

# 重建分区表与 FAT32\_DBR 研究与实现

陈培德<sup>1,2</sup>, 吴建平<sup>1,2</sup>, 王丽清<sup>1,2</sup>

(1. 云南大学 信息学院, 云南 昆明 650223;

2. 云南省高校数字媒体技术重点实验室, 云南 昆明 650223)

**摘要:**通过对硬盘 MBR 分区表结构、FAT32 文件系统整体布局以及 FAT32\_DBR 结构的分析,以实验的方式对硬盘 MBR 分区表、FAT32\_DBR 以及 FAT1 同时被破坏后进行修复研究。提出了恢复硬盘数据的两种方法:修复硬盘 FAT32\_DBR 和 FAT1 以及修复硬盘 MBR 分区表、FAT32\_DBR 和 FAT1。实验结果表明:当硬盘 MBR 分区表、FAT32\_DBR 和 FAT1 同时被破坏后,可以采用这两种方法中的一种来恢复 FAT32 文件系统的全部数据,解决了由于硬盘 MBR 分区表、FAT32\_DBR 和 FAT1 同时被破坏后数据难以恢复的难题。方法一只能恢复只有一个 MBR 分区表中的数据,而方法二可以恢复多个 MBR 分区表中的数据。

**关键词:** FAT32 文件系统;分区表;子目录;文件分配表

中图分类号: TP311.12

文献标识码: A

文章编号: 1673-629X(2016)10-0188-04

doi: 10.3969/j.issn.1673-629X.2016.10.041

## Research and Implementation of Rebuilding Partition Table and FAT32\_DBR

CHEN Pei-de<sup>1,2</sup>, WU Jian-ping<sup>1,2</sup>, WANG Li-qing<sup>1,2</sup>

(1. School of Information Science and Engineering, Yunnan University, Kunming 650223, China;

2. Key Laboratory of Digital Media Technology of Universities in Yunnan Province, Kunming 650223, China)

**Abstract:** To analyze and research the MBR partition table, the whole layout of FAT32 File System and DBR in the FAT32 File System, the MBR partition table, FAT32\_DBR and FAT1 are repaired when they are damaged at one time by the mode of experiment. The two methods are put forward for recovering the data, the first one is to repair FAT32\_DBR and FAT1 in hard disk, the second is to repair MBR partition table, FAT32\_DBR and FAT1 in hard disk. The results of experiment indicate that one of two methods could be used to recover all data in FAT32 File System. Solved the problem that the data could be recovered difficultly as MBR partition table, FAT32\_DBR and FAT1 damaged at one time. The practice is proved that the first method is only used to recover the all data in a partition table and the second one is to used to recover the all data in many partitions.

**Key words:** FAT32 file system; partition table; subdirectory; file allocation table

## 0 引言

FAT32 文件系统广泛应用于 U 盘、硬盘、移动硬盘、照相机(或摄像机)的 SD 卡、手机的 TF 卡等外存储器上,但是由于用户操作不当、计算机病毒、断电、误操作、振动等各种原因导致 FAT32 文件系统被破坏的现象时有发生,从而引发数据丢失。其中:当 MBR(即主引导记录)分区表、FAT32\_DBR(即 DOS 操作系统的引导记录)和 FAT1(即 FAT32 文件系统的第 1 个文件分配表)同时被破坏后,重建的难度非常大。

当 MBR 分区表被破坏后,在资源管理器中无法看到盘符,使用计算机管理中的磁盘管理查看时,出现

“未分配”的提示<sup>[1]</sup>。

如果分区表完好,而只是 FAT32\_DBR 被破坏,在资源管理器中可以看到盘符,但当单击该盘符时,出现“磁盘未格式化”的提示<sup>[2]</sup>。

经过长期的实验研究,总结了重建 MBR 分区表、FAT32\_DBR 和 FAT1 的基本思路与方法。

## 1 MBR 分区表的基本结构

在 Windows 平台下能够支持的分区结构包括 MBR 分区、GPT 磁盘分区和动态磁盘分区<sup>[1]</sup>。其中 MBR 分区仍然是外存储器的主要分区形式。在 MBR

收稿日期: 2015-12-31

修回日期: 2016-04-21

网络出版时间: 2016-09-26

基金项目: 云南省 2015 年度高校数字媒体技术重点资助项目(2015KFKT002)

