

微型计算机 使用技巧汇编

weixinjisuanjishiyongjiqiaohuibian
weixinjisuanjishiyongjiqiaohuibian
weixinjisuanjishiyongjiqiaohuibian
weixinjisuanjishiyongjiqiaohuibian

微型计算机使用技巧汇编

岱凌云 主编

科学技术文献出版社

(京)新登字 130 号

内 容 提 要

本书汇编了微型计算机的使用技巧及实用程序。内容包括：操作系统的使用经验，BASIC、PASCAL、COBOL、C 等语言的编程技巧和应用实例，dBASE Ⅲ、FoxBASE 等数据库的应用实例，打印技巧和制表方法及实用程序，文件的删除、恢复、查找、保护、排序等方法，图形处理方法与技巧。

本书适用于广大的微机用户。

微型计算机使用技巧汇编

岱凌云 主编

科学技术文献出版社出版

(北京复兴路 15 号 邮政编码 100038)

世界知识印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

787×1092 毫米 16 开本 26 印张 600 千字

1993 年 2 月第 1 版 1993 年 2 月第 1 次印刷

印数：1—7100 册

ISBN 7-5023-1766-X/TP·96

定价：15.60 元

前 言

目前，我国的微型计算机装机量与日俱增，应用领域也日趋广泛。但低水平的重复开发现象十分普遍，耗费了大量的人力物力。

为此，笔者将微机使用技巧、编程技巧及实用程序汇编成册，以便读者学习、查阅和选用，达到节约开发时间，提高微机使用效率的目的。

在本书的汇编工作中，力求内容的先进性及实用性。由于编者水平有限，其中难免有错误，恳请广大读者指正。

编者 1992. 3

目 录

第一章 DOS、MS-DOS、CCDOS2. 13、 XENIX 操作系统使用经验

- | | |
|---|--|
| 1. 1 动态选择安装 DOS 设备驱动程序的方法..... (2) | 1. 17 修正 2. 13D 汉字操作系统驱动程序不能调整行距一例..... (22) |
| 1. 2 检查 DOS 内存的程序..... (5) | 1. 18 将 CCDOS4. 0 的造字程序用于 CCBIOS2. 13E 系统造字..... (23) |
| 1. 3 DOS 各版本的比较和选用技巧... (8) | 1. 19 用五笔字型的压缩字库在没有硬盘的 PC 机上使用 CCDOS 2. 13H..... (23) |
| 1. 4 实现高随机度随机序列的一种简单方法..... (10) | 1. 20 在硬盘上安装 2. 13H 某种字体高点阵字库的方法..... (24) |
| 1. 5 执行 RESTORE 时出现的问题及解决办法..... (11) | 1. 21 CCDOS2. 13H 使用 CEGA 卡上的汉字字模的程序..... (24) |
| 1. 6 几种转向操作符的使用实例..... (11) | 1. 22 2. 13H 系统 CH25. COM 文件的修正..... (26) |
| 1. 7 灵活运用 COPY 命令扩大其使用范围..... (12) | 1. 23 修改完善 2. 13H 系统的 LP 功能的方法..... (26) |
| 1. 8 MS-DOS3. 0 以上系统的使用技巧..... (13) | 1. 24 修改 2. 13H 提示信息的方法... (27) |
| 1. 9 灵活、方便、安全地使用硬盘的方法..... (14) | 1. 25 在 2. 13 中实现五笔字型多种功能的方法..... (28) |
| 1. 10 三种 CTRL-C 中断的屏蔽方法和一个实用屏蔽程序..... (14) | 1. 26 忘记了 root 口令后的一种操作简单、运行方便的处理方法..... (28) |
| 1. 11 使 SVR3. 2. 0 支持 VGA 的方法和技巧..... (15) | 1. 27 用 SHELL 语言编写的在 XENIX 下实现类似 DOS 中的 F3 功能的程序..... (29) |
| 1. 12 CCDOS2. 13H 内外部词组共享方法..... (17) | 1. 28 XENIX 系统 TAR 命令中存在的问题及解决方法..... (30) |
| 1. 13 CCDOS2. 13A 与 GWDOS3. 20 的硬盘共存法..... (19) | 1. 29 在 XENIX 中一次拷贝多个 DOS 文件的程序..... (31) |
| 1. 14 CCDOS2. 13A 常用字库的使用经验..... (20) | 1. 30 XENIX 系统的一种维护方法..... (32) |
| 1. 15 对 CCDOS2. 13A 系统显示文件死机的处理方法..... (20) | 1. 31 续谈 XENIX 系统的维护..... (33) |
| 1. 16 CCDOS2. 13 下显示 TURBO PASCAL 图形单元的简单易行方法..... (21) | 1. 32 恢复 XENIX 引导程序的一种方 |

