

PC系列机 工具箱(四)

最新
出版

- 张显贵 姚玉珍 编
- 电子工业出版社



PC 系列机工具箱(四)

张显贵 姚玉珍 编

电子工业出版社

(京)新登字 055 号

内容提要

本系列丛书分五册介绍 PC 系列机使用经验和编程技巧。本册内容包括四个部分: C 语言编程技巧, Pascal 使用技巧, BASIC 技巧程序和其它语言编程技巧。书中介绍了具体应用实例 200 多个, 大部分文章除有文字说明外, 还有程序实例。利用书中介绍的编程经验和技巧, 能解决实际编程中或计算机使用过程中的很多问题, 对开发应用软件有很大帮助。

本书适合从事计算机方面工作的工程技术人员 大专院校计算机专业师生阅读。

ISBN 7-5053-2371-7

PC 系列机工具箱(四)

张显贵 姚玉珍 编

陈铁钢 审

责任编辑 张丽华

*

电子工业出版社(北京市万寿路)

电子工业出版社发行 各地新华书店经销

山东电子工业印刷厂印刷

*

开本: 787×1092 毫米 1/16 印张: 26 字数: 648 千字

1994 年 6 月第 1 版 1994 年 6 月第 1 次印刷

印数: 0001—12000 册 定价: 19.80 元

ISBN 7-5053-2371-7/TP·674

前 言

很久就想编一本书,一本有关计算机方面的工具性很强的书,因考虑不成熟,始终未能如愿。经几年实践积累,现推出笨拙之作奉献给广大读者,但愿它的诞生能给从事计算机方面工作的人员以帮助,成为他们的得力助手,充分发挥“工具箱”的作用。

书中所包含的内容是很多工程技术人员经验和智慧的结晶,也是我们编者辛勤劳动的结果。所有程序都进行了仔细调试和验证,所有的维修经验也都一一验证过。无可非议,此书的出版,将对我国计算机事业的发展有很大的帮助。

当然,因现在 PC 系列机种类繁多,难免有些软件因机型不同而产生不完全兼容的问题,很可能在一种机器上运行完好的程序,在另一种机器上实现不了。因此,希望读者在使用书中的技巧时,一定要注意其运行环境。

在本书的编写过程中,得到彭清、袁征、陈俊、马晓晓、孙继刚、严明、叶端红、尚光洁、张利民、胡芳芳、赵辉、刘天军、许广义、童日辉等同志的热心帮助,在此表示深深的感谢。因水平有限,书中可能会有一些疏漏之处,请广大读者给予指正,并诚恳地希望能将问题反馈给我们,以便及时修正,更希望同我们交流经验。

编 者

1993 年 8 月于北京

目 录

第一部分 C 语言编程技巧	(1)
1. 保存与恢复汉字屏幕窗口的方法	(1)
2. 在 Turbo C 2.0 图形方式下显示彩色汉字	(3)
3. 用 C 语言实现的文本文件阅读器	(5)
4. 磁盘文件的递归显示与查询实用程序	(9)
5. DOS 环境下的文件连接程序	(11)
6. 用 Turbo C 编写的一条 move 命令	(12)
7. Turbo C 语言可变参数的引用	(14)
8. 开发 TVGA 彩卡高分辨(1024×768)图形功能	(15)
9. 在文本文件中打印机控制码的使用	(18)
10. 一个简单的 C 语言音乐函数	(20)
11. 硬盘分区表的保存与恢复	(21)
12. 动态屏幕设计	(27)
13. Turbo C 实现弹出式菜单	(29)
14. 开机启动时自动关闭 Num Lock	(32)
15. 屏幕高低显示转换程序	(33)
16. 用 C 语言修复 dBASE Ⅲ 库文件	(34)
17. 一种制作立体投影窗口的方法	(35)
18. 图象屏幕硬拷贝中断服务程序	(37)
19. 汉化软件工具的设计与实现	(39)
20. C 语言可执行代码的调用	(42)
21. 用 C 语言实现放大汉字	(43)
22. 巧用 C 中 union 实现二进制文件的传输	(45)
23. 模拟键盘输入的通用子程序设计及应用	(48)
24. Turbo C 字符方式下的颜色设置	(51)
25. 中西文字符串快速显示	(51)
26. 利用 PC 机进行精确定时	(53)
27. 利用 C 语言对数据库加密	(55)
28. C 语言程序外部变量的引用	(56)
29. 去掉 WS 分页符的程序	(57)
30. Turbo C 在 CH 方式下绘图的方法	(58)
31. Turbo C 在长城机上实现汉字显示	(59)
32. 像 Pascal 语言一样使用 C 数组	(61)
33. 使用 Turbo C 标准函数显示彩色汉字	(64)

