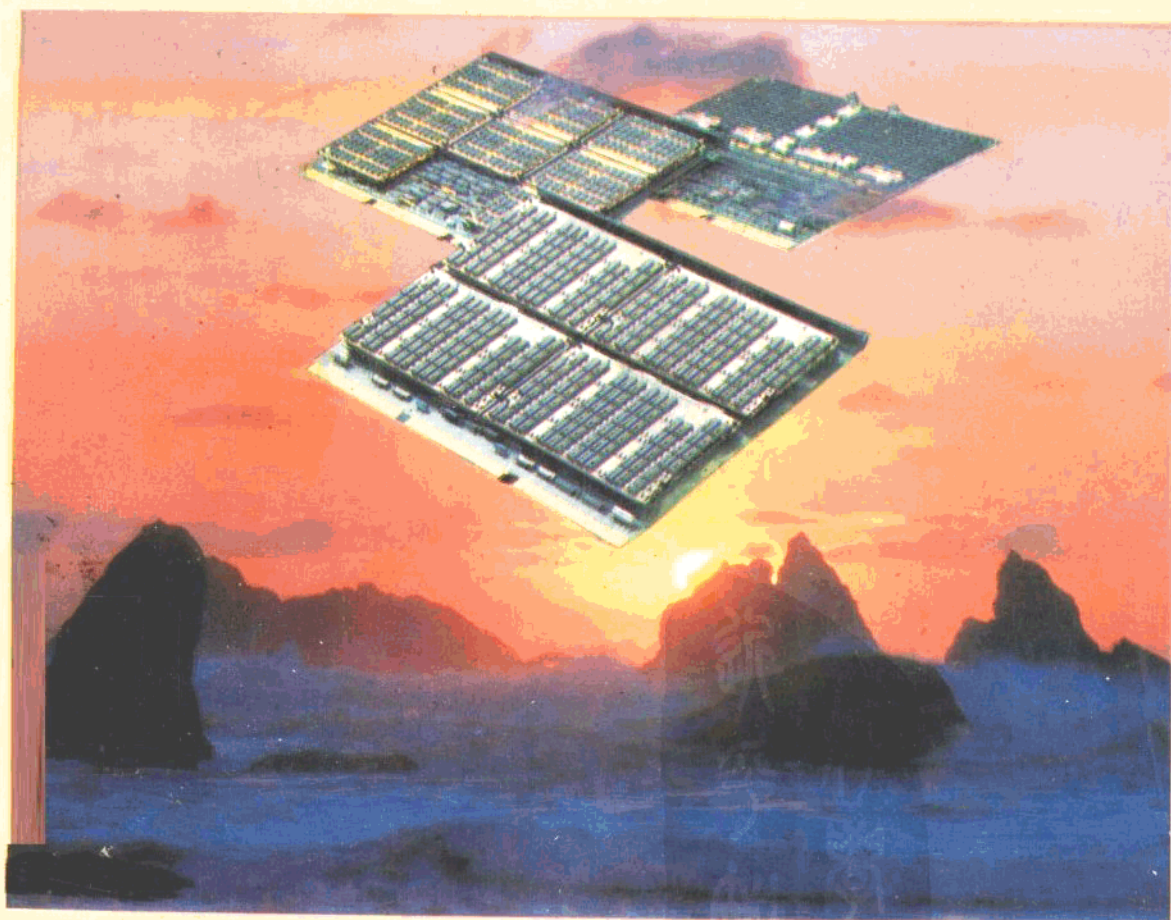




四川计算机软件资料社技术丛书

DOS 6.0 技术、技巧 与实用程序大全

(下)



第 二 部 分

目 录

第十一章 可变的配置

§ 11.1 在引导时的可选操作	(225)
§ 11.1.1 用 F8 键清除引导	(225)
§ 11.1.2 CONFIG.SYS 中的用户可选项	(226)
§ 11.2 多重 PC 特性	(228)
§ 11.2.1 多套匹配的启动文件	(228)
§ 11.2.2 第三方 DOS 引导管理程序	(228)
§ 11.2.3 第三方管理程序 DOS 6 方法	(229)
§ 11.3 更多的多重特性	(231)
§ 11.4 AUTOEXEC.BAT 中的菜单	(231)
§ 11.4.1 CONFIG.SYS 中的菜单	(232)
§ 11.4.1.1 CONFIG.SYS 菜单的设置	(232)
§ 11.4.2 使用 CONFIG.SYS 菜单的 AUTOEXEC.BAT 文件	(235)
§ 11.5 警告和建议	(235)
§ 11.5.1 制作安全引导盘	(235)
§ 11.5.2 什么是安全引导盘	(235)
§ 11.5.3 安全引导盘的制作	(236)
§ 11.5.4 安全引导盘的测试	(237)
§ 11.5.5 清除安全引导网	(237)
§ 11.5.6 引导问题的诊断	(238)
§ 11.5.7 [COMMON]检测	(239)
§ 11.5.8 启动文件变化的发现	(239)
§ 11.5.9 /K 和 [TRASH] 替换	(239)
§ 11.6 MemMaker 和 CONFIG.SYS 的菜单	(241)
§ 11.7 用于可变配置的 DOS 命令	(242)
§ 11.7.1 CONFIG.SYS 指令	(242)
§ 11.7.2 DOS 可装入的设置驱动程序	(242)

§ 11.7.3 CONFIG.SYS 菜单命令	(242)
§ 11.7.4 AUTOEXEC.BAT 配置命令	(242)
§ 11.8 用于可变配置的 Power Tools 命令	(242)