法	(33)	序工作原理	(39)
1. 33 在 XENIX 系统中正确设置终端类型的几种方法	(34)	1. 38 XENIX SYSTEM V 中 haltsys 命令的改进方法	(40)
1. 34 关闭 XENIX 系统的一种简捷方法	(35)	1. 39 XENIX 系统与 DOS 系统的交叉开发方法	(40)
1. 35 XENIX 中终端并行打印机的安装方法	(36)	1. 40 增强 XENIX 对 DOS 文件的操作功能的一种实施方法	(42)
1. 36 设置 XENIX 超级用户的后备口令之法	(38)	1. 41 在 XENIX 环境下直接执行 DOS 应用程序	(45)
1. 37 XENIX SYSTEM V 的 DOS 程			

第二章 文件的删除、恢复、修改、查找、保护、排序等方法及实用程序

2. 1 快速删除空文件的方法	(48)	程序	(68)
2. 2 UNIX 系统中的 rco 文件被删后的解决办法	(49)	2. 16 一个自动得到命令文件间调用关系图的程序	(69)
2. 3 从 XENIX 的盘块管理看被删文件恢复的几个问题	(49)	2. 17 利用系统调用增强批文件的灵活性	(71)
2. 4 两种备份文件恢复中的一些体会	(51)	2. 18 PC/XT 兼容机的虚拟盘设备驱动程序	(72)
2. 5 软盘文件系统的抢救方法	(54)	2. 19 磁盘文件管理的若干经验	(77)
2. 6 EXE 文件的文件头分析和使用	(55)	2. 20 解决不同 DOS 版本间备份文件安装的新途径	(78)
2. 7 修改 EXE 文件的一种简便方法	(57)	2. 21 自由选择运行环境的 AutoEXEC. BAT 文件	(78)
2. 8 修改 EXE 文件中说明部分的新方法	(58)	2. 22 不同机型间二进制数据文件的共享法	(80)
2. 9 EXE 文件与 COM 文件的比较及转换	(60)	2. 23 用 INKEY (X) 函数功能控制输入密码时间	(80)
2. 10 只需一行命令即可实现从众多的文本文件中查到所需文件	(62)	2. 24 XENIX 系统下的文件加密法	(81)
2. 11 识别关键字的简单方法	(63)	2. 25 独享硬盘的一种方法	(82)
2. 12 用 PCTOOLS 查找忘记文件名的文件	(64)	2. 26 反跟踪加密一例	(85)
2. 13 通用的文件属性置取程序	(64)	2. 27 主硬盘自举记录的口令设置	(87)
2. 14 一个简单实用的供用户动态选择安装 CONFIG 文件的程序	(66)	2. 28 也谈主硬盘口令设置	(88)
2. 15 命令文件的检查与自动缩格转化		2. 29 建立自己的硬盘工作区的原理和方法	(91)

- 2. 30 一种方便可靠的 3.5 英寸磁盘加密法 (92)
- 2. 31 带标号的 BAT 文件的加密 (93)
- 2. 32 子目录的隐藏及定位方法 (94)
- 2. 33 隐含文件的巧用 (95)
- 2. 34 隐藏子目录一法 (96)
- 2. 35 简便实用的子目录隐藏方法 (97)
- 2. 36 能使整个盘文件隐含的一种简单方法 (97)
- 2. 37 dBASE Ⅲ 数据文件加密方法和源程序 (98)
- 2. 38 纯西文键盘输入文件的压缩加密程序 (101)
- 2. 39 在 UNIX 系统上的加密文件程序 (102)
- 2. 40 对抗 PCtools 的小技巧 (104)
- 2. 41 一种反拷贝的方法 (104)
- 2. 42 防拷贝的一种方法 (105)
- 2. 43 文件加密的几种方法和比较 ... (107)
- 2. 44 逻辑 D 盘的加锁方法及程序 ... (108)
- 2. 45 DRDOS6.0 版为用户提供了较完善的系统安全措施 (111)
- 2. 46 实现文件双重保护的自动化方法 (112)
- 2. 47 高实用性的文件保护程序 (114)
- 2. 48 分组排序和查找算法可减少平均工作量 (116)
- 2. 49 用 BASIC 语言调用 8087 汇编语言子程序来完成快速实数排序的方法 (117)
- 2. 50 快速排序的 BASIC 程序 (118)
- 2. 51 在数据库中实现按姓氏笔划排序的方法 (一) (119)
- 2. 52 在数据库中实现按姓氏笔画排序的方法 (二) (120)
- 2. 53 分级排序方法与树搜索的结合应用 (121)
- 2. 54 一种新的快速排序方法——分类浮沉法 (123)
- 2. 55 对《WS》编辑的文件进行排序的方法 (125)
- 2. 56 在 C—dBASE Ⅲ 上实现汉字字段总笔画数排序的映射法 (126)

