

# IBM PC 微计算机中被误删除文件的恢复

刘 洵 于 前 洋

**摘要:** 本文提出了恢复IBM PC/XT和PC/AT中, 被误删除文件的恢复方法。指出了文件能够完全恢复或部分恢复的条件。用例子说明了软磁盘和硬磁盘上恢复文件的具体步骤。

## 一、问题的提出

目前, IBM PC/XT及IBM PC/AT 微计算机的应用日益广泛。有些用户在使用过程中, 因操作不慎会偶而误删了无备份的有用文件。若这些文件是在以前的某一特定条件下建立的, 而目前这些条件已无法恢复, 又没有留下文件副本, 这样不仅使用户前功尽弃, 有时还会造成不可挽回的信息损失和巨大的时间上的浪费。因此, 探讨恢复IBM PC微机中被误删文件的方法, 正为广大用户所关注。文献<sup>[1-4]</sup>提出了恢复小容量文件的方法。文献<sup>[5]</sup>提出了恢复大容量文件的方法, 可惜无汇编程序文本可供用户直接使用。而且, 所有上述文献均未提及PC/AT上的文件恢复问题。本文提出了用DEBUG恢复PC/XT, PC/AT上误删文件的方法, 并用实例说明了具体的操作步骤。为便于用户使用, 本文也系统地叙述了文件的存放方法等有关问题。

## 二、文件在磁盘上的存放方法

如何把已删除的文件重新恢复起来, 这与磁盘文件的管理方式有关。因此, 在介绍文件恢复方法之前, 应首先了解 IBM PC DOS 文件管理系统的有关知识。

### 1. 磁盘空间的分配

在 IBM PC机上使用的软盘或硬盘经DOS格式化后, 采用的都是每扇区512字节的标准格式, 而且磁盘空间(对于软盘来说是整个软盘, 对于硬盘是指DOS分区)被分为四个区域:

- (1) 引导记录区
- (2) 文件分配表(FAT)区(有两个副本)
- (3) 根目录区
- (4) 数据区

这四个区域的大小及在磁盘上的实际位置, 对于软盘和硬盘来说是不同的, 如表1所示。其中, PC/AT的硬盘上的文件结构, 是根据引导记录中的BPB(BIOS参数块)参数得来的。

表 1 磁盘上文件的分区

| 介 质                                  | 各 区 的 长 度 |      |      | 各区的起始位置 (逻辑扇区号) |      |     |     | 扇区/簇 |
|--------------------------------------|-----------|------|------|-----------------|------|-----|-----|------|
|                                      | 引导区       | FAT区 | 目录区  | 引导区             | FAT区 | 目录区 | 数据区 |      |
| 5 <sup>1</sup> / <sub>4</sub> in单面软盘 | 1 扇区      | 2 扇区 | 4 扇区 | 0               | 01H  | 03H | 07H | 1    |
| 5 <sup>1</sup> / <sub>4</sub> in双面软盘 | 1 扇区      | 4 扇区 | 7 扇区 | 0               | 01H  | 05H | 0CH | 2    |
| PC/XT上的10M硬盘                         | 1 扇区      | 16扇区 | 32扇区 | 0               | 01H  | 11H | 31H | 8    |
| PC/AT上的20M硬盘                         | 1 扇区      | 82扇区 | 32扇区 | 0               | 01H  | 53H | 73H | 4    |