第十二章 设备驱动程序

§ 12.1 设备驱动程序怎样运行	(243)
§ 12.1.1 设备链	(243)
§ 12.1.2 历史回顾	(243)
§ 12.1.3 DOS 2.0 版的改进	(244)
§ 12.2 DOS 默认的设备驱动程序	(244)
§ 12.3 可安装的 DOS 设备驱动程序	(246)
§ 12.4 其他的可安装设备驱动程序	(246)
§ 12.5 设备驱动程序链的连接	(246)
§ 12.6 非设备驱动程序	(249)
§ 12.7 DOS 提供的可安装设备驱动程序	(250)
§ 12.8 修改 DOS 默认设备驱动程序	(261)
§ 12.9 DOS Power Tools 与设备驱动程序	(261)

第十三章 DOS 的环境

§ 13.1 DOS 环境介绍	(262)
§ 13.1.1 把程序装入内存	(262)
§ 13.1.2 DOS 环境的结构	(263)
§ 13.2 DOS 环境的使用	(263)
§ 13.2.1 COMSPEC	(263)
§ 13.2.2 PATH	(264)
§ 13.2.2.1 COMMAND.COM 如何执行命令	(264)
§ 13.2.2.2 DOSPATH 的结构	(266)
§ 13.2.2.3 用文件保存路径定义	(267)
§ 13.2.2.4 PATH 最大长度	(268)
§ 13.2.2.5 对 PATH 的控制	(269)
§ 13.2.2.6 使用 PATH 的诀窍	(270)
§ 13.2.3 PROMPT	(271)
§ 13.2.3.1 用文件保存 PROMPT 的定义	(271)
§ 13.2.3.2 重定义 PROMPT 的一例	(271)
§ 13.2.4 DIRCMD	(272)
§ 13.3 其它要用到 DOS 环境的程序	(272)
§ 13.3.1 批文件与环境	(272)
§ 13.3.1.1 SET 命令	(272)
§ 13.3.1.2 由批文件得到环境的定义	(273)

§ 13.4	应用和实用程序与环境	(274)
§ 13.5	DOS 的主环境与子环境	(276)
§ 13.5.1	拥有和没有自己的环境的程序	(277)
§ 13.5.2	每一个环境块的大小	(277)
§ 13.5.3	避免子环境浪费过多的内存	(278)
§ 13.5.4	如何为子环境块保留空间	(278)
§ 13.5.5	释放自己的环境的程序	(279)
§ 13.5.6	环境定义的有效范围	(279)
§ 13.6	与环境有关的 DOS 命令	(280)
§ 13.7	与环境有关的 DOS Power Tools 程序	(280)

第十四章 批文件和 DOSKEY 宏命令

§ 14.1	批文件的编程	(281)
§ 14.1.1	简单的批文件	(281)
§ 14.1.1.1	建立批文件	(281)
§ 14.1.1.2	批文件的基础	(281)
§ 14.1.1.3	一些有用的单行批文件	(282)
§ 14.1.2	比较复杂的批文件	(284)
§ 14.1.2.1	控制命令行是否回显	(284)
§ 14.1.2.2	暂停	(284)
§ 14.1.2.3	批处理的 ANSI ESCAPE 命令	(285)
§ 14.1.2.4	一个简单的菜单系统	(286)
§ 14.1.2.5	菜单系统实现的另一种方法	(287)
§ 14.1.2.6	一个较好的菜单体系	(288)
§ 14.1.2.7	最后一点补充	(292)
§ 14.1.3	循环、分层和其它复杂结构	(294)
§ 14.1.3.1	可替换参数 %0 到 %9	(294)
§ 14.1.3.2	由返回结果选择操作	(295)
§ 14.1.3.3	调用其它批文件的批文件	(296)
§ 14.1.3.4	用 SHIFT 来控制很多的命令行参数	(297)
§ 14.1.3.5	批文件嵌套——CALL 与 COMMAND /C	(298)
§ 14.1.4	FOR 命令	(299)
§ 14.1.5	把你的批文件保存在哪里	(301)
§ 14.1.5.1	批文件编译器	(302)
§ 14.1.6	DOSKEY 宏命令	(302)
§ 14.1.6.1	DOSKEY 命令的语法	(302)
§ 14.1.6.2	DOSKEY 宏命令与批文件的相同之处	(303)
§ 14.1.6.3	DOSKEY 宏命令与批文件的不同之处	(303)
§ 14.1.6.4	创建 DOSKEY 宏命令	(303)

§ 14.1.6.5	DOSKEY 宏命令的存贮	(304)
§ 14.1.6.6	重新装入宏命令	(304)
§ 14.1.6.7	重定义 DOS 命令	(304)
§ 14.1.7	其它的命令处理程序	(305)
§ 14.2	DOS 批文件命令	(305)
§ 14.3	DOS Power Tools 的批文件程序	(306)

