

微型计算机 应用经验汇编(2)

金泉涌等 编

清华大学出版社

微型计算机应用经验汇编(2)

金泉涌 编

清华大学出版社

内 容 简 介

《微型计算机应用经验汇编(1)》出版后,受到广大微机用户的欢迎,本书是继《微型计算机应用经验汇编(1)》之后的第(2)集,共汇集了近 200 个使用微机的小经验、小技巧,是广大微机用户的良师益友。内容包括:磁盘与操作系统的使用技巧;键盘与打印机使用技巧;BASIC、C 等语言的应用例程;数据库的应用例程;多种语言间的互相调用;图形及屏幕显示技巧。绝大部分程序已在 PC 机上通过调试。

本书适用于广大微机操作者、应用开发者。

(京)新登字 158 号

微型计算机应用经验汇编(2)

金泉涌 编

☆

清华大学出版社出版

北京 清华园

北京密云胶印厂印刷

新华书店总店科技发行所发行

☆

开本: 787×1092 1/16 印张: 28 字数: 661 千字

1993 年 8 月第 1 版 1993 年 8 月第 1 次印刷

印数: 00001—10000

ISBN 7-302-01215-6/TP·456

定价: 16.60 元

前 言

目前,我国微型计算机的装机量不断增加,用户遍布全国各行各业,甚至家庭。应用领域越来越广,应用水平不断提高。为适应广大微机用户的需要,本书汇集了微机应用、开发者的实践经验近 200 条,也可以说是编程技巧、用机技巧、解决实际问题的创造性和灵活性的集中反映。所以实用价值大,饶有趣味。广大读者不仅取之则用,而且能从文中的思路得到启发,为自己的思路打开联想的天窗,从而能节省开发时间,提高应用水平。

全书内容精选自多种报、刊发表过的文章,在编辑过程中绝大部分程序已在 PC 机上通过调试,同时每篇文章后署有原作者姓名。内容包括:磁盘与操作系统的使用技巧;键盘与打印机的使用技巧;BASIC、C 等语言的应用例程;数据库的应用例程;多种语言的混合应用;图形与屏幕显示技巧。

本书适用于广大微机操作者、应用开发者。

参加本书编辑工作的还有尹锋、乔迁、岱东、张秀、海燕等。

编 者

1993. 11.

目 录

专题一 磁盘与操作系统的使用技巧

1.1 扩大软磁盘容量的三种方法	1
1.2 方便实用的软盘格式化程序	4
1.3 特殊扇区的格式化和读写方法	5
1.4 磁盘逻辑“坏簇”的自动辨认和回收程序	6
1.5 在校正软驱磁头中巧用 DISKCOPY 文件	8
1.6 测试软盘的实用程序	10
1.7 用设置虚拟磁盘的方法来延长微机寿命和提高使用效率	11
1.8 快速判断命令参数中驱动器号正确性的实用程序	13
1.9 一种迅速确定硬盘类型的方法	14
1.10 硬盘物理盘符漂移及其处理一例——巧用 ASSIGN 命令	15
1.11 DOS 环境的概念、使用方法和技巧	16
1.12 DOS 实用程序 MOVE 的实现	19
1.13 切换操作系统简便方法	22
1.14 在低版本 DOS 下使用 APPEND 命令的方法和 PATH APPEND 及 FASTOPEN 的综合运用	23
1.15 不同 DOS 版本下系统程序的执行方法	26
1.16 在应用程序环境下运行所有 DOS 命令的方法	28
1.17 用批命令连接多种操作系统	30
1.18 可改变 DOS 版本号的实用程序	32
1.19 用 ANSI.SYS 扩展 DOS 的屏幕和键盘	33
1.20 DOS 文件操作的几点小经验	35
1.21 微机操作技巧四则	37
1.22 为 MS-DOS 增加修改子目录名命令	40
1.23 在 MS-DOS 5.0 环境中,正常运行 2.13H 汉字程序的一种方法	41
1.24 实现文本阅读器的简便方法	43
1.25 XENIX 系统的软盘启动和维护	45
1.26 XENIX 的软盘启动方法	46
1.27 恢复 XENIX 系统核心的一种简便方法	49
1.28 硬盘逻辑损坏的维修方法	49
1.29 IBM PC 机硬盘引导失败的自动恢复	52
1.30 简易的磁头校正程序及使用方法	54