作者简介: 陈培德(1966-),男,工程师,研究方向为文件系统与数据恢复技术。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20160926.0951.002.html>

分区中,整个硬盘 0 扇区由主引导记录、磁盘签名、分区表和结束标志四部分组成。其中,最重要的就是分区表。在硬盘 0 号扇区中最多只能存放 4 个分区表。每个分区表项长度为 16 个字节<sup>[2]</sup>,分为 6 部分,说明如下:

- (1) 引导标志占 1 个字节,其取值为 0x00 或 0x80,如该分区不引操作系统,则取值为 0x00<sup>[3]</sup>。
- (2) 在 CHS(即柱面、磁头和扇区)存储方式下作为起始地址占 3 个字节,而在 LBA(即逻辑块存取方式)下这 3 个字节未定义,可以使用任意数据,目前 Windows 操作系统下硬盘使用的存取方式为 LBA<sup>[4]</sup>。
- (3) 分区标志占 1 个字节,FAT32 文件系统为 0x0C 或者 0x0B。
- (4) 在 CHS 存储方式下作为结束地址占 3 个字节,而在 LBA 下这 3 个字节未定义,可以使用任意数据<sup>[5]</sup>。
- (5) 相对扇区占 4 个字节,即分区表到 DBR 扇区数。
- (6) 总扇区数占 4 个字节,即分区所占扇区总数<sup>[6]</sup>,如果分区为 FAT32 文件系统,总扇区数等于 FAT32\_DBR 中的总扇区数。

## 2 FAT32 整体布局及 FAT32\_DBR 结构

从整体结构上看,FAT32 文件系统由保留扇区(注:DBR 位于保留扇区的第 1 个扇区,即 FAT32 文件系统的 0 号扇区)、FAT 表(包括 FAT1 和 FAT2)和数据区三部分组成。FAT32 文件系统的总体布局<sup>[7]</sup>如图 1 所示。

保留扇区	FAT 表		数据区
	FAT1	FAT2	
0 号扇区、1 号扇区、…	…	…	…
			2 号簇、3 号簇、…

图 1 FAT32 文件系统的总体布局

从图 1 可知,FAT32\_DBR 位于 FAT32 卷的 0 号扇区,占一个扇区。一般情况下,在 FAT32 文件系统的 6 号扇区存储着 FAT32\_DBR 备份<sup>[8]</sup>。

FAT32\_DBR 在 FAT32 文件系统占有极其重要的地位,在 FAT32\_DBR 中存储着 FAT32 文件系统的重要参数<sup>[9]</sup>。如:每个扇区的字节数、每个簇的扇区数、保留扇区数、每个 FAT 表所占扇区数等,这些参数一旦被破坏或不正确,FAT32 卷将会出现这样或那样一些故障,并显示在屏幕上。如:出现“磁盘未格式化”、“文件或目录损坏无法读取”、“此卷不包含可识别的文件系统”以及“目录出现乱码”等提示信息<sup>[10]</sup>。

FAT32\_DBR 完成对 FAT32 文件系统参数的定义和引导系统的作用,FAT32\_DBR 结构定义如表 1 所示<sup>[11-12]</sup>。

表 1 FAT32\_DBR 的结构

字节位移	字节数	含义
0x00	2	跳转指令
0x01	1	空操作指令
0x03	8	厂商标志和操作系统版本号
0x0B	2	字节数/扇区
0x0D	1	扇区数/簇
0x0E	2	保留扇区数
0x10	1	FAT 表的个数,一般为 2
0x11	2	未用,一般为 00
0x13	2	未用,一般为 00
0x15	1	媒体描述符
0x16	2	未用,一般为 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 所占扇区数
0x28	2	扩展标志
0x2A	2	文件系统版本
0x2C	4	根目录起始簇的簇号(一般为 2)
0x30	2	文件系统信息扇区号,一般为 1
0x32	2	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
0x40	1	物理驱动器号
0x41	1	保留,一般为 00
0x42	1	扩展引导标签,一般为 0x28 或 0x29
0x43	4	卷标序号
0x47	11	一般为“NO NAME”
0x52	8	一般为“FAT32”
0x53	427	引导代码
0x1FE	2	有效结束标志