第十五章 DEBUG 的使用

§ 15.1	运行 DEBUG	(307)
§ 15.1.1	DEBUG 命令	(307)
§ 15.2	用 DEBUG 建立和运行程序	(308)
§ 15.2.1	建立 BEEP 程序	(308)
§ 15.2.2	用 DEBUG 运行 BEEP	(310)
§ 15.2.3	BEEP 的分步执行	(310)
§ 15.2.4	将 BEEP.COM 存盘	(312)
§ 15.3	使用 DEBUG 和描述文件	(313)
§ 15.4	禁用 Print Screen 的描述文件	(314)
§ 15.4.1	利用 E 命令禁用 Print Screen	(314)
§ 15.4.2	建立 NOPRTSCR 程序	(315)
§ 15.4.3	增强 NOPRTSCR 的功能	(315)
§ 15.5	测试任务交换程序	(317)
§ 15.6	用 DEBUG 运行 ROM 程序	(318)
§ 15.7	利用 DEBUG 修改磁盘上的程序	(318)
§ 15.7.1	将 EDIT 装入 DEBUG	(319)
§ 15.8	DEBUG 命令	(320)

第十六章 通信

§ 16.1	串行口和并行口	(322)
§ 16.2	接插件与电缆	(324)
§ 16.2.1	直接与跨接(交叉)电缆	(325)
§ 16.3	数据传输速率	(327)
§ 16.4	两台计算机间的直接连接	(327)
§ 16.5	调制解调器连接	(328)
§ 16.5.1	使用调制解调器的原因	(328)
§ 16.5.2	调制解调器命令	(329)
§ 16.5.3	内部与外部调制解调器	(329)
§ 16.5.4	端口地址和中断	(329)
§ 16.5.5	数据调制解调器与传真调制解调器	(330)

§ 16.5.6 串行数据通信应用程序.....	(331)
§ 16.6 应用 DOS 命令进行简单的远程控制连接	(331)
§ 16.7 局域网	(331)
§ 16.8 避免存取冲突的共享	(332)
§ 16.9 计算机间通信的其他方法	(333)
§ 16.9.1 Interlnk	(334)
§ 16.9.1.1 Interlnk 所用电缆	(334)
§ 16.9.1.2 装载 Interlnk	(336)
§ 16.9.1.3 运行 Interlnk	(337)
§ 16.10 与通信有关的 DOS 命令	(338)
§ 16.11 与通信有关的 DOS Power Tools	(338)

第十七章 注意与提示

§ 17.1 共享(SHARE)	(339)
§ 17.1.1 需要 SHARE 的情况	(339)
§ 17.1.2 装载 SHARE	(339)
§ 17.2 使用 MSBACKUP	(340)
§ 17.2.1 安装 MSBACKUP	(340)
§ 17.2.2 备份设备.....	(340)
§ 17.2.3 备份循环/周期	(341)
§ 17.2.4 选择备份文件.....	(241)
§ 17.2.5 备份选项.....	(342)
§ 17.2.6 备份记录.....	(342)
§ 17.2.7 比较和恢复文件.....	(342)
§ 17.3 麻烦的避免	(343)
§ 17.3.1 从来不用的程序.....	(343)
§ 17.3.2 不能(或不应该)裁载于高址的程序.....	(344)
§ 17.3.3 在 Windows 中不使用的程序	(345)
§ 17.3.3.1 APPEND	(345)
§ 17.3.3.2 JOIN 和 SUBST	(345)
§ 17.3.3.3 CHKDSK /F 和 DEFRAG	(346)
§ 17.3.3.4 磁盘和内存管理程序.....	(346)
§ 17.3.3.5 MLSEFUNC	(346)
§ 17.3.3.6 VSAFE	(346)
§ 17.3.4 关于磁盘压缩和磁盘高速缓存的一些说明.....	(346)
§ 17.4 从麻烦中退出	(348)
§ 17.4.1 UNDELETE	(348)
§ 17.4.1.1 UNDELETE 安全网的限制	(349)
§ 17.4.2 UNFORMAT	(349)

§ 17.5	计算机病毒	(350)
§ 17.5.1	什么是计算机病毒?	(350)
§ 17.5.2	它们如何地普通?	(351)
§ 17.5.3	PC 机如何受传染?	(351)
§ 17.5.4	一项简单的,但并不非常有用的预防措施	(351)
§ 17.5.5	检测并阻止计算机病毒的技术	(352)
§ 17.5.6	你应该采取哪种反病毒保护措施	(353)
§ 17.5.7	DOS 反病毒程序	(354)
§ 17.6	不要惊慌	(354)
§ 17.6.1	寻求帮助	(355)

第十八章 命令参考指南

§ 18.1	语法规则	(356)
§ 18.2	ANSI.SYS 命令	(359)

第十九章 DOS Power Tools 实用程序

§ 19.1	什么是共享软件	(449)
§ 19.2	程序汇总	(449)

第二部分 高级技术和告诫

第十一章 可变的配置

你在 PC 上执行的多样任务中的每一个,其最优配置可能是不同的。在这一章里你将学会怎样用几种不同的方法建立可变的配置。你还可学会当你进行某种尝试而不能达到目的时如何重新开始。

§ 11.1 在引导时的可选操作

有时你可能需要一种在引导时和 PC 交互的方法:例如,取消设备驱动程序的装入,或从几个可选路径中选择一个来跟踪 AUTOEXEC. BAT 文件。在以前的 DOS 版本中这不容易做到。幸运的是,DOS 6 包括了三个不同的工具,每一个都有助于实现这一过程。

§ 11.1.1 用 F8 键清除引导

DOS 6 的一个重要的改进之处是引入了清除引导(Clean Boot)概念。现在你可以在 IO. SYS 执行 CONFIG. SYS 的任何一行之前,和在 COMMAND. COM 运行 AUTOEXEC. BAT 文件之前,按你的想法指示 IO. SYS。

