入到表 5.26 对应单元格中。

(4) 将长文件夹名所对应的短文件夹名填入到表 5.27 对应单元格中。

表 5.27 H 盘根目录下长文件夹名所对应的短文件名

序号	长文件夹名	长文件夹名所对应的短文件夹名
1	A12(存放 400 个文件索引目录)	
2	A13(存放 100 个文件索引目录)	
3	a100 长文件名	
4	A600(存放 600 个文件索引目录)	
5	a10000(存放 10000 个文件 0000~9999)	
6	abcd(1000 个文件)	
7	abcd(2000 个文件)	
8	abcd_a101~a500(1300 个文件)	
9	abcd_a101~a500(存放 1300 个文件)	
10	abcd_a101~b299(1200files)	

5.12 在题 5.10 中,在 H 盘根目录下存储的最后一个目录项如图 5.45 所示,该目录项描述的是一个文件。

008005E0	53 48 41 4E 47 48 41 49 44 4F 43 20 18 8D 0E 50	SHANGHAIDOC ...P
008005F0	BC 46 BC 46 0B 00 91 8D 72 44 02 00 00 6A 00 00	糉糉..?rD...j..

图 5.45 H 盘根目录存储的最后一个目录项

请回答下列问题。

(1) 写出该目录项在资源管理器所查看到的文件名。并将该文件的基本信息填入表 5.28 中右侧对应单元格中(注:假设该文件内容在 H 盘上的存储是连续的)。

表 5.28 某文件的基本信息

在资源管理器中查看到的主文件名和扩展名	_____.	_____
文件属性代码	0X	_____
文件名和扩展名大小写代码	0X	_____
文件创建日期和时间	_____ 年 _____ 月 _____ 日 _____ : _____ :	_____
文件修改日期和时间	_____ 年 _____ 月 _____ 日 _____ : _____ :	_____
文件开始簇号和结束簇号	文件开始簇号: _____	文件结束簇号: _____
文件实际大小	_____ MB(_____ 字节)	_____
文件所占磁盘空间	_____ MB(_____ 字节)	_____
文件开始簇号在 FAT1 表的扇区号及偏移地址	扇区号: _____	偏移地址: _____ ~ _____
文件结束簇号在 FAT1 表的扇区号及偏移地址	扇区号: _____	偏移地址: _____ ~ _____
文件开始簇号在 FAT2 表的扇区号及偏移地址	扇区号: _____	偏移地址: _____ ~ _____
文件结束簇号在 FAT2 表的扇区号及偏移地址	扇区号: _____	偏移地址: _____ ~ _____

(2) 画出该文件在 H 盘的存储结构图。

(3) 假设回收站已空,将该文件放入回收站后,该目录项、回收站文件夹、FAT1 表、FAT2 表、磁盘已使用空间和自由空间有何变化?(注:操作系统为 Windows 7)

(4) 将该文件从回收站中清空后,该目录项、回收站文件夹、FAT1 表、FAT2 表、磁盘已使用空间和自由空间又有何变化?(注:操作系统为 Windows 7)

5.13 某逻辑盘两个子目录开始扇区前 64 字节分别如图 5.46 和图 5.47 所示,请回答下列问题:

Drive H:	85% free	Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
File system:	FAT32	00820000	2E	20	20	20	20	20	20	20	20	20	20	AA	73	4C	.		
Volume label:	新加卷	00820010	BC	46	BC	46	00	00	74	4C	BC	46	22	00	00	00	00	MF4F..tL4F....	
Default Edit Mode	original	00820020	2E	20	20	20	20	20	20	20	20	20	10	00	AA	73	4C	..	
Sector 16640 of 6285312		00820030	BC	46	BC	46	00	00	74	4C	BC	46	00	00	00	00	00	MF4F..tL4F....	
		Offset:	82001F		= 0	Block:	82001F	-	82001F	Size:								1	

图 5.46 第 1 个子目录前 96 字节

Drive H:	85% free	Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
File system:	FAT32	1D99F000	2E	20	20	20	20	20	20	20	20	20	10	00	8D	54	4E	.	
Volume label:	新加卷	1D99F010	BC	46	BC	46	01	00	55	4E	BC	46	A1	D1	00	00	00	00	
Default Edit Mode	original	1D99F020	2E	20	20	20	20	20	20	20	20	20	10	00	8D	54	4E	..	
Sector 969976 of 6285312		1D99F030	BC	46	BC	46	01	00	55	4E	BC	46	A0	D1	00	00	00	00	
		Offset:	82001F		= 0	Block:	82001F	-	82001F	Size:								1	

图 5.47 第 2 个子目录前 96 字节

(1) 请分别将这两个子目录的开始簇号填入到表 5.29 对应单元格中。

(2) 请分别将这两个子目录上一级目录的开始簇号填入到表 5.29 对应单元格中。

(3) 通过这两个子目录的开始扇区号和开始簇号,请计算出该逻辑盘中每个簇的扇区数,通过这两个子目录上一级目录的开始簇号和每个簇的扇区数,计算出这两个子目录上一级目录的开始扇区号并将结果填入到表 5.29 对应单元格中。

表 5.29 两个子目录及其上一目录(父目录)开始位置

子目录	两个子目录开始位置		上一级目录(即父目录)开始位置	
	开始扇区号	开始簇号	开始扇区号	开始簇号
第 1 个	16640			
第 2 个	969976			

(4) 这两个子目录的目录名是否存储在根目录下? 为什么?