一般情况下,重建 FAT32\_DBR 所需计算的参数见表 2,其他参数可以使用同一版本的 FAT32 系统中的 DBR 参数<sup>[13]</sup>。

表 2 重建 FAT32\_DBR 所需计算的参数(1)

字节位移	字节数	含义
0x0D	1	扇区数/簇
0x0E	2	保留扇区数
0x20	4	该分区总扇区数
0x24	4	每 FAT 所占扇区数

## 3 重建 FAT1 的基本思路和方法

当 FAT32 文件系统的 FAT1 被破坏后,可以通过

FAT32 文件系统的 FAT2 来恢复,其思路如下:

- (1)查找第 1 个子目录所在位置,并记录下扇区号和所在簇号;
- (2)查找第 2 个子目录所在位置,并记录下扇区号和所在簇号;
- (3)通过这两个子目录所在扇区号和所在簇号,计算出每个簇的扇区数;
- (4)计算出 2 号簇所在扇区号<sup>[14]</sup>,即 FAT32 根目录的开始扇区号;
- (5)查找并记录下 FAT2 开始扇区号;
- (6)通过 2 号簇所在扇区号和 FAT2 的开始扇区号,计算出每个 FAT 表所占扇区数;
- (7)通过每个 FAT 表所占扇区数和 FAT2 的开始扇区号,计算出 FAT1 的开始扇区号,并通过 FAT2 恢复 FAT1<sup>[15]</sup>。

下面以实验的方式讨论重建 FAT32 文件系统 FAT1 的基本方法:

- (1)使用 WinHex 软件菜单栏上的工具→打开物理磁盘功能打开该硬盘。
- (2)查找第 1 个子目录在硬盘位置,在整个硬盘 8 328 号扇区找到,如图 2 所示。从图 2 可知,第 1 个子目录在 FAT32 的簇号为 0x00000003(即 3)。

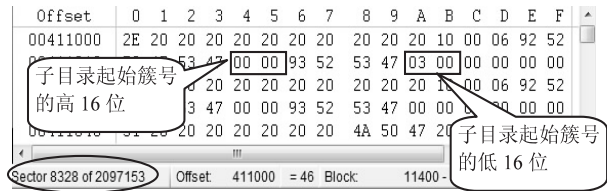


图 2 第 1 个子目录所在硬盘中的位置

- (3)查找第 2 个子目录在硬盘位置,在整个硬盘 14 600 号扇区找到,第 2 个子目录在 FAT32 的簇号为 0x0313(即 787)。
- (4)每个簇的扇区数等于两个子目录所在硬盘开始扇区号之差除以两个子目录在 FAT32 开始簇号之差。
- 每个簇的扇区数 =  $(14\ 600 - 8\ 328) \div (787 - 3) = 8$
- (5)假设 2 号簇在硬盘的开始扇区号为  $x$ ,由  $(14\ 600 - x) \div (787 - 2) = 8$ ,计算出  $x = 8\ 320$ ,FAT2 的结束扇区号为 8 319。
- (6)查找 FAT2 在硬盘的开始扇区号,在 6 285 号扇区找到,如图 3 所示。

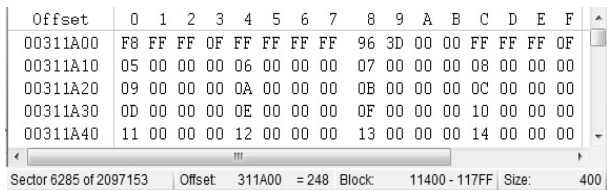


图 3 数据区 2 的开始扇区号所在位置

每个 FAT 表所占扇区数 = FAT2 结束扇区号 - FAT2 开始扇区号 + 1 =  $8\ 319 - 6\ 285 + 1 = 2\ 035$

FAT1 开始扇区号 = FAT2 开始扇区号 - 每个 FAT 表所占扇区数 =  $6\ 285 - 2\ 035 = 4\ 250$

(7)将光标移动到 6 285 号扇区的开始位置处定义块首,将光标移动到 8 319 号扇区最后一个字节处定义块尾。单击“复制”按钮。

(8)将光标移动到 FAT1 开始扇区(即 4 250 号扇区)的开始位置处,单击“粘贴”按钮,并存盘,即可恢复 FAT1。

## 4 重建 FAT32\_DBR 基本思路和方法

由于分区表被破坏,无法从分区表中获得 FAT32\_DBR 在硬盘中的位置,也无法从分区表中获得 FAT32\_DBR 中的总扇区数。

因此,FAT32\_DBR 的位置可以放置在硬盘 0 号扇区至 FAT1 之间的任一扇区。如果将 FAT32\_DBR 存放在硬盘 0 号扇区,说明该硬盘没有分区表,此时 FAT32\_DBR 中保留扇区数为 FAT1 开始扇区前的扇区数;如果 FAT32\_DBR 不存放在硬盘的 0 号扇区,说明该硬盘存在分区表,还需要恢复分区表。

