第一部分　DOS、CCDOS 操作使用

1.01　在 WORDSTAR 中重新定义 Home 和 End 两键

江苏省大丰县委办公室（224100）陈上权

Home 和 End 两键位于键盘右边，在 WORDSTAR 中分别表示（Ctrl＋Q＋E）光标移至屏顶、(Ctrl＋Q＋X）光标移到屏底；而在金山汉卡等一些汉字处理软件中，将这两键分别定义为光标移至行首和光标移至行尾；这样给修改文稿带来了极大的方便。修改 WORDSTAR 中 Home 和 End 两键的定义，其具体过程如下：

```
A>DEBUG WS.COM
-E4BD
682A：04BD  11.11  05.13        ；将 05（Home）改为 13（Ctrl＋Q＋S 将光标移至行首）
-E6F8
682A：06F8  11.11  58.04        ；将 58（End）改为 04（Ctrl＋Q＋D 光标移至行尾）
-W
-Q
A>
```

完成上述修改之后，再使用 WORDSTAR 的 Home 和 End 两键，修改文稿就方便多了。（在按上文修改时，注意 WS 版本，或注意检验横线标注的值。——编者）

（1992 年第 19 期第 99 版）

1.02　巧用 CHKDSK 命令一例

湖南长沙铁路分局勘察设计院（410001）胡　卓

在 C 盘上运行一个 BASIC 程序时，碰到这样的故障现象：在 BASIC 状态，程序刚一运行就显示：Disk Media Error（磁盘介质错误）。屡次将软盘中程序拷入 C 盘再试，出现同样的信息。根据 BASIC 手册的解释，这种错误信息表示原来的磁盘坏了，只能重新格式化坏磁盘，若格式化失败，磁盘只好报废；因不愿格式化 C 盘，于是怀疑是否有病毒，用 SCAN3.0 检测 C 盘，检测到某个文件时，中断显示：

Read found error device C

Abort，Retry，Ignore，Fail?

检测内存无病毒，据此分析，可能是 C 盘中程序文件太破碎。应用 DOS 中 CHKDSK/

F 命令，改正目录或文件分配表中的错误，将软盘中程序拷入 C 盘再试，故障排除，运行正常。由此可见有些故障外表上是硬故障，但实际却是软故障，而这时 CHKDSK 将提供您很多的帮助。

（上文故障一般是由于病毒或非正常关机造成文件分配簇链被破坏，其值指向一个非物理存在的扇区所致。——编者）

（1992 年第 37 期 131 版）

1.03 为 MS-DOS 增加修改子目录名命令

广西柳州市统计局计算站（545001）曾庆涛

众所周知，MS-DOS 有一个修改文件名的命令 RENAME，但没有修改子目录名的命令，这给许多工作带来不便。虽然使用某些软件（如 PCTOOLS 等）可以修改子目录名，但是在日常使用中显得繁琐。为此，笔者用汇编语言编写了一个修改子目录名的程序 RENDIR. ASM，该程序经汇编、连接，再经 EXE2BIN 命令转换为 RENDIR. COM，即可作为 MS-DOS 的一个外部命令使用，格式为：

[D:] [PATH] RENDIR [d:] subdirectory1 subdirectory2

这样就可把第一个子目录名修改为第二个子目录名。

RENDIR. ASM 程序清单如下：

```
        TITLE RENDIR.ASM
CODE    SEGMENT
        ORG     05CH
FILE1   DB      12 DUP(?)
        ORG     06DH
FILE2   DB      11 DUP(?)
        ORG     100H
        ASSUME  CS:CODE,DS:CODE
BEGIN:  MOV     DI,OFFSET FCB1
        MOV     SI,OFFSET FILE1
        MOV     CX,0CH
        CLD
        REPZ    MOVSB
        ADD     DI,5
        MOV     SI,OFFSET FILE2
        CMP     BYTE PTR [SI],20H
        JNZ     LOOP
        MOV     AH,9
        MOV     DX,OFFSET PEMSG
        INT     21H
        JMP     EXIT
```

```
LOOP:   MOV     CX,0BH
        CLD
        REPZ    MOVSB
        MOV     DX,OFFSET FCB
        MOV     AH,17H
        INT     21H
        CMP     AL,0
        JZ      EXIT
        MOV     AH,9
        MOV     DX,OFFSET NCMSG
        INT     21H
EXIT:   INT     20H
PEMSG   DB      'Parameter error'
        DB      13,10,'$'
NCMSG   DB      'Name not changed'
        DB      13,10,'$'
FCB     DB      0FFH,5 DUP(?),10H
FCB1    DB      28 DUP(?)
CODE    ENDS
        END     BEGIN
```

