

集粹

电脑用户 实用技巧集粹

陈德满 主编

俞沛之 路文 副主编



化学工业出版社

CHINA CHEMICAL INDUSTRY PRESS

电脑用户实用技巧集粹

陈德满 主 编

俞沛之 路 文 副主编

化 学 工 业 出 版 社

· 北 京 ·

(京)新登字 039 号

图书在版编目(CIP)数据

电脑用户实用技巧集粹/陈德满主编. —北京:化学工业出版社,1995.4

ISBN 7-5025-1472-4

I. 电… II. 陈… III. 微型计算机—计算机应用 IV. TP
36

中国版本图书馆 CIP 数据核字(95)第 03167 号

出版发行 化学工业出版社(北京市朝阳区惠新里 3 号)

社长:俸培宗 总编辑:蔡剑秋

经 销 新华书店北京发行所
印 刷 北京通县京华印刷厂
装 订 三河东柳装订厂装订
版 次 1995 年 2 月第 1 版
印 次 1995 年 2 月第 1 次印刷
开 本 787×1092 1/16
印 张 25
字 数 611 千字
印 数 1—5,000
定 价 27.80 元

目 录

第一章 DOS 系统应用技巧	1
§ 1.1 DOS 内部的秘密	1
§ 1.2 DOS 对磁盘的初始化过程	8
§ 1.3 怎样降低和提高硬盘的 DOS 版本?	10
§ 1.4 怎样解决操作不当出现的问题?	12
§ 1.5 系统配置以及容易出现错误	14
§ 1.6 虚拟盘的设置与使用	16
§ 1.7 DOS 版本的变迁	17
§ 1.8 DOS 不同版本的兼容使用	18
§ 1.9 COMMAND.COM 的使用技巧	19
§ 1.10 灵活使用 DOS 环境变量	20
§ 1.11 文件扩展名标识	21
§ 1.12 DOS3.3 至 DOS5.0 的增扩功能及其应用	26
§ 1.13 DOS5.0 的 EMS 模拟器及其应用	42
§ 1.14 MS-DOS6.0 的新特性	43
§ 1.15 MS-DOS6.0 启动系统设置技巧	45
§ 1.16 怎样在 MS-DOS6.0 操作系统下正常运行程序	46
§ 1.17 怎样避免“Incorrect DOS Version”	49
§ 1.18 怎样安全修改磁盘分区表	50
§ 1.19 怎样方便地复制软盘	51
§ 1.20 怎样限制 FORMAT 命令的使用	52
§ 1.21 怎样防止意外格式化硬盘	53
§ 1.22 怎样恢复硬盘被意外格式化后的文件	54
§ 1.23 怎样较好地使用 DOS 命令 CHKDSK.COM	54
§ 1.24 怎样编制 DOS 系统级菜单、合并 BAT 文件	55
§ 1.25 怎样排序磁盘文件目录	60
§ 1.26 批处理文件中读取按键值程序的设计和应用	60
§ 1.27 巧用 RESTORE 实现从任意一张盘开始恢复文件	62
§ 1.28 PROMPT 命令的多种用途	63
§ 1.29 DRIVER.SYS 设备命令的几种用法	64
§ 1.30 DRDOS 系统驱动程序的灵活配置	65
§ 1.31 DR-DOS6.0 的硬磁盘压缩功能	66
§ 1.32 磁盘增容的好方法	67
§ 1.33 如何在 MS-DOS5.0 版中使用低版本程序	67
§ 1.34 CMOS 内容的保存与恢复	68
§ 1.35 AMI BIOS 微机 CMOS 参数的含义与设置技巧	69
§ 1.36 怎样开发和利用 384KB 内存资源	71
§ 1.37 怎样优化高位内存区的使用	72
§ 1.38 怎样让汉字系统合理驻留在高端内存块	73
§ 1.39 怎样利用 DOS3.3 管理 1M 以上内存	75

§ 1.40 怎样压缩汉字库扩大可用内存	76
§ 1.41 怎样解决屏幕锁死在末行	77
第二章 汉字操作系统	78
§ 2.1 中英文“乒乓”开关的设计	78
§ 2.2 不同汉字系统共存于硬盘与启动方法	81
§ 2.3 显示汉字库的共用	83
§ 2.4 MS-DOS 5.0 下 2.13H 汉字系统使用技巧	84
§ 2.5 DOS 3.3/5.0 下使用 2.13H 及 SPDOS 的方法	85
§ 2.6 2.13H 屏幕死锁的解决办法	86
§ 2.7 3070C.EXE 在使用中的一些问题及解决方法	87
§ 2.8 在 WMDOS 系统中使用 2.13H 中的打印功能	89
§ 2.9 获取 CCDOS2.13H 系统中五笔字型词组的方法	90
§ 2.10 CV26 屏幕背景黑线问题的解决办法	90
§ 2.11 HIMEM 环境下 SPDOS 显示字库的高内存装载	92
§ 2.12 如何减少 SPDOS5.0 和 5.1 频繁读硬盘字库	93
§ 2.13 WPS 怎样利用 2.13 的显示字库	94
§ 2.14 DOS5.0 下 SPDOS5.1 的不兼容性及其解决办法	95
§ 2.15 怎样解决使用 SPDOS6.0F 时内存不够的问题	96
§ 2.16 解决 SPDOS6.0F 与高版本 DOS 中 EMM386 的冲突	96
§ 2.17 SPDOS6.0F 汉字系统内存的充分利用	97
§ 2.18 如何实现汉字系统的自动切换	98
§ 2.19 如何在 WPS 中自由地使用自定义纸张	99
§ 2.20 怎样在 SPDOS6.0F 下使用打印控制命令	100
§ 2.21 巧用 SPDOS6.0F	101
§ 2.22 怎样在进入五笔字型的同时进入纯中文状态	102
§ 2.23 解决 SPDOS6.0F 挂接五笔时内存不够的方法	104
§ 2.24 怎样在 SPDOS 下灵活控制光标	104
§ 2.25 怎样在 SPDOS 下实现功能键的自动切换	105
§ 2.26 怎样在退出 SPDOS5.0 系统时释放所有驻留程序	105
§ 2.27 怎样在无硬盘的微机上运行 CCDOS2.13	107
§ 2.28 怎样在没有硬盘的微机上启动五笔字型	108
§ 2.29 HIMEM 与 2.13H 读虚盘程序 FILE3 的兼容	108
§ 2.30 DOS5.0 中 2.13H 不能使用虚盘的解决办法	109
§ 2.31 怎样在 MS-DOS6.0 下运行 2.13 汉字系统	110
第三章 文字处理系统技巧	112
§ 3.1 怎样在大硬盘上运行 WPS	112
§ 3.2 怎样修改 SPDOS 部分缺省参数	114
§ 3.3 怎样在没有硬盘的 PC 机上运行 WPS	115
§ 3.4 怎样在 WPS 文字处理系统中使用 9 针仿 24 针 TX-850	116
§ 3.5 怎样将硬回车换成软回车	117
§ 3.6 怎样在高版本 WPS 中使用个人词汇库	118

