硬盘维护与数据恢复

主编　孙维连　于莉莉　单琳娜

Computer

中国水利水电出版社
www.waterpub.com.cn

硬盘维护与数据恢复

主　编　孙维连　于莉莉　单琳娜
副主编　高　晶　何丽丽　张宗利
　　代治国　张　则

中国水利水电出版社
www.waterpub.com.cn

内 容 提 要

本书主要内容包括：调试程序 DEBUG、DOS 的磁盘与文件管理、硬盘的发展与文件结构、BIOS 设置、硬盘分区与格式化、硬盘工具软件(Disk Genius、Ghost、PQ8.0、Winhex)、硬盘维护与数据修复、TC2.0 实用工具软件开发技术等。

本书可作为软件维修工具书，可作为电气类及有关专业的高校教材，也可供从事计算机科研和工程技术工作人员参考。

图书在版编目（CIP）数据

硬盘维护与数据恢复 / 孙维连，于莉莉，单琳娜主编.
北京：中国水利水电出版社，2007
ISBN 978-7-5084-4626-4

Ⅰ. 硬… Ⅱ. ①孙…②于…③单… Ⅲ. ①硬磁盘－高等学校：技术学校－教材②数据管理－安全技术－高等学校：技术学校－教材 Ⅳ. TP333.3 TP309.3

中国版本图书馆 CIP 数据核字（2007）第 071399 号

书　　名	**硬盘维护与数据恢复**
作　　者	主编　孙维连　于莉莉　单琳娜
出版　发行	中国水利水电出版社（北京市三里河路 6 号　100044） 网址：www.waterpub.com.cn E-mail：sales@waterpub.com.cn 电话：（010）63202266（总机）、68331835（营销中心）
经　　售	北京科水图书销售中心（零售） 电话：（010）88383994、63202643 全国各地新华书店和相关出版物销售网点
排　　版	中国水利水电出版社微机排版中心
印　　刷	北京市兴怀印刷厂
规　　格	787mm×1092mm　16 开本　18 印张　427 千字
版　　次	2007 年 6 月第 1 版　2007 年 6 月第 1 次印刷
印　　数	0001—4000 册
定　　价	**38.00** 元

凡购买我社图书，如有缺页、倒页、脱页的，本社营销中心负责调换

前　言

在计算机应用过程中，用户难免会遇到许多技术问题需要查阅大量资料及相关程序。但往往由于资料不足或短时间内找不到相关程序，使问题不能迅速得到解决，进而影响计算机的使用效果。在用户遇到的这些问题当中，大部分与软件维修技术相关。所以，为使用户能迅速地解决这些问题，作者根据多年的教学与实践研究及软件维修经验编成此书，奉献给广大读者。

本书以 DEBUG 下的汇编语言为工具，通过大量实例，深入剖析了 Windows 操作系统内部数据结构、参数与磁盘中断，并给出应用技巧。目前计算机真正的硬件故障不多，但由于软件原因造成的计算机不能正常工作，或对软件故障处理不当，造成重大损失的情况，则随处可见。为此，本书又列举了大量有价值的实例，对计算机软件故障的排除，予以详尽分析 ，并给出了具体解决的方法及步骤，例如：硬盘、软盘软件故障的排除。书中又对计算机的安全技术问题，予以详尽讨论，并给出与计算机安全相关的各种方法策略及具体的计算机软件安全防护措施。

本书针对计算机常用的存储设备，软盘、硬盘及文件系统进行深入剖析，并给出 20 多个 C 语言开发维修工具的实例，应用这些程序，可实现对存储软盘、硬盘、文件等中的数据，就像看书一样，任我们随意去翻阅每一页（扇区）的单词、语句，并根据需要进行浏览、修改或重写。同时，本书还根据应用技术要求深入剖析了大容量硬盘的多种读写技术；并提供维护软盘、硬盘多种技术手段，对软盘硬盘数据丢失，给出多种恢复技术的实践操作。

本书通过对软盘、硬盘结构的分析，给出对硬盘引导扇区、分区表、分区引导扇区丢失或损坏的修复技术，对不能启动的硬盘上恢复文件的方法，对硬盘的保护，对 FAT16. FAT32. 文件系统，都进行了详尽剖析，为解决许多实际问题列举了大量实用性极强的例题。