• II •

1.31	修复故障盘数据的一种实用程序	56
1.32	用批处理文件实现的 DOS 实用例程	59
1.33	快速获得汉字区位码的程序	61
1.34	一个能快速地动态撤离驻留于内存的汉字系统及 TSR 的实用程序	62

专题二 键盘与打印机使用技巧

2.1	利用计算机打印条形码程序	69
2.2	实现高密盘向低密盘直接拷贝的 C 程序	71
2.3	实现打印控制的直接代码法	72
2.4	在汉字打印机驱动程序执行后打印机标准换页问题的解决办法	76
2.5	TAB 格式文件的转换和打印程序	80
2.6	用 Turbo C 实现打印命令功能的程序	81
2.7	用于屏幕拷贝的不可打印字符过滤程序	83
2.8	FoxBASE 下编制打印程序新方法	86
2.9	获取键盘及汉字的扫描代码程序	87
2.10	可以得到键盘扫描码的程序	88
2.11	键盘的重新定义方案及实现方法	90
2.12	实现键盘重定义又一法	92
2.13	设置终端自定义功能键的方法	94
2.14	汉化中部十个光标控制键程序与重新定义汉字输入方式的选择键	95
2.15	用单个功能键仿真 BROW 命令中的组合功能键	97
2.16	“Do”键的设计方法	99
2.17	检测热启动操作的简便方法	102
2.18	有效地提高键盘速率的程序	104
2.19	CCDOS 下功能键的切换程序	105
2.20	提高键盘速度的一种简单方法	106
2.21	Turbo C 2.0 键盘输入功能的扩充	107
2.22	提高汉字输入速度的两种方法	109
2.23	用软件实现“双回车”的方法	110
2.24	BASIC 键盘译码器	111

专题三 几种语言的编程技巧

3.1	用 BASIC 语言的作图命令来模拟时钟的程序	113
3.2	模拟时间报点程序	114
3.3	用 GWBASIC 语言编写的简易模拟计算器	115
3.4	递归调用在 BASIC 语言中的实现	116
3.5	改进 ECHO 功能的 BASIC 程序	117
3.6	利用 BASIC 语言的 33H 中断调用鼠标的方法	119
3.7	用 BASIC 中的 INKEY\$ 函数实现菜单的彩条驱动	122

3.8	实用的.COM、.EXE文件转换、还原程序	123
3.9	编译 BASIC 7.1 版的改进	125
3.10	应用 BASIC 的几个小技巧	128
3.11	多种语言源程序的自动批量编译	131
3.12	用 DEBUG 完善 ED 编辑器的方法	133
3.13	用 C 语言和 DOS COPY 命令来实现对多个文件的一次显示	135
3.14	改变鼠标光标形状的程序	139
3.15	C 语言宏代换功能的实现方法	141
3.16	修改子目录名称查看子目录名及删除子目录的程序	144
3.17	用 Turbo C 实现 DOS SHELL	147
3.18	用 Turbo C 编写的删除目录树程序	148
3.19	用 Turbo C 2.0 编写的文件删除程序	150
3.20	用 Turbo C 编写的文件分割程序	151
3.21	Turbo C 语言可变参数的引用方法	154
3.22	检出和分离 C 语言源程序错误的一种简单方法	155
3.23	对 Microsoft C 5.0 Turbo C 2.0 可执行代码调用的尝试	156
3.24	程序执行时间的测量方法	158
3.25	反汇编文件的自动转换程序	160
3.26	C 语言中指针使用应注意的几个问题	162
3.27	用 C 语言解决 COBOL 中的批处理问题	165
3.28	一个实用的源程序统计程序	166
3.29	Turbo C++ 中文本及图形方式下的汉字处理程序	167
3.30	为 Turbo C 2.0 增加提取子字符串函数	169
3.31	快速移动文件程序	170
3.32	谈 COBOL 语言中索引文件遭破坏后的恢复	173

