

# 用高级语言实现 DOS 环境下的控制功能

韩立毛

(盐城工学院计算机工程系, 盐城, 224003)

**摘要** 在 DOS 环境下实现多种控制功能, 如: 屏幕控制功能、光标控制功能、键盘重新定义功能等, 这些功能的实现对于 DOS 环境下的某些操作来得更加方便, 对于改善人机界面将起到重要作用。提出用高级语言实现 DOS 环境下控制功能的方法。

**关键词** 高级语言 DOS 环境 控制功能

**分类号** TP3

## 一、实现原理及方法

IBM PC 微机及其兼容机应用 DOS 磁盘操作系统时, 可用 ANSI. SYS 驱动程序的“扩充的屏幕的键盘控制功能”来代替标准输入和标准输出支持, 使屏幕和键盘模仿 ANSI (American National Standards Institute 美国国家标准协会) 的终端设备。这些设备能用 ANSI 研制的转义 (ESCape) 序列在不同硬件、监视器和键盘上完成指定的功能, 包括屏幕控制、键盘设置、发声控制等等, 这些控制都是以 ESC 开头的, 把这种控制与设置称之为 ESC 控制序列或 ANSI 转义序列。

例如下列命令即为 ESC 控制序列中的三条:

ESC[2J           清屏

ESC[2K           清除从当前光标位置开始的直到行尾的所有字符

ESC[m,nH       光标定位于 m 行, n 列

在高级语言中没有代表“ESC”的符号字符, 因此不能直接向 DOS 系统发出控制请求, 但可以用高级语言提供的字符实现这一目的。

在 Turbo PASCAL 中有一字符函数 CHR(I), 其中 I 为一字符的 ASCII 码, 这一函数的功能为将 ASCII 码 I 转换为其代表的字符, 如 CHR(65) 代表为 A, CHR(75) 代表为 K。查一下 ASCII 码表, 即可找到“ESC”所对应的 ASCII 码为 27, 这样就可将“ESC”用 wrtie 语句输出, 从而突破了这个问题的瓶颈。

在 FORTRAN77 中有一字符内部函数 CHAR(I), 同样可用 WRTIE 语句输出“ESC”; 在 C 语言中也有字符处理, 可用 PRTNTE 语句输出“ESC”。

下面就 Turbo PASCAL、FORTRAN77、Turbo C 三种高级语言, 讨论 DOS 环境下控制功能的实现方法。

## 二、DOS 环境下控制功能的实现

### 1、屏幕控制功能

用上述原理和方法可实现屏幕控制功能。举例如下：

#### (1)定义前景颜色

用 Turbo PASCAL 语言：

```
write(chr(27),'[31m');——红色   write(chr(27),'[34m');——蓝色  
write(chr(27),'[32m');——绿色   write(chr(27),'[35m');——淡紫  
write(chr(27),'[33m');——桔色   write(chr(27),'[36m');——青色  
write(chr(27),'[37m');——白色
```

用 FORTRAN77 语言：

```
WRITE(*,'(1X,2A)') CHAR(27),'[31m'——红色
```

其余同上。

用 Turbo C 语言

```
printf("\33[31m");——红色
```

其余同上。

#### (2)定义背景颜色

用 Turbo PASCAL 语言：

```
write(chr(27),'[41m');——红色   write(chr(27),'[44m');——蓝色  
write(chr(27),'[42m');——绿色   write(chr(27),'[45m');——淡紫  
write(chr(27),'[43m');——桔色   write(chr(27),'[46m');——青色  
write(chr(27),'[47m');——白色
```

用 FORTRAN77 语言：

```
WRITE(*,'(1X,2A)') CHAR(27),'[41m'——绿色
```

其余同上。

用 Turbo C 语言

```
printf("\33[41m");——绿色
```

其余同上。

### 2、光标控制功能

用上述原理和方法可实现屏幕控制功能。举例如下：

用 Turbo PASCAL 语言

```
write(chr(27),'[2J');——清屏  
write(chr(27),'[5A');——光标保持列不变而上移 5 行  
write(chr(27),'[K');——清除从当前光标位置开始的直到行尾的所有字符  
write(chr(27),'[m;nH');——光标定位于 m 行 n 列
```

其中 m, n 为数字, 分别表示行数和列数。在执行上述命令将光标定位后, 若要在光标之后跟一串字符时, 只需在最后加上单引号括起来的字符串。如：

```
write(chr(27),'[20,30H','TREE');将光标定位于 20 行 30 列, 并在其后显示字符串
```

TREE。

用 FORTRAN77 语言:

WRITE(\*, '(1X,2A)') CHAR(27), '[2J'——清屏

WRITE(\*, '(1X,2A)') CHAR(27), '[5A'——光标保持列不变而上移 5 行

WRITE(\*, '(1X,2A)') CHAR(27), '[K'——清除从当前光标位置开始的直到行尾的所有字符

WRITE(\*, '(1X,2A)') CHAR(27), '[m;nH'——光标定位于 m 行 n 列

注意:在执行上述命令将光标定位后,若要在光标之后跟一串字符时,需要将上述语言中的“2A”改为“3A”,并在最后加上单引号括起来的字符串。如:

WRITE(\*, '(1X,2A)') CHAR(27), '[20,30H', 'TREE';将光标定位于 20 行 30 列,并在其后显示字符串 TREE。

用 Turbo C 语言:

printf("\33[2J")——清屏

printf("\33[5A")——光标保持列不变而上移 5 行

printf("\33[K")——清除从当前光标位置开始的直到行尾的所有字符

printf("\33[m;nH")——光标定位于 m 行 n 列

### 3. 键盘控制功能

用上述原理和方法可实现键盘重定义功能。其两种语句格式为:

用 Turbo PASCAL 语言

(1)write(chr(27), '[O;K;"string";13P');

(2)write(chr(27), '[O;K;"string";P');

其中 K 为所定义键盘的 ASCII 码, string 为所定义的内容, 格式(1)为所定义的键最后带回车, 格式(2)为所定义的键最后不带回车。

用上述两种格式就可以对所要定义的键重新进行定义, 通常是进行功能键的重新定义, 标准型的 83 键键盘有 F1 到 F10 共 10 个功能键, 扩展到 101 键键盘多了 F11 和 F12 两个功能键, F1 到 F10 所对应的扩展 ASCII 码为 59 到 68, F11 和 F12 所对应的扩展 ASCII 码为 133 和 134, DOS 环境下定义了 F1 到 F6, 余下的 6 个功能键没有定义, 用户可根据需要对其进行定义。如将 F8 定义为“DIR/P”且后面跟回车, 可用下面的语句来定义:

write(chr(27), '[0;66;"DIR/P";13P');

其中, 0 表示使用了扩展 ASCII 码, 66 为 F8 的扩展 ASCII 码, 双引号里的字符为所要定义的内容, 13 代表回车。如不要回车只需要将 13 连同它前面的分号一同去掉, 变为 write(chr(27), '[0;66;"DIR/P"')即可。

用户可以将欲定义的功能键编制为一个程序, 将其编译成可执行文件并放入自动批处理文件中, 这样每次启动计算机后, 功能键均按程序中的设定设置好。

用 FORTRAN77 语言:

(1)WRITE(CHAR(27), '[O;K;"string";13P';

(2)WRITE(CHAR(27), '[O;K;"string"');P';

用 Turbo C 语言

```
(1)printf("\33[0;K;"string";13P");
```

```
(2)printf("\33[0;K;"string"P");
```

#### 4. 输出语句执行后光标的定位

在程序中常常要从键盘输入数据,在输入之前往往需要先显示一些提示信息,然后在提示信息之后输入,此时光标应该停留在提示信息之后,这在 Turbo PASCAL 用 write 语句;在 Turbo C 中用 printf 即可。而在 FORTRAN 语言中执行 WRITE 语句时光标是移到下一行的行首,有时为了连贯,要求将光标定位于提示信息之后,此时可用 FORTRAN 语句来实现,方法如下:

```
.....
CHARACTER * 8 NAME
WRITE(*,100)'请输入姓名:'
100  FORMAT(1X,A,$)
READ(*,'(A)') NAME
.....
```

程序运行在屏幕上提示“请输入姓名:”之后,光标停留在冒号的后面,等待使用者从键盘输入 NAME 的值,其中 FORMAT 语句中的 \$ 所起的即为抑制换行。注意 FORMAT 句中 \$ 控制符只在其后紧跟 READ 语句时才能起作用,若跟的是其它语句,则不能起到抑制换行的作用。

#### 5. 发声控制功能

在程序中可用下面的语句来控制扬声器发出鸣叫。

用 Turbo PASCAL 语言:

```
write(chr(7));
```

用 Turbo C 语言:

```
printf("\7);
```

FORTRAN 语言中用:

```
WRITE(*,"(1X,A)") CHAR(7)
```

若要指定扬声器发出一定的鸣叫次数,则可用子程序与循环来实现:

```
SUBROUTINE SOUND(N)
DO 10,I=1,N
WRITE(*,'(1X,A)') CHAR(7)
10  CONTINUE
END
```

### 三、实现条件

上述屏幕控制功能、光标控制功能、键盘设置功能用其它 DOS 环境下的其它控制功能要在 IBM PC 及其兼容机上实现时,其启动的系统盘必须含有系统配置文件 CONFIG.SYS 和驱动程序 ANSI.SYS 两个文件,且在 CONFIG.SYS 中必须安装驱动程序:DEVICE=<盘符><路径>ANSI.SYS。