DOS 6 在你首次引导时显示新信息:Starting MS-DOS...在出现其它画面之前它将在屏幕上停留几秒钟。你一看见这一信息就可按 F8。此时将出现另一条信息:

MS-DOS will prompt you to confirm each CONFIG. SYS command
键入你关于如何处理 CONFIG. SYS 文件的选择。

如果你在 CONFIG. SYS 文件中指定了 DOS=HIGH 和/或 DOS=UMB,IO. SYS 将首先询问你是否要遵循这些指示。和下面所有的问题一样,必须按 Y 或 N 键来回答。所有其它的回答被忽略,包括 Ctrl-Break。

IO. SYS 将仔细查看你的 CONFIG. SYS 文件,每遇到一个程序行就对你提问。如果这一行指定了一个设备驱动程序的装入,且你按了 Y 键,你将看到驱动程序装入时的信息。如果有 INSTALL 语句或 SHELL 语句,对它们的处理将推迟到你的 CONFIG. SYS 文件中的所有其它行处理完之后。

接着开始 SHELL 语句,然后是一些 INSTALL 语句。最后,选择命令解释程序是否处理 AUTOEXEC. BAT 文件。此时你只能选择完全处理或完全不处理。(当然,此时你可以通过按 Ctrl-Break 退出批文件。在关键时刻也可以通过使用新的 CHOICE 命令建立用户的交互作用,下面我们将会讲到。)

这个进程对查找一个新近修改的 CONFIG. SYS 文件的故障很有帮助。在本章后面的“Diagnosing Bootup Problems(诊断引导问题)”部分中我们将讨论怎样使用它。F8 键的新用法只在 CONFIG. SYS 文件处理过程中(以及在处理 AUTOEXEC. BAT 文件的开始)应用。以后,你可以重新定

义或按其它方式使用它。

§ 11. 1. 2 CONFIG.SYS 中的用户可选项

如果你知道你的 CONFIG.SYS 文件由那些你每次都要执行的程序行组成,而其中只有一两行你有时要略过,DOS 6 就能使你建立 CONFIG.SYS。CONFIG.SYS 文件的所有程序行都是下面的格式

```
ACTION=DETAILS
```

这里的 ACTION 可能是 BREAK,BUFFERS,COUNTRY,DEVICE,DEVICEHIGH,DOS,DRIVEPARM,FILES,FCBS,INSTALL,LASTDRIVE,SHELL,STACKS,或 SWITCHES。(这些几乎是所有的有效行。只有一个 SET,在语法上有一点不同——它可用于批文件和 DOS 提示符下。你可能还会注意到有一些以 REM 开头的注释行,REM 后有一个空格,后面跟你所注释的文本。)

你可以在等号前加一个问号来指示 IO.SYS 提示你在它运行任一 ACTION=DETAILS 行前回答。不幸的是,使用这一功能也就意味着放弃了很多用户从不知道的 CONFIG.SYS 文件的另一功能——改善文件“格式”的能力。

你可以通过插入“白空间”来增强 CONFIG.SYS 文件的可读性。正常情况下,IO.SYS 忽略你加在等号前后的空格。这样,你可以让你的 CONFIG.SYS 文件象这样:

```
FILES=80
BUFFERS=30
BREAK=ON
STACKS=9,256
LASTDRIVE=Z
DOS=HIGH,UMB
DEVICE=C:\DOS\HIMEM.SYS
DEVICE=C:\dos\EMM386.EXE RAM 3072 FRAME=D400 I=E400-F7FF
X=A000-D1FF
DEVICEHIGH=C:\setup\STACKER.COM/NB/EMS @@@@
DEVICEHIGH=C:\setup\drivers\ASPI2DOS.SYS/Z/Y-/P140
DEVICEHIGH=C:\setup\drivers\CUNI_ASP.SYS/ID:4/N:1/D:MSCD001
DEVICEHIGH=C:\uv\ANSI-UV.SYS
DEVICE=C:\setup\drivers\MVSOUND.SYS D:3 Q:10 T:1 V:65
DEVICE=C:\wfwg\PROTMAN.DOS/I:C:\wfwg
DEVICE=C:\wfwg\WORKGRP.SYS
DEVICE=C:\wfwg\EXP16.DOS
INSTALL=C:\dos\SHARE.EXE/f:4096
INSTALL=C:\mouse\ballpnt\MOUSE.COM
SHELL=C:\dos\COMMAND.COM C:\dos /P /e:624
```

或者象这样:

```
FILES    =   80
BUFFERS  =   30
BREAK    =   ON
```

```

STACKS = 9,256
LASTDRIVE = Z
DOS = HIGH,UMB
DEVICE = C:\dos\HIMEM.SYS
DEVICE = C:\dos\EMM386.EXE RAM 3072 FRAME=D400
        I = E400-F7FF
        X = A000-D1FF
DEVICEHIGH = C:\setup\STACKER.COM/NB/EMS@@@
DEVICEHIGH = C:\setup\drivers\ASPI2DOS.SYS/Z/Y-/P140
DEVICEHIGH = C:\setup\drivers\CUNT-ASP.SYS/ID:4 /N:1 /D:MSCD0001
DEVICEHIGH = C:\UV\ANSI-UV.SYS
DEVICE = C:\setup\drivers\MVSOUND.SYS D:3 Q:10 T:1 V:65
DEVICE = C:\wfwg\PROTMAN.DOS/I:C:\wfwg
DEVICE = C:\wfwg\WORKGRP.SYS
DEVICE = C:\wfwg\EXP16.DOS
INSTALL = C:\dos\SHARE.EXE /f:4096
INSTALL = C:\mouse\ballpnt\MOUSE.COM
SHELL = C:\dos\COMMAND.COM C:\dos/P /e:624