第三章 BASIC、PASCAL、COBOL、C、WORDSTAR 等语言的应用技巧

- 3. 1 在 CCDOS 下使用 True BASIC 的方法 (128)
- 3. 2 True BASIC 汉化的扩展 (128)
- 3. 3 用 BASIC 语言编写的 16 点阵汉字屏幕编辑式造字程序 (129)
- 3. 4 BASIC 语言中程序行的复制方法 (131)
- 3. 5 直接读取 FoxBASE 数据库的 BASIC 程序 (132)
- 3. 6 BASIC 语言调用 COBOL 数据文件的方法 (133)
- 3. 7 IBM PC BASIC 语言矢量字库及其应用 (134)
- 3. 8 BASIC 数据文件两级压缩技术的实现方法 (136)
- 3. 9 通用软件封面程序 (136)
- 3. 10 C 语言函数的参数传递过程 ... (140)
- 3. 11 用 Turbo C++ 编写去掉 Wordstar 分页符的程序 (142)
- 3. 12 用 C 语言编的在 DOS 环境下的文件连接程序 (142)
- 3. 13 Turbo C 字符方式下的颜色设

置..... (143)	3. 35 用 Turbo PASCAL5. 0 处理 dBASE Ⅲ 的 DBF 文件 (176)
3. 14 用 Turbo C 标准函数显示彩色 汉字的方法 (144)	3. 36 在 Turbo 中读写 dBASE Ⅲ 的 DBF 文件的两种方法 (179)
3. 15 Turbo C 在长城机上实现汉字 显示与查询实用程序 (145)	3. 37 用 PASCAL 语言实现不同汉字 系统的内码转换法 (179)
3. 16 用 Turbo C 编写的磁盘文件的 递归显示与查询实用程序 (146)	3. 38 实现对 COBOL 的自动批量编译 的程序 (181)
3. 17 使用 Turbo C 输入函数时遇到 的问题及解决办法 (147)	3. 39 COBOL 语言中不可恢复的 I/O 错误的调试方法 (182)
3. 18 Turbo C 的屏幕模式设置权的 获取法 (148)	3. 40 实现 COBOL 程序批量编译和字 符查找的简便方法 (183)
3. 19 用 Turbo C 将 WS 文书文件转 换成 TXT 文件的方法 (150)	3. 41 在 640K 的 IBM PC 机上不能运 行 COBOL 的两种解决方法 (183)
3. 20 用 Turbo C 2. 0 编写的一条 move 命令 (151)	3. 42 在内存 640K 的 IBM PC 机上不能 运行 COBOL 的简便解决方法 (184)
3. 21 简便实用的文件分割程序 (152)	3. 43 在 XENIX 交叉开发中用汇编语 言编写的文件格式转换程序 (185)
3. 22 子目录改名的一种简便方法 (153)	3. 44 提高编程效率的方法——建立汇 编语言工具箱 (187)
3. 23 用 C 语言编写的 CAT 文件连接 程序 (154)	3. 45 在 UNIX 环境下用 C 语言调用汇 编语言子程序的方法 (190)
3. 24 用 C 语言编写的删除目录实用 程序 (154)	3. 46 在 UNIX 上实现 FORTRAN 对 C 函数的调用方法 (191)
3. 25 键盘缓冲区直接存取例程 (156)	3. 47 用 WORDSTAR 编辑的小汇编程 序转入到 DEBUG 编译的方法 (194)
3. 26 C 语言中指针使用的常见错误... (159)	3. 48 自适应显示行数的 Wordstar ... (195)
3. 27 利用 PC 机自身的定时功能进行 精确定时 (160)	3. 49 为中文 WS 增加路径选择的一个 简便方法 (195)
3. 28 C 语言外部变量引用经验点滴... (162)	3. 50 WS 内存文件的恢复方法..... (196)
3. 29 实现外部硬件中断的编程方法... (162)	3. 51 WORDSTAR 工作参数的修改 方法 (198)
3. 30 汉字环境下运行 Turbo PASCAL 程序 (166)	3. 52 使 WS 在任意机型及汉字操作 系统下运行 (198)
3. 31 Turbo PASCAL 中传递过程参数 的有效途径 (168)	3. 53 对 WORDSTAR 编辑状态行宽 的定义 (199)
3. 32 用 Turbo PASCAL6. 0 编写的 DOS 加载程序 (170)	3. 54 介绍 WORDSTAR 中一条未公
3. 33 用 PASCAL 语言编写的判断两 图是否同构的程序 (173)	
3. 34 实用程序 README. COM 的汉 化方法 (175)	

- 布的命令..... (199) 3. 56 WS 放弃文本回收程序..... (200)
3. 55 Wordstar 未公布的 14 条命令 ... (199)