(1992 年第 7 期第 59 版)

1.04 妙用 ASSIGN 命令

青海省医药公司计算机室 (810007) 黄文平

在使用印刷电路板设计软件 ORCAD 时，每次运行都要在 B 驱调用库文件，速度很慢，对于现在 286 以上的微机，都有扩展内存，设置一个虚拟磁盘，用 ASSIGN 命令将在 B 驱的操作改为对虚拟磁盘的操作，运行速度大大加快。具体作法是：

1. 在系统配置文件 CONFIG.SYS 设置虚拟盘：

   DEVICE=ANSI.SYS

   DEVICE=RAMDRIVE.SYS 384 512 64/E

2. 建立一个批处理文件：

   D:\ORCAD>TYPE ORCAD.BAT

   ECHO OFFECHO() The *.lib file will work at RAM-Disk

   PAUS

   EASSIGN B=E

   COPY *.LIB B:

   DRAFT

   ASSIGN(要根据系统分配的虚盘符号修改上面的 ORCAD.BAT 文件。)

(1992 年第 45 期第 133 版)

1.05 一个文件“虚”拷贝程序

浙江省兴钢铁厂计算机室（312000）马宏兴

PC-DOS 单用户操作系统下的文件共享，往往通过 PATH、APPEND 命令来实现。但是，在某些情况下，这种方法并不奏效。笔者介绍一个用 TURBO BASIC 语言编写的文件拷贝程序。运行此程序后，在目标处并不真正形成一个文件拷贝，而只是生成一个目录项，其中的起始簇号和文件长度项均与共享文件的相同，故称之为文件“虚拷贝”程序。

程序分二部分：第一部分主要是从命令行中截取源文件和目标文件的路径文件名参数，并建立相应的目标文件。这部分功能由过程 Commandline、OpenFile 实现。第二部分是从源文件和目标文件的路径文件名参数中分离驱动器符、路径名和文件名，由此构造出源文件、目标文件的 FCB，并根据各自的路径名分别进入相应的新目录。这部分功能由过程 ProcessF 和 FNMakeFCB$ 实现。第三部分由过程 RSFCB 和 RWFCB 构成，主要利用 DOS 的 0FH 功能调用分别打开源文件、目标文件，并修改目标文件的 FCB 中的文件长度、文件起始簇号项，使其与源文件 FCB 中的相同，同时置目标文件 FCB 中的设备属性字节的第六位为 1，然后利用 DOS 的 INT21H 的 10H 功能调用关闭文件。这样，目标文件的目录项即被更新，其中的文件长度、文件的起始簇号项与源文件的相同。