§ 3.7	怎样减少 WPS 的读盘操作	121
§ 3.8	怎样使 WS 在任意机型及汉字操作系统下运行	122
§ 3.9	怎样使汉化 WS 适配各种监视器	122
§ 3.10	怎样扩充 WS 定时存盘功能	124
§ 3.11	消除汉字 WS 自动换行符与分页符	125
§ 3.12	怎样加快中文 WS 的启动速度	126
§ 3.13	怎样直接打印文章段落	126
§ 3.14	怎样修改 WS 的退格键和删除键	126
§ 3.15	怎样使光标一次移动两个字节	127
§ 3.16	文件不能存盘的解决方法	127
§ 3.17	怎样为中文 WS 增加路径选择	128
§ 3.18	怎样在多种系统下使用 CCED	129
§ 3.19	怎样在 CGA 卡及 2.13H 下运行 CCED	129
§ 3.20	CCED 与 *.DBF 数据传递的方法	130
§ 3.21	怎样在只有软驱的微机上使用 CCEDV4.0	131
§ 3.22	怎样在不同硬盘目录下共享各种编辑软件	132
第四章	数据库应用技巧	134
§ 4.1	FOXBASE 主控模块的程序设计	134
§ 4.2	自动生成数据库的结构库	136
§ 4.3	FOXPRO(FOXBASE+)的打印机初始化程序	136
§ 4.4	FOXBASE 反编译的简单方法	137
§ 4.5	dBASE III 向 FOXBASE 全自动转换程序	137
§ 4.6	关于编译 dBASE III 新增功能的用法	143
§ 4.7	数据库通用型立体下拉菜单	144
§ 4.8	FOXPLUS 显示 SPT 图形	146
§ 4.9	页面重叠式通用菜单的设计	148
§ 4.10	消去 MFOXBASE 中每行下黑线的方法	151
§ 4.11	通用输入输出程序	151
§ 4.12	模拟电视游动字幕程序	153
§ 4.13	FOXBASE+菜单管理	154
§ 4.14	建立下拉式和弹出式菜单	157
§ 4.15	立体阴影窗口的设计	158
§ 4.16	完善 FOXBASE 的汉字屏幕的存取	158
§ 4.17	在 dBASE 系统中绘制图形	161
§ 4.18	建立直观的屏幕显示	165
§ 4.19	清屏与显示方式举要	166
§ 4.20	几种检索方法的优劣比较	168
§ 4.21	按随机条件检索数据库记录	169
§ 4.22	模糊检索几点技巧	170
§ 4.23	任意组合条件的数据查询	172

§ 4.24	DBF 文件结构信息	175
§ 4.25	随时中断功能的实现	176
§ 4.26	多用户 FOXBASE+ 错误捕捉	176
§ 4.27	巧用 INKEY() 函数	178
§ 4.28	精巧的 FOX 内存管理器 FOXSWAP	179
§ 4.29	FOX 文件的反编译	181
§ 4.30	数据库文件加密三法	184
§ 4.31	恢复被误清除的数据库记录	186
§ 4.32	dBASE III 功能键的自动切换与按键捕捉	187
§ 4.33	dBASE III 和 BASIC 的功能互补	188
§ 4.34	dBASE III 调用 BASIC 图形文件	188
§ 4.35	dBASE III 与 TURBO PASCAL 的数据交模	189
§ 4.36	dBASE III PLUS, FOXBASE PLUS 调用汇编语言程序	191
§ 4.37	dBASE III 与 COBOL 语言的数据交换	192
§ 4.38	自动生成数据字典中汉字拼音首字母的方法	192
§ 4.39	巧用 WS 修复损坏的数据库文件	193
§ 4.40	FOXBASE+、dBASE 中打印换页问题	194
§ 4.41	实现数据库记录中“0”不打印的两种方法	196
§ 4.42	FOXBASE 中实现报表的无格式打印	197
第五章	程序设计技巧	202
§ 5.1	怎样了解系统资源	202
§ 5.2	如何利用程序识别显示卡	204
§ 5.3	获得键盘编码的通用方法	206
§ 5.4	怎样快速移动键盘光标	207
§ 5.5	键盘功能键的自动切换	208
§ 5.6	如何了解系统的驱动器数	209
§ 5.7	怎样灵活控制打印	210
§ 5.8	利用驻存程序直接获得键码	211
§ 5.9	如何将 BAT 文件转为 COM 文件	213
§ 5.10	执行程序的结构与加载	217
§ 5.11	怎样扩充 COM 文件	220
§ 5.12	菜单程序的设计	222
§ 5.13	怎样避免重复驻留内存	225
§ 5.14	怎样驻留内存并退出	231
§ 5.15	如何释放特定内存块	232
§ 5.16	如何有选择地清理驻存软件	233
§ 5.17	屏幕图形文件的立即执行	235
§ 5.18	怎样使用扩展键盘读功能	235
§ 5.19	利用汉卡提高汉字显示速度	238
§ 5.20	怎样用程序关闭屏幕显示	239