FAT32\_DBR 中的总扇区数可以通过每个 FAT 所占扇区数和每个簇的扇区数估算出来。

重建 FAT32\_DBR 可以分为两种情况来考虑。

情况一:将 FAT32\_DBR 存放在硬盘的 0 号扇区,其方法如下:

- (1)由于每 FAT 表项占 4 个字节,而每个扇区为 512 个字节,所以每个扇区可以容纳 128 个 FAT 表项。
- (2)该 FAT32 文件系统每个 FAT 表占 2 035 个扇区,所以该 FAT 表可以容纳 260 480 个 FAT 表项,由于 FAT32 文件系统的开始簇号为 2,所以该 FAT32 的最大结束簇号为 260 479。

可以估算该 FAT32 文件系统的总簇数为 260 469,比最大结束簇号少 10 个簇,所以数据区所占簇号范围为 2 ~ 260 470,共计 260 469 个簇,由于每个簇的扇区数为 8,所以,数据区所占扇区数为 2 083 752。

(3)由 FAT32\_DBR 存放在硬盘的 0 号扇区,所以保留扇区数为 FAT1 的开始扇区号,即 4 250。

(4)DBR 中总扇区数 = 数据区所占扇区数 + 每个 FAT 所占扇区数 × FAT 表的个数 + 保留扇区数 =  $2\ 083\ 752 + 2\ 035 \times 2 + 4\ 250 = 2\ 092\ 072$ 。

(5)将计算出的每个簇的扇区数、保留扇区数等值,转换为在 FAT32\_DBR 中的存储形式,如表 3 所示。

(6)将同一版本的 FAT32\_DBR 复制到硬盘的 0 号扇区,并将每个簇的扇区数、保留扇区数等值修改为表 3 中的值后存盘,如图 4 所示。

表 3 重建 FAT32\_DBR 所需计算的参数(2)

字节位移	字节数	含义	在 DBR 中的存储形式
0x0D	1	扇区数/簇	08
0x0E	2	保留扇区数	9A 10
0x20	4	该分区总扇区数	28 EC1F 00
0x24	4	每 FAT 所占扇区数	F307

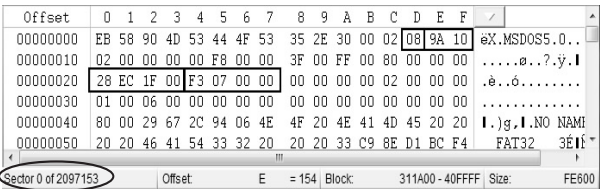


图 4 修改 FAT32\_DBR 中的参数(1)

重新启动 Windows 操作系统后,在资源管理器中即可看该硬盘的盘符,打开该硬盘即可看到恢复出来的文件。

情况二:FAT32\_DBR 没有存放在硬盘的 0 号扇区,其方法如下:

方法(1)、(2)同情况一。

(3)由于 FAT32\_DBR 没有存放在硬盘的 0 号扇区,可以假设 FAT32\_DBR 存放在硬盘的 128 号扇区(即 FAT32 的 0 号扇区,当然也可以存储在其他扇区,但该值必须小于 FAT1 开始扇区号)。

FAT32\_DBR 中的保留扇区 = FAT1 的开始扇区号 - FAT32\_DBR 所在扇区号 = 4 250 - 128 = 4 122

(4)FAT32\_DBR 中总扇区数 = 保留扇区数 + 每个 FAT 表所占扇区数 × FAT 表的个数 + 数据区所占扇区数 = 4 122 + 2 035 × 2 + 2 083 752 = 2 091 944。

(5)将计算出每个簇的扇区数、保留扇区数等值,转换为在 DBR 中的存储形式,如表 4 所示。

表 4 重建 FAT32\_DBR 所需计算的参数(3)