专题四 数据库应用实例和技巧

4.1	在 dBASE III PLUS 中实现随意翻页查询的方法	177
4.2	用 dBASE III 实现多个项目模糊检索的一种方法	182
4.3	对索引文件关键字的修改程序	184
4.4	通用的数据库文件制表程序	186
4.5	一个方便、实用、快速的制表程序	189
4.6	适用于多数据库操作的程序	191
4.7	字符串批量修改实用程序	194
4.8	ZAP 命令删除数据库记录的恢复	197
4.9	一种自动得到命令文件调用关系图的程序	200
4.10	完成有特殊要求排序功能的方法	202
4.11	使程序格式标准化的程序	204
4.12	用 TOTAL 命令实现对多个关键字段同类求和的方法	205
4.13	解决多个关键字段分类求和的简单途径	207

4.14	解决双重分类求和问题的通用子程序	209
4.15	dBASE 或 FoxBASE 中工作区的使用技巧	210
4.16	用 FoxBASE 的数组实现窗口功能	212
4.17	FoxBASE 的窗口管理程序	215
4.18	在 FoxBASE 下实现菜单的翻页显示模块	217
4.19	FoxBASE+ 屏幕颜色的选择程序	218
4.20	多个 READ 命令并存时实现全屏幕编辑程序	220
4.21	用全屏幕编辑方式实现数据录入与修改时的检测功能	222
4.22	FoxBASE 全屏幕编辑报表通用程序	223
4.23	在 FoxBASE 下通用汉字制框程序	226
4.24	FoxBASE 系统下日期使用方法	227
4.25	超多库数据登录程序的实现方法	228
4.26	FoxBASE+ 通用数据输入程序	232
4.27	利用错误处理程序来简化程序设计	237
4.28	FoxBASE 中识别 F2~F10 的一种方法	239
4.29	FoxBASE 状态下测试外设工作状态的实用程序	240
4.30	dBASE III 命令文件转入 FoxBASE 环境下运行时存在的问题	242
4.31	at () 函数及 stuff () 函数在菜单选择中的应用	243
4.32	实现“FoxBASE+”共享方法及补充	244
4.33	用函数实现数据库的动态输入	246
4.34	数据输入程序中汉字与西文状态自动切换功能的实现方法	249
4.35	对“数据录入程序中汉字与西文状态的自动切换功能的实现方法”一文的补充	254
4.36	自动清除当前目录下所有数据库文件中数据的程序	255
4.37	数值型字段清零通用程序	257
4.38	数据库文件中的半个汉字字符的剔除程序	259
4.39	用 SCATTER、GATHER 命令设计数据库编辑系统实例	260
4.40	INFORMIX 中一些语句的使用技巧	262
4.41	提高统计求和速度的几种方法	263
4.42	在 dBASE 状态下运行 DOS 程序	267
4.43	dBASE III 系统下的变长记录技术	267
4.44	为 dBASE+ 和 FoxBASE+ 补充两个用于窗口设计的命令	271
4.45	对任何数据库结构进行修改的程序	275
4.46	通用数据库记录置空方法	278
4.47	用 MFoxBASE 编制的 PRG 文件自动整理程序	279

专题五 多种语言的联合应用

5.1	在 IBM 微机上实现 FORTRAN、PASCAL、C 语言相互调用的机制和方法	281
5.2	实现 XENIX 多用户 COBOL 语言与 DOS 下 BASIC 语言数据文件的共享	286
5.3	用 Turbo C 直接访问 dBASE III 和 FoxBASE 数据库的程序	287
5.4	用 C 语言把 DOS 环境下的文件转到 XENIX 下运行	289

5.5 C语言与dBASE III之间的数据交换	290
5.6 大模式C程序转换为汇编程序存在的问题及解决方法	292
5.7 用BASIC随机读取dBASE III和FoxBASE+的DBF数据文件的简便方法	295
5.8 读取任一FoxBASE+和dBASE III、N数据库文件(DBF)的方法	296
5.9 在Turbo BASIC中汇编语言代码的自动转换程序	299
5.10 汇编调C浮点运算的一种简便方法	303
5.11 程序间传递参数的最简方法	306
5.12 dBASE III与ORACLE数据交换的简便方法	307
5.13 在高级语言中直接存取(.DWG)文件的方法	308
5.14 将DOS文件向XENIX批量传送的程序	316
5.15 XENIX与DOS系统汉字间的相互转换	318
5.16 在FORTRAN程序中调用DOS命令的方法	320