5.14 某 U 盘只有一个 MBR 分区,对应的文件系统是 FAT32, FAT32_DBR 前 80 字节如图 5.48 所示,请根据 FAT32_DBR 中的相应参数计算该 FAT32 每个簇的扇区数,并将该值填入到图 5.48 扇区偏移 0XD0 下画线处(注: 素材文件名为 zy5_14.vhd)。

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00000000	EB	58	90	4D	53	44	4F	53	35	2E	30	00	02	_	06	18	隔. MSDOS5.0....
00000010	02	00	00	00	00	F8	00	00	3F	00	FF	00	00	00	00	00??.
00000020	00	00	20	00	FD	03	00	00	00	00	00	00	02	00	00	00	.. .?.....
00000030	01	00	06	00	00	00	00	00	00	00	00	00	00	00	00	00
00000040	80	00	29	77	A8	D8	3A	4E	4F	20	4E	41	4D	45	20	20	€.)w ?NO NAME

图 5.48 某 FAT32_DBR 前 96 字节

5.15 某 U 盘的文件系统为 FAT32, FAT32_DBR 前 80 字节如图 5.49 所示,请根据该 FAT32_DBR 中 BPB 相应参数,计算出该 FAT32 文件系统总扇区数的范围,即该 U 盘的最小总扇区数是多少? 最大总扇区数是多少? 并验证该 U 盘的总扇区数是否在该范围内。

5.16 设计一个实验,以 FAT32_DBR 中的 BPB 参数为依据,通过统计 FAT 表中已分配簇和未分配簇,计算 FAT32 文件系统中磁盘已用空间和自由空间。

5.17 根据你对 FAT32 文件系统的理解和认识,你认为计算 FAT32 文件系统中每个簇

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00000000	EB	58	90	4D	53	44	4F	53	35	2E	30	00	02	04	3E	20	隔.MSDOS5.0...>
00000010	02	00	00	00	00	F8	00	00	3F	00	FF	00	00	00	00	00??......
00000020	<u>00</u>	<u>00</u>	<u>20</u>	<u>00</u>	E1	OF	00	00	00	00	00	02	00	00	00	..?.....	
00000030	01	00	06	00	00	00	00	00	00	00	00	00	00	00	00	00
00000040	80	00	29	82	4A	0D	94	4E	4F	20	4E	41	4D	45	20	20	€.)僕.撫0 NAME

图 5.49 某 FAT32_DBR 前 96 字节

的扇区数有哪几种方法?

5.18 某硬盘 0 号扇区偏移 0X01BE~0X01FD 处存储有 4 个 MBR 分区表, 对应的文件系统均为 FAT32。由于用户操作不慎, 将 4 个 MBR 分区删除。用户使用 WinHex 软件, 通过 FAT32_DBR 特征值向下查找 FAT32_DBR, 分别在 128 号、134 号、198784 号、198790 号、362624 号、362630 号、567424 号和 567430 号这 8 个扇区中查找到。请回答下列问题(注: 素材文件名为 zy5_18.vhd)。

(1) 你怎样判断这 8 个扇区中哪 4 个扇区存储的是 FAT32_DBR? 哪 4 个扇区存储的是 FAT32_DBR 备份?

(2) 请将这 4 个分区对应 FAT32_DBR 和 FAT32_DBR 备份所在扇区号填入到表 5.30 对应单元格中，并分别计算出前 3 个分区的结束扇区号以及对应逻辑盘的容量；并将结果填入到表 5.30 对应单元格中(提示：由于 4 个 MBR 分区表均存储整个硬盘 0 号扇区，4 个分区的划分是尾首相连的，即第 1 个分区结束扇区号的下 1 个扇区号为第 2 个分区的开始扇区号)。

表 5.30 某硬盘 FAT32 DBR 及 FAT32 DBR 备份所在扇区号

分区	FAT32_DBR 在整个硬盘扇区号	FAT32_DBR 备份 在整个硬盘扇区号	分区结束 扇区号	容量 (单位: MB)
第 1 个				
第 2 个				
第 3 个				
第 4 个				221

(3) 第 4 个分区的容量为 221MB, 请计算出第 4 个分区的结束扇区号, 并将结果填入到表 5.30 对应单元格中。

(4) 请根据 4 个 FAT32_DBR 所在扇区号,计算出 4 个分区表的相对扇区,并将结果填入到表 5.31 对应单元格中。

(5) 请根据 4 个 FAT32 文件系统开始扇区号和结束扇区号,计算出 4 个分区总扇区数,并将结果填入到表 5.31 对应单元格中。

表 5.31 某硬盘 0 号扇区 4 个分区表的相对扇区和总扇区数

分区表	分区标志	相对扇区			总扇区数		
		十进制	十六进制	存储形式	十进制	十六进制	存储形式
第1个	0B						
第2个	0B						
第3个	0B						
第4个	0B						