34. 中文方式下 Turbo C 及其图形功能的使用	(65)
35. Turbo C 实现具有立体投影效果的选择菜单	(67)
36. C 语言 Printf() 函数中的“++”运算符	(69)
37. 用 C 语言巧写中断服务程序	(71)
38. Turbo C 程序设计中的汉字处理技术	(73)
39. Turbo C 语言的打印机输出	(81)
40. 用 C 语言给源程序加行号	(82)
41. C 语言直接处理汉字的实现	(83)
42. 微机硬盘锁定一法	(84)
43. 利用 Turbo C 的图形页技术实现动画显示	(84)
44. Turbo C 2.0 键盘输入功能的扩充	(87)
45. 子目录改名的一种简便方法	(88)
46. 用 C 语言实现光条菜单	(89)
47. 用 Turbo C 实现全屏幕图形编辑	(90)
48. 自编 C 语言三个字串处理函数程序	(93)
49. Turbo C 2.0 图形库函数的一点不足	(94)
50. 用 C 语言扩充 DOS 命令参数	(95)
51. 简便实用的文件分割程序	(97)
52. 文件目录显示功能的改进	(99)
53. 在 DOS 下直接调用 BIOS 中断例程	(101)
54. DOS 状态下移动文件的几种方法	(103)
55. 利用 2.13 高点阵字库实现屏幕汉字放大	(105)
56. 窗口环境下带过滤的字符串输入	(108)
57. 长城 CVGA 卡 ZOOM 功能的一种特殊用途	(111)
58. 打印控制程序的设计	(114)
59. 获取磁盘参数的通用数据结构	(121)
60. 预防“定时炸弹”病毒	(124)
61. 完全清除内存新方法	(128)
62. 中文 Windows 使用技巧	(129)
63. 在 ORACLE 中如何实现中文格式报表的输出	(131)
64. C 语言程序的优化	(133)
65. 自动添加段结束符 \$	(136)
66. 通用打印程序	(138)
67. 一个实用的源程序统计工具	(140)
68. Turbo C++ 文本及图形方式下的汉字处理方法	(142)
69. 在配有 CEGA 卡的微机上 Turbo C 的汉字显示	(143)
70. C 语言使用扩展的屏幕和键盘控制	(146)
71. C 语言读取 dBASE 数据库文件的通用方法	(149)
72. 如何从程序内部对转换进行控制	(151)
73. 用 C 语言对金额数据进行处理	(152)