程序中使用了 DOS 保留项，因此对 DOS 版本有一定的要求，本程序运行于 DOS 3.30。

程序清单如下：

```
%DS=8:%DX=4:%AX=1
CALL Commandline
CALL Openfile
CALL ProcessF(SFile$,SDriver%)
SFCB$=FNMake FCB$(SFile$,SDriver%)
CALL RSFCB
CALL ProcessF(Dfile$,DDriver%)
DFCB$=FNMakeFCB$(DFile$,DDriver%)
CALL RWDFCB
END
SUB Commandline
  SHARED SFile$,dfILE$
  LOCAL CParameter$,i%,CR$,V%,F%,Ver%
  CR$=CHR$(13)+CHR$(7):F%=0
  REG %AX,&H3000
  CALL INTERRUPT &H21:V%=REG(%AX)
  Ver%=(V% AND &HOOFF)*100+(V% AND &HFF00)/256
  IF (Ver%<>500 AND Ver%<>330 AND Ver%<>331) THEN
    PRINT CR$+"Incorrect DOS Version !":STOP
END IF
CPara$=COMMAND$+" "
```

```
SFile$ =MID$ (CPara$ ,1,INSTR(CPara$ ," ")-1)
CPara$ =MID$ (CPara$ ,INSTR(CPara$ ,"")+""
FOR i%=1 TO LEN(CPara$ )
  IF MID$ (CPara$ ,i%,1)<>"" THEN
    DFile$ =MID$ (CPara$ ,i%,INSTR(I%,CPara$ ,"")-i%)
    EXIT FOR
  END IF
NEXT i%
IF (LEN(SFile$ )=0 OR LEN(Dfile$ )=0 OR INSTR(1,DFile$ ,":")<>0) THEN
    PRINT CR$ +"Usage:VCOPY [d:][path]<SourceFile> [path]<TargetFile>"
    STOP
END IF
IF (INSTR(1,DFile$ ,"\")=0 AND INSTR(1,SFile$ ,"\")<>0) THEN
    FOR i%=LEN(SFile$ ) to 1 step -1
    IF MID$ (SFILES$ ,I%,1)="\" THEN
      DFILE$ =LEFT$ (SFILE$ ,I%)+DFILE$
      EXIT FOR
    END IF
  NEXT I%
END IF
IF (INSTR(1,SFile$ ,":")<>0 AND INSTR(1,DFile$ ,":")=0) THEN
    DFile$ =LEFT$ (SFile$ ,2)+dfILE$
  END IF
END SUB
SUB ProcessF(FileName$ ,Driver%)
  LOCAL I%,Flag%
  Flag%=1
  IF MID$ (FileName$ ,,2,1)=":" THEN
    Driver%=((ASC(UCASE$ (LEFT$ (FileName$ ,1)))-&H40) AND &H00FF)
  ELSE
    Driver%=0 :Flag%=0
  END IF
  IF INSTR(FileName$ ,"\")<>0 THEN
    FOR i%=LEN(FileName$ ) TO 1 STEP -1
          FOR i%=LEN(FileName$ ) TO 1 STEP -1
        IF (MID$ (FileName$ ,i%,1))="\"THEN
          IF (i%=3) OR (i%=1) THEN
            CHDIR LEFT$ (FileName$ ,i%)
          ELSE
            CHDIR MID$ (FileName$ ,1,i%-1)
          END IF
          FileName$ =MID$ (FileName$ ,i%+1)
          EXIT FOR
```


试读结束：需要全本请在线购买： www.ert

```
            END IF
            EXIT i%
          ELSE
            IF Flag%=1 THEN
              FileName$=MID$(FileName$,3)
            END IF
          END IF
        END SUB
        SUB OpenFile
          SHARED SFile$,DFile$
          LOCAL S$,CR$
          CR$=CHR$(13)+CHR$(7)
          ON ERROR GOTO DispErr1
          OPEN SFile$ FOR INTUP AS #1:CLOSE #1
          ON ERROR GOTO CreatFile
          OPEN DFile$ FOR INPUT AS #1:CLOSE #1
          PRINT CR$+"File:<"+Dfile$+"> is exist,Overwrite? (Y/N)"
          S$=INKEY$
          WHILE LEN(S$)=0:S$=INKEY$:WEND
          IF UCASE$(S$)="Y" THEN
            EXIT SUB
          END IF
          STOP
          Creatit:
          ON ERROR GOTO DispErr2
          OPEN DFile$ FOR output as #1:CLOSE #1
          EXIT SUB
          CreatFile:
          RESUME Creatit
          DispErr2:
          PRINT CR$+"File:<"+DFile$+"> can't be created !":STOP
          DispErr1:
          PRINT CR$+"File:<"+SFile$+"> not found !":stop
        END SUB
        DEF FNMakeFCB$(FileName$,Driver%)
          LOCAL P%
          IF INSTR(FileName$,".")=0 THEN
            Filename$=left$(Filename$+Space$(11),11)
          ELSE
            P%=INSTR(Filename$,".")
            FileName$=LEFT$(FileName$,P%-1)+SPACE$(8),8)+LEFT$(MID$(File-
Name$,P%+1)+SPACE$(3),3)
          END IF
```

```
  FNMakeFCB$=CHR$(Driver%)+FileName$+STRING$(25,0)
END DEF
SUB RSFCB
  SHARED Fc$,Fl$,SFCB$
  REG %DS,PEEK(0)+(256*PEEK(1)):DEF SEG=VARSEG(SFCB$)
  REG %DX,PEEK(VARPTR(SFCB$)+2)+256*PEEK(VARPTR(SFCB$)-3)
  REG %AX,&H0F00
  CALL INTERRUPT &H21
  REG %AX,&H1000
  CALL INTERRUPT &H21
  DEF SEG:Fc$=MID$(SFCB$,&H1C,2):Fl$=MID$(SFCB$,&H11,4)
END SUB
SUB RWDFCB
  SHARED Fc$,Fl$,DFCB$
  REG %DS,PEEK(0)+256*PEEK(1):DEF SEG=VARSEG(DFCB$)
  REG %DX,PEEK(VARPTR(DFCB$)+2)+256*PEEK(VARPTR(DFCB$)+3)
  REG %AX,&H0F00
  CALL INTERRUPT &H21
  DEF SEG:MID$(DFCB$,&H1C,2)=Fc$:MID$(DFCB$,&H11,4)=Fl$
MID$(DFCB$,&H1B,1)=CHR$(ASC(MID$(DFCB$,&H1B,1))XOR&H0040)
  DEF SEG=VARSEG(DFCB$)
  REG %AX,&H1000
  CALL INTERRUPT &H21
  DEF SEG
END SUB
```