## 2. 磁盘文件的管理

为了有秩序地管理文件，IBM PC 的文件管理系统特别在磁盘上设立了文件分配表和根目录，用来记载磁盘空间的分配状态及文件的有关信息，以便于对磁盘文件进行操作。

根目录区中存放着每个文件的目录项，每个目录项为32个字节，其字节含义如下：

0—7字节：文件名。这个字段的首字节还指示此文件的状态，如第一字节为：

0 0H表示未曾用过的目录项；

E 5H表示此文件已被删除；

2 EH表示是一个子目录项。

8—10字节：文件扩展名

11 字 节：文件的属性

12—21字节：保留

22—23字节：文件建立的时间

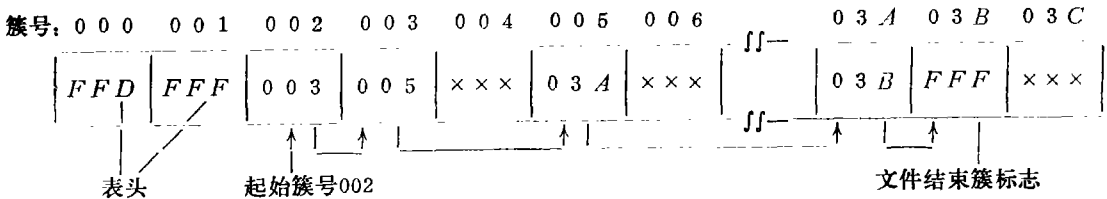
24—25字节：文件建立的日期

26—27字节：文件起始簇号

28—31字节：文件所占的字节数

每当在磁盘上建立一个文件时，DOS 都要在目录区中为此文件设立一个目录项，并且在数据区中为文件分配存贮空间。存贮空间是以簇为单位分配的。一次分配一簇，如果文件长度超过一簇的容量，就要给它分配两簇或者更多簇，这样就产生了簇与簇之间的链接问题。这种链接是通过文件分配表来实现的。

文件分配表是一个以顺序簇号为下标的一维数组(起始簇号为000)，它以簇号作为输入，读出的是文件所占用的下一簇号，比如：



在IBM PC/XT上使用的磁盘(软盘和硬盘)每个簇号为12 bit(1.5字节)，PC/AT机硬盘每个簇号为16 bit(2字节)。

DOS 根据文件分配表就能知道哪些空间未用，哪些已占用。它总是把未占用的最小簇

号分配给新建立的文件，并将跳过已占用的空间之后的自由空间继续分配给此文件。因而，文件并不一定是存放在连续的扇区里。

文件分配表中的每一个簇号都与磁盘上的扇区有相应的对应关系，对于软盘和不同的硬盘，其簇号与扇区转换关系如下：

5<sup>1/4</sup>in双面软盘：扇区号 = (簇号 - 2) × 2 + 0CH

PC/XT 10M硬盘：扇区号 = (簇号 - 2) × 8 + 31H

PC/AT 20M硬盘：扇区号 = (簇号 - 2) × 4 + 73H

### 三、误删除文件的状态及恢复的可能性

当删除磁盘上的文件时，DOS要进行下列操作：1)把被删文件目录项的首字节置为E5H(字母“e”的ASCII码)；2)修改文件分配表，把该文件曾占用的簇号全部置为0，释放为自由空间。

由此可见，文件刚被删除后，存贮在文件数据区中的文件实体并没有被破坏。而且，寻找文件的一个重要线索—文件起始簇号仍原样保留在此文件目录中，这就为恢复文件提供了可能性。然而，这个可能性是在误删文件之后没有进行写盘操作这个前提下存在的。如果进行了写操作，这样误删文件曾占用的空间就很可能被复盖，文件便无法恢复了。所以，当用户一旦误删了重要文件后，一定不要进行写操作。只要正确掌握文件恢复方法，被误删的文件是可能部分或全部恢复的。

对于不同容量的文件，恢复的难易程度也不同。如果文件的容量较小，只占用一或两个簇号，那么把这几个簇号从自由空间中分离出来并不困难。如果文件容量很大，再加上文件所在的磁盘经常进行建立与删除文件操作，这样就使文件在磁盘上的存放比较零散，恢复起来就比较困难。

### 四、小容量文件的恢复

了解文件在磁盘上的存放方法及被删文件的状态后，就可以利用DOS中的DEBUG程序来恢复文件。为了从易到难，我们首先介绍文件所含字节数小于一簇的小容量文件的恢复方法。其操作过程可概括为两步：1)恢复文件名，将文件目录项的首字节由E5H改成实际文件名第一个字母的ASCII码。2)把文件的起始簇号作为文件分配表的下标号查表，在此单元里填上文件结束簇标志(FFF)。