本书共分 7 章，主要内容包括：调试程序 DEBUG、DOS 的磁盘与文件管

理、硬盘的发展与文件结构、BIOS 设置、硬盘分区与格式化、硬盘工具软件（Disk Genius、Ghost、PQ8.0、Winhex）硬盘维护与数据修复、TC2.0 实用工具软件开发技术。

全书由孙维连、于莉莉、单琳娜任主编，由何丽丽、高晶、张宗利、代治国、张则任副主编。第 1 章由于莉莉编写，第 2 章由单琳娜编写，第 3 章第 3.3、3.4、3.5、3.6、3.7 节由张宗利编写，第 4 章由张则编写，第 5 章由高晶编写，第 6 章第 6.5 节和附录由孙维连编写，第 7 章由何丽丽编写，第 3 章第 3.1、3.2 节，第 6 章第 6.1、6.2、6.3、6.4 节由代治国编写。全书由佳木斯大学公共计算机教研部孙维连教授统稿，张启来主审。王锐、贺雷、石全飞、马晓君、张敬强、戴德伟、高志杨、刘涛、苏晓光等也参与了本书的部分编写工作。作者对参与和支持本书出版的人员，在此表示衷心感谢。

作者竭力想奉献给读者一本既有价值，又有特色的软件维修工具书，但时间仓促，水平所限，书中错误在所难免，敬请广大读者批评指正，愿与广大读者相互交流，提高软件维修的理论与实践技术水平。

作　者

2007 年 6 月 18 日

目　录

第1章 调试程序DEBUG

1.1 引 言

DEBUG是一个DOS实用程序，是软件工作人员在调试程序中时经常使用的工具。DEBUG的功能很强，它包含19个子命令。这些子命令不仅能跟踪运行程序的踪迹，以便了解程序中每条指令的执行结果，从而分析程序流程的正确性；而且能直接与磁盘文件或它的指定扇区进行对话，以便读写磁盘文件或它的某个扇区的具体内容，为显示、分析和修改磁盘文件提供了方便。此外，它还可以显示和修改指定范围的内存内容，以及CPU内部寄存器和标志位的内容等。

因此DEBUG程序可用于以下几方面。

（1）提供一个可控制的测试环境，让用户能监视和控制被调试程序的执行情况，还可在程序中直接指定问题，然后立即执行它，并判定所指问题是否已经解决，因此，不必重新汇编程序即可查看所做的修改是否有效。

（2）装入、修改或显示任何文件（EXE、HEX文件需改名）。

（3）执行目标文件，目标文件是用机器语言格式表示的可执行程序。

（4）检查、搜索内存及内存映象存盘。

（5）检查修改磁盘扇区中的内容。

（6）修改文件分配表FAT、目录项表FDT。

（7）用DEBUG汇编命令或通过DOS重定向输入文本文件来建立短小COM文件。

（8）恢复内存中的文本文件或二进制文件。

（9）恢复被删除的文件。

（10）对文件名、子目录名修改及加密或解密。

（11）清除计算机病毒。

（12）修复磁盘被破坏的文件。

本章先介绍如何启动DEBUG程序，以及有关DEBUG命令的通用信息，然后按功能分类逐一详细介绍它的19个子命令的用途、格式及有关的示例，最后综合举例说明这些命令的使用。

1.2 怎样启动DEBUG程序

1.2.1 DEBUG程序的启动

在DOS提示符A〉下，键入如下的命令：

A) DEBUG [d:][path][filename[.exe]][parm1][parm2]

其中 DEBUG 是调试程序的文件名，后面跟的是被调程序的文件说明。如果输入了 filename，DEBUG 程序就把该指定文件装入内存。此后，就可以打入命令修改、显示或执行所指定文件的内容。

如果没有输入 filename，则只能针对当前内存中的内容进行修改、显示或执行。可用命名命令 Name 或装入命令 Load 将所需要的文件装入内存，然后就可以使用命令对内存中的内容进行修改、显示和执行。

可选参数 parm1 和 parm2，表示命名文件说明时可选择的参数。例如：

DEBUG DISKCOMP.COM A: B:

在这个命令中，A: 和 B: 是 DEBUG 为 DISKCOMP 程序准备的参数。

当 DEBUG 启动成功后，显示提示符“-”（短横线）。这说明现在系统在 DEBUG 程序的管理之下。所有的 DEBUG 命令，只有跟在提示符“-”后键入才有效。