(1992 年第 44 期第 131 版)

1.06 文件目录显示功能改进一则

黄石市统计局 (435000) 杨斌

通常我们用 DIR 命令来查看文件目录，并根据不同需要给 DIR 命令带上“/W”、“/P”等参数，如用“DIR/W”可横向排列文件、子目录以便于查看众多的文件或目录。但由于子目录无任何后缀标志，与不带后缀的文件名很难区别开，给查看者带来不便。另外 DIR/W 命令未提供专门查看子目录的功能，不能将子目录与文件分开来显示，这也给某些用户带来不便。为此，笔者用 TURBO C2.0 编制了一个名为 DD.EXE 的程序，实现了上述功能。在显示时，子目录名用“[ ]”括起，如带上“D”或“F”参数可分别显示子目录或文件。

其命令格式为：

DD <DRIVE：> [D/F]

DRIVE 是指驱动器名，它可取 A～C；D，显示目录；F，显示文件。

在 AUTOEXEC.BAT 文件中加上一行 PATH=C:\; 然后将 DD.EXE 拷入 C:\中，重新启动机器，即可象使用 DIR/W 一样运行 DD.EXE。

该程序在 HP386、AST386、AST286、SUPER286、LC0520、IBM PC/XT、M24 等机型上运行通过。

DD.C 程序清单如下：

```
#include "stdio.h"
#include "dir.h"
#include "dos.h"
#include "io.h"
#include "string.h"
struct ffblk f;
int done;
char path[MAXDIR];
main(argc,argv)
int argc;
char *argv[];
{
int i=1,j=0,k=0;
int drive;
char mid[MAXDIR];
if (argc! =1)
{
  if((argc==3)&&strcmp(argv[2],"D")&&strcmp(argv[2],"d")&&strcmp(arg v[2],"f")
&&strcmp(argv[2],"F"))
    {printf(" Usage:dd [drive:] [d/f]\n");exit(0);}
  if(! strcmp(argv[1],"a:")||! strcmp(argv[1],"A:")) drive=0;
  else {
    if(! strcmp(argv[1],"b:")||! strcmp(argv[1],"B:")) drive=1;
    else {
      if(! strcmp(argv[1],"C:")||! strcmp(argv[1],"C:")) drive=2;
      else {printf(" Invatid parameters.\n");exit(0);}
      }
    }
  setdisk(drive);
  getcwd(path,MAXDIR);
  printf("[%s]\n",path);
  done=findfirst("*.*",&f,FA_RDONLY|FA_HIDDEN|FA_DIREC|FA_SYSTEM|FA_
ARCH);
  WHILE(! DONE){
    if(argc! =3){
      if(f.ff_attrib==FA_DIREC)
        {strcpy(mid,f.ff_name);strcat(mid,"]");printf("[%-11s",mid);++j;}
```

```
        else {printf(" %-12s",f.ff_name);++k;}++i;
        }
      else
        {

if((! stremp(argv[2],"d")||! stremp(argv[2],"D"))&&(f.ff_attrib==FA_DIREC))
        {strcpy(mid,f.ff_name);streat(mid,"]");printf(" [%-11s",mid);
        ++j;++i;
      }

if((! stremp(argv[2],"f")||! stremp(argv[2],"F"))&&(f.ff_attrib! =FA_DIREC))
        {printf(" %-12s",f.ff_name);
        ++k;++i;
        }
      }
      if(i>5){
      printf("\n");
      i=1;
      }
      done=findnext(&f);
      }
    printf("\n");
    if(argc! =3){
      printf("        %d subdirectories\n",j);
      printf("        %d file(s)\n",k);
    }
    else{
      if(! stremp(argv[2],"d")||! stremp(argv[2],"D"))printf(" %d subdirectories\n",j);
      if(! stremp(argv[2],"f")||! stremp(argv[2],"F"))printf(" %d file(s)\n",k);
      }
    drive=2;setdisk(drive);        }
```

