软盘零磁道故障分析与数据修复

阮高华　　(华资公司武汉分公司)

摘要:本文分析了软盘零磁道故障现象,针对驱动器读盘错误,致使盘内文件无法进行读写操作等问题,论述了修复盘内数据文件的两种方法。即 *DEBUG* 修复法和编程修复法。

一、前言

零磁道故障现象是发生在旧软盘中的一种常见故障,其原因是软盘使用或保管不当所致。故障形式有物理故障和逻辑故障两类。物理故障是零磁道出现介质缺陷,物理性损坏。逻辑故障是扇区格式受到破坏,属于软损坏。根据维修经验,我们发现通常遇到的软盘零磁道故障形式,大部分属于逻辑故障。发生这类故障时,除了使用磁盘格式化命令和 DEBUG 动态调试命令以外,使用任何其它的 DOS 命令时,系统均给出"扇区未找到,A 驱动器读错误"(SECTOR NOT FOUND ERROR READING DRIVE A, ABORT、RETRY、IGNORE)信息,敲 R 键再试也无效,敲 I 键强制执行则数据文件面目全非,故只有击 A 键退出。由于命令不能进入软盘,因此对盘内文件的显示读写等一系列操作均告失败。此时,只有使用格式化命令重新进行格式化处理才能恢复故障盘的启用。然而如此处理盘内文件的损失后果是可想而知的。为了挽救盘内的数据文件,必须要设法将其先移出来,而后进行格式化,最后再将其移回去。为此,笔者介绍两种行之有效的方法来修复盘内数据。一种是借助动态调试程序进行处理的 DEBUG 修复法,另一种是采用 8086/8088 汇编语言程序进行处理的编程修复法。

二、DEBUG 修复法

磁盘文件的完整结构是由文件的目录区和数据区两部分组成的。因此,修复故障盘数据主要分二个步骤,一是移出盘文件目录部分,二是移出文件数据部分。

1.修复盘内文件目录

(1)将故障盘插入 A 驱动器,用 DEBUG 装入命令(LOAD)把盘文件目录部分装入内存中。

```
C>DEBUG
-L  10005  7
```

上面 L 命令表示从 A 驱动器的盘上装入数据,并把数据存放在以 100 单元开始的内存中:从 A 盘的第 5 扇区开始,传送 7 个连续扇区的数据。

(2)将另一张格式化过和软盘(简称非故障盘)插入 A 驱动器中,用写命令(WRITE)将内存中的盘文件目录信息写到非故障盘上。

```
-W10005  7
```

上面 W 命令表示从 100 单元开始的数据写到 A 驱动器的磁盘中,由第 5 扇区开始,连续写 7 个扇区的数据。这里要注意上面 W 命令所带的四个参数一定要与前面 L 命令所带的参数相一致,否则,将导致数据混乱。至此,盘文件目录部分即被移至非故障盘上。

2.修复文件数据

(1)将故障盘插入 A 驱动器中,根据盘文件目录,先用命名命令(NAME)将第一个文件装入到内存中。

```
-NA:×××(文件名)
-L
```

上面 L 命令不必指定参数,表示从默认驱动器的磁盘上,装入由 NAME 命令指定的文件,并把它存放在 100 单元开始的内存中。

(2)将非故障盘插入 A 驱动器中,将内存中的第一个文件内容写到非故障盘上。

```
-W
```

上面 W 命令也不必指定参数,表示把从 100 单元开始的文件以文件说明 FILESPEC 写到默认驱动器(即 A

驱动器)的磁盘上。这样,第一个文件的信息就全部地移至非故障盘上了。

然后,重复进行步骤二,将故障盘上的全部文件均移至非故障盘上。再对故障盘进行格式化操作,最后将各个文件从非故障盘上拷贝回来即可。到此为止,就可以对软盘上的文件进行读写等各种操作了。

DEBUG 修复法简便易行,不受软、硬件环境的限制。但此方法在操作上要直接涉及到磁盘的目录地址、驱动器号和扇区号等参数,操作稍有不慎,后果则不勘设想。为此,再介绍另一种用汇编语言编程修复数据的方法。

三、编程修复法

程序的功能设计主要分为读写目录和读写文件两方面。主要设计步骤如下:

1. 显示功能菜单,接收用户的键选输入。

2. 选择读写目录功能时,提示用户将故障盘插入 A 驱动器,非故障盘插入 B 驱动器,准备完毕后任按一键。

3. 用中断 INT25H 将故障盘中的目录扇区读入内存。

4. 用中断 INT26H 将内存中的目录扇区内容写入非故障盘中。

至此,故障盘中的文件目录扇区内容即修复至非故障盘中,可用 DIR 命令显示查看。当然此时只有文件名,而无实际内容。

5. 选择读写文件功能时,提示输入文件名。

6. 用中断 INT21H,0AH 读取待修复的文件名。

7. 用中断 INT21H,42H 取待修复的文件长度。

8. 用中断 INT21H,3FH 将待修复的文件内容读入内存。

9. 用中断 INT21H,0AH,42H,40H 将文件名、文件长度和文件内容写入非故障盘中。

至此,即完整地修复了一个文件,接着可进行下一个文件的修复操作。

本程序采取汉字菜单驱动,提示信息帮助,操作简单,使用方便。此外程序还具有较好的容错功能,即使出现误操作也不会导致不良后果。该程序已在 PC/XT、长城—286EX 等微机上通过,数据修复效果明显。现将该程序清单介绍给大家,只要将其编译、连接成 EXE 文件,即可执行之。

程序清单如下:

```
stack      segment para stack'stack'
stapn    db 256 dup(?)
top      equ length stapn
stack    ends
data     segment para public'data'
mbuff   db 49
          db ?
          db 50 dup(?)
messl db 30 dup(' '),'数据修复程序',38 dup(; ')
       db 30 dup(' '),'1—读写目录',38 dup(' ')
       db 30 dup(' '),'2—读写文件;,38 dup(' ')
       db 24 dup(' '),'请输入 12(q 键退出):','¥'
dxslp db 15 dnp(' ')读文件, 请输入盘符、文件名:','¥'
dxslpl db 15 dup(' '),'写文件,请输入盘符、文件名:','¥'
softl db 15 dup(' '),'请插入源盘至 A,目的盘至 B,任敲一键继续:','¥'
meseg db 15 dup(' '),'文件名错:请敲一键重新输入!','¥'
handle  dw  ?
datbuffl db 60416 dup (?)
datbuff db 3584 dup (?)
data ends
mcode segment
mproc proc far
       assume  ds:mcode,es:data,es:data
clear macro
      mov ax, 7
      int 10h
emdm
mcal  macro mes.numb,dxfs,wjm,wjc
       mov dx,offset mes
       mov ah,numb
       mov al,dxfs
       mov bx,wjm
       mov cx,wjc
       int 21h
endm
cpdx  macro qdqh,sqh,sqs,dxcz
       mov al,qdqh
```

```
        mov dx,sqh
        mov bx,offset datbuff
        mov cx,sqs
        int dxcz
endm
qwjm macro
        mov dx,offset mbuff
        mov ah,0ah
        int 21h
        mov bl,mbuff+1
        mov bh,0
        mov [mbuff+bx+2],0
endm
qwjc  macro
        mov ah,42h
        mov aL,2
        mov bx,handle
        mov cx,0
        mov dx,0
        int 21h
endm
start: push ds
        xor ax,ax
        push ax
        mov ax,data
        mov ds,ax
        mov es,ax
lop: clear
      mcal messl,9
      mcal 0,1
      cmp al,  q′
      jz exit
      cmp al,  1′
      jz rmla
      jmp open
exit: mcal 0,4ch
      ret
rmla: clear
       mcal soft1,9
       mcal 0,8
       cpdx 0,5,7,25h
       add sp,2
wmlb: cpdx 1,5,7,26h
       add sp,2
       jmp lop
error: clear
       mcal meseg,9
       mcal 0,8
open: clear
       mcal dxslp,9
       qwjm
       mcal mbuff+2,3dh,2
       mov handle,ax
       qwjc
       mov cx,ax
       push cx
       mcal mbuff+2,3dh,0
       jc error
       mov handle,ax
       pop cx
read: mov ah,3fh
       mov bx,hamdle
       mov dx,offset datbuffl
       int 21h
       push cx
write: clear
        mcal dxslp1,9
        qwjm
        mcal mbuff+2,3ch, , ,0
        jc errorl
        mov handle,ax
        qwjc
        pop cx
        mov ah,40h
        mov bx,handle
        mov dx,offset datbuff1
        int 21h
close: mcal 0,3eh,handle
       jmp lop
error1: clear
        mcal meseg,9
        mcal 0,8
        jmp write
mproc endp
mcode ends
        end start
```