第四章 dBASE III 和 FoxBASE 等数据库的应用技巧

- | | |
|--|---|
| <p>4. 1 dBASE III 数据库自动校核程序 ... (202)</p> <p>4. 2 通用 dBASE III 数据库检索程序
的设计方法..... (204)</p> <p>4. 3 自动处理 dBASE III 文件中的非
法字符程序..... (207)</p> <p>4. 4 在 dBASE III 状态下用 WORDSTAR
来编写程序..... (208)</p> <p>4. 5 在 dBASE III 状态下使用其它软件
来编写程序..... (209)</p> <p>4. 6 生成报表程序的一种简单方法... (209)</p> <p>4. 7 dBASE III 多栏目多字段分类统计
报表的快速统计方法..... (212)</p> <p>4. 8 磁盘文件长度、剩余空间的统计
方法..... (212)</p> <p>4. 9 对全屏幕编辑命令 BROWSE 做
一点补充..... (213)</p> <p>4. 10 关于 ROUND () 函数的简单修
正法 (214)</p> <p>4. 11 分行处理较长的中西文混合字符
串的方法 (214)</p> <p>4. 12 用 SEEK 命令实现不等条件定
位 (216)</p> <p>4. 13 在 dBASE III 中 Seek 和 skip 命令
的联合使用可代替 Count
命令计数 (216)</p> <p>4. 14 ALTERNATE 命令的使用技巧... (217)</p> <p>4. 15 解决 dBASE III 编译工具 clipper
对 Total 命令的缺陷 (218)</p> <p>4. 16 解决 CLIPPER 不能编译子程
序的技巧 (219)</p> <p>4. 17 实现数据库中字符的替换程序... (219)</p> | <p>4. 18 dBASE III 程序实现锯齿结构的
自动转换 (220)</p> <p>4. 19 CHR[\$](14)的巧用与 dBASE III
图表功能的实现 (221)</p> <p>4. 20 用 dBASE III 进行程序设计中值得
注意的几个问题..... (224)</p> <p>4. 21 在 dBASE III 使用中容易忽视的
三个问题 (225)</p> <p>4. 22 用 SEEK 或 find 命令可提高连续
查找速度 (226)</p> <p>4. 23 正确实现 C-dBASE III 提供的索
引功能的方法 (226)</p> <p>4. 24 对动态数据库以菜单形式动态
输入、修改查询的方法 (227)</p> <p>4. 25 建立数据库结构有新法 (228)</p> <p>4. 26 在 dBASE IV 中利用内存变量改
变查询条件 (228)</p> <p>4. 27 实现 SET COLOR 语句的程序 ... (229)</p> <p>4. 28 提高汉字 dBASE III 执行速度的
方法 (229)</p> <p>4. 29 优化 FoxBASE+ 操作 (231)</p> <p>4. 30 实现通用多重模糊查询的一种
方法 (232)</p> <p>4. 31 FoxBASE 组合查询的实现程序... (233)</p> <p>4. 32 实现数据库检索方法之我见 ... (235)</p> <p>4. 33 模拟 pc shell 用户界面操作相关
数据库的设计技巧 (236)</p> <p>4. 34 在 FoxBASE 中产生随机数的
算法 (237)</p> <p>4. 35 FoxBASE 的三种容错处理的
流程 (241)</p> |
|--|---|

4. 36 实现 FoxBASE 窗口式全屏幕编辑的程序	(212)	4. 48 解决 FoxBASE 下汉字输入冲突的有效方法	(261)
4. 37 多个 READ 命令并存时实现全屏幕编辑的方法	(215)	4. 49 加强汉字 FoxBASE ⁺ 屏幕功能的汇编子程序	(262)
4. 38 FoxBASE ⁺ 下屏幕数据的编辑输入法	(217)	4. 50 提高 FoxBASE ⁺ 应用运行效率的一些途径	(267)
4. 39 FOX 程序的两种反编译方法 ...	(218)	4. 51 FoxBASE2. 1 函数的分类	(270)
4. 40 一种更为简便的 FOX 程序的反编译方法	(219)	4. 52 对 FoxBASE 程序的断电处理 ...	(272)
4. 41 FoxBASE ⁺ 反编译方法及程序...	(219)	4. 53 任意长字符串的编辑和存储的程序	(273)
4. 42 在 FoxBASE 下弹出式多窗口的实现	(253)	4. 54 通用的 dBASE (FoxBASE) 和高级语言的参数传递方法	(276)
4. 43 用 FoxBASE ⁺ 实现只在特定窗口内操作的程序	(255)	4. 55 FoxBASE ⁺ 的 CALL 命令及其参数传递实例	(277)
4. 44 FoxBASE 中也可方便地实现在窗口内显示移动信息	(255)	4. 56 dBASE 数据转入 ORACLE 系统的两种方法.....	(279)
4. 45 在 FoxBASE 中直接显示立体直方图的一种简单方法	(257)	4. 57 在 FoxBASE 中仿堆栈功能子程序	(284)
4. 46 用 FoxBASE ⁺ 实现“边拉边现”画面程序	(258)	4. 58 为 FoxBASE ⁺ 增加一条数据库结构函数	(285)
4. 47 完成菜单建立、管理、帮助的程序	(259)	4. 59 错误处理程序在调试程序中的应用	(286)