现以双面软盘上存放的文件为例。

例1：恢复一个长度小于一簇的文件，其文件名为ABC.BAS。

具体操作如下：

1) 在系统盘下调入DEBUG程序

```
C>DEBUG
```

2) 在A驱动器中插入被删文件所在的软盘，键入命令：

```
-L 100 0 1 B
```

把软盘上的文件分配表和根目录调入内存。

3) 用 *D* 命令显示读入的信息, 并可把与被删文件有关的信息打印出来。

- *D* 100 1700

在文件目录区中找到被删文件的目录, 记下目录第一字节的当前地址 *09A0*, 由目录可知文件起始簇号为 *0032*, 簇号  $\times 1.5$  便是此簇号相对文件分配表的偏移地址 (在这里还要加上文件分配表的起始地址 *100H*)。

$$32H \times 1.5 + 100H = 14BH$$

4) 用 *E* 命令修改需要改动的地址单元里的内容:

- *E* 09A0 41

在目录的首字节中填上原文件名的第一个字母 *A* 的 *ASCII* 码。

- *E* 014B FF

- *E* 014C 4F

在簇号所对应的地址单元里填上文件结束簇标志 *FFF* (由于每个簇号占 *1.5* 字节, 所以改写时要涉及到两个地址单元)。

5) 用 *W* 命令把修改后的内容写回磁盘:

- *W* 100 0 1 B

6) 退出 *DEBUG*

- *Q*

到此为止 *ABC.BAS* 文件已被恢复。

注意, 为避免在写入新文件时将 *ABC.BAS* 文件复盖, 还应当将此文件复制到另一张软盘上, 然后删除原来软盘上的这一文件。最后, 再把刚才复制到另一软盘上的 *ABC.BAS* 文件复制回来<sup>[1]</sup>。

*PC/XT* 硬盘上的文件恢复方法与软盘相同, 区别只是硬盘上的文件分配表与根目录在磁盘上的位置与软盘不同, 因此在把它们调入内存时命令如下:

- *L* 100 2 1 30

写盘命令也相应为:

- *W* 100 2 1 30

上述的文件恢复方法也同样适用于 *PC/AT* 机上的 *20M* 硬盘, 关于它的文件分配表与根目录位置参阅表 1。由于这两部分较长可把它们分别调盘, 先修改目录项后再修改文件分配表。

## 五、大容量文件的恢复

大容量文件恢复的基本思想, 就是把被删文件曾占用的簇链从自由空间中分离出来。由于用户并不知道此文件原来是否存贮在连续的扇区上, 因而只能用试探的方法。先假设文件是连续存放的, 把紧挨着起始簇号的自由空间先分离出来, 然后查看是否为要恢复的文件。如果不是, 就要跳过已占用的空间接下去继续查找。下面以双面软盘和 *PC/AT* 机硬盘为例, 介绍大容量文件的恢复方法。

例 2: 双面软盘上误删的大文件的恢复 (假设该文件名为 *VTT.R.BAS*)

1) 进入 *DEBUG*, 把软盘插入 *A* 驱动器并键入命令:

- *L* 100 0 1 B

把软盘上的文件分配表、根目录调入内存。

```
- D 900 1700
```

显示根目录。

从被删文件目录项中，可查到其目录首字节所在地址  $0A20H$ ，文件起始簇号  $0061H$ ，文件长度  $4280H$ 。由于文件占用的簇数必须是一个整数，即大于  $\lceil \text{文件长度} / (\text{每簇扇区数} \times 512 \text{字节}) \rceil$  的一个整数，所以此文件占用的簇数为：