专题六 图形与显示技巧

6.1 TVGA卡图形方式的编程方法	324
6.2 截获VGA高品质图像程序	329
6.3 存储和调用VGA屏幕图像的一对程序	334
6.4 在FoxBASE+环境EGA、VGA显示器屏幕图像的存储与恢复程序	337
6.5 用Turbo C为CdBASE III作圆饼图	339
6.6 中文窗口的保存与恢复程序	343
6.7 在屏幕任一位置开一窗口的子程序	354
6.8 屏幕动态图形的随机转储方法	355
6.9 动态图形的一种简捷实现方法	359
6.10 在dBASE III下的动态画面设计方法	361
6.11 用FoxBASE编写的四种趣味动态引导画面	362
6.12 设计动画封面程序	366
6.13 在FoxBASE+下调用BIOS图形功能及其参数传递程序	367
6.14 用dBASE III PLUS调用BIOS绘图功能	371
6.15 电力能量损耗图的画法和编程技巧	376
6.16 对真实感图像进行颜色插值的一种实现方法	379
6.17 屏幕色彩随意设置通用程序	383
6.18 实用的调色板程序	386
6.19 对《实用的调色板程序》一文的补充	388
6.20 一个不破坏回显色的程序	391
6.21 一个实用的光标加速和制动程序	392
6.22 直接用点阵显示汉字的方法	395
6.23 无中文操作系统支撑下的汉字显示方法	399
6.24 在配有CEGA卡的微机上实现图形汉字的显示	403
6.25 C语言图形状态下汉字的显示	405
6.26 实现汉字竖行显示的程序	407

6.27	扩充 MS-PASCAL 的彩色汉字显示程序	408
6.28	不同显示模式下汉字系统的自动选择	411
6.29	应用程序中屏幕锁行的显示方法	413
6.30	控制 dBASE III 显示行数的实用程序	414
6.31	PC 机显示适配器与快速字符显示方法	416
6.32	在屏幕上显示各种字体的实现	421
6.33	用 CCDOS 2.13H 的特殊显示功能来美化屏幕	425
6.34	CCDOS 2.13H 特殊显示功能应用实例	426
6.35	用 CCDOS 2.13H Microsoft C 6.0 和 MASM 显示特殊中文标题	430
6.36	翻页显示数据库记录程序	433
6.37	显示器保护程序	435

磁盘与操作系统的使用技巧

本部分介绍磁盘与操作系统的使用方法、技巧及实用程序。

1.1 扩大软磁盘容量的三种方法

本文介绍三种方法,利用高密度驱动器,使用 MS-DOS 3.3 的格式化程序在 360K 软盘上格式化 720K 乃至更多的磁盘容量(720K 为 MS-DOS 3.3 下的标准格式),将原来磁盘容量增加一倍,而且格式化过程一次完成,且没有任何坏扇区标志。笔者已在 AST-286 及 COMPAQ 计算机上经过一年的使用,从未出现任何读写异常的问题。用这些方法,拥有高密度驱动器的用户可以利用手中的低密度软盘存储更多的数据。

凡使用过 DOS 3.0 以上版本的用户都知道,利用 FORMAT A: /4 命令可以在高密度驱动器内格式化 360K 软盘,在这里关键的是改变了驱动器的步长,在 360K 时,其步长取的是高密格式的 2 倍,这就给我们一个启发,如果能够使其步长取高密格式而每道扇区数取低密格式,就能够得到 80 道而每道扇区数仍为 9 的磁盘。因此,主要目的就是设法仅仅改变驱动器的步长,而使其每道扇区数不变或稍微增加为机器能够接受的扇区数。下面三种方法都是基于这个目的而加以实现的。

方法一:

(1) 用 MS-DOS 3.3(或 PC-DOS 3.3)格式化一张系统盘:

```
FORMAT A: /4/S
```

(2) 拷入系统的 DRIVER.SYS, FORMAT.COM, DEBUG.COM 文件于此系统盘上。

(3) 用行编辑 EDLIN 或 WORDSTAR 在此盘上建立 CONFIG.SYS 文件,其中必须包含以下语句:

```
DEVICE=DRIVER.SYS/D:0/F:2
```

此处假设 A 驱为高密,若高密为 B 驱动器,上一句中/D:0 改为/D:1 即可,CONFIG.SYS 文件中的其它内容可根据自己的应用情况增加。这样,系统引导时,即建立了一个扩展驱动器,其盘符由系统连接的硬盘数量按 D,E,F……的顺序递增(在此,不妨假设为 E)。

(4) DEBUG A: FORMAT.COM(因 PC-DOS 3.3 稍有差别,故列出一段):

.....