字节位移	字节数	含义	在 DBR 中的存储形式
0x0D	1	扇区数/簇	08
0x0E	2	保留扇区数	1A 10
0x20	4	该分区总扇区数	A8 EB1F 00
0x24	4	每 FAT 所占扇区数	F307

(6)将同一版本的 FAT32\_DBR 复制到硬盘的 128 号扇区,并将每个簇的扇区数、保留扇区数等值修改为表 4 中的值后存盘,如图 5 所示。

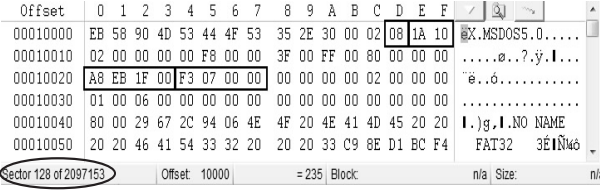


图 5 将数据复制到 FAT32\_DBR 中的参数(2)

5 重建 MBR 分区表基本思路和方法

针对情况一,不需要重建 MBR 的分区表即恢复出 FAT32 文件系统中的所有数据。

针对情况二,还需要重建 FAT32 文件系统 MBR 分区表,其基本思路如下:

从 FAT32\_DBR 所在扇区得到分区表的相对扇区和 FAT32 文件系统的总扇区数。

由图 5 可知,FAT32\_DBR 在整个硬盘的 128 号扇区,在分区表中的存储形式为“80 00 00 00”;而总扇区数为 2 091 944,在分区表中的存储形式为“A8 EB 1F 00”。

由于目前硬盘的存储形式均为 LBA,所以起始地址和结束地址均未定义,可以为任意值。

FAT32 文件系统的分区标志为 0X0C。

由于此硬盘不是引导盘,所以引导标志为 00。

由此可以计算出 NTFS 文件系统的分区表如下:

0001 01 00 07 FE FF FF 80 00 00 00 A8 EB 1F 00

将该值填入到整个硬盘的 0 号扇区偏移地址 0X01BE ~ 0X01CD 处后,存盘并退出 WinHex。

重新启动 Windows 操作系统后,在资源管理器中即可看该硬盘的盘符,打开该硬盘即可看到恢复出来的文件。

6 结束语

文中提出两种方法的核心在于通过分析 MBR 分区表和 FAT32 文件系统的结构,根据子目录的开始扇区号和开始簇号计算出每簇的扇区数,并计算 FAT32 文件系统 2 号簇(即根目录的开始簇号)的开始扇区号,从而计算出每个 FAT 表所扇区数,估算出 FAT32 的总扇区数,根据这些参数计算出 FAT32\_DBR 中的相应参数,最终恢复了 FAT32 文件系统的全部数据;用户可以通过资源管理器看到恢复出来的全部数据。通过实践证明,该方法不仅方便快捷,而且简单实用。

参考文献:

[1] 刘 伟.数据恢复技术深度揭秘[M].北京:电子工业出版社,2010.

[2] 张钟澍,陈代军,李新萌.修复和维护你的硬盘[M].北京:北京希望电子出版社,2002:44.

[3] Schulman A,Michels R J,Kyle J,et al.Undocumented DOS a programmer's guide to reserved MS-DOS functions and data structures[M].New York:Addison-Wesley Publishing Company,Inc,1990:129.

[4] 马 林.数据重现:文件系统原理精解与数据恢复最佳实践[M].北京:清华大学出版社,2009:44.



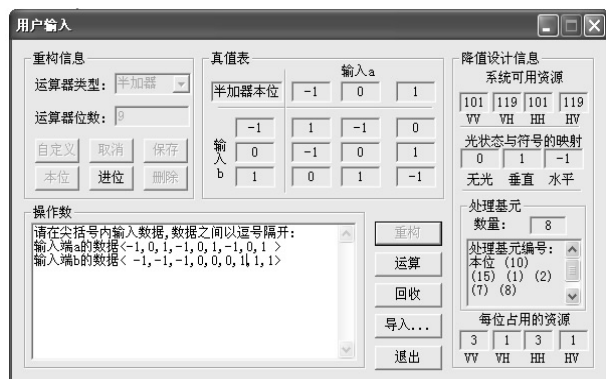


图 6 监控系统运行界面

嵌入式系统控制液晶以及解码器,以 ARM7 嵌入式开发板为平台,用 C 语言和汇编语言开发。它根据监控系统传送过来的控制信息和重构指令控制编码器、运算器以及解码器的工作,并将运算器得到的运算结果传送给监控系统。