§ 5.21	怎样存取 EGA/VGA 图形	240
§ 5.22	怎样提高图形存取与打印的速度	242
§ 5.23	怎样快速存取 GWBIOS 屏幕	244
§ 5.24	怎样存取部分屏幕	246
§ 5.25	怎样实现逐层清屏	248
§ 5.26	用刷新法实现动画显示	249
§ 5.27	用 C 语言实现对 BIOS 的调用	250
§ 5.28	后台程序的设计方法	252
§ 5.29	如何在高密单软驱上使用 TURBO C2.0	255
§ 5.30	如何加快应用程序的运行速度	255
§ 5.31	提高解释 BASIC 运行速度的小技巧	258
§ 5.32	对 BASIC 加密的解密	258
§ 5.33	用 BASIC 实现汉字放大显示及动画	259
§ 5.34	如何西文 DOS 下直接显示汉字	261
§ 5.35	如何使应用软件兼容各类汉字系统	263
§ 5.36	如何在 DOS 下调用 SPT 图形	266
§ 5.37	如何在彩色立体窗口中显示汉字	267
§ 5.38	如何在西文 Auto CAD 下标注汉字	268
§ 5.39	怎样在 DOS6.0 下运行汉化 Auto CAD10.0	269
§ 5.40	无协处理器巧用 Auto CAD	270
§ 5.41	怎样用 Auto CAD 做三维旋转	271
§ 5.42	利用 DOS 加载编写反跟踪函数	272
§ 5.43	橡皮筋技术的开发	273
§ 5.44	怎样制作通用立体菜单	275
第六章	常用工具软件应用技巧	276
§ 6.1	关于工具软件的二次开发应用	276
§ 6.2	ARJ 文件压缩程序命令及开关功能表	277
§ 6.3	高性能数据压缩/还原软件 LHARC	281
§ 6.4	PKLITE 压缩/加密可执行文件软件	283
§ 6.5	磁盘碎片与 COMPRESS 程序的正确使用	284
§ 6.6	使 DEBUG 进入单步连续监视状态	286
§ 6.7	使用 PCTOOLS7.0 恢复 ZAP 数据库	287
§ 6.8	硬盘安全分区及分区写保护处理方法	288
§ 6.9	给 BAT 文件增加菜单的方法	289
§ 6.10	增强 DEBUG 的跟踪能力	290
§ 6.11	DEBUG 的反汇编存盘与压缩输出	291
§ 6.12	DEBUG 的几点实用性改进	292
§ 6.13	防硬盘误格式化的方法	293
§ 6.14	低级格式化硬盘的四种方法	293
§ 6.15	恢复软盘上被误删除的文件	295

§ 6.16	挽救受损软盘数据的方法	297
§ 6.17	巧用虚拟图形设备 GRD.SYS	299
§ 6.18	联想汉字系统中 ALLOC.COM 程序的使用技巧	301
§ 6.19	DRDOS6.0 下使用金山 SPDOS 的几点技巧	303
§ 6.20	关于 DRDOS6.0 的 EDITOR 功能键	304
第七章	加密解密技巧	305
§ 7.1	文件加密的主要方法	305
§ 7.2	磁盘加密技术的原理及其应用	306
§ 7.3	建立绝对保密的子目录	308
§ 7.4	怎样防止硬盘上执行文件被拷贝	308
§ 7.5	如何根据 BIOS 参数块查找起始地址	310
§ 7.6	用程序方法防止非法复制	310
§ 7.7	更有效地利用装配程序防止非法复制	312
§ 7.8	常用的反跟踪技术	315
§ 7.9	怎样加密软盘反跟踪	317
§ 7.10	怎样对子目录消隐加密	319
§ 7.11	怎样对硬加密软盘进行拷贝	320
§ 7.12	软件时序保护法	322
§ 7.13	怎样用密匙法加密	323
§ 7.14	方便的磁盘文件硬加密法	325
§ 7.15	有效加密文件的技巧	327
§ 7.16	用汇编语言为 FOXBASE 文件加密	328
§ 7.17	位操作用于文件加密	329
§ 7.18	查看 CMOS 中开机密码的程序	330
§ 7.19	怎样备份加密软件	331
§ 7.20	软件加/解密技术分类与特点	332
§ 7.21	解密工具软件 LLGZ 与 SOFE-ICE 的比较	335
§ 7.22	如何使用 LLGZ 破译加密软件	337
第八章	电脑维护技巧	341
§ 8.1	电脑一般性维护	341
§ 8.2	怎样维护硬盘驱动器	343
§ 8.3	如何有效使用硬盘	344
§ 8.4	怎样看待扩充软盘容量的做法	345
§ 8.5	软盘修复小技巧	346
§ 8.6	排除软盘驱动器的软故障	347
§ 8.7	修复引导失败的硬盘	348
§ 8.8	恢复逻辑损坏的硬盘	349
§ 8.9	程序修复硬盘逻辑损坏	351
§ 8.10	修复 0 道物理损坏的硬盘	354
§ 8.11	用 DM 软件恢复硬盘	355