$$4280H / (2 \times 512D) = 17$$

2) 恢复文件名

```
- E 0A20 56
```

把文件名第一个字母 ( $V$  的  $ASCII$  码为  $56H$ ) 重新写回。

3) 恢复文件分配表

以起始簇号为下标找到它所在的空簇，这个空簇对应文件分配表的偏移地址为：

$$61H \times 1.5 + 100H = 191H$$

从  $191H$  地址起在每个空簇中填入下一空簇的簇号，在第17个空簇里填上文件结束簇标志  $FFF$ 。

由于每个簇号为1.5个字节，读写很不直观，为了正确填写簇号，这里介绍两个规则：

a. 如果簇号乘以1.5为整数，即以整数为文件分配表的偏移地址，其地址单元里装的是12 bit簇号的低8位，下一地址里的低半字节是它的高4位。

b. 如果簇号乘以1.5为小数，则取整后作为偏移地址，此地址里的高半字节是12 bit簇号的低4位，高8位在下一地址单元里。

根据这两个规则我们将簇号061—070中，分别填入062—071的内容，如下表所示：

|           |       |     |     |     |     |     |                               |
|-----------|-------|-----|-----|-----|-----|-----|-------------------------------|
| 簇号        | 060   | 061 | 062 | 063 | 064 | 065 | .....                         |
| 0D40:0190 | FF    | 2F  | 06  | 63  | 40  | 06  | 65 60—06 67 80 06 69 A0 06 6B |
| 簇号        | ..... |     |     |     |     | 070 | 071                           |
| 0D40:01A0 | C0    | 06  | 6D  | E0  | 06  | 6F  | 00 07—71 F0 FF.....           |

用  $E$  命令把上述内容逐个写入内存。

4) 把已经修改的目录和文件分配表写回磁盘

```
- W 100 0 1 B
```

5) 退出  $DEBUG$

```
- Q
```

对恢复后的文件要进行检查，看在此文件中是否夹杂了其它信息。如果混进了其它内容，就要把这部分内容所占的簇号重新置成原来的状态，然后再继续向后寻找应属于此文件的簇号。

此例介绍的方法同样适用于  $PC/XT$  机硬盘。

例3.  $PC/AT$  机硬盘文件的恢复，设被恢复的文件名为  $KYBD.DGS$

1) 在  $DEBUG$  下键入命令：

```
- L 100 2 53 20
```

把根目录调入内存

```
- D 100 4100
```

显示根目录

从目录项中查出被删文件的目录首字节地址为03E0H，文件起始簇号为20FFH，文件长度为1F8CH。

2) 恢复文件名

- E 03E0 4B

先在内存中修改，然后写回磁盘

- W 100 2 53 20

3) 恢复文件分配表

- L 100 2 1 29

把文件分配表调入内存

- D 42C0 4350

显示起始簇号对应文件分配表的偏移地址附近的部分内容

起始簇在文件分配表中的偏移地址为： $20FFH \times 2 + 100H = 42FEH$

由文件长度和每簇所含的字节数(2K)可知此文件需占4簇，即为20FF 2100 2101 2102。因此先在起始簇及之后的几个空簇里依次填入下一空簇的簇号，在第4个空簇里填文件结束簇标志FFFF。修改的部分如下：

|                                  |       |
|----------------------------------|-------|
| 簇号 .....                         | 20FF  |
| 0D40:42F0 .....                  | 00 21 |
| 簇号 2100 2101 2102                |       |
| 0D40:4300 01 21 02 21 FF FF..... |       |

先用E命令把改动的部分写入内存，然后把文件分配表写回磁盘

- W 100 2 1 29

4) 退出DEBUG

- Q

对恢复后的文件检查方法与上例相同，不再赘述。

## 六、讨 论

应当指出，用DEBUG程序来恢复误删文件，具有一定的风险。因为一次不慎的操作，可能影响操作系统中文件的完整性，导致整机无法正常运行。因此必须十分细心，特别是在使用W命令时更要小心。

大容量误删文件的恢复有时需要反复修改文件分配表(FAT)，因此在修改后最好把它的两个副本对照一下，以防在修改过程中误改了其它文件的簇号。

由于恢复文件簇号时通常只改动了文件分配表的第一副本，为可靠起见最好把恢复好的文件先拷贝到其它盘上，然后把它从原来所在的盘上删掉，如有必要再拷贝回来，这样它就和新建的文件一样，所有信息都完整地存在盘上了。