(1992 年第 45 期第 135 版)

1.07 一条语句实现 DOS shell

四川永川 1102 信箱七室 (632163) 魏东

许多流行的软件如 TURBO 系列、ACAD 等都具有 DOS shell 功能，即可不退出程序，运行 DOS 命令，如 dir、copy 等。这大大方便了用户，增加了软件的实用性。

这里提供一个在 TURBO C 中实现 DOS shell 的简捷方法，只需多加一条 system ("") 语句。注意这里函数 system 中的参数是空串。这是 TURBO C 参考手册中未公布的功能，

但在 TURBO PROLOG 中有这种用法。它的实质是调入了第二个命令解释程序 COMMAND.COM，所以从 shell 中返回原程序只需键入 Exit。

文中提供了一个示范程序，在 TURBO C V2.0 中编译通过。笔者在编写的 PC 机仿真 SUN/4110 终端的程序中应用此方法效果良好。

```
/*
            PROG SHELL.C 1993
            USE SYSTEM("")
*/
#include <stdio.h>
#include <dos.h>
main(){
    printf("\n WELCOME USE DEMO SHELL press any key to DOS SHELL\n");
getch();
printf("\n NOW GO TO SHELL press EXIT to return");
system("");
printf("\n WELCOME YOU BACK FROM SHELL ");
}
```

1.08 浅谈 PATH、APPEND 及 FASTOPEN 的综合运用

开滦矿务局基建公司安装处微机室(063100) 杨兆明

在执行带有 .EXE、.COM 和 .BAT 后缀的文件时，通常是先进入该文件所在目录，然后再执行。此时若想执行其他目录的其他文件，需退出当前目录进入要执行的文件所在子目录中，这种方法很繁琐，若在每一用户目录下均放各种应用软件，则在硬盘上各用户编辑的文件与系统实用软件的文件混合存放，寻找起来很不方便，容易与应用软件的文件发生混淆，而且还会引起系统运行效率降低。这里介绍一种简便易行的方法，它解决了单用户、单任务操作系统 DOS 不具备多用户管理功能的问题，即在任意目录下可运行任何文件。这样，不仅可以节省硬盘空间，减少在硬盘上保留同一软件的多个拷贝的文件数目，而且还能提高系统的运行效率。

具体办法是，利用 PATH 和 APPEND 命令编辑一个批处理文件，放入 DOS 子目录中，然后用 FASTOPEN 命令建立文件名的内存缓冲区，存储从前曾经打开过的文件和路径在磁盘中所处的位置，以减少以后再次打开这些文件时所花的时间。

当执行一个文件需要系统搜索路径，FASTOPEN 命令能够减少这些额外的查找时间，可使程序的加载速度加快，有的批处理文件能更快的执行；同时能使包含很多目录的大容量硬盘使用效率更高。

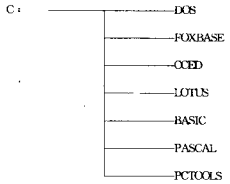
将 FASTOPEN 命令加在 AUTOEXEC.BAT 文件中。其格式为：

```
PATH C:\DOS;
FASTOPEN driver:[=entries] […]
```

其中：driver，指明要建立文件名缓冲区的驱动器(1～24 硬盘驱动器)；Eentries，指定驱

动器在内存中保留的文件名和目录名的个数(范围从1～999)。命令中参数的最佳值随用户的硬盘大小、内存情况及想要保留的文件名和目录名的个数的不同而定。

设在机器的C盘根目录下,有下列几个子目录:



如果在执行完FOXBASE,欲马上执行CCED,只需编辑一个调用CCED的文件,就能满足上述要求,即在任意目录执行任意文件。

其文件内容如下:

```
*CE.BAT
    @ECHO OFF
    PATH C:\DOS;C:\CCED;
    APPEND C:\CE;
    C:\CCED\CCED %1
    APPEND;
    PATH C:\DOS;
    ECHO ON
```

产生的CE.BAT批文件存放到DOS子目录中,在任何情况下,只要键入CE,在任何目录内部都可调用CCED,而不需要退出当前目录,如同欲执行的文件就在当前目录一样。同样,在任何目录下皆可任意调用象FOXBASE、LOTUS1-2-3、BASIC、PASCAL、PCTOOLS等应用程序,又如调用FOXBASE文件如下:

```
*FB.BAT
    @ECHO OFF
    PATH C:\DOS;C:\FOXBASE;
    APPEND C:\FB;
    C:\FOXBASE\FOXPLUS %1 %2
    APPEND;
    PATH C:\DOS;
    ECHO ON
```