2D28:1052 B80D44	MOV	AX,440D
2D28:1055 8A1EC203	MOV	BL,[03C2]
2D28:1059 FEC3	INC	BL
2D28:105B B94208	MOV	CX,0842
2D28:105E 8D164902	LEA	DX,[0249]
2D28:1062 CD21	INT	21
2D28:1064 803E490200	CMP	BYTE PTR [0249],00
2D28:1069 7428	JZ	1093
2D28:106B 8D16A807	LEA	DX,[07A8]
2D28:106F 803E490202	CMP	BYTE PTR [0249],02
2D28:1074 7417	JZ	108D
.....		

修改地址2D28:1069一句为 JMP 1093后存盘。

(5) 用此系统盘启动,然后用 FORMAT E:,即可在高密驱动器内将低密软盘格式化为720K 容量了。

方法二:

同方法一,但不用修改 FORMAT.COM 程序,而用以上系统盘启动机器以后,利用 PCTOOLS, 进入格式化磁盘功能,然后选择(在语句 DEVICE=DRIVER.SYS 下建立的)驱动器盘符,则可以进行720K 格式化。

方法三:

进入 DEBUG.COM,用 A 命令编制以下文件(长度60H,命名为 RE-INT.COM)

1179:0100 EB3A	JMP	013C
1179:0102 50	PUSH	AX
1179:0103 1E	PUSH	DS
1179:0104 31C0	XOR	AX,AX
1179:0106 8ED8	MOV	DS,AX
1179:0108 C70678002205	MOV	WORD PTR [0078], 0522
1179:010E C7067A000000	MOV	WORD PTR [007A], 0000
1179:0114 B8FD02	MOV	AX,02DF
1179:0117 A32205	MOV	[0522], AX
1179:011A B82502	MOV	AX, 0225
1179:011D A32405	MOV	[0524], AX
1179:0120 B80A1B	MOV	AX,1B0A
1179:0123 A32605	MOV	[0526], AX
1179:0126 B8FF32	MOV	AX,32FF
1179:0129 A32805	MOV	[0528], AX
1179:012C B0F6	MOV	AL,F6
1179:012E A22A05	MOV	[052A], AL
1179:0131 90	NOP	
1179:0132 B057	MOV	AL,57
1179:0134 A29004	MOV	[0490], AL
1179:0137 1F	POP	DS
1179:0138 58	POP	AX
1179:0139 CD68	INT	68
1179:013B CF	IRET	
1179:013C B81335	MOV	AX,3513
1179:013F CD21	INT	21

1179:0141 FA	CLI
1179:0142 89DA	MOV DX,BX
1179:0144 06	PUSH ES
1179:0145 1F	POP DS
1179:0146 B86825	MOV AX,2568
1179:0149 CD21	INT 21
1179:014B 0E	PUSH CS
1179:014C 1F	POP DS
1179:014D BA0201	MOV DX,0102
1179:0150 B81325	MOV AX,2513
1179:0153 CD21	INT 21
1179:0155 FB	STI
1179:0156 B80131	MOV AX,3101
1179:0159 BA2000	MOV DX,0020
1179:015C CD21	INT 21
1179:015E3 CD20	INT 20
RCX	
60	
NRE-INT.COM	
W	