(6) 请根据表 5.31 中的相对扇区和总扇区数的存储形式,写出存储在硬盘 0 号扇区偏移 0X01BE~0X01FD 处的 4 个分区表,并将结果填入到表 5.32 对应单元格中。

表 5.32 某硬盘 0 号扇区的 4 个分区表

分区表	扇区偏移	存储在硬盘0号扇区的分区表(存储形式)												
第1个	0X01BE~0X01CD	80	01	01	00	0B	FE	FF	FF					
第2个	0X01CE~0X01DD	00	01	01	00	0B	FE	FF	FF					
第3个	0X01DE~0X01ED	00	01	01	00	0B	FE	FF	FF					
第4个	0X01EE~0X01FD	00	01	01	00	0B	FE	FF	FF					

(7) 最后将表 5.32 中的 4 个分区表填入到整个硬盘 0 号扇区偏移 0X01BE~0X01FD 处,然后存盘并退出 WinHex(实际操作题)。

5.19 某硬盘 0 号扇区的 4 个 MBR 分区表所对应的文件系统均为 FAT32，由于用户操作不慎将 4 个 MBR 分区表删除。并且在每个 FAT32 文件系统中，要么是 FAT32_DBR 被破坏、要么是 FAT32_DBR 备份被破坏。用户使用 WinHex 软件，通过 FAT32_DBR 的特征值向下查找 FAT32_DBR，分别在 134 号、204934 号、389248 号和 716934 号这 4 个扇区中查找到，从这 4 个扇区中获得的隐藏扇区数存储形式分别为“80 00 00 00”“80 20 03 00”“80 F0 05 00”和“80 F0 0A 00”。请回答下列问题（注：素材文件名为 zy5_19.vhd）。

(1) 你怎样判断这 4 个扇区中存储的是 FAT32_DBR 还是 FAT32_DBR 备份?

(2) 请通过这 4 个扇区中存储的 FAT32_DBR 或者 FAT32_DBR 备份计算出对应的 FAT32_DBR 备份或者 FAT32_DBR 扇区号，并将结果填入到表 5.33 对应单元格中，分别计算出前 3 个分区的结束扇区号以及前 3 个分区所对应逻辑盘的容量；并将结果填入到表 5.33 对应单元格中(提示：由于 4 个 MBR 分区表均存储整个硬盘 0 号扇区，4 个分区的划分是尾首相连的)。

(3) 第 4 个分区对应的容量为 98MB, 请计算出第 4 个分区的结束扇区号, 并将结果填入到表 5.33 对应单元格中。

表 5.33 某硬盘 FAT32 DBR 及 FAT32 DBR 备份所在扇区号

分区	FAT32_DBR 在整个硬盘扇区号	FAT32_DBR 备份 在整个硬盘扇区号	结束扇区号	容量 (单位: MB)
第 1 个				
第 2 个				
第 3 个				
第 4 个				98

(4) 请根据 4 个 FAT32_DBR 所在扇区号,计算出 4 个分区表的相对扇区,并将结果填入到表 5.34 对应单元格中。

表 5.34 某硬盘 0 号扇区 4 个分区表中的相对扇区和总扇区数

(5) 请根据 4 个 FAT32 文件系统开始扇区号和结束扇区号, 计算出 4 个分区总扇区数, 并将结果填入到表 5.34 对应单元格中。

(6) 请根据表 5.34 中的相对扇区和总扇区数的存储形式, 写出存储在硬盘 0 号扇区偏移 0X01BE~0X01FD 处的 4 个分区表, 并将结果填入到表 5.35 对应单元格中。

表 5.35 某硬盘 0 号扇区的 4 个分区表

分区表	扇区偏移	存储在硬盘 0 号扇区的分区表(存储形式)													
第 1 个	0X01BE~0X01CD	80	01	01	00	0B	FE	FF	FF						
第 2 个	0X01CE~0X01DD	00	01	01	00	0B	FE	FF	FF						
第 3 个	0X01DE~0X01ED	00	01	01	00	0B	FE	FF	FF						
第 4 个	0X01EE~0X01FD	00	01	01	00	0B	FE	FF	FF						

(7) 请将表 5.35 中的 4 个分区表填入到整个硬盘 0 号扇区偏移 0X01BE~0X01FD 处, 然后存盘(实际操作题)。

(8) 如果用户查找到的是 FAT32_DBR 备份, 请通过 FAT32_DBR 备份恢复对应的 FAT32_DBR; 如果用户查找到的是 FAT32_DBR, 请通过 FAT32_DBR 恢复对应的 FAT32_DBR 备份, 然后存盘并退出 WinHex(实际操作题)。