## 5 结束语

文中在降值设计理论的指导下设计了三值光计算机实验系统,该系统不仅证明了三值逻辑光计算机原理以及降值设计理论的正确性,更为下一步实用型三值光计算机的研制奠定了坚实的基础。

## 参考文献:

- [1] 金 翊,何华灿,吕养天. 三值光计算机的基本原理[J]. 中国科学:E 辑,2003,33(2):111-115.
- [2] 李 梅,金 翊,何华灿,等. 基于三值逻辑光学处理器实现向量矩阵乘法[J]. 计算机应用研究,2009,26(8):2839-2841.
- [3] 孙 浩,金 翊,严军勇. 三值光计算机编码器与解码器原

理的实验研究[J]. 计算机工程与应用,2004,40(16):82-83.

- [4] 严军勇,金 翊,孙 浩. 三值光计算机多位编码器与解码器的可行性实验研究[J]. 计算机工程,2004,30(14):175-177.
- [5] 黄伟刚,金 翊,艾丽蓉,等. 三值光计算机百位编码器的设计与构造[J]. 计算机工程与科学,2006,28(4):139-142.
- [6] 包九龙,金 翊,蔡 超. 三值光计算机百位量级编码器的实现[J]. 计算机技术与发展,2007,17(2):19-22.
- [7] 李 军,金 翊,尹逊玮. C51 单片机在三值光计算机编码器中的应用[J]. 计算机技术与发展,2008,18(9):180-182.
- [8] Jin Yi, He Huacan, Ai Lirong. Lane of parallel through carry in ternary optical adder[J]. Science in China: Information Sciences, 2005, 48(1):107-113.
- [9] Zuo K Z, Jin Y, Xue T. Tri-state light decoder of ternary optical computer[C]//Proceedings of the 3rd international conference on computer science & education. Xiamen: [s. n.], 2008:442-444.
- [10] Wang H J, Zhou Y, Jin Y. Design and implementation of 1 pixel bit reconfigurable ternary optical processor[J]. Journal of Shanghai University, 2011(5):430-437.
- [11] Huang H X, Masahide I, Toyohiko Y. Modified signed-digit arithmetic based on redundant bit representation[J]. Applied Optics, 1994, 33(26):6146-6156.
- [12] Li Mei, He Huacan, Jin Yi, et al. An optical method for MSD addition[J]. Acta Photonica Sinica, 2010, 39(6):1053-1057.
- [13] 严军勇,金 翊,左开中. 无进(借)位运算器的降值设计理论及其在三值光计算机中的应用[J]. 中国科学:E 辑, 2008, 38(12):2112-2122.

(上接第 191 页)

- [5] 汪中夏,张京生,刘 伟. RAID 数据恢复技术揭秘[M]. 北京:清华大学出版社,2010:103-105.
- [6] 戴士剑,涂彦晖. 数据恢复技术[M]. 北京:电子工业出版社,2005:226.
- [7] Ivens K, Gardinier K. Windows2000: the complete reference 2001[M]. Beijing: China Machine Press, 2001:531.
- [8] 刘乃琦,郭建东,张 可. 系统与数据恢复技术[M]. 成都:电子科技大学出版社,2008:77.
- [9] 陈培德,张共胜. 微机组装与维修实用技术教程[M]. 成都:电子科技大学出版社,1999.
- [10] 郭文武,李宗远,于 泳. 硬盘数据恢复及维修[M]. 北京:

国防工业出版社,2015.

- [11] 李晓中,乔 晗,马 鑫,等. 数据恢复原理与实践[M]. 北京:国防工业出版社,2011.
- [12] 迅维网,罗 工. 硬盘维修及数据恢复不是事儿[M]. 北京:电子工业出版社,2015.
- [13] 韩松峰,常俊超. 数据恢复技术与应用[M]. 北京:电子工业出版社,2014.
- [14] Hogan T. The programmer's PC sourcebook[M]. Redmond Washington, USA: Microsoft Press, 1988:60.
- [15] Solomon D A. Inside Windows NT[M]. 2nd ed. Redmond Washington, USA: Microsoft Corporation, 1998:330.