74. 多功能打印源程序清单	(154)
第二部分 Pascal 使用技巧	(163)
1. Pascal 与 dBASE Ⅲ 数据通信通用接口模块的设计	(163)
2. Pascal 直接读取 dBASE/FoxBase 数据库数据的通用方法	(171)
3. dBASE(FoxBase)和高级语言的参数传递方法	(175)
4. dBASE Ⅲ 与 Pascal 参数相互传递	(177)
5. 如何获知系统配置的逻辑盘数	(178)
6. 利用 ANSI.SYS 重新定义功能键	(179)
7. 怎样使长城 0520CH 一屏显示 28 行表格	(181)
8. 应用程序坚固性的讨论	(183)
9. 一个方便实用的硬盘测试程序	(187)
10. LQ—1600K 打印机控制程序	(189)
11. 从一种新途径来解决源程序文件的打印质量问题	(190)
12. 计算机自动识谱音乐程序	(192)
13. 滚动式菜单简易设计	(194)
14. Pascal 实现单键菜单选择的方法	(195)
15. 用 Turbo Pascal 控制打印机	(196)
16. 用 Turbo Pascal 实现中断服务例程	(197)
17. 用 Turbo Pascal 4.0 实现图形动态显示	(198)
18. 计算机模拟电视字幕移动	(199)
19. 实现 EGA 显示器每屏显示行数的转换	(201)
20. Turbo Pascal 4.0 汉字显示	(202)
21. 显示器保护程序	(204)
22. 《渴望》主题歌音乐程序	(205)
23. 谈 Turbo Pascal 4.0 汉字显示	(207)
24. 再谈 Turbo Pascal 4.0 汉字显示	(207)
25. 汉化 Turbo Pascal 程序在 CCDOS 环境下不能显示汉字的解决办法	(208)
26. 多份文件快速阅读及打印程序	(210)
27. 修改 Turbo.COM 适应 EGA 卡 80×13 的显示方式	(212)
28. Pascal 语言对 dBASE Ⅲ 的 DBF 文件的直接访问	(212)
29. 给 PC 及 PC/XT 机添加“日历”	(214)
30. 在 DOS 下打印输出金山中文系统的图象文件	(216)
31. 全屏幕图象硬拷贝程序单元	(219)
32. Turbo Pascal 中的窗口制框	(220)
33. 多物体动画设计探索	(221)
第三部分 BASIC 技巧程序	(225)
1. 将 BASIC 图形文件转换成数据库文件	(225)
2. 用 BASIC 对 dBASE Ⅲ 源程序自动再编辑	(226)
3. 在 BASIC 系统下调用其它软件	(227)
4. 在 EDLIN 行编辑指令下修改 BASIC 程序	(229)

5. dBASE III 与 BASIC 语言通过内存传递数据	(230)
6. 巧用 BASIC 陷阱技术	(232)
7. 放大字体	(232)
8. 加速编制表格打印程序的一个方法	(235)
9. 英文高级语言软件汉字接收显示	(237)
10. 用 TRUE BASIC 实现动画技术	(237)
11. 编制 TRUE BASIC 键盘彩色绘图程序	(239)
12. 在长城 0520CH 机高分辨率下用 BASIC 语言绘图	(241)
13. GWBASIC 功能键的修改方法	(242)
14. Turbo BASIC 输入语句的改进	(242)
15. 病毒的消除及磁盘免疫程序	(243)
16. 随机文件中两个 FIELD 语句的连接	(245)
17. 均衡调用打印针	(246)
18. 实用自动生成空表格程序	(248)
19. 分页打印正文文件	(249)
20. 将文件分段或取某一段打印的实用程序	(251)
21. 用 BASIC 语言编写的双页打印程序	(252)
22. 在 LQ-2500K 打印机上打印彩色字	(253)
23. 实现文件的对折打印	(254)
24. 彩色动态字幕及伴音引导程序设计	(256)
25. 检测硬盘适配器	(257)
26. 音乐伴奏日历实时钟的建立	(258)
27. LQ-1600K 打印机断针检测程序	(260)
28. 中分辨率 CCDOS 下 BASIC 字符串反相显示	(260)
29. 屏幕图形数字标注通用子程序	(262)
30. DOS 下将数据库转换成表格输出	(263)
31. 巧用字符串模拟堆栈操作	(265)
32. 词组文件的高效制作及不同版本间的格式变换	(266)
33. 计算机键盘模拟电子琴演奏	(268)
34. 确定个人计算机的机型	(268)
35. 利用汉字库在屏幕上生成美术字	(269)
36. 自动生成音乐程序的方法	(270)
37. Turbo BASIC 在汉字系统下的光标定位输入	(272)
38. 修改 TRUE BASIC 语言使之能在单显系统中使用图形方式和显示汉字	(274)
39. 制表符转换程序	(275)
40. 简单易行的查找汉字区位码的方法	(276)
41. 《TRUE BASIC 标准》中多数据项记录存储示例的失误	(277)
42. TRUE BASIC 语言数组定义技巧	(278)
43. BASIC 语言读取 dBASE III 数据库文件的通用程序	(279)
44. 更换目录名小程序	(280)