```

如果你想直观地突出一组动作,你也可以加一些空行或缩进排一些行。实际上,CONFIG.SYS 文件是一个程序,使列出的程序格式化以便更容易读的这一想法极普通,是很好的编程练习。

但是,MS-DOS 6 使你能查看 CONFIG.SYS 文件,而且这一新功能还允许你能在引导时使某些行成为可选行。按规定,CONFIG.SYS 文件的程序行可通过在等号前加一个问号而成为可选的,但是在问号和 ACTION 名字末尾之间一定不能有空格。如果你对 CONFIG.SYS 文件的一些而非全部行进行了这些操作,则当你下一次引导 PC 时,会问你是否执行带问号的每一行,而不是其它行。这时你不必在指定时刻(显示 Starting MS-DOS...信息时)先按任何指定键(象 F8 键)。

如果你在单词 DEVICEHIGH 和等号之间使用了 DEVICEHIGH 语句的可选参数,你不能使用问号。不过除了这个限制,你可以用一个程序行提醒用户在引导时间内它是否被执行。(注意你可以在等号后面加空格,但不能在该行的等号前面加空行。)

SET 命令是另一个特殊情况。这个命令的语法和 CONFIG.SYS 文件中其它有效行有一点不同。它是:

```
SET name=definition
```

这里的 name 将被看成全是大写字母,但定义并非如此。这个命令在两个方面很特殊,都要处理空格。和其它所有的 CONFIG.SYS 命令不同,SET 命令不会忽略等号两边的空格。程序行

```
SET ADAM=EVE
```

和

```
SET ADAM = eve
```

被看成是完全不同的。五字符名字“ADAM ”和“ EVE”相等,而四字符名字“ADAM”和“eve”相等——等号右边的空格和字符都很重要。

你不能在 SET 命令后跟 ? = 的理由和不能让带可选参数的 DEVICEHIGH 运作的理由一样。在这两种情况下 ACTION 后都必须有一个空格,而特殊的 ? = 操作不能处理那个空格。

正常情况下,当你在 CONFIG.SYS 文件中增加了一些新程序行而不能确保句法是否正确时,你可以使用? =结构。如果你回答 Y 且一切正常,那么你就可能做对了。取消问号(并需要加一些白空间),该程序行就成为 CONFIG.SYS 文件中永久的一部分了。

如果在执行该行情况下引导 PC 失败,那么重新引导并在被提问时回答 N。这时不使用该行,而使用所有其它行引导 PC。现在你可以编辑那一行,或做你以前能做的工作,没有必要使用专门的 DOS 引导软盘来避免 PC 系统事故,因此也没有必要让引导盘装载那些访问网络才使用的或你过去常做的一切。

§ 11.2 多重 PC 特性

你不必使用 DOS 6 中的这些新功能来赋予一台 PC 多重特性。在这一部分我们将介绍几种普及的高级用户方案。

§ 11.2.1 多套匹配的启动文件

改变 PC 特性的最简单的方法是改变它的启动文件 CONFIG.SYS 和 AUTOEXEC.BAT。最简易的途径是制作若干替换引导软盘,每一张都有特有的一对匹配 CONFIG.SYS 和 AUTOEXEC.BAT 文件。不过一个更好的方法是对每个文件建立若干替换格式,并把它们用好记的名字存储到某个子目录,也许是 C:\CONFIGUR。你可以把文件命名为 CF-BARE 和 AE-BARE,它们分别是 CONFIG.SYS 和 AUTOEXEC.BAT 文件的缩写。CF-WIN 和 AE-WIN 可能是你计划运行 Windows 时的一对文件。CF-NET 和 AE-NET 可能请求联入网络,而不运行 Windows。CF-MAX 和 AE-MAX 可能请求联网、运行 Windows,也支持 CD-ROM 驱动器和声音卡。

下一步是建立几个批文件,把它们叫做 SET-BARE.BAT, SET-WIN.BAT, SET-NET.BAT 和 SET-MAX.BAT。它们看起来很相似。例如,这里是第一个文件:

```
C:
CD\
COPY CONFIGUR\CF-BARE CONFIG.SYS
COPY CONFIGUR\AE-BARE AUTOEXEC.BAT
CFG Warmboot
```

第一行确定 C 驱动器为 DOS 的缺省驱动器。下一行进入根目录。接着两行把启动文件的指定匹配对拷贝到根目录下,并给它们正确的名字。最后一行调用 DOS Power Tool 程序的一个命令行,实质上就是热启动(按 Ctrl-Alt-Del)。

这个方案的一个变化是建立一个批文件来调用一个存储器干扰程序,象 GEM 接口的 ventura Publisher,但是实际上并不是指示批文件调用该程序,而是保存当前启动文件,载入新设置,并重新引导。这套新的启动文件将在下面的环境中运行极需内存的应用程序,该环境释放了所有的设备驱动程序, TSR 程序,和其它在其间分配的存储器用户。一旦退出该应用程序, AUTOEXEC.BAT 文件将恢复最初那套启动文件,并重新引导 PC。图 11-1 所示的就是这样一组文件。