1.2.2 DEBUG 程序初始化

当启动 DEBUG 时，对要调试的程序设置寄存器和标志位其数值如下。

（1）四个段寄存器（CS，DS，ES 和 SS）置于内存空闲块的底部，即 DEBUG 程序结束以后的第一个段地址。

（2）指令指针 IP 置为 100H。

（3）堆栈指针（SP）置到该段的结尾处，或者是装入程序的临时底部，取决于哪一个较低。

（4）其余的寄存器（AX，BX，CX，DX，BP，SI 和 DI）置为 0。但当启动 DEBUG 时，如果使用一个要调试程序的文件标识符，则该文件长度包含在 CX 中（以字节表示）；若文件大于 64K 则该文件长度包含在 BX 和 CX 中（高位在 BX 中）。

（5）设置标志寄存器为被清除的状态。

（6）约定的磁盘缓冲区置于代码段 80H 处。

注意：若由 DEBUG 调入的程序具有扩展名 EXE，则 DEBUG 必须进行再分配，把段寄存器、堆栈指针置为文件中规定的值。

经过上述初始化后，在屏幕上显示 DEBUG 的提示符“-”，等待键入命令。

1.2.3 有关 DEBUG 命令的一些通用信息

在详细介绍 DEBUG 命令以前，现将有关它的通用信息介绍如下。

（1）DEBUG 的每个命令为一个字母，通常后面有一个或多个参数。

（2）命令和参数可以用大写、小写或大小写混合方式输入。

（3）命令和参数可以用定界符（空格或逗点）分隔开，然而，只有在两个连续的十六进制数之间必须使用定界符，因此，下面的命令是等价的：

DCS: 100 110

DCS: 100 L10

D,CS: 100,110

（4）可以用 Ctrl+Break 键来停止一个命令的执行，返回 DEBUG 提示符。

（5）每个命令只有在按了 Enter 键之后，输入命令才起作用，并开始执行。

（6）对于产生大量输出的命令，可用 Ctrl+Numlock 键去暂停显示，以便在这些输出从屏幕上滚消失之前看清楚，按任何键可继续下面的显示。

（7）在使用 DEBUG 程序期间，能使用控制键和 DOS 编辑键。

（8）如果遇到一个语法错误，则显示出该行并指出错误之所在，例如：

```
DCS:100 CS:110
        ^error
```

在这个例子里，显示命令期望第二个地址里仅包括一个十六进制位移值，但当它发现 c 时，识别出它不是一个十六进制，故出错。

1.2.4　DEBUG 命令中的地址和地址范围参数

DEBUG 命令中的参数多，现将常用的地址和地址范围参数介绍如下。

DEBUG 中的地址格式为：[〈段地址〉:]〈位移量〉。据此，它有如下的三种形式。

（1）用字符表示的段寄存器名称，加上一个位移值，其间用冒号隔开，例如：

```
CS: 100
```

（2）一个段地址，加上一个位移量，其间仍用冒号隔开，例如：

```
4BA: 0100
```

（3）只有一位移值（段地址用约定值），例如 100，此时对子命令 A，G，L，T，U 和 W，约定段为 CS，对所有其他子命令的约定段为 DS。

为确定一个地址范围的低地址和高地址，可输入下面两种格式之一。

（1）〈段地址〉:〈始位移量〉〈终位移量〉，例如：

```
CS: 100 110
```

其中第二地址仅用位移量，且与前面地址需用空格符分隔开。

（2）〈段地址〉:〈始位移量〉〈长度〉，例如：

```
CS: 100  L11
```

在这里，要注意，所有的数字值都可为 1～4 个字符的十六进制值，而且地址和地址范围指定的内存位置必须是内存实际存在的。如果企图去存取一个不存在的存贮单元，将会产生预想不到的后果。

以下各节介绍 DEBUG 的子命令及其应用。

1.3　汇编与反汇编命令

1.3.1　汇编命令 A（Assemble Command）

用途：把 8086/8087/8088 助记符直接汇编到存贮器内。

格式：A[〈地址〉]