§ 8.12	用 DEBUG 修复硬盘自举失败	356
§ 8.13	软盘受损后的补救	359
§ 8.14	零道受损软盘格式化小技巧	360
§ 8.15	用 HD 软件修复 0 道坏软盘	361
§ 8.16	汉字显示故障排除	361
§ 8.17	怎样消除 WPS 的“假死机”	362
§ 8.18	PCTOOLS8.0 的抗病毒功能介绍	362
§ 8.19	DM 软件恢复硬盘自举	363
§ 8.20	怎样备份硬盘引导记录	364
§ 8.21	如何设置 CR3240 打印机的电子开关	364
§ 8.22	系统不确认硬盘故障排除	365
§ 8.23	电脑病毒现象及其防治	365
§ 8.24	如何防治系统引导型病毒	368
§ 8.25	如何防治文件型病毒	370
§ 8.26	怎样防止系统引导型病毒感染硬盘	373
§ 8.27	利用内存管理检测病毒	376
§ 8.28	驻存检测病毒法	378
§ 8.29	使用 BOOTSAFE 防治系统引导型病毒	381
§ 8.30	VSAFE 程序的使用方法	383
§ 8.31	清除硬盘系统引导型病毒	386
§ 8.32	怎样清除死而复生的病毒	388

第一章 DOS 系统应用技巧

电脑的各组成部分是怎样互相配合、协调一致地工作的呢？这是通过操作系统去控制的。操作系统是电脑系统的重要组成部分，是电脑所有软硬件资源的组织者和管理者，任何一个用户都是通过操作系统使用电脑的。操作系统是为提高电脑利用率，方便用户，缩短电脑响应时间，对电脑系统进行控制与管理的大型程序，它由许多具有控制和管理功能的子程序组成。最常用的操作系统是磁盘操作系统，即 DOS。DOS 系统的基本任务一般包括以下几个方面：管理好电脑的全部资源（如中央处理机、存储器、各种外设、程序和数据等），使它们能够被充分利用，从而有效地进行工作；同时，它是用户与电脑的接口，让用户使用方便，操作顺利，不必过问电脑硬件的具体细节，就能使它成为一台功能较强的电脑。

随着电脑技术的不断发展，DOS 的版本不断更新，功能不断增强，在短短的 10 余年时间里，其从 DOS3.1 发展至 DOS6.21，管理功能起了飞跃性的变化。这些变化对于一般电脑用户来说，许多已经非常熟悉了，但仍有不少内容对许多用户在不同程度上显得生疏或不知所措，譬如高版本 DOS 的内存管理问题、大硬盘上安装汉字系统问题、倍密磁盘容量问题等，可以说这些难以入门的地方正是高版本 DOS 系统功能闪光的部分，为了跟上电脑飞速发展的步伐，必须攻克这些大大小小的难题。也正是出于这一目的，我们从广大电脑用户的角度出发，本着抓住关键、实用入手、深入浅出、通俗易懂的原则，精心选择了广大用户比较关心的 DOS 实用技术方面的问题，通过对 DOS 功能应用的实例分析，对一些典型问题进行了归纳和总结，以帮助各位用户解决电脑使用中出现的各种难题。

§ 1.1 DOS 内部的秘密

摘要：DOS 是磁盘操作系统 (Disk Operating System) 的简称，它是电脑发挥功能的核心和灵魂。那么，DOS 究竟是怎样工作的，其运行原理与核心结构是什么？作为电脑用户，首先必须对此有一个深入的了解。请跟我们一起去看一下 DOS 系统的内核及其秘密。

一、DOS 的基本功能

为使自己成为电脑的明白的主人，就需了解其操作系统。这里提供一次走马观花的机会，以了解什么是操作系统。尽管我们不可能学会对操作系统动大手术，把它拆开，再重新组装，但能学会对它的一些增减修补。

一个操作系统的目的是顺利地运行一台电脑。按该术语的全部含义，操作系统是电脑系统中首要的和最重要的程序，通常亦是最复杂的程序。这最复杂的计算机程序被简单地用来管理计算机，实在是很了不起。更了不起的是，迄今人们造成的最强有力的工具——计算机已被很有效地用于计算机自己。你无法用一个锯把计算机锯成两半，但却可以用一台计算机去操作另一台计算机，这得感谢操作系统。

一个操作系统所做的大部分工作是使你能避免某些非常凌乱和乏味的细节。举例说，让我们看 DOS 的 COPY 命令。假如我们用这条命令把一个软盘拷贝到另一个软盘上去。这看起来似乎很简单。但是，为完成拷贝操作，系统将不得不做下列事情，并且还只是所要做的事情的一小部分。你可能会大吃一惊，为完成像拷贝这样的简单工作，操作系统所必需做的工作竟是那么多。