§ 11.2.2 第三方 DOS 引导管理程序

多年以来就有许多第三方“DOS 引导管理程序”,这些程序是均分和共享的商业软件通用的。其中一些和上面讲述的自动执行过程没有什么不同。另一些指示你建立一对 CONFIG.SYS 和

AUTOEXEC. BAT 文件,然后选择在引导时将使用哪个子设备。

VP. BAT	
@echo off	
echo.	
echo	VP. BAT -- Setup and run Ventura Publisher
echo.	
echo	First we will save the old CONFIG. SYS and AUTOEXEC. BAT files,
echo	then copy new ones from C:\CONFIGUR directory and reboot.
echo.	
C:	
cd\	
if exist cf. old ECHO * * * Warning: CF. OLD is about to be deleted. * * * *	
if exist cf. old PAUSE	
if exist cf. old DEL cf. old	
if exist cf. sav REN cf. sav cf. old	
REN config. sys cf. sav	
COPY C:\configur\cf--vp config. sys	
if exist ae. old ECHO * * * Warning: AE. OLD is about to be deleted. * * * *	
if exist ae. old PAUSE	
if exist ae. old DEL ae. old	
if exist ae. sav REN ae. sav ae. old	
REN autoexec. bat ae. sav	
COPY c:\configur\ae--vp autoexec. bat	
reboot	
CF--VP	
FILES	= 20
BUFFERS	= 16
BREAK	= ON
AE--VP	
@echo off	
echo	AE--VP -- Special AUTOEXEC. BAT file for Ventura Publisher
echo.	
echo	Invoking programs
echo.	
PATH C:\DOS;C:\UT\MAIN;C:\SETUP	
PROMPT \$P \$G	
c:\mouse\MOUSE	
D:	
CD \VENTURA	
DRV MRGR VP %1 /S=SDFAST86. VGA /M=32 /X=D; /A=10	
echo.	
c:\ut\nor\beep/F440 /D2 /R3	
echo	You have 5 seconds to decide:
c:\ut\main\TIMER--VP L 5	
if errorlevel 50 goto stayhere	
if errorlevel 25 goto goback	
if errorlevel 10 goto goback	
:goback	
c:\setup\GOBACK	
:stayhere	

图 11-1 可变的 CONFIG. SYS 和 AUTOEXEC. BAT

DOS Power Tools 的程序 CFG 则有点不同。它允许你在 CONFIG. SYS 和 AUTOEXEC. BAT 文件中加入各种功能,其中一些把复制功能加到 DOS 6;另一些则不能。(见十八章关于 CFG 程序的介绍。)

§ 11. 2. 3 第三方管理程序的 DOS 6 方法

在大多数的第三类多重配置产品中,基本策略和前面部分介绍的批文件设置相似。你选择一个

配置,放置适当的 CONFIG.SYS 和 AUTOEXEC.BAT 命令,然后重新引导系统。

它们中有一些使用了不同的方法。其中的特殊之处值得注意。Hans Salvisberg 的 BOOT.SYS 通过一个 DEVICE 行控制你的 PC。它接着处理你在装入 BOOT.SYS 的 DEVICE 行后写入 CONFIG.SYS 文件的一组菜单。这些菜单出现在屏幕上,而你的输入将被记录。最后,BOOT.SYS 完成菜单处理,并让 IO.SYS 重新处理 CONFIG.SYS 文件,包括它已编辑来适合菜单选择的那一部分。

BOOT.SYS 最象新的 DOS 6 的 CONFIG.SYS 菜单方法,但在一点上有所不同。用 DOS 5 启动的 DOS 命令被包括在 CONFIG.SYS 文件中。(程序行 DOS=HIGH,UMB 是这个命令的一个典型例子。)这一行在所有的 DEVICE 行前由 IO.SYS 处理,这意味着 BOOT.SYS 不能控制该行。

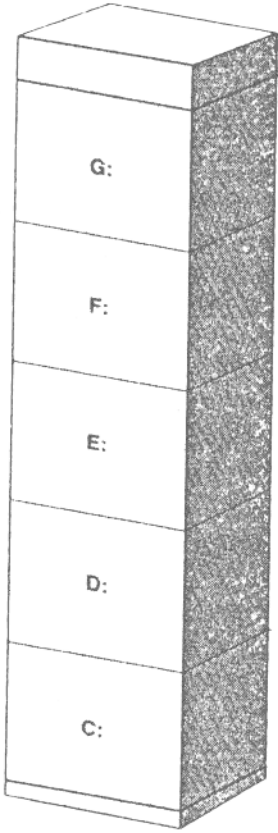


图 11—2 OS/2 引导管理程序提供的用于 OS/2 和 DOS 的硬盘

来自 V Communications 的 Boot commander 试图处理 DOS 5 和后面读到的情况,并按照 CONFIG.SYS 文件之前的 DOS 行执行。这可能是在 DOS 6 以前开发多重配置的最好和最安全的办法(如果你想不同于前面介绍的匹配批文件方法)。

如果你的计算机除了有 DOS 还安装了 OS/2,你可以安装双重引导或引导管理程序机制用来在两种操作系统间切换。

OS/2 双重引导精密模仿匹配启动文件的方法来改变 DOS 配置。不同的是 OS/2 双重引导方法不仅替换启动文件,也替换隐藏的系统文件和引导记录。每次引导时你将进入你上次使用的操作系统。DOS 批文件和 OS/2 CMD 文件使你从一个操作系统进入另一个操作系统,然后再返回。