45. 屏幕图形文件改为可执行文件	(281)
46. BASIC 语言调用 COBOL 数据文件	(283)
47. 在 CCDOS 下使用 TRUE BASIC	(284)
48. TRUE BASIC 汉化的扩展	(285)
49. 纯西文键盘输入文件的压缩加密	(286)
50. 汉字及对应内码的查询	(287)
51. 应用软件封面的简单制作	(288)
52. 快速查找汉字国标字符机内码的一种方法	(290)
53. 3.5 英寸与 5.25 英寸磁盘文件的相互复制	(290)
54. 追回失去的 BASIC 程序	(292)
55. 几种新发现的 BASIC 命令的作用和句法	(295)
56. 变序多重加密法	(297)
57. 在带硬字库打印机上直接打印各种字体字	(299)
58. 实用的整理 PRG 文件程序	(300)
59. 利用汇编语言为 BASICA 增加局部清屏功能	(302)
60. PC BASIC 与 TRUE BASIC 的比较	(303)
61. 给 BASIC 增加一个类似于 TYPE 的“命令”	(307)
62. BASIC 中 INKEY \$ 函数的几种用法	(308)
63. 3070 打印机字/行间距控制程序	(309)
64. 用 BASIC 语言实现在打印机上打印屏幕图形	(311)
65. IBM PC 文本文件密写程序	(311)
66. BASIC 语言中移动光标修改数据的一种方法	(312)
67. 用 BASIC 语言作图	(313)
68. 用软中断解决 Turbo BASIC 的汉字输入问题	(314)
69. 用 BASIC 语言进行不同精度变量混合运算的一点经验	(316)
70. 编译 BASIC 环境下实现屏幕内容的自动打印	(317)
71. TRUE BASIC 键盘彩色绘图程序	(318)
72. 用 TRUE BASIC 隐形输入密码	(325)
73. BASIC 的 INVIRON 语句与 ENVIRON \$ 函数	(327)
74. 打印实用程序	(327)
第四部分 其它语言编程技巧	(331)
1. COBOL 语言的编程技巧	(331)
2. COBOL 程序批量编译和字符查找的简便方法	(333)
3. 如何实现 COBOL 程序批量编译	(333)
4. 提高二级 COBOL 数字数据接收能力的简易方法	(334)
5. COBOL 语言功能的新扩展	(335)
6. 如何得到键盘扫描码	(337)
7. 实用系统安全“黑盒子”程序	(339)
8. 在 DOS 下直接执行 BASIC 屏幕图形文件	(342)
9. 长城 0520CH 机 COBOL 语言如何调用汇编语言	(344)