- 在原软盘上有所给出名字的文件吗？
- 拷贝的目标是否不是一个软盘文件(比如是一个打印机)？
- 在目标软盘上是否已经有该名字的文件？
- 在目标软盘上是否有放该文件的空间(把盘上所有可用的自由空间都计算在内,并且,如果那里已有该文件的拷贝,则这个老的拷贝所用的全部空间也算在内)？
- 如果该文件对目标软盘是新的,那是否有空间来放一个新的目录条目,或者目录是否满了？
- 原软盘是单面还是双面,或者另一种格式？
- 目标软盘是单面还是双面,或者另一种格式？
- 该文件是被拷贝给自己的吗(这是不允许的)？
- 原软盘和目标软盘驱动器是确实存在的吗?如果不是,那这是否是一个单盘系统,从而不得不虚构一个驱动器 B?
- 是从目录规定的文件长度去找到原文件的结尾呢,还是按文件标志的一个逻辑结尾(例如 ASCII 文本文件)去找？

- 原盘的文件位置分配表是在内存吗？
- 目标盘的文件位置分配表是在内存吗？
- 原文件在其文件位置分配表内是否正确地描绘其踪迹？
- 目录给的文件尺寸与分配表给的文件尺寸是一致的吗？
- 在内存中有多少空间可用来缓冲存储该文件的拷贝？
- 缓冲存储的空间是大于还是小于 64K？
- 我们是否需要擦除命令程序以获得更多的缓冲存储空间？

到此为止,想必你对这张表已感到厌烦了。但这是刚开始哪!那还只是问题的逻辑方面,并且就是在这里,我们亦已略去许多细节了。下面列出问题的物理方面,这里所略去的实际细节甚至更多:

- 盘驱动器的马达在运行吗？
- 寻找臂定位在正确磁道上吗？
- 在这个磁道上我们要读/写多少个数据区？
- 驱动器是否已准备好去执行一个命令？
- 软盘是否表现不好?我们是否需要重新启动,重新检验？
- 如果我们正在重新尝试,那在抱怨以前,我们所尝试的次数是否已足够多？
- 对我们抱怨的反应会是什么?是再试一次,不去管它,或是宣告异常结束？
- 我们要等待盘驱动器吗？
- 读/写/寻道过程是否顺利完成?目标盘是写保护的吗？

这里所列的还只是某些凌乱细节的粗略梗概,远不是拷贝一个文件会碰到的全部问题。并且拷贝还是一项相对说来算简单的 DOS 操作(虽然这个操作把几乎每一件事都做一点)。顺便提一下,这里所作的逻辑和物理之间的区分是操作系统中的一个非常重要的问题。

像 DOS 这样的操作系统的主要工作是使你不必为所有这些细节操心。此项任务最繁重部分包括输入输出设备,如软盘驱动器、打印机或通过电话线相连的异步通信等设备的繁多而细小情况的处理。

此外,DOS 还提供较高一级的服务,如软盘目录的搜索、文件的拷贝和程序的加载。

操作系统设计成功的关键之一是模块化。操作系统的设计者把所需做的工作分解成功能性质明显不同的各部分时,操作系统就变得简单了,效能亦更高。随后,还需把这些部分组织到一个精心定义的层次结构中去。在层次结构中,每一层都处理其所分配到的细节,从而使它上面的各层不必再关心这些细节(并且依此推论,亦不必关心下面各层的细节)。所以,下面我们将讨论 DOS 的各模块部分。

二、DOS 的基本组成

DOS 基本上有六个模块组成。第一个是 ROM-BIOS。ROM-BIOS 是内装在 IBM/PC 里面的,并可用作任一操作系统的一部分。部分 ROM-BIOS 的工作是提供计算机所需要的大多数基本服务。因为 IBM/PC 的 ROM-BIOS 是内装在它里面的,所以 ROM-BIOS 不是 DOS 的一个独有部分,它是每一种 IBM/PC 操作系统的一部分。

下一部分是软盘的“引导记录”。引导记录是一个很简单的程序,它放在每一软盘的第一扇区上。引导记录的工作是当 PC 刚加电启动或再启动用(Ctrl-Alt-Del 键)时,开始操作系统的加载过程。引导记录把操作系统的下二部分读入内存,从而使它们能做完加载 DOS 的工作。

这下二部分是两个软盘文件,IBMBIO.COM 和 IBMDOS.COM。它们都由引导记录进行加载,并且在 DOS 工作时都留在内存里,所以它们之间的区别不明显。

IBMBIO.COM 用作 ROM-BIOS 的一个可变换的扩充部分。从 DOS 2.00 开始,IBMBIO.COM 可以用另一些部分加以扩充,我们把它们叫作独立的设备驱动程序。后面还将更多地提及它们。ROM-BIOS、IBMBIO.COM 以及设备驱动程序一起就组成了操作系统的“物理”部分。

IBMDOS.COM 提供 DOS 的核心服务项目。这是操作系统输入输出的“逻辑”部分。

这两个文件都是“隐”“系统”文件。作为隐文件,当你用 DIR 目录命令列出软盘上的文件时,它们是见不到的。然而 CHKDSK 命令(DOS 1.10 以上版本中有)能使你确信其存在,但却仍不告诉你它们的名字。如用实用程序命令 Disk Look 来检查软盘,那 IBMBIO.COM 和 IBMDOS.COM 就会按名字列出。浏览这二份文件就能知道它们包含有什么内容。IBMBIO.COM 的工作之一是把操作系统的下一个部分装入内存。

第五部分是叫作 COMMAND.COM 的软盘文件。COMMAND.COM 的基本工作是处理你从键盘敲入的 DOS 命令。类属于内部命令的那些 DOS 命令——如 TYPE,COPY 和 DIR——实际上都放在 COMMAND.COM 里面。COMMAND.COM 本身又分成二部分,一部分成为 IBMDOS.COM 的扩充,而另一部分成为暂时性的命令处理程序。