其他文件也是如此这里就不在一一赘述了。只要将批文件在硬盘上安装一次,就可实现磁盘管理井然有序,减少繁琐的操作,节省时间,提高效率。

本程序已在AST-286、386、486、COMPAQ-386等机器及MS-DOS3.3、MS-DOS 5.0操作系统上运行通过。

(1992年第14期第103版)

试读结束:需要全本请在线购买: www.ert

1.09 用 ANSI.SYS 扩展 DOS 的屏幕和键盘

无锡市 35 信箱 31 分箱(214083) 毛卓振

ANSI.SYS 是 DOS 操作系统 2.0 以上版本提供的可安装的设备驱动程序，该程序让用户使用 ANSI 换码序列。ANSI 换码序列是由美国国家标准学会所规定的字符序列，它对屏幕显示和键盘的控制提供了扩展功能。开发应用软件时，可在程序中用规定的 ANSI 换码字符串序列来控制显示屏幕上光标的位置和颜色，也可对键盘的各个键重新进行定义(包括定义 DOS 的功能键等)，使用起来十分方便灵活。请注意，ANSI 换码序列中的控制字符串序列只对 DOS 功能调用 1,2,6,9 有效。

ANSI 换码序列提供了很多种规定的控制字符，高版本 DOS 功能更强些。它主要是用于屏幕设置和键盘定义，使用十分简单，下面举几个非常实用的例子来说明问题。

1. 设置屏幕光标位置

ESC[2J　　清屏幕；

ESC[#；#H　设置光标位置，#为十进制数的参数，前面的#为行号，后面的#为列号；

ESC[#A　光标在同列内上移#行，如果光标已经在顶行，DOS 就忽略此序列；

ESC[#B　光标在同列内下移#行，如果光标已经在底行，DOS 就忽略此序列；

ESC[#C　光标在同行内右移#列，如果光标已经在最右，DOS 就忽略此序列；

ESC[#D　光标在同行内左移#列，如果光标已经在最左，DOS 就忽略此序列；

ESC[K　　清行，从光标位置一直清到行尾；

ESC[6n　　读光标当前位置，并以 ESC[#.#]形式显示。

2. 设置显示器方式和屏幕颜色

ESC[=#h　，显示器方式，#号及对应显示方式如下：

0	40x25	黑白	4	320x200	彩色
1	40x25	彩色	5	320x200	黑白
2	80x25	黑白	6	640x200	黑白
3	80x25	彩色			

ESC[#；…；#m，屏幕颜色、#号及对应颜色和特性如下：

图形功能		前景颜色		背景颜色	
0	关闭属性	30	黑	40	黑
1	显示粗线	31	红	41	红
5	闪烁	32	绿	42	绿
7	屏幕反相	33	黄	43	黄
8	隐含	34	蓝	44	蓝
		35	粉红	45	粉红
		36	青	46	青
		37	白	47	白

3. 扩展键盘功能

ESC[#；#p

或　　ESC['string';'string'p

或　　ESC[#;'string';#p

前面的#号是指定要定义的字符键的ASCII码或字符，后面的#号或字符串string是将要定义成的字符ASCII码或内容。

一般用得较多的是对不常用的功能键进行定义，定义一些常用的DOS命令或用户自己的命令或需要经常从键盘敲入的字符串等，如可以定义F8为DIR/P，F9为PCTOOLS等。常用功能键的组合代码如表1-1所示。

在DEBUG下输入如下程序：

```
>DEBUG
-A100
136B:0100 MOV DX,114
136B:0103 MOV AH,A
136B:0105 INT 21
136B:0107 MOV AL,1B
136B:0109 MOV [115],AL
136B:010C MOV DX,115
136B:010F MOV AH,9
136B:0111 INT 21
136B:0113 INT 3
136B:0114 DB 30,0
136B:0116
-G=100
```

然后从键盘上输入控制序列(ESC程序中设置，不必再输入)：

[2J $ <CR>　　清屏；

或[15;15H $ <CR>　　光标定位在15行15列；

表1-1　常用功能键的组合代码表(十进制)