另外，如果使用通配符“\*”或“?”误删了一批文件，这时应先恢复单簇和小容量文件，然后再恢复大容量文件。

### 附示例1~3

例1：PC/XT上单簇文件(ABC.BAS)的恢复

C>DEBUG

-L 100 0 1 B

-D 100 1C0

|           |   |                  |
|-----------|---|------------------|
| 0D40:0100 | FD FF FF 03 40 00 05 60-00 FF BF 00 09 A0 00 0B | .....            |
| 0D40:0110 | C0 00 0D E0 00 0F 00 01-11 20 01 13 40 01 15 60 | .....            |
| 0D40:0120 | 01 17 F0 FF 19 A0 01 1B-C0 01 1D E0 01 1F 00 02 | ..p.....         |
| 0D40:0130 | 21 20 02 23 40 02 25 60-02 27 80 02 29 F0 FF 2B | !.#%&'()*p.+     |
| 0D40:0140 | C0 02 2D F0 FF 2F 00 03-31 F0 FF 00 40 03 35 60 | @-p/..ip..0.5    |
| 0D40:0150 | 03 37 80 03 39 A0 03 3B-C0 03 3D E0 03 3F 00 04 | .7..9.;@.='.?..  |
| 0D40:0160 | 41 20 04 43 F0 FF 45 60-04 47 80 04 49 A0 04 4B | A.Cp.E'.G..I.K   |
| 0D40:0170 | C0 04 4D E0 04 4F 00 05-51 20 05 53 40 05 55 60 | @.M'.D..@.S@.U'  |
| 0D40:0180 | 05 57 80 05 59 A0 05 5B-C0 05 5D E0 05 5F 0F 06 | .W..Y..[@.]'.... |
| 0D40:0190 | FF 2F 06 63 40 06 65 60-06 67 80 06 69 A0 06 6B | /.c@.e'.g...i.k  |
| 0D40:01A0 | C0 06 6D E0 06 6F 00 07-71 F0 FF 73 40 07 75 F0 | @.m'.o..qp.@.up  |
| 0D40:01B0 | FF 00 00 00 00 00 00-00 00 00 00 00 00 00 00    | .....            |
| 0D40:01C0 | 00  |                  |

|           |   |                  |
|-----------|---|------------------|
| 0D40:0900 | 49 42 4D 42 49 4F 20 20-43 4F 4D 27 00 00 00 00 | IBMBIO COM'....  |
| 0D40:0910 | 00 00 00 00 00 00 00 60-54 07 02 00 80 12 00 00 | .....'T.....     |
| 0D40:0920 | 49 42 4D 44 4F 53 20 20-43 4F 4D 27 00 00 00 00 | IBMDOS' COM'.... |
| 0D40:0930 | 00 00 00 00 00 00 00 60-54 07 07 00 80 42 00 00 | .....'T.....B..  |
| 0D40:0940 | 43 4F 4D 4D 41 4E 44 20-43 4F 4D 20 00 00 00 00 | COMMAND COM'.... |
| 0D40:0950 | 00 00 00 00 00 00 00 60-3E 08 18 00 80 45 00 00 | .....'>.....E..  |
| 0D40:0960 | 53 41 4D 50 4C 45 53 20-42 41 53 20 00 00 00 00 | SAMPLES BAS'.... |
| 0D40:0970 | 00 00 00 00 00 00 00 40-0E 09 2A 00 15 00 00 00 | .....@..*.....   |
| 0D40:0980 | 44 4F 4E 4B 45 59 20 20-42 41 53 20 00 00 00 00 | DONKEY BAS'....  |
| 0D40:0990 | 00 00 00 00 00 00 00 40-0E 09 2E 00 00 0E 00 00 | .....@.....      |
| 0D40:09A0 | ES 42 43 20 20 20 20-42 41 53 20 00 00 00 00    | eBC BAS'....     |
| 0D40:09B0 | 00 00 00 00 00 00 79 1F-21 00 32 00 A9 03 00 00 | .....y.!..2.)... |
| 0D40:09C0 | 41 52 57 20 20 20 20-42 41 53 20 00 00 00 00    | ARW BAS'....     |
| 0D40:09D0 | 00 00 00 00 00 00 00 40-0E 09 33 00 80 42 00 00 | .....@..3..B..   |
| 0D40:09E0 | 41 4E 47 45 20 20 20 20-42 41 53 20 00 00 00 00 | ANGE BAS'....    |
| 0D40:09F0 | 00 00 00 00 00 00 00 40-0E 09 44 00 00 69 00 00 | .....@..D..i..   |