以上一段程序,修改13号中断向量为68号,且执行后留驻内存,软盘驱动器参数表的一部分参数被设置成 DF 02 25 02 0A 1B FF 32 F6,即磁盘每道10个扇区,扇区间隙为32(这是一种特殊的、非标准格式,正常使用的低密度盘为每道9扇区,扇区间隙为50,程序中的参数可根据自己的机器情况作适当调整)。这组参数共11个,放在由0:78~0:7B为指针所指出的连续单元中,大部分机器的磁盘参数表放在0:522单元开始的11个单元中,上述程序中的参数也是从0:522单元开始修改,为稳妥起见,可以在执行程序前先查找0:78~7B的指针,以确定地址,若不同,要修改程序中的相应地址:522,524,526,528,52A 五个单元为其机器的地址。另外,修改了0:490单元的内容,一般360K 盘的值为74,当格式化720K 时应该修改为54,此参数是控制高密驱动器的步进是否加倍、数据传输率等一些参数,这一步是关键。0:490为 A 驱动器,0:491为 B 驱动器,若你的高密度盘为 B 驱动器,请修改程序中[490]为[491](请比较以上程序执行前后你的机器的软盘参数表的改变,也可以修改程序中部分参数为完全标准的参数)。

假设取名为 RE-INT.COM,在系统下执行 RE-INT 程序,让其驻留内存,然后用未作过修改的 FORMAT.COM 程序格式化,但同时应当在格式化中指出每道扇区数,也可以同时指定(或省略)最大磁道数。例如:FORMAT A:/N:10/T:84,可以格式化840K 非标准磁盘容量(其中 N 后带的参数为每道扇区数,T 后带的参数为最大磁道数,详细情况可参考 MS DOS 3.3 磁盘操作系统手册中 FORMAT 命令的说明)。

比较以上三种方法,方法一和方法二不修改系统软盘参数,简单易行,修改后的 FORMAT 程序对以往的使用无任何影响,而且对系统来说由于逻辑上扩充了一个驱动器,为在高密盘上的文件传送带来很多方便(例如使用命令:XCOPY A:*. * E:,可在同一高密度盘内互相拷贝文件),但注意这两种方法在第一次执行格式化前,高密驱动器必须有过一次正常的读盘动作(仅仅一次,例如:DIR A:*. *),才能正常执行 FORMAT 程序。方法三修改系统软盘参数,有一定的灵活性,但修改过程其个别参数可能要根据自

己的机器做一些调整(譬如扇区间隙),而且非标准参数对不同机器的兼容性会有差异。以上三种方法格式化后的软盘,就像1.2M的高密度盘一样,只能在高密驱动器内才能使用。而且,根据笔者的使用经验,AST 286机使用三种方法都可以,而且格式化后的磁盘不再需要 RE-INT 程序修改磁盘参数表就可以进行读写,其系统盘也可以作为引导盘使用。COMPAQ 系列和其它286、386机器建议采用第三种方法,而且由于系统的软盘参数需要经过 RE-INT 程序修改后才能使用此类规格的软盘,因此,这种格式的系统盘将不可能作为引导盘使用。

本文介绍的第三种方法对于 PC 大部分机器作为360K 盘扩充为420K 盘可能起到一些作用,格式化命令为:FORMAT A:/N:10/T:42,可以在 PC 机上突破以往有些文章介绍的格式化400K 的界限,但是一定不要忘了使用 DOS 3.3以上版本,而且读写这种盘时,也同样需要 RE-INT.COM 程序的支持。

(李维宪)

1.2 方便实用的软盘格式化程序

现在有很多微机配置了两种尺寸的软盘驱动器,A 驱动器用130mm(5英寸)磁盘,B 驱动器用90mm(3英寸)磁盘。这样每次格式化软盘时,需要翻阅 DOS 手册查找合适的命令开关,如在 B 驱动器中格式化720K 磁盘。

为简单起见,这里提供一个批处理文件 FORMAT.BAT。使用时,将格式化程序(如 DOS 中的 FORMAT.COM 或 PCTOOLS 中的 PCFORMAT)换名为 FORMAT!.COM。这样就可以用 format 360、format 12、format 720和 format 144等命令随意格式化软盘了。

程序: FORMAT.BAT

```
ECHO OFF
REM FORMAT.BAT
REM RENAME DOS'S FORMAT.COM TO FORMAT!.COM
GOTO %1
```

·HELP

```
ECHO You must specify what type of disk you want to format.
ECHO FORMAT 360 will produce a 360KB disk in drive A:
ECHO FORMAT 12 will produce a 1.2MB disk in drive A:
ECHO FORMAT 720 will produce a 720KB disk in drive B:
ECHO FORMAT 144 will produce a 1.44MB disk in drive B:
GOTO DONE
```