说明：汇编命令 A 将用户输入的汇编语句从[〈地址〉]参数所指定的地址开始汇编到

存贮器中连续的单元里以得到可执行的机器码。例如：

```
A) DEBUG
-A200
08B4:0200 XOR AX,AX
08B4:0202 MOV [BX],AX
08B4:0204 RET
08B4:0205〈按 ENTER 键〉
```

若在命令中没有指定地址，但前面用过汇编命令，则接上一个汇编命令的最后一个单元开始存放；若前面没有用过汇编命令，则从 CS：100 单元开始连续存放。

当所有需要的语句都输入完后，这时提示输入下一个语句，在此提示下按 ENTER 键作为响应，于是就返回到 DEBUG 提示符。

注意：对于 DEBUG 在装入 EXE 文件后，首次使用 A 命令，且后面没有指定地址，则从 CS: IP 开始，连续存放程序；若指定地址，则从指定地址开始存放。

```
A) DEBUG ASM.EXE
-A 〈按 ENTER 键〉
XXXX:XXXX(CS:IP) MOV AX,0
```

例如：

```
-A5000:0
5000:0 MOV AX,1
```

又如：

```
-A1000
CS:1000 开始
```

若输入语句有错，DEBUG 就显示：^Error 并重新显示当前汇编行的地址，等待新的键入。

DEBUG 支持标准的 8086/8088 汇编语言的语法（和 8087 指令系统）并具有以下的一些规则。

（1）所有输入的数字值全为十六进制数，可输入 1～4 个十六进制数字字符。

（2）前缀助记符，必须在相关指令之前输入，也可另起一行。

（3）段超越助记符为 CS:、DS:、ES: 和 SS:。

（4）字符串操作助记符必须明确地说明字符串长度，例如，MOVSW 必须用于去传送字串，而 MOVSB 必须用于去传送字节串。

（5）汇编程序将自动地汇编短转移、近转移或远转移和调用，这取决于对目标地址的字节位移，也能用 NEAR 和 FAR 前缀取代它们，例如：

```
0100：0500 JMP 520                    ；二字节的短转移指令
0100：0502 JMP NEAR 505               ；三字节的近转移指令
0100：0505 JMP FAR 50A                ；五字节的远转移指令
```

（6）交叉段远返回的助记符是 RETF。

（7）DEBUG 无法说明一些操作数是属于字存贮单元还是字节存贮单元，必须用前缀

“WORDPTR”（可缩写为“WO”）或“BYTE PTR”（可缩写为“BY”）来明确说明数据的类型，例如：

```
NEG BYTE PTR[128]
DEC WO[SI]
```

（8）DEBUG也无法说明操作数是属于存贮单元还是属于立即数，因此DEBUG中把存贮单元的地址放在括号中，例如：

```
MOV AX,21                    ; 把21H送入AX中
MOV AX,[21]                  ; 把地址为21H以及21H+1的存贮单元的内容送至AX
```

（9）也能包含两个常用的伪指令：DB操作码将把字节数值直接汇编到存贮器内；DW操作码将把字的数值直接汇编到存贮器内。例如：

```
DB 1,2,3,4 "THIS IS AN EXAMPLE"
DW 1000,2000,3000, "BACH"
```

（10）DEBUG支持所有形式的寄存器间接寻址命令，例如：

```
ADD BX,34[BP+2][SI-1]
POP [BP+DI]
```

（11）DEBUG支持所有操作码的同义词，例如：

```
LOOPZ 100
JA  200
```

A命令主要用于小段程序的汇编、调试程序中发现其中一部分需要改写或要增补一段等，就可直接在DEBUG下编程序，变成机器码并调试和运行，亦可存入磁盘。这样就省去了调编辑程序、汇编程序、连接程序的复杂手续，给用户提供了方便。下面是说明A命令使用的例子。

【例1.1】 数据块传送（将200开始的16个字节单元的数据传送到220开始的单元中去）。

```
A> DEBUG
-A100
08F8:0100 MOV SI,0200
08F8:0103 MOV DI,0220
08F8:0106 MOV CX,10
08F8:0109 REPZ MOVSB
08F8:010B INT 3
08F8:010C
-A200
08F8:0200 DB 60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75
-G=100
-D220 L10
08F8:0220 60 61 62 63 64 65 66 67-68 69 70 71 72 73 74 75
```

【例1.2】 读软盘A:DOS引导区。

```
-A100                        ;从100H输入程序
```