例 2: PC/XT 上大容量文件 (VTTR, BAS) 的恢复

DEBUG

-L 100 0 1 B

-D 100 1CF

|           |   |                   |
|-----------|---|-------------------|
| 0D40:0100 | FD FF FF 03 40 00 05 60-00 FF BF 00 09 A0 00 0B | .....             |
| 0D40:0110 | C0 00 0D E0 00 0F 00 01-11 20 01 13 40 01 15 60 | .....             |
| 0D40:0120 | 01 17 F0 FF 19 A0 01 1B-C0 01 1D E0 01 1F 00 02 | ..p.....          |
| 0D40:0130 | 21 20 02 23 40 02 25 60-02 27 80 02 29 F0 FF 2B | !.#%&'()*p.+      |
| 0D40:0140 | C0 02 2D F0 FF 2F 00 03-31 F0 FF 00 40 03 35 60 | @-p/..ip..0.5     |
| 0D40:0150 | 03 37 80 03 39 A0 03 3B-C0 03 3D E0 03 3F 00 04 | .7..9.;@.='.?..   |
| 0D40:0160 | 41 20 04 43 F0 FF 45 60-04 47 80 04 49 A0 04 4B | A.Cp.E'.G..I.K    |
| 0D40:0170 | C0 04 4D E0 04 4F 00 05-51 20 05 53 40 05 55 60 | @.M'.D..@.S@.U'   |
| 0D40:0180 | 05 57 80 05 59 A0 05 5B-C0 05 5D E0 05 5F 0F 06 | .W..Y..[@.]'....  |
| 0D40:0190 | FF 0F 00 00 00 00 00 00-00 00 00 00 00 00 00    | .....@.....       |
| 0D40:01A0 | 00 00 00 00 00 00 00 00-00 00 00 73 40 07 75 F0 | .....@.....up     |
| 0D40:01B0 | FF 77 80 07 FF AF 07 7B-C0 07 7D F0 FF 7F 00 0B | .w.../..[@.]p.... |
| 0D40:01C0 | 81 F0 FF 00 00 00 00 00-00 00 00 00 00 00 00    | p.....            |

|           |   |                 |
|-----------|---|-----------------|
| 0D40:09C0 | 41 52 57 20 20 20 20-42 41 53 20 00 00 00 00    | ARW BAS'....    |
| 0D40:09D0 | 00 00 00 00 00 00 00 40-0E 09 33 00 80 42 00 00 | .....@..3..B..  |
| 0D40:09E0 | 41 4E 47 45 20 20 20 20-42 41 53 20 00 00 00 00 | ANGE BAS'....   |
| 0D40:09F0 | 00 00 00 00 00 00 00 40-0E 09 44 00 00 69 00 00 | .....@..D..i..  |
| 0D40:0A00 | 49 4D 47 50 52 4F 20 20-42 41 53 20 00 00 00 00 | IMPROB BAS'.... |
| 0D40:0A10 | 00 00 00 00 00 00 00 40-0E 09 5F 00 6C 07 00 00 | .....@.....l... |
| 0D40:0A20 | ES 54 54 52 20 20 20 20-42 41 53 20 00 00 00 00 | eITR BAS'....   |
| 0D40:0A30 | C7 00 00 00 00 00 00 40-0E 09 61 00 80 42 00 00 | .....@..a..B..  |
| 0D40:0A40 | 52 41 4D 54 53 20 20 20-42 41 53 20 00 00 00 00 | RAMTS BAS'....  |
| 0D40:0A50 | 00 00 00 00 00 00 00 40-0E 09 72 00 2D 0E 00 00 | .....@..r..7... |