10. 在长城 0520CH 机上利用 FORTRAN 语言绘图	(34)
11. 用 FORTRAN 语言使用绘图仪	(34)
12. 扩充 MS Pascal 的彩色汉字显示功能	(34)
13. 使 Turbo Pascal 5.0 在 CH 机上正常显示	(350)
14. 一个实用的音乐报时程序	(351)
15. 双机共享打印机系统	(355)
16. DIR I 型病毒的免疫程序设计	(359)
17. DIR I 病毒的清除	(362)
18. 286/386 机器扩充软磁盘容量的一种新方法	(371)
19. 子目录改名	(373)
20. 汉字 FoxBase+ BIOS 图形功能调用的通用程序	(375)
21. 四通打字机文件与计算机文件转换	(378)
22. 实现 COBOL 软件的脱机打印	(384)
23. 实现 MS COBOL 索引文件到 LEVEL I COBOL 索引文件的转换	(385)
24. 在 2.13A 系统下 COBOL 语言打印实线表格	(387)
25. 利用 COBOL 语言处理变长记录	(388)
26. dBASE 数据库和 COBOL 语言的结合使用	(390)
27. 给 BAT 文件增加菜单选择功能	(393)
28. 用 FORTRAN 77 在微机上实现部分功能控制	(394)
29. FORTRAN 源程序的处理	(396)
30. FORTRAN 与 AUTO LISP 的数据接口	(398)
31. 利用 AUTO LISP 语言建立分级菜单的几个问题	(399)
32. Turbo PROLOG 在文本方式下的汉字显示	(400)
33. 在 PC 机上用不同语言编程及进行数据传送	(401)
34. 在高级语言中巧用程序段前缀信息	(403)
35. 节省内存空间和时间的方法	(405)

第一部分 C 语言编程技巧

1. 保存与恢复汉字屏幕窗口的方法

方法(一)

通过分配内存来实现屏幕和窗口的保存与恢复,见程序(一)。内存的大小由屏幕窗口的大小决定,通过 malloc() 函数分配,调用 save_video() 函数存贮,调用 restore_video() 函数恢复屏幕窗口。该方法适用于对部分屏幕或窗口的保存与恢复。

方法(二)

利用显示缓冲区中的空余页面进行存贮,见程序(三)。调用 save_video_page() 函数存贮,调用 restore_video_page() 函数恢复。适用于对整个视屏的存贮与恢复,改变函数的段地址可改变页面号,最多可保存 6 个视屏。该方法的特点是速度快,不需另行分配内存。

方法(三)

把汉字屏幕的整个视屏或窗口永久地保存在文件中,见程序(二)。由于要进行磁盘的读写操作,速度稍慢。

演示程序分别用三种方法对屏幕和窗口进行存贮、恢复和移动操作,采用快速的直接视屏存取,对中西文屏幕都适用。一些没有提供对汉字屏幕存贮和恢复功能的高级语言,也可调用方法(二)或方法(三)的功能。程序用 Turbo C 2.1 编译,在浪潮 286 (CEGA 卡) 和东海 286 (CEGA 卡) 上运行通过。

程序清单

```
#include<stdlib.h>
#include<stdio.h>
#include<conio.h>
#include<mem.h>
#include<string.h>
main()
{
    int size, startx=10, starty=19, endx=13, endy=33;
    int startx1=15, starty1=40, endx1=18, endy1=54;
    char *p1, *p2;
    FILE *fd;
    size=(endx-startx+1)*(endy-starty+1)*2;
    p1=malloc(size);
    p2=malloc(size);
    display();
    getch();
    /* 程序(一) */
    save_video(startx, starty, endx, endy, p1, p2);
    clrscr();
    restore_video(startx, starty, endx, endy, p1, p2);
    getch();
}
```

```

/* 程序(二) */
fd=fopen("scr1.xwf","w+b");
save_video(startx, starty, endx, endy, p1, p2);
fwrite(p1, size, 1, fd);
fwrite(p2, size, 1, fd);
fclose(fd);
fd=fopen("scr1.xwf","r+b");
fread(p1, size, 1, fd);
fread(p2, size, 1, fd);
restore_video(startx1, starty1, endx1, endy2, p1, p2);
fclose(fd);
getch();
/* 程序(三) */
save_video_padge();
clrscr();
getch();
restore_video_padge();
free(p1);
free(p2);
}
display()
{
register int i, j;
for (i = 1, j = 15, i < 26; i++)
{
gotoxy(j, i);
printf("THIS IS THE TEST OF THE CHINESE SCREEN PROGRAM");
}
gotoxy(20, 11);
printf("中国计算机用户");
gotoxy(20, 12);
printf("中国计算机用户");
gotoxy(20, 13);
printf("中国计算机用户");
gotoxy(20, 14);
printf("中国计算机用户");
}
save_video(int startx, int starty, int endx, int endy, char * p1, char * p2)
{
register int i, j;
char far * v1 = (char far *) 0xB8000000; char far * t1;
char far * v2 = (char far *) 0xB8000000; char far * t2;
for (i = startx; i < endx + 1; i++)
for (j = starty; j < endy + 1; j++)