第五章 打印、制表方法及实用程序

5. 1 修改 CCDOS 4. 0V 打印驱动程序可变换字型.....	(288)	打印.....	(293)
5. 2 在其它汉字操作系统中调用 2. 13H 打印驱动程序的方法.....	(288)	5. 8 在 XENIX 下 FoxBASE 中的高级透明打印程序.....	(294)
5. 3 完善 GWBIOS 2. 13D 汉字系统打印机驱动程序功能的方法.....	(289)	5. 9 也谈多用户操作系统下 FoxBASE ⁺ 中高级透明打印技术.....	(295)
5. 4 为 LP. COM 增加分页功能	(290)	5. 10 用一条命令实现大批文本文件脱机打印的方法	(296)
5. 5 低速主机与高速打印机的匹配方法.....	(291)	5. 11 dBASE 文本文件大批量分页输出程序	(296)
5. 6 在 FoxBASE ⁺ 中实现终端打印的简单方法.....	(292)	5. 12 用 BASIC 形成供批量输出标准文件名程序	(297)
5. 7 用 FoxBASE 实现备注字段的表格		5. 13 用文本文件输出好的程序清单...	(298)

- | | |
|---|--|
| 5. 14 C—dBASE Ⅲ 自动连续分页打印
(一) (299) | 5. 30 用 WORDSTAR 编辑文件时实现
折页打印的方法 (319) |
| 5. 15 C—dBASE Ⅲ 自动连续分页打印
(二) (299) | 5. 31 WS 的折页打印 (319) |
| 5. 16 在 C—dBASE Ⅲ 下打印整个汉字字
库的程序 (300) | 5. 32 用中文 WS 进行高级打印 (320) |
| 5. 17 在带硬字库打印机上直接打印各
种字体字 (300) | 5. 33 在 CCDOS2. 13A 下用 COBOL 语
言打印实线表格的方法 (320) |
| 5. 18 在 AR—3240 打印机上实现图像
打印功能的方法 (301) | 5. 34 在 0520CH 下实现双线框和单线
框 (321) |
| 5. 19 解决源程序文件打印质量问题的
一种新途径 (303) | 5. 35 用 8088 汇编语言编写的快速制
表程序 (322) |
| 5. 20 多份文件快速阅读及打印程序... (304) | 5. 36 用 MFoxBASE+2. 00(C)中的 GRD
语句实现屏幕表格的改进... (323) |
| 5. 21 用假脱机和后台打印方式提高
打印速度 (305) | 5. 37 在 DOS 下将数据库转换成表格
输出的程序 (324) |
| 5. 22 打印机控制码序列转换程序 ... (308) | 5. 38 用宏汇编语言编写的汉字制表符
(六区↔九区)转换程序 (326) |
| 5. 23 LQ—1600K 打印机功能的设置
程序 (310) | 5. 39 弥补 CCED 编辑软件不足的
两种方法 (328) |
| 5. 24 在 LQ—1600K 上实现加宽打
印程序 (312) | 5. 40 OFFICE 制表软件使用经验
四则 (329) |
| 5. 25 LQ—1600K 打印机的一个控制
程序 (313) | 5. 41 用 CCED 将 dBASE 数据库进行
横向报表输出的方法 (330) |
| 5. 26 ROM BIOS 数据区空白区域的
应用 (314) | 5. 42 CCED 中文编辑软件运行中版本
到期的解决方法 (331) |
| 5. 27 五笔字型扫描表分析及码本打
印程序 (316) | 5. 43 一个新的制表程序 (332) |
| 5. 28 使 WPS 适应硬盘的多系统分区...
(317) | 5. 44 通用打印程序 (336) |
| 5. 29 WS 软件文章标题打印位置居中
的计算 (318) | 5. 45 一个实用的 BASICA 图形打印
程序 (337) |

第六章 图形处理方法与技巧

- | | |
|---|--|
| 6. 1 在 DOS 下直接执行 BASIC 屏幕图
形文件的方法 (340) | 6. 4 为 Quick BASIC 程序建立图形汉字
库 (342) |
| 6. 2 在 0520CH 系列机上用 BASIC 语言
放大屏幕彩色图形的方法 (340) | 6. 5 CH 机上的一个键盘绘图程序 ... (344) |
| 6. 3 用 BASIC 绘图功能弥补 dBASE Ⅲ
绘图功能的不足 (341) | 6. 6 用虚拟图形设备在 0520CH 机上 |