键名	单键	Shift	Ctrl	Alt
F1	0,59	0,84	0,94	0,104
F2	0,60	0,85	0,95	0,105
F3	0,61	0,86	0,96	0,106
F4	0,62	0,87	0,97	0,107
F5	0,63	0,88	0,98	0,108
F6	0,64	0,89	0,99	0,109
F7	0,65	0,90	0,100	0,110
F8	0,66	0,91	0,101	0,111
F9	0,67	0,92	0,102	0,112
F10	0,68	0,93	0,103	0,113
F11	0,133	0,135	0,137	0,139
F12	0,134	0,136	0,138	0,140

或[7m $ <CR>	屏幕反相；
或[31;47m $ <CR>	前景红色，背景白色；
或[=4h $ <CR>	显示器 320x200 彩色方式；
或['A';'B'p $ <CR>	把 A 键定义成 B 键；
或[66;'A'p $ <CR>	把 B 键定义成 A 键；
或[0;68;'dir/p';13p $ <CR>	把 F10 键定义成 dir/p<CR>；
或[0;133;'pcshell';13p $ <CR>	把 F11 键定义成 pcshell<CR>；
或[0;134;'gwbasic';13p $ <CR>	把 F12 键定义成 gwbasic<CR>。

其他按需要可用类似上面的方法进行定义，也可不用键盘输入，在程序中直接定义请使用功能调用 9 。

（1992 年第 31 期第 105 版）

1.10 如何整理屏幕帮助信息

水电部六局技术处(118216) 陈玉山

目前许多单位都拥有不少 IBM PC 及其兼容机的系统软件，一方面由于缺少说明书而不能正常使用或只能使用其中的一些基本功能；另一方面，这些软件在启动后都有着丰富的联机帮助信息。如果能将这些帮助信息整理出来，那么就能为使用这些软件提供很大的帮助。

一般而言，这些软件的帮助文件是不能直接或经简单处理而在打印机上打印出来的。为此，本人想出一个办法，就是当屏幕上出现帮助信息时，通过简单方法把屏幕信息移到一个文本文件中去，每屏信息对应一个文本文件，再把相关信息的文本文件合并成一个文本文件，用文本编辑程序进行编辑，编辑完毕即可打印出一本帮助手册。

当屏幕出现所需信息时，如能简单地通过按键来实现屏幕信息的转储则是很方便的。显然，通过修改 INT5 来实现这个功能较为适宜。这样，需要修改 INT5 的入口地址，使之指向实现上述功能的中断服务程序。当需要的信息在屏幕上出现时，按下屏幕硬拷贝键即可实现屏幕信息的转储。

IBM PC 机 CGA 显示控制器有字符和图形两种显示方式。在图形显示方式下，显示缓冲区中的每 1 个、2 个或 4 个二进位，分别与高分辨率、中分辨率和低分辨率的每个象素相对应，因而显示缓冲区的内容是不能用上述方法进行转储而得到对应信息的文本文件的；而在字符显示方式下，屏幕上的每个字符均占 2 个字节：一个是字符的 ASCII 码，另一个是字符的显示属性。所以，在字符显示方式下通过处理是能够实现上述功能的。

实现原理为：显示缓冲区是从 B800:0H 开始的 16KB RAM。在字符显示方式下，有 80×25 和 40×25 两种显示方式。在 80×25 方式下，每屏信息需占 80×2×25=4000 字节(约为 4KB)，这样显示缓冲区共可容纳 4 屏信息，每屏算作一页，页号为 0～3；在 40×25 方式下，每屏信息需占 40×2×25=2000 字节，显示缓冲区共可容纳 8 页信息，页号 0～7。由于大多数软件都是使用第 0 页，而且 80×25 方式第 0 页的内存范围覆盖了 40×25 方式第 0

页的内存范围，所以仅需考虑 80×25 方式第 0 页的转储即可。

在 80×25 显示方式下，每屏共显示字符数为 80×25=2000 个，每屏信息占用显示缓冲区 2×2000=4000 字节。显示缓冲区首址为 B800:0H，屏幕上字符的 ASCII 码及字符的属性，从左到右、从上到下地在显示缓冲区中顺序存储。因此，只要把 B800:0H 开始的 4000 个字节进行整理，包括压缩掉每个字符的属性字节，为每行加回车换行符，再写到相应文本文件中去即可。显然，每个屏幕文本文件的长度均为 2000+25×2=2050 字节。