```

```

    {
        t1=(v1+(i*160)+j*2);
        *p1++=*t1--;
        *p1--=*t1;
        t2=(v2+(i*160)+j*2);
        *p2++=*t2++;
        *p2--=*t2;
    }
}

restore_video (int startx, int starty, int endx, int endy, char *p1, char *p2)
{
    register int i, j;
    char far *t1=(char far *) 0xB8000000; char far *v1;
    char far *t2=(char far *) 0xB8000000; char far *v2;
    for(i=startx; i<endx+1; i++)
    for(j=starty; j<endy+1; j++)
    {
        v1=t1;
        v1+=(i*160)+j*2;
        *v1++=*p1++;
        *v1=*p1--;
        v2=t2;
        v2+=(i*160)+j*2;
        *v2--=*p2--;
        *v2=*p2++;
    }
}

save_video_page ()
{
    movedata (0xb800, 0x0000, 0xb900, 0000, 4000);
    movedata (0xb000, 0x0000, 0xb100, 0000, 4000);
}

restore_video_page ()
{
    movedata (0xb900, 0x0000, 0xb800, 0000, 4000);
    movedata (0xb100, 0x0000, 0xb000, 0000, 4000);
}

```

徐卫飞

2. 在 Turbo C 2.0 图形方式下显示彩色汉字

Turbo C 2.0 提供了丰富的图形函数库,因而用它来开发用户界面相当完美。然而,由于它是西文软件,不可避免地在汉字处理方面就要有许多不足之处。

Turbo C 2.0 在图形方式下,提供了 outtextxy 函数,其主要功能是将一串字符送到指定

位置。但此函数不能输出汉字字符串,一般用户只能用 printf 函数来显示汉字串。因此,就不能像 outtextxy 函数那样在图形下正确地定出汉字串的显示位置,也不能显示出彩色汉字。为此,研究了一套方法,不但能完成 outtextxy 函数的一切功能,而且还能显示出彩色汉字。经过一些处理后,还能放大和产生美术字体,从而大大地增强了用户界面的完善性。

程序清单

```
# include <graphics.h>
# include <stdio.h>
# include <stdlib.h>
# include <conio.h>
# include <dos.h>
# include <string.h>

void drawhz(int x,int y,char finame [15], int color) /* 输出汉字串函数 */
{
    short int coordx, coordy,i;
    FILE * fp;
    if((fp=fopen(finame, "rb"))== 0)    /* 判断汉字串文件存在否? */
    {
        printf("Can not open the file\n");
        return;
    }
    while(! feof(fp))    /* 读入汉字串文件内容 */
    {
        i=fscanf(fp, "%d, %d", &coordx, &coordy);
        if (i<=0) {fclose (fp);return;}
        coordx=x+coordx; coordy=y+coordy;
        putpixel(coordx,coordy,col); /* 显示汉字串 */
    }
    fclose(fp);
}

main()
{
    int maxx,maxy,x1,y1,x2,y2,errorcode;
    int graphdriver=DETECT,graphmode;
    struct arccoordstype arcinfo;
    initgraph(&graphdriver,&graphmode,""); /* 设置图形显示器方式 */
    errorcode=graphresult();
    if (errorcode!= 0) /* 判断图形文件存在否? */
    {
        gotoxy(10,10);printf ("图形文件错误: %s\n",grapherrormsg(errorcode));
        gotoxy(10,11);printf ("按任一建退出...");
        getch();
        closegraph(); exit (1);
    }
}
```

```

drawhz (80,130, "jqxxxt.hzc",4); /* 调用显示汉字串函数 */
/* 注:从坐标(80,130)处开始显示一串汉字,"jqxxxt.hzc"为汉字串文件,存的为汉字串的点阵坐标值,"
4"为要显示汉字串的颜色值(此处为红色)。 */
getch(); closegraph(); exit(0);
}