第六部分,亦是 DOS 的最后一部分,包括了所有外部命令,如 FORMAT 和 DISKCOPY。这些命令的工作是各不相同的,执行这些工作的程序只是在需要时才将它们加载到存储器里。和上面已述及的其它五个部分不同,虽然这些程序是和 DOS 一起提供的,但严格地说,外部命令程序并不是 DOS 操作系统的一个不可分割的组成部分。

外部命令不总是放在内存里,它们是 DOS 的非常驻部分。两个系统文件 IBMBIO.COM 和 IBMDOS.COM,与一些设备管理程序一起,是 DOS 的常驻部分。COMMAND.COM 是半常驻的,属于特别类型,引导记录只是被临时用一下,所以它不是 DOS 的常驻部分。

三、DOS 的基础部分:ROM-BIOS

DOS 的第一部分是放在只读存储器中的 ROM-BIOS,或叫基本输入/输出系统。ROM-BIOS 提供某些最基本、最初步的操作系统的服务。

因为 ROM-BIOS 是内装在 IBM/PC 里面的,并且只有换系统硬件时才能改变它,所以对 PC 来说它是非常基本的。作为 IBM/PC 的内装部分,ROM-BIOS 就不是 DOS 本身一部分;它是在 IBM/PC 上运行的任一操作系统的一部分。

ROM-BIOS 由好几个部分组成,其中多为程序(其余是一些重要的数据表格)。要执行的第一个 ROM-BIOS 的程序是加电测试子程序。当电源开关接通时,此程序就对存储

器和接在 IBM/PC 上的设备进行测试。

所执行的 ROM—BIOS 的下一个部分是一个启动操作系统的程序——用其引导程序把操作系统上推。这个程序要检查是否安装有一个软盘驱动器,然后再从软盘读出“引导记录”。读了引导记录后,启动程序把控制转给引导记录,从而使后者能够去读操作系统的其余部分。

除了我们已经提到的两个部分:加电自测和启动引导加载程序外,在 ROM—BIOS 里还有许多其它程序,而这些程序正是我们最感兴趣的部分。这些程序包括对 IBM/PC 上所有标准外围设备的支持程序,这些是为键盘、显示屏、软盘、异步通信转换器、打印机以及盒式磁带接口等完成主要操作功能的程序。用 IBM/PC 实现大多数有意思的技巧时,亦是离不开这些程序的。

第二个模块是启动加载,引导记录。引导记录被用来启动 DOS。一个引导记录包含为读出和启动操作系统各主要部分所必需的最小数量。像前面说过的那样,计算机用它的引导程序把操作系统上推,它亦因此得名。这是“引导加载程序”,为简便亦叫“引导程序”。

当加电时或通过按 Ctrl—Alt—Del 键来启动 IBM/PC 时,ROM—BIOS 启动子程序把启动盘上的第一个记录读入存贮器的一个标准位置(在地址 31744,十六进制 7C00 上)。读出引导记录后,ROM—BIOS 转向地址 31744 转移的方向,把主控权转移给引导程序,引导记录随后必须继续将操作系统装入内存。

对 DOS 来说,引导记录的主要工作仅是装入 IBMBIO.COM 和 IBMDOS.COM 文件。引导记录的尺寸大小是一个标准的软盘扇区,即 512 字节,这对一个很复杂的程序是不够大的。为简化引导程序的工作,把 IBMBIO.COM 和 IBMDOS.COM 这两个文件放在启动盘预先确定的位置上,这就使引导程序省去搜索它们的麻烦,而读所有其它文件时都是需经过搜索的。

顺便指出,这就是“系统格式化”软盘与一般软盘不同的原因。它含两个系统文件:IBMBIO.COM 和 IBMDOS.COM,并且把两个文件存放在标准的、预先确定的位置上。这亦是不能随便把一个一般的软盘改变为系统软盘的原因。

虽然引导记录程序无法找到文件位置,但它完全可以检查目录,以确知这两个系统文件是否被正确地列出。因为这两个文件一定出现在软盘目录上,它们被标上“隐”和“系统”两种记号而受到保护,不会遭到擦除或更改。

由于其任务简单,所以引导记录是 DOS 操作系统的一个比较稳定的部分,只有在系统文件改变其大小尺寸或位置的情况下,它才不得不随之改变。

你可以发现不同的 DOS 版本中的引导记录是不同的,如果你有实用程序 Disk Look,用它能很方便地检验引导记录:启动 DiskLook,按功能键 F7,从键盘敲入引导记录的位置:0 面、00 磁道和扇区 1,随后再按键 F6,引导记录即被显示出来。

四、DOS 的设备管理:IBMBIO.COM

两个系统文件中的第一个 IBMBIO.COM,其任务是扩充 ROM—BIOS。ROM—BIOS 与 IBMBIO.COM 二者都处理输入输出管理事项,也称设备管理。这项工作包括照料操作各种 I/O 设备(如软盘)时所碰到的所有比较繁琐的细节,其任务包括差错检测和校正。

和 ROM—BIOS 不同,IBMBIO.COM 中的程序是比较容易改变的。IBMBIO.COM 的任务是做三件 ROM—BIOS 所不能做的事。第一件是根据特定操作系统的特殊需要而裁剪拼接。任何一个操作系统,都可使用 ROM 中通用的 BIOS;但不同的操作系统还必须提