制图的技巧.....	(346)	现动态显示	(373)
6. 7 实用的离散结点绘图程序.....	(348)	6. 21 动态图形快速刷新显示方法 ...	(374)
6. 8 在 dBASE III plus、FoxBASE 下实 现绘图的一种简单方法.....	(351)	6. 22 EGA/VGA 屏幕存储与 恢复程序.....	(376)
6. 9 在 dBASE III 下直接作图的实现 方法.....	(352)	6. 23 EGA/VGA 平滑滚动及实用 程序.....	(377)
6. 10 FoxBASE 状态下的一种新的作图 方式	(354)	6. 24 屏幕图象放大快速平滑算法及 BASIC 程序	(380)
6. 11 中文方式下 Turbo C 及其图形功 能的使用	(355)	6. 25 IBM PC 机图形的先压缩再存储 的方法.....	(383)
6. 12 利用 Turbo C 的图形页技术实现 动画显示	(358)	6. 26 用 AutoLISP 编制绘图程序的技 巧	(384)
6. 13 用 Turbo PASCAL 绘图子程序 实现 FoxBASE+ 图形功能的 扩充	(359)	6. 27 将屏幕图形转换为 Auto CAD 命 令组文件	(385)
6. 14 通用数据库统计图形生成程序...	(361)	6. 28 加快 Auto CAD 绘图速度的 方法.....	(388)
6. 15 在 FoxBASE+ 系统设计中应用 系统图形驱动文件 GRD·SYS 的技巧	(362)	6. 29 图形压缩存储和快速调用软件...	(389)
6. 16 在 FoxBASE 下的通用 绘图程序	(364)	6. 30 压缩存储各时刻静态图实现动画 的方法	(391)
6. 17 解决 FORTRAN 绘图功能的简单 方法	(365)	6. 31 快速调用画面的方法	(393)
6. 18 利用内存数组直接形成图形数据 的绘图方法	(368)	6. 32 生成画面的实用技术	(394)
6. 19 在配 CGA 显示卡的计算机上实现 两页作图的方法	(372)	6. 33 具有多级放大功能的屏幕图形拷 贝程序的设计与实现	(396)
6. 20 用位面显示法在 EGA 图形板上实 现动态显示	(373)	6. 34 实用的汉字屏幕硬拷贝程序 ...	(399)
		6. 35 高分辨率屏幕图形拷贝程序 ...	(400)
		6. 36 对图形硬拷贝的改进办法	(401)
		6. 37 微机绘图中任意图形的变换 ...	(402)

第一章 DOS、MSDOS、CCDOS2.13、XENIX 操作系统使用经验

1. 1 动态选择安装 DOS 设备驱动程序的方法

从 DOSV2. 00 版本起便给用户提供了安装附加的设备驱动程序的方法, 即使用 CONFIG. SYS 文件。每当机器启动时, 在 CONFIG. SYS 文件中指明的各个设备驱动程序便被依次安装进入内存。但有时会遇到这种情况, 即希望有选择地安装或不安装某些驱动程序。例如在网络环境下 (如 3+网), 很多服务功能都是通过设备驱动程序的方式提供的。如果暂不使用网络功能而将机器作为单机启动, 这些驱动程序仍将被安装进来并占去相当大一部分内存。通常的解决办法就是先修改 CONFIG. SYS 文件, 删去那些不希望安装的

DEVICE=.....

语句, 然后重新启动。但这种做法显得较麻烦且容易出错。

是否能够在机器启动时动态地决定安装或不安装某些设备驱动程序? 可以, 而且并不复杂。这先要从 DOS 对设备驱动程序做一些了解。

DOS 在自举过程中将 CONFIG. SYS 文件的内容读入内部工作区, 并转换为内部格式, 然后才逐一将指定的驱动程序装入。所谓内部格式即:

- ①所有字母转换为大写;
- ②去掉不必要的空格和制表符;
- ③将每个关键词用一个特写字母代替。

其中关键词与相应字母的对应关系如表所示。

例如, 对 CONFIG. SYS 中的语句行

DEVICE=ansi. sys

其转换后的内部格式为:

DANSI. SYS<NULL><LF>

又如对语句行

DEVICE=VDISK. SYS380

则转换成为

DVDISK. SYS<NULL><SPACE>380<CR><

LF>

其中<NULL>、<CR>、<LF>、<SPACE>分别为 ASCII 空、回车、换行和空格字符。

表 关键词与相应字母的对应关系

关键词	对应字母
BREAK=	C
BUFFERS=	B
COUNTRY=	Q
DEVICE=	D
FCBS=	X
FILES=	F
LASTDRIVE=	L
SHELL=	S
STACK=	Z

总之, DOS 总是在等号后面的第一个参数 (对 DEVICE 语句就是设备驱动程序的文件名) 之后插入一个<NULL>字符, 以便于执行 open 系统调用。当调用成功后 (驱动程序文件已被打开并读入内存), DOS 便将<NULL>字符用空格代替, 以便于驱动程序自身读取相应参数。此后, DOS 在其内部工作区构造一个 RH (Request Header), 并对该驱动程序以命令号 0 执行远调用, 即初始化 (initialization)。关于 DOS 与设备驱动程序之间的调用细节本文不详细讨论, 读者如感兴趣可参见 DOS 技术手册。值得注意的是在 RH 中包含有指向 CONFIG. SYS 内部格式缓冲区的远指针, 其本意是用来供驱动程序提取命令行参数用的 (如上例中 VDISK. SYS 后面的 “380”)。但若谨慎地对这一 DOS 内部缓冲区进行操作, 例如将其用空格充填, 则会使 DOS 认为已经没有待装入的设备驱动程序, 从而达到不用修改 CONFIG. SYS 文件而跳过其中指明的设备驱动程序的目的。