```

汉字串文件的实现方法是在 BASIC 或 C 等高级语言下,先显示一串汉字,然后取出此串汉字的点阵坐标,存于汉字串文件中,该汉字串文件就可使用了。

此方法在 AST 386 等微机上运行通过。

王斌

3. 用 C 语言实现的文本文件阅读器

为了对文件进行归类或要查询某一个文件,经常需要阅读各类文本文件,目前尚未见到阅读文本文件的专用软件,比较流行的是采用 Borland 公司的 Turbo 系列软件上所附的文本阅读器软件 README.COM,或 MS DOS 系统提供的 TYPE 命令实现。前者是为西文系统设计的阅读软件,是比较完善的文本阅读器,具有上下翻行、翻页、打印某页、显示移到文件头和字符搜索等功能;后者可用于文本文件的显示,但功能极弱,不能适应日常工作中文本文件的显示与阅读。由于 README.COM 是针对西文的 ASCII 码文件设计的,将其应用于中文文件的阅读时存在许多问题,尽管国内近来对 README.COM 的汉化做了一些工作,但由于汉化工作是面对可执行文件而非源程序进行,因此,始终存在一些难以解决的问题。主要是以下三个问题:

①不支持带通配符“?”或“*”的文件名,一次只能对一个文件进行阅读,而不能对一批文件名有规律的文本文件进行依次逐个阅读。

②难以辨识出文件中的各种控制符(在各类文字处理系统建立的中文文件中,含有各种用于分页、分行、字体选择和字型变化的控制符,这些控制符的 ASCII 码值通常介于 80H—A0H 之间,在阅读时,西文软件将其作为扩展 ASCII 码中的图形符显示)。由于受文本文件中控制符的影响,极易使显示的文件发生混乱。

③不能正常显示用 SUPER WPS 系统的 D 命令编辑的文书文件。

鉴于汉化 README.COM 存在诸多不足,因此用户只好使用文字处理系统来阅读文件,但该方法又因频繁进入与退出文字处理系统,不但需要频繁的读写磁盘操作,而且需不断键入新的文件名才能完成多个文件的查询与阅读。为此,用 Turbo C 2.0 编制了一个程序(READMEFILE.C),它可以实现文本阅读的基本功能。将该程序与文字处理系统结合起来,可以完成一个增强的文本阅读功能,即将 READMEFILE.C 程序用于文件的搜索、查询与阅读,而将文字处理系统作为搜索某一字符串、打印文件的某一页和修改文件,互相取长补短,使功能倍增。

1. 程序的功能

READMEFILE 程序具有如下功能:

①支持带通配符“?”、“*”的文件名,实现了用通配符一次阅读一批文件的功能。程序根据输入的文件名(可含盘符、路径和文件名)自动在指定范围内搜索满足条件的文件,并依次逐个读入内存,在屏幕上显示出来供用户阅读。

②在显示完文件的第一屏内容(22 行文本)后,屏幕底部出现提示信息 and 命令行,此时可键入下列命令实现阅读文件所需的不同功能:

PGUP:向前翻一屏,即重新显示前一屏的内容。若当前显示的是文件第一屏的内容,向前翻屏仍显示原内容。

F:将文件指针移到文件开头,即重新显示文件第一屏的内容。

E:当前显示的文件结束,开始显示下一个满足条件的文件的第一屏内容。

ESC:结束文件的阅读,返回 DOS 状态。

其它任意键:继续显示文件下一屏的内容。

③可以正常显示 SUPER WPS 的文书文件与非文书文件。程序通过判别文件头是否为“01FF”,以区别 WPS 系统的二类文件。若判断文件为文书文件,则自动将文件的读写指针移到从文件头开始偏移地址 1024 的位置(即文件内容开始存放的位置),以避免文件头。若为非文书文件,则从文件头开始显示。