OS/2 引导管理程序方法则完全不同。要使用这个策略你必须重新划分磁盘并指定 1MB 的分

区给引导管理程序以专门使用 OS/2 FDISK 程序。

OS/2 引导管理程序的分区是有效的,且它保持一个其它分区“可启动”的记录。至于涉及到 FDISK 时,那些其它分区无效;这一点 OS/2 引导管理程序了解得更清楚。当你用你安装的 OS/2 引导管理程序引导计算机时,你可以得到一个菜单,它类似于 DOS 6 中可得到的 CONFIG.SYS 菜单。在一段限制时间之后(如果你定义了的话),它就默认地从你的一个可启动分区引导;或者你可以选择一个分区,从那里引导。

OS/2 引导管理程序也可以在任一逻辑卷内装入 OS/2 操作系统文件。和 DOS 限定在 A 或 C 驱动器不同,OS/2 可被安装在任何你需要的地方,而且一旦被安装在那里,它将从那个位置引导。图 11—2 显示了一个系统的逻辑卷,既有 OS/2 也有 DOS 和 OS/2 的引导管理程序。

§ 11.3 更多的多重特性

引导管理程序并不是发展 PC 多重特性的唯一方法。在 DOS 6 之前,AUTOEXEC.BAT 文件中的菜单使用得很广泛。它们的范围从简单的起始聚集调制到象 CONED 一样的功能强大的共享程序。DOS 6 提供的可选项很容易在 CONFIG.SYS 文件中建立菜单而不用使用 DOS 中没有的项。

§ 14.4 AUTOEXEC.BAT 中的菜单

很多高级用户在他们或朋友的机器上的 AUTOEXEC.BAT 文件中建立了批处理文件菜单。最简单的就是只把一个简单的文本文件和几个小的批处理文件结合起来。要想知道这有多么简单,可以看下面的例子:假定你建立了一个包含下面这些行的名为 MENU.TXT 的文本文件:

MY MENU

Enter your choice from among the following:

1. Run BASIC interpreter
2. Run checkbook program
3. Run word processor

同时你还建立了三个批处理文件 1.BAT, 2.BAT 和 3.BAT。这些批文件中的每一个都做菜单中指定的一件事,且每个文件都以返回 MENU 目录,清屏幕并重新显示 MENU.TXT 文件结束。下面是 1.BAT 可能的内容:

```
@ECHO OFF
REM      This is 1. BAT
D:
CD\BASIC
QBASIC
C:
CD\MENU
CLS
TYPE MENU.TXT
```

它假定你的 D 驱动器中有一个叫 BASIC 的目录,且你想从那个目录运行 QBASIC。另两个批处理文件是类似的,不同之处只是它们调用各自相应的应用程序。

要启动整个过程,PC 中的 AUTOEXEC. BAT 文件应该以下面这几行结束:

```
C:
CD\MENU
CLS
TYPE MENU. TXT
```

这种菜单方法不是很复杂。它毕竟没有提供什么可靠性,因为用户实际上只在他或她选择菜单项时在 DOS 提示符下键入选择,这和键入其它 DOS 命令一样容易。不过,在很多情况下这已足够好了。

§ 11. 4. 1 CONFIG. SYS 中的菜单

DOS 6 中真正被看作可变配置的是它对菜单命令的支持。现在我们可以让 CONFIG. SYS 文件有驱动菜单,工作颜色,并能控制你每次引导时 PC 配置的建立。

DOS 6 菜单方法的唯一限制(不能由第三类引导管理程序实现的)就是 IO. SYS 在它开始处理 CONFIG. SYS 文件之前装入 DBLSPACE. BIN。因此,你可以选择在某些配置下装入 DBLSPACE. BIN,在另一些配置下则不装入。这是老式的基于软盘的方法运作的一种情况(如果你使用 SYS 增加系统文件,你可能得从引导软盘删除 DELSPACE. BIN),但是当你在没有激活 DBLSPACE 情况下工作时,一定要注意不要破坏或删除你的 CVF。

§ 11. 4. 1. 1 CONFIG. SYS 菜单的设置

在 CONFIG. SYS 文件中设置菜单是很容易的。这种要求的结构和大多数 Microsoft 公司的应用程序的 INI 文件或 Windows 下的 WIN. INI 文件很相似。这种语法是程序块的一个简单重复,每一块前面都有用方括号括起的名字:

```
[block_name]
items for block
...
[next_block]
...
```

第一层或顶层菜单总是叫做[MENU],它必须被方括号括起,且必须是程序行的第一项。(你可以键入[menu],[Menu],等等;它不区分大小写。)另一个被保留的菜单名是[COMMON],它将在后面讨论。

在每个程序块名之间,可以有一组 CONFIG. SYS 命令和参量(一个配置块)或更多的菜单命令(一个菜单块),但每一块必须是配置块或菜单块,且顶层[menu]块必须是菜单块。注意在一个块中你不能把菜单命令和其它的 CONFIG. SYS 命令或参量混合起来。

只有五个菜单命令:MENUITEM, SUBMENU, MENUDEFAULT, MENUCOLOR 和 INCLUDE(它们都不区分大小写),且它们只在一个块中有效。因为一个块是从一个块名项到下一项之间扩充,包括托管的第一个块名项——[menu]——足够使这些命令合法。

注意:Microsoft 公司说明 NUMLOCK 是另一种菜单命令,它们可能指定你只能在一个配置块中使用它,但是实际上在没有块标号的 CONFIG. SYS 文件它也运行得很好,这里没有列出菜单命令。