·360

```
C:\DOS\FORMAT! A:/N:09 /T:40 %2 %3 %4
GOTO DONE
```

·12

```
C:\DOS\FORMAT! A:/N:15 /T:80 %2 %3 %4
GOTO DONE
```

```

:720
C:\DOS\FORMAT! B:/N:09 /T:80 %2 %3 %4
GOTO DONE
:144
C:\DOS\FORMAT! B:/N:18 /T:80 %2 %3 %4
DONE

```

其中,format 360用在 A 驱动器格式化360K 盘;format 12用在 A 驱动器格式化1.2M 盘;format 720用在 B 驱动器格式化720K 盘;format 144用在 B 驱动器格式化1.44M 盘。

——Ed Quillen

注:该程序假设 FORMAT!.COM 在 \DOS 目录中,可以将 DOS 换为 FORMAT!.COM 所在目录名。最好将 FORMAT.COM 放在 path 所列出的某个目录中,就更方便了。

1.3 特殊扇区的格式化和读写方法

很多软件用特殊的磁道或扇区存储达到防搬护的目的,如扇区在磁道中的位置不是顺序排列而是杂乱无章,一个扇区不是512位而竟是1024位等。下面就谈谈这种特殊扇区是怎样格式化和读写的。

笔者通过对 IBM PC/XT BIOS 中的 int 13H 程序进行分析,知道磁盘读写格式化等操作所用的基本参数,如每扇区的字节数、每道的扇区数等被存放在磁盘参数表中(BIOS 提供的 int 1eH 是其起始地址)。所以先分析一下驻留在 BIOS 中默认的该参数表(其中 DOS 指2.10版本)。

```

ORG OFFC7H
DISK-BASE LABEL BYTE
EFC7 DB 11001111B; SRT=C(8ms), HWT=F(480ms), DOS 下改为 DF
EFC8 DB 02; HLT=1(4ms), MODE=0, DMA 方式
EFC9 DB 25; 操作后等待马达关闭的计数值
EFCA DB 02; 128 * N * 2字节/扇区, 当 N=2时为512字节/扇区
EFCB DB 08; 每道的扇区数, DOS 下改为09
EFCC DB 2A; 读写时的间隔长度
EFCD DB FF; 数据长度
EFCE DB 50; 格式化的间隔长度
EFCF DB F6; 格式化填充字节
EFD0 DB 19; 磁头定位时间, DOS 改为0F
EFD1 DB 04; 马达启动时间, DOS 改为02

```

现在终于明白了 DOS 使用的360KB 软盘都格式化为512字节/扇区、每道09扇区的原因了。如果通过修改这个参数表的有关参数,即可格式化出特殊的扇区或磁道了。

下面就格式化一个特殊的磁道(A 驱0面第20磁道):扇区含1K 位,有4个扇区且磁道按 A1、A3、A2、A4的扇区格式排列。

首先找磁盘参数表的位置,以便修改。它由 int 1eH 指定,即0:0078~0007B,用 debug 可得到

-d078, 7B

0000:0070.....22 05 00 00.....

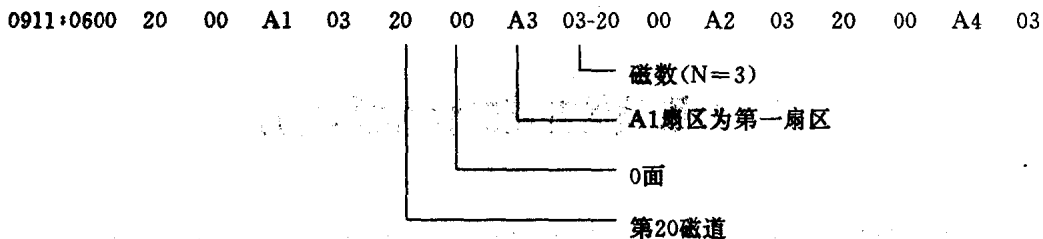
即0:0522开始的即为参数表内容,正常情况下为上表所示。现将其改为:

-d0:520,52f

0000:0520..DF 02 25 03 04 2A-FF50 F6 00 02...