④解决了汉字的显示问题,杜绝了将一个汉字分别显示在二行的情况。该项功能是通过判断用双字节(其机内码均大于 A0H)表示的汉字的前一字节的显示位置,使其不显示在屏幕的第 79 列(屏幕显示列为 0~79)来实现的。

⑤程序可滤掉原文件中含有的字型控制符(ASCII 码大于 7FH 小于 A0H)、分页符(ASCII 码为 0C0AH)、硬回车(ASCII 码为 0D0AH)和软回车(ASCII 码为 8D8AH)等特殊意义的符号,并将后三种符号作为统计显示行数的标准。由于成功地滤掉了文件中的控制符,从而彻底地解决了由这些符号引起的汉字误组合使显示文件内容混乱的问题。

⑥可由用户根据需要指定在屏幕上显示文件的行数,实现了部分文件内容查询的功能。若指定的显示行数小于屏幕允许的显示行数,则将不停地依次显示满足条件的文件的内容。

2. 程序的使用

该程序可放在硬盘根目录下作为 DOS 的外部命令使用,也可以在 SPDOS、2.13 和 UC DOS 等中文系统下使用,它具有比 TYPE 命令强大得多的功能。另外,尽管该程序是针对 SUPER WPS 系统下文件的显示和查询,但由于解决了汉字文件显示中的常见问题,也完全适用于其它系统文本文件的显示,只是因为各文字处理系统的分页符、软回车和硬回车的编码可能存在差异,使屏幕显示的行数与要求的行数有一定的误差。

程序可以按以下方式使用:

①C>READFILE 文件名 行数<回车>

②C>READFILE 文件名 <回车>

③C>READFILE <回车>

后两种使用方式不指定显示文件的行数,则隐含为显示文件的全部内容。第一种使用方式指定显示行数,则每个文件显示完指定的行数后就自动结束,系统将继续显示下一个满足条件的文件。另外,程序还能根据是否输入了文件名而进行提示。文件名中可以包括驱动器、路径和通配符“*”和“?”,此时为一批文件的阅读。

程序在 AST 386SX/20 微机上调试通过,也适用于其它型号的微机,程序编译的软件环境为 Turbo C 2.0。

程序清单

```
# include "stdio.h"
# include "dos.h"
# include "dir.h"
# include "ctype.h"
```



```

main(argc,argv)
int argc;
char * argv[];
{
    char ch1,ch3,dir[80];
    FILE * fp;
    struct ffblk f;
    long int ll[50];
    int i=0,i1=0,ch,k1=0,k2=0,k3=0,l1,l2,p1,ch2;
    printf("\n 中西文文本文件阅读器\n");
    printf(" 版本 V1.0\n");
    if (argc<2)
    {
        printf("\n 请输入文件名,");
        scanf("%s",argv[1]);
        ch2=getchar();
    }
    if (argc==3) p1=atoi(argv[2]);
    else
    p1=1000;
    getcwd(dir,80);
    if ((dir[0] != argv[1][0] && (argv[1][1] != '\0'))
    {
        ch3=tolower(argv[1][0] - 'a');
        setdisk(ch3);
    }
    l1=findfirst(argv[1],&f,0);
    if (l1<0) printf("文件名输入错,磁盘上无此文件\n");
    while(! l1)
    {
        k1=0;
        k3=0;
        if(((fp=fopen(f.ff-name,"rb"))=NULL)
        {
            printf("文件 f.ff-name 不能打开! \n");
            exit(-1);
        }
        printf("文件 %s 的内容为:\n",f.ff-name);
        ll[0]=0;
        if (((getc(fp)==0X01) && (getc(fp) ==0xff)))
        {
            fseek(fp,1024L,0);
            ll[0]=ftell(fp);
            ch1='y';

```