本文后面所附的是一个按上述思路设计的简单但已足够实用的程序清单 SKIP. ASM。将其经汇编、连接和二进制转换:

MASM SKIP
LINK SKIP
EXE2BIN SKIP

生成 SKIP. BIN 后拷入 C: 盘根目录中, 并对 CONFIG. SYS 文件作如下修改:

(1) 在需要进行动态选择安装的设备驱动程序命令行之前行插入一行, 格式为:

DEVICE=SKIP. BIN 字符提示

其中“字符”是任一非空格的 ASCII 字符, “提示”是任一字符串。

(2) 在需要进行动态选择安装的设备驱动程序命令行之后也插入一行, 格式为:

DEVICE=SKIP. BIN

(注意后面一定不要带参数)

这样在重新启动时, 当 DOS 将第一个 SKIP. BIN 作为设备驱动程序装入后, 便会将给定的“提示”内容显示出来, 并等待键盘输入一个字符。如所输入的恰是给定“字符”(忽略大小写), 则所有驱动程序将被照常装入, 正如没有 SKIP 语句行时一样; 而如果按任一其它键(如回车), 则位于两个 SKIP 语句行之间的所有设备驱动程序将被忽略而不装入。

下面是一个实际的 CONFIG. SYS 文件的例子。为便于说明, 在前面加注了行号:

```
1: FILES=35
2: BUFFERS=10
3: DEVICE=ANSI. SYS
4: DEVICE=VDISK. SYS buffer=380 KB, sector=
  512B, dir=8, /E: 8
5: DEVICE=SKIP. BIN Y Do you want to start PC-
  NFS (Yor [N])?
6: DEVICE=F: \NFS\PCNFS. SYS/x
7: DEVICE=F: \NFS\SOCKDRV. SYS
8: DEVICE=F: \NFS\SLIP. SYS/p2/b9600
9: LASTDRIVE=V
10: DEVICE=SKIP. BIN
```

每当机器启动时, 第 1~4 行照常执行, 而在执行第 5 行时, DOS 提示

DO YOU WANT TO START PC - NFS (YOR [N])?

并等待输入。如按“Y”键, 则第 6~9 行将照常执行(与 SUN 工作站联网软件被装入), 而第 10 行对 DOS 无任何影响。如按任一其它键, 则第 6~10 行的内容全被跳过。可见两个 SKIP

语句行实际上起到了 IF 和 ENDIF 的作用。

有兴趣的读者不妨一试, 也许 SKIP 程序能为您提供些帮助或启发。以 SKIP. ASM 为原型, 读者也很容易设计出具有更强功能(如多重选择)或更完美人机界面(如菜单式交互)的选择程序来。

本文所提供的 SKIP 程序是在 IBM PC-DOS V3. 30 版本下, 用宏汇编 V5. 00 实现的。对于其它版本如万一不能顺利使用, 仿照此思路进行修改也是不困难的。

```
; *****SKIP. ASM*****
; Program to enable device drivers to be selectively
; loaded.
;
; Author: LuKe
;
; USAGE:
;
; DEVICE=SKIP. BIN Character Prompt-String
; DEVICE=...
; .....
; DEVICE=...
```

```
RH          STRUC ; request header
RH__LEN     DB ?   ; len of packet
RH__UNIT    DB ?   ; unit code (block devices only)
RH__CMD     DB ?   ; device driver command
RH__STATUSDW ? ; returned by device driver
RH__RES1    DD ?   ; reserved
RH__RES2    DD ?   ; reserved
RH          DNDS
```

```
RHO          STRUC initialization (command 0)
RHO__RH     DB     SIZE RH DUP (?); fixed portion
RHO__NUNITS DB ? ; number of units (block devices
                  only)
RHO__BRK__OFS DW ? ; offse for break
RHO__BRK__SEG DW ? ; segment for break
RHO__BPB__TBO DW ? ; offset of pointer to BPB
                  ; array
RHO__BPB__TBS DW ? ; segment of pointer to BPB
                  ; array
```

```

RHO __ DRV __ LTR DB ? ; fits available drive (block,
                        devices only)
RHO                ENDS

```

```

NULL EQU 0
TAB EQU 9
LF EQU 0AH
CR EQU 0DH
SPACE EQU 20H

```

```

CODE SEGMENT
    ASSUME CS: CODE

```

```

    ORG 0

```

```

NEXT __ DEV DD -1
ATTRIBUTE DW 8000H ; char device
STRATEGY DW DEV __ STRATEGY
INTERRUPT DW DEV __ INTERRUPT
DEV __ NAME DB 'SKIPSKIP'

```

```

RH __ ADDR LABEL DWORD
RH __ ofs DW ?
RH __ SEG DW ?