供其各自不同的 BIOS 部分。

IBMBIO.COM 的第二项任务是在必要时去修正 ROM-BIOS 中的差错。像 ROM-BIOS 这样的程序是经过非常仔细的测试的,因为一旦把它放入 ROM,本质上讲就不能再变了。如果此后在 ROM-BIOS 中还发现有差错,通常就得通过改变 IBMBIO.COM 来进行校正。这由 IBMBIO.COM 重置中断向量来实现,使得 BIOS 操作首先运行 IBMBIO.COM,然后再运行 ROM-BIOS。调用 ROM-BIOS 中的程序的规则始终包含使用中断,而不是转移到 ROM 中的程序位置。使用中断的一个很重要的理由是使 ROM-BIOS 的程序可由 IBMBIO.COM 中的程序来替代,否则,我们就无法超越 ROM-BIOS 程序。

第三件 ROM-BIOS 所不能做而 IBMBIO.COM 却必须做的事,是管理新的外围设备,如大容量硬盘,或 8 英寸软盘,或绘图仪,或可能接在 PC 上的几百种计算机设备中的任一种。当 IBM/PC 加接新的输入输出设备时,这些新的设备的支持程序可以加在 IBMBIO.COM 或它的附属程序里,而不需置换有 ROM-BIOS 的 ROM 存储器片子。

在 DOS 的早期版本中,加接新设备的任务包括对 IBMBIO.COM 的内容作修改,并且还可能要对其它的基本 DOS 程序的内容作修改。虽然,这对 Microsoft 或 IBM 公司不算一回事,但对那些想在 DOS 上增加设备支持程序的其他人来说实在是件麻烦事。对你或我,用这种方法在 DOS 上增加设备支持程序就意味着做某些完全是修修补补的工作——一件艰巨而冒险的工作。

由于计算机获得成功的很重要的标志之一就是提供增加新设备的自由度,所以,从 DOS2.00 版开始,就作了改变,以使增加新设备支持程序的工作做起来更方便。当 IBMBIO.COM 首次进入运行时,它就对软盘中的配置文件进行检测。如找到配置文件,就把它读出用作指南,其中的部分内容可用来设定各种不同的系统参数。

配置文件中的指南包括需要包含到 BIOS 中去的各种设备控制器程序的名字。随之,每一个设备控制程序都作为附加程序被加载到 IBMBIO.COM 中去。这种方案使增加新设备的工作可按模块化方式进行,比较切实易行,而不需打扰 DOS 的系统文件。

通常,在 IBM/PC 上运行的程序都希望使用 BIOS 的传统 DOS 版本。但偶尔亦有某个程序要求对 I/O 操作有特殊管理。由于 IBMBIO.COM 是一个软盘文件,如需要可以改变它,从而形成 IBMBIO.COM 的用户版本。

五、DOS 的核心部分:IBMDOS.COM

两个系统文件中的第二个是 IBMDOS.COM。

该文件由 DOS 服务性例行程序组成,这些程序不是直接支持输入和输出的程序。虽然把 IBMBIO.COM 与 IBMDOS.COM 的功能分开不是本质性的问题,但把它们分成二个不同的程序文件却使 DOS 更模块化,并可把那些特定的只适用于某一种计算机的部分与适用于所有 DOS 计算机的共用部分分开。

DOS 的服务性例行程序又分为由其自己的专用中断调用的例行程序和那些共享一个中断码的例行程序(第 33 号,十六进制的 21)。按 DOS 的专门术语,第一组叫 DOS 中断,第二组称 DOS 功能调用。在所有这两种情况下,都用软件中断来调用它们,这样做的理由与 BIOS 程序使用中断的理由相同:模块化。中断 32~63(十六进制的 20~3F)这个范围留给 DOS 使用,但其中只有少数几个用上了,余下的都留作将来需要时用。DOS 的中断服务包括读写软盘上的扇区,亦包括在 DOS 出现差错时以及 Ctrl-Break 键盘操作时对控制的访问。

DOS 的功能性服务在中间一级提供了大多数的输入输出服务,例如读键盘输入、对显示屏的输出、对异步通信线路的输入和输出以及打印机输出等。亦提供有各种软盘逻辑操作,如开启和关闭一个文件、文件目录的搜索、文件的擦除和建立以及读写数据等,凡一个程序为控制文件以及文件中的数据所需要的基本操作,这些例行程序几乎都提供了。程序就不用着自己去做目录译码、文件位置分配表等。如你的程序对文件和数据的管理需要作的控制超出了你所用的程序设计语言许可范围,但你又不希望触及软盘数据格式这样的困难问题,那 DOS 文件服务功能就正是你所需要的。

较高级别的 DOS 程序频繁使用这些 DOS 服务例行程序中的大多数。例如 DIR 与 COPY 命令就要用目录搜索服务。命令解释程序用它来寻找程序文件。

六、命令:COMMAND.COM 与内部命令

DOS 的下一个部分是 COMMAND.COM,不论从它们为我们做些什么,还是从它是怎样操作的这两方面来讲,它都是最有意思的一个。COMMAND.COM 有好几种功能。首先它是“命令处理器”,这就是说,它负责读出我们由键盘输入的命令,并确定要怎样对待它们。

假如我们输入一条内部命令,如 DIR,TYPE,REM,或 PAUSE,那么我们就是在要求一种内装在 COMMAND.COM 里的服务。而且这条内部命令可以立即执行。

为识别内部命令,COMMAND.COM 有一个命令名字表,如果能浏览一下持有 COMMAND.COM 的软盘文件的内部情况,就能找到这些命令的名字,可以用 DEBUG 或 Disk Look 来检验 COMMAND.COM。在其中还能发现 DOS 开始操作时所显示的起始信息。如需要可以用 DEBUG 或类似 Sec Mod 的实用程序来改变这些信息,使 DOS 在运行开始时就以你的名字来欢迎,或显示你公司的标志。也可以改变内部命令的名字;如你不改变名字的长度,那是很容易的。