这些菜单命令行的语法是：

MENUIITEM=configuration__block__name[,Text]

SUBMENU=menu__block__name[,Text]

MENUDEFAULT=block__name[,time]

MENUCOLOR=forecolor,backcolor

INCLUDE=configuration__block__name

这里的 configuration__block__name 和 menu__block__name 是你想转向的其它块的名字；block__name 是在菜单的 menuitem 或 submenu 中提到的一个块的名字。text 项可能是你希望的文本，不超过 70 个字符。（实际上，它可能更长，但只显示前面 70 个字符。）Time 是在选择 block__name 项后到继续 CONFIG.SYS 文件之间的若干秒间隔时间。Forecolor 和 backcolor 是代表你所希望的菜单当前颜色方案的数字。这个颜色安排将应用到用你安装的设备驱动程序，或用后面块中的另一个 menucolor 命令，或在 DOS 提示符下的清屏命令(CLS)重新启动为止。

在 QBASIC 颜色命令中使用下面约定的有效颜色代号(不同于 ANSI 协定)。Forecolor 值可能从 0 到 15；backcolor 值只从 0 到 7。

0	black	黑	8	dark grey	暗灰
1	blue	蓝	9	bright blue	亮蓝
2	green	绿	10	bright green	鲜绿
3	cyan	青蓝	11	bright cyan	亮青蓝
4	red	红	12	bright red	鲜红
5	magenta	洋红	13	bright magenta	亮洋红
6	brown	褐色	14	bright yellow	鲜黄
7	dim white	灰白	15	bright white	亮白

MENUIITEM,SUBMENU,MENUCOLOR 和 MENUDEFAULT 命令只在菜单块中有效。所有的 MENUIITEM 行必须指向一个配置块；所有的 SUBMENU 行必须指向一个菜单块。

INCLUDE 行可能只放在配置块中。它们使 IO.SYS 把块中所有等号后跟有名字的语句拷贝到当前块中。

下面这个最小化的菜单实际上很有用。

[menu]

menuitem=onward,Welcome to CONFIG.SYS menus!

menucolor=10,0

menudefault=onward,1

[onward]

...

它只给你一个选择：向前运行；每隔一秒它将自己向前运行。它似乎不太有用。这个菜单的实用性在于它让你显示一个文本行(本例中是 Welcome to CONFIG.SYS menus!)，并按后面的设置设颜色。你可能认为你能把时间值设为 0，这样菜单颜色不必受屏幕上的影响。其实，如果这样这个块中的行一个也不能执行(除了超时，会立即实现)，不管该块中行的顺序怎样。

图 11—3 显示了一个更复杂的 CONFIG.SYS 菜单，它有一个顶层菜单，其中有三个 MENUITEM 和一个 SUBMENU。子菜单中有两个菜单项。一共有九个块。图示的最后一块 [COMMON]有一个特殊的任务。组成这一块命令不管选择哪个菜单都要执行。它被放在 CON-

FIG. SYS 文件末尾,也是一个很有用的块,我们将作一个简短的介绍。

```
rem Here you can put some reminder to yourself of what version menu this is.
rem =====

[MENU]
MENUITEM                =CLEAN,Real minimalist configuration
SUBMENU                 =MAIN,Usual configuration alternatives
MENUITEM                =BERNOULI,Load Bernoulli support
MENUITEM                =TEST,Test modifications to system
MENUCOLOR               =14,1
MENUDEFAULT             =MAIN,60

[CLEAN]
rem      Here is the truly minimalist configuration!
STACKS                =0,0

[MAIN]
rem      offer a choice between loading the network drivers
rem      or simply using the system wethout network access
MEMUCOLOR             =10,0
MENUITEM              =USUAL,Mormal setup with network drivers
MENUITEM              =OFTEN,No Network connection
MENUDEFAULT           =USUAL,60

[BERNOULI]
INCLUDE USUAL
DEVICE                =c:\setup\drivers\rnd.sys

[TEST]
rem      Put here anything you want to test,but don't want
rem      to use other than when you are doing those tests.
INCLUDE USUAL

[USUAL]
DOS                   =umb
FILES                 = 80
FCBS                  =16,0
BUFFERS               =30,0
LASTDRIVE             = Z
DEVICE                =c:\dos\HIMEM.SYS
DEVICE                =c:\dos\EMM386.EXE noems highscan
DEVICEHIGH            =C:\DOS\DBLSPACE.SYS /MOVE
DEVICE                =C:\DOS\RAMDRIVE.SYS 1024 512 128/e
INCLUDE OFTEN
INCLUDE NetStuff

[OFTEN]
DOS                   = high
DEVICEHIGH/L:1,12048 =c:\dos\SETVER.EXE
BREAK                 = on
STACKS                =9,256
INSTALL              =c:\dos\SHARE.EXE /f:4096
DEVICE               =c:\dos\SMARTDRV.EXE/double_buffer

[NetStuff]
DEVICE                =c:\workgrp6\PROTMAN.EOS /i;c:\workgrp6
DEVICEHIGH /L:1,7280  c:\workgrp6\WORKGRP.SYS
DEVICEHIGH /L:1,11168 c:\workgrp6\EXP16.DOS

[COMMON]
SHELL                 c:\dos\COMMAND.COM c:\dos\ /e:624 /p

rem At htis point the possibilities for the CONFIG environment variable are:
rem      CLEAN,USUAL,OFTEN,BERNOULI,and TEST
```

图 11—3 一个较复杂的 CONFIG. SYS 菜单