```

```

VERIFY DB 'DSKIP. BIN'
VERIFYL EQU $ __ OFFSET VERIFY

```

```

DEV __ STRATEGY:

```

```

    MOV CH, RH __ ofs, BX
    MOV CS, RH __ SEG, ES
    RETF

```

```

DEV __ INTERRUPT:

```

```

    PUSH AX
    PUSH BX
    PUSH CX
    PUSH DX
    PUSH SI
    PUSH DI
    PUSH DS
    PUSH ES
    JMP INIT ; the only possible call to SKIP is;
                initialization

```

```

EXIT:

```

```

    LES BX, CS, RH __ ADDR
    MOV ES, [BX]. RH0 __ BRK __ ofs. 0; do

```

not reside in memory

```

MOV ES, [BX]. RH0 __ BRK __ SEG, CS
MOV ES, [BX]. RH __ STATUS, 0100H;
done

```

```

POP ES
POP DS
POP DI
POP SI
POP DX
POP CX
POP BX
POP AX
RETF

```

```

INIT:

```

```

    CLD
    LES BX, CS, RH __ ADDR
    LDS SI, DWORD PTR
        ES, [RX]. RH0 __ BPB __ TBO
    ADD SI, VERIFYL

```

```

INIT1: ; determine the existence of parameters

```

```

    LODSB
    CMP AL, SPACE
    JZ INT 1
    CMP AL, TAB
    JZ INT 1
    CMP AL, CR
    JZ EXIT
    CMP AL, LF
    JZ EXIT
    CMP AL, NULL
    JZ EXIT
    MOV CL, AL ; CL=the Character
    CALL NEWLINE

```

```

INIT2 ; display the Prompt—String

```

```

    LODSB
    CMP AL, CR
    JZ INIT3
    CMP AL, LF
    JZ INIT3
    CMP AL, NULL
    JZ INIT3
    CALL DISPLAY
    JMP INIT2

```

```

INIT3: ; wait for input

```

```

MOV     AH, 0
INT     16H
CALL    DISPLAY
CALL    NEWLINE
CMP     AL, 'a' ; convert to uppercase
JB      INIT4
CMP     AL, 'z'
JA      INIT4
AND     AL, NOT 'a' - 'A'

INIT4:
CMP     AL, CL ; if it is the same Character,
        let
DOS continue
JZ      EXIT
MOV     DI, OFFSET VERIFY
MOV     AH, SPACE ; filling character
INIT5:
MOV     BX, 0
INIT6:
LODSB
MOV     [SI-1], AH ; blank DOS buffer until;
        the next SKIP line
CMP     AL, CS: [DI+BX]
JNZ     INIT5
INC     BX
CMP     BX, VERIFYL
JB      INIT6
JMP     EXIT

```

```

NEWLINE PROC
PUSH    AX
MOV     AL, CR
CALL    DISPLAY
MOV     AL, LF
CALL    DISPLAY
POP     AX
RET

```

NEWLINE ENDP

```

DISPLAY PROC
PUSH    AX
PUSH    BX
MOV     AH, 0EH
MOV     BX, 7

```

```

INT     10H
POP     BX
POP     AX
RET
DISPLAY ENDP
CODE ENDS
END

```

(吕珂)

1. 2 检查 DOS 内存的程序

笔者设计了这个程序，能够检查出内存中每个驻留程序的位置、大小等情况。

1. DOS 的内存分配原理

DOS 将各个存储块用链式结构存放。在每个存储块前有 1 个 16 字节长的存储器控制块，存储器控制块的前五个字节是有效的，含义如下：

字节 0 (1 字节) 如果是链中的最后一块则为 'Z'，否则为 'M'；

字节 1 (2 字节) 使用这个内存块的程序的前缀段段址，如果为 0 则表示该块已被撤销分配；

字节 3 (2 字节) 存储块大小 (单位为节——16 字节)；

第 1 块的地址放在 1 个内部变量中，DOS 通过这个变量找到第 1 个已分配的存储块的控制块，从其中的信息可找到下 1 个存储块，依次类推 (图 1. 1)。因此，只要找到头 1 个存储块的地址，就能找到所有的存储块，并得到存储块的有关信息。

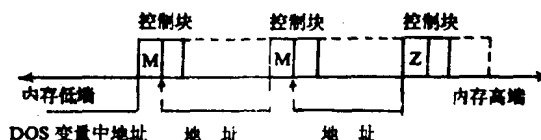


图 1. 1

当系统装入一个程序时，DOS 首先为它建立 1 个环境块 (后面介绍)，然后 DOS 就确定可