如果一条命令不在内部命令表里,那就是一条外部命令,也就是放在软盘文件中由 COMMAND 寻找的命令。COMMAND.COM 根据我们的要求,在相应的软盘中搜索一个命令处理文件,并把它投入执行。

有三类命令处理文件:用其文件名的扩展部分来区别的,按其优先权次序,分别为:“.COM”,它用两种程序格式中的一种来表示一个程序文件;“.EXE”,它用这两种程序格式中余下的一种来表示;和“.BAT”,表示一个批处理文件。COMMAND.COM 以严格的次序搜索这三类文件。

当 COMMAND.COM 找到一个程序文件,不论是什么格式的,都把该程序加载到内存里去并进行必要的变换。把程序加载入内存并为其加上程序段前缀之后,COMMAND.COM 就把控制权转移给程序,从而使程序能做它自己的工作。

如果命令处理是一个“.BAT”批处理文件,那它就是由 ASCII 文本文件格式的一串命令组成,这些命令将被执行,就像它们由键盘输入的那样。COMMAND.COM 有许多任务,其中一项是记住其在批处理文件中的位置,从而当文件中的一个命令结束时,文件中的下一个命令就能被读出。如果一个批处理文件中的一个命令引起了另一个批处理文件投入运行,那就不可能再返回第一个批处理文件;批处理文件是不能嵌套的,但它们可以一个个连接在一起。

在有些计算机系统上,所有输入的命令均可被重新引向一个文件,这也适用于从键盘获得其输入的程序,以及命令解释程序。但 DOS 的情况不是这样的,只有 COMMAND.

COM 命令解释程序自动地从批处理文件中读出。

顺便指出,程序可以把修改情况写入一个批处理文件,从而控制下一个要执行的命令或程序是哪一个,在复杂的应用中这是普遍使用的手段,并且可把它当作程序连接技术的一种天然代用品。

到目前为止我们还只接触 COMMAND.COM 内容的很小一部分。COMMAND.COM 实际上分成三个部分。第一部分放在内存,其位置在那些属于 IBMBIO.COM 和 IBMDOS.COM 程序的后面,并且和那些程序一样,亦成了 DOS 常驻部分的一员。实际上,COMMAND.COM 的这一部分和 IBMDOS.COM 是没有区别的。

COMMAND.COM 的第二部分仅被暂时使用。在启动时,它被用来寻找并执行一个 AUTOEXEC.BAT 批处理文件。一旦这些事做完,它就没用了。

COMMAND.COM 的第三部分,是最有意思的驻留部分,是 DOS 的最巧妙性能之一。这部分包括命令解释,其中有执行内部命令的程序。当前,一个像命令解释程序那样复杂的程序是不可能很小的。一方面,我们很希望命令解释程序总是在内存里,另一方面却又不希望它总是占有着存贮空间,特别是当内存不是很大的情况下。

该问题的有意思的解决办法是把 COMMAND.COM 的这一部分放在内存的高端(通常是最后部分),并允许其它程序写入内存的这部分。每当要用命令解释程序时,COMMAND.COM 的常驻部分就先检验,看看命令解释程序是否仍在那里,而未被其它程序冲掉。如果它已不在那里,则需从软盘重新加载(顺便指出,这就是为什么在你的大多数软盘片上都需有 COMMAND.COM)的拷贝,即使有些软盘片不是系统格式化的也罢。如你有时得到这样的信息“Insert DOS disk……(插入 DOS 盘……)”,这是因为在你所用的盘片上找不到 COMMAND.COM。

为检测该临时部分是否在内存里,那就由 COMMAND.COM 的常驻部分计算一下,该临时部分所在位置的检验和数是多少,如果检验和数和预期的不相符,那就重新装入 COMMAND.COM。顺便提一下,当 COMMAND.COM 被重新装入时,亦计算和检验,如有偏差,则 DOS 就会提出来。即使偏差只是在 DOS 的起始信息上(The IBM Personal Computer DOS……),而且前面曾提过,你是可以改变这个起始信息的,DOS 还是会提出来的,所以,如果你真的进行了这样的改变,或对 COMMAND.COM 作了另外的改变,那你就应在所有的软盘上作一模一样的改变。

COMMAND.COM 做成一个独立的文件,而不和两个 IBM 的系统文件组合在一起的原因之一,是想让它更便于写在用户版。这是为适应特殊需要 IBM/PC 所采用的主要方法之一。如果需要把某些用户命令变成内部的,或者如果需要把命令解释程序的操作方式加以改变,那就需要写一个特殊的 GOMMAND.COM。

IBM 的手册指出,可以通过按 Ctrl+Alt+Del 键来重新引导操作系统。这是从读出引导记录开始的一次完全的系统重新启动。然而,有另一种重新启动方法:打入 COMMAND.COM 命令,则 COMMAND.COM 将被重新加载,那么不用重新加载 IBMBIO.COM 与 IBMDOS.COM 就能使系统被重新启动。这样,就可获得一份命令解释程序的刷新了的拷贝。并执行一个 AUTOEXEC 批处理文件。

七、DOS 的外部命令

DOS 操作系统的最后部分由外部命令组成。之所以叫外部命令,是因为它们不包括在 DOS 常驻 IBM/PC 内存的那部分。外部命令驻留在磁盘上的程序文件里。