即[0525]=03,[0526]=04;每个扇区 $128 * 3 * 2 = 1024$,有04个扇区。

而格式化时识别的数据是从BX所指地址依次按磁道、磁头、扇区、磁数四位一组排列的。只要改变这些数据的次序,就能格式化出所要的扇区了。在其中放以下数据(BX=0600)。



这样,得到格式化这种特殊扇区的程序为:

0911:0100 B80405 MOV AX,0504; AH=05格式化功能,

AL=04为格式化扇区数

0911:0103 BA00 00 MOV DX,0000; A 驱0面

0911:0106 B90120 MOV CX,2001;第20磁道,01为起始扇区

0911:0109 BB 00 06 MOV BX,0600;格式化识别数据的地址

0911:010C CD 13 int13

同样地,只要在读写一个扇区之前修改磁盘基本参数表的相关参数,用 int 13H 的读写功能就可以自由地读写这种特殊扇区了。

(姜 维)

1.4 磁盘逻辑“坏簇”的自动辨认和回收程序

磁盘的坏簇分为两种:一种是物理坏簇,因盘介质局部损伤所致,由 FORMAT 命令发现并进行统计;另一种通常由操作系统型病毒侵害所致,并非物理损伤,称为逻辑“坏簇”。前一种是不可恢复的,后一种却能设法回收,以便恢复磁盘的容量,供正常存储文件使用。但是,两种坏簇在 FAT 表的对应簇域中的标志都是 FF7H(或 FFF7H),怎样区分呢?

笔者编写了双面双密软盘逻辑“坏簇”的辨认、回收程序。主要思路是,物理坏簇中并未写有正常信息,读时必然出错,逻辑“坏簇”则与盘的其它簇一样,事先经过格式化,并且写有正常格式信息,因此可以正常读出。程序中,先读出盘文件分配表 FAT1 的内容,从 002 簇域开始,逐一检测各簇域值是否为坏簇标志“FF7H”。如果是,则将相应簇号转换为

逻辑扇区号,以便从盘文件区读对应簇(两个相邻扇区);之后,再按上述思路辨别是否为逻辑“坏簇”;若是,则将相应簇域内容清为000H,予以回收。如此循环,直到 FAT 内容检测处理完毕;最后把处理过的 FAT1内容写回盘的 FAT1和 FAT2。须注意的是,对于软盘和总簇数不大于4078的硬盘,因其簇域长度为1.5字节,应视奇数还是偶数簇域区别处理。

市售的消毒软件通常并不对病毒所致的“坏簇”进行回收,这样,磁盘在经过感染、消毒、感染……之后,逻辑“坏簇”会不断增多,显著影响盘容量,因此,运行本程序是很必要的。但须注意的是,应先运行操作系统型病毒的消除软件,再运行本程序,否则不能恢复盘的引导扇区;如果已知盘被操作系统型病毒感染,并且先对该盘运行了本程序,就应再用健康盘通过 DEBUG 的扇区读写功能,对被感染盘的引导扇区进行恢复。

使用时,本程序经汇编、链接并转化为 COM 文件,写在 A 盘;待检盘作为 B 盘。对程序进行适当修改,也可检测和维护硬盘。源程序清单如下(本程序在 IBM PC/XT 及其兼容机上通过运行)。

```
code segment
    org     100h
    assume  cs:code,ds:code
begin proc    near
    mov     al,1           ;检测 B 盘
    mov     cx,2
    mov     dx,1
    mov     bx,offset buf1
    int     25h           ;读出 FAT1
    add     sp,2
    mov     si,1           ;si:簇号计数器
    add     bx,3           ;Bx:FAT 中的字节位移
rep:  mov     ax,[bx]
    mov     dx,ax
    inc     si
    call    search        ;判偶数簇域
    inc     bx
    mov     ax,[bx]
    mov     dx,ax
    mov     cl,4
    shr     ax,cl
    inc     si
    call    search        ;判奇数簇域
    add     bx,2
    cmp     si,164h       ;盘文件区共356簇
    jb     rep            ;对 FAT 搜索完否?
    mov     dx,1
    call    write         ;写 FAT1
    mov     dx,3
    call    write         ;写 FAT2
    mov     ax,4c00h
    int     21h           ;返回 DOS
    ret
```