汇编语言源程序为 NEWINT5.ASM。经编译并转换后产生 NEWINT5.COM 即可执行。程序中还对存取显示缓冲区引起的 CPU 与视频控制器争用内存，使 CRT 上出现"雪花"问题进行了相应的处理。

使用时，先在 DOS 状态下执行 NEWINT5.COM。这样，该程序中的中断服务程序取代屏幕硬拷贝中断服务程序而驻留内存，然后启动相应软件。每当屏幕上出现要转储信息时，按下屏幕硬拷贝键，这时执行的将是把屏幕缓冲区内容整理并存到磁盘文件中去的功能。在 NEWINT5.COM 驻留内存后，第一次按下屏幕拷贝键生成的文件名为 INFOR.01，第二次为 INFOR.02，……，第九十九次为 INFOR.99。这些文件均放在 C 盘的 MSG 子目录下，因此，使用本程序之前，应在 C 盘上先建一个名为 MSG 的子目录。最后将生成的 INFOR.××文件进行适当的合并，再用文本编辑程序进行编辑，就可以打印出帮助手册。

本程序在 GW0520A 机上调试通过。通过对 TURBO PASCAL 4.0 程序的验证，效果良好。

程序清单如下：

```
CODE    SEGMENT
        ASSUME CS:CODE,DS:CODE,ES:CODE
        ORG 100H
START:  JMP BEGIN
NEWINT5 PROC FAR
        JMP SSS
ENVSEG  DW  0                   ;驻留程序环境块段地址
INT_5   EQU THIS DWORD          ;存原 INT5 向量
INT5    DW  0
INT5SG  DW  0
TRSSEG  DW  0                   ;驻留标志
TEN     DB  10
FILE    DB "C:\MSG\INFOR.XX",0  ;文件名
HANDLE  DW  0                   ;文件指针
COUNT   DB  0                   ;计数器
BUFFER  DB  4000 DUP(' ')       ;缓冲区
SSS:    CLI                     ;关中断
        PUSH AX                 ;保护现场
        PUSH BX
        PUSH CX
        PUSH DX
        PUSH SI
```

```
        PUSH DI
        PUSH DS
        PUSH ES
        PUSH BP
        PUSH CS
        POP DS
        INC COUNT                ;计数加1(初值为0)
        NOV AL,COUNT
        CMP AL,100
        JAE RETU
        MOV AH,0
        DIV TEN
        ADD AL,30H
        ADD AH,30H
        MOV [FILE+13),AL         ;产生相应序号的文件名
        MOV [FILE+14),AL
        MOV AX,0B800H            ;传送屏幕缓冲区内容到
        MOV DS,AX                ;本程序缓冲区
        MOV SI,0
        MOV CX,2000
        PUSH CS
        POP ES
        MOV DI,OFFSET BUFFER
        CLD
        MOV DX,03DAH             ;本段程序用以消除雪花
MOVE:   IN AL,DX
        AND AL,1
        JNZ MOVE
WAIT1:  IN AL,DX
        AND AL,1
        JZ WAIT1
        LODSW
        STOSW
        LOOP MOVE
        PUSH CS
        POP DS
        MOV AX,OFFSET BUFFER     ;将缓冲区内容进行压缩
        MOV SI,AX                ;去掉属性字节并为每行加
        MOV DI,AX                ;回车换行符
        MOV CX,2000
        MOV BH,80
PACK:   MOV AL,[SI]
        MOV [DI],AL
```

```
        INC SI
        INC SI
        INC DI
        DEC BH
        JNZ GO_ON
        MOV BYTE PTR [DI],0DH
        INC DI
        MOV BYTE PTR [DI],0AH
        INC DI
        MOV BH,80
GO_ON:  LOOP PACK
        MOV AH,3CH              ;建立文件
        LEA DX,FILE
        MOV CX,20H
        INT 21H
        JC RETU
        MOV BX,AX
        MOV HANDLE,AX
        MOV AH,40H              ;写文件
        MOV CX,2050
        MOV DX,OFFFSET BUFFER
        INT 21H
        MOV AH,3EH              ;关闭文件
        MOV BX,HANDLE
        INT 21H
RETU:   POP BP                  ;恢复现场
        POP ES
        POP DS
        POP DI
        POP SI
        POP DX
        POP CX
        POP BX
        MOV AL,20H
        OUT 20H,AL
        POP AX
        STI                     ;开中
        IRET                    ;返回断点
NEWINT5 ENDP
;
BEGIN:
        PUSH CS
        POP DS
```


试读结束：需要全本请在线购买： www.ert