

如何充分使用 FoxPro 提供的色彩

邓建军 刘怀宇 (中国矿业大学)

FoxPro 为用户提供了强大的颜色及其控制功能,然而大多用户却仅使用了其中很少一部分。主要是对于具体的颜色操作很少有书做系统介绍,从而给我们如何编制出色彩丰富的用户界面带来一定困难,为此,下面讨论一下如何操作控制 FoxPro 的颜色。

一、颜色组成

对于 For DOS 的彩色屏幕提供了 10 种颜色,具体代码见表 1:

表 1

颜色代码		RGB 值			
颜色	代码	红色的数量值	绿色的数量值	蓝色的数量值	
BLACK	黑	N	0	0	0
CYAN	青蓝	BG	0	255	255
WHITE	白	W	255	255	255
BLANK	空白	X			
GREEN	绿	G	0	255	0
YELLOW	黄	GR+	255	255	0
BLUE	蓝	B	0	255	0
MAGENTA	紫红	RB	255	0	255
BROWN	棕红	GR	255	255	0
RED	红	R	255	0	0

对于 For Windows,还为用户提供了 RGB()函数,从而提供了更加丰富的颜色。RGB 是 Red, Green Blue 三个英文字的首字母组合而成,即采用红、绿、蓝为基本色,用数字 0-256 表示其起作用的强弱程度,0 表示该基色不起作用,随着其值的增加,该基色所起作用程度加深,从而可得到 $256 \times 256 \times 256 = 2^{24}$ 种颜色。例 RGB(255, 0, 0, 255, 255, 255)的前三个数值表示的是前景色(红色),后三个数值表示的是背景色(白色)。因 For Windows 的其他命令与 For DOS 的基本一致,下面只以 For DOS 为例作介绍。

二、颜色定义

首先介绍几个相关概念一色对(Color Pair)、调色盘(Color Scheme)、色组(Color Set)。

色对:由前景色、背景色和分隔符"/"组合而成。例 GR+/ BX 表示前景色是黄色,背景色是兰色,在计算机上显示为兰底白字。

调色盘:由 10 个色对和一个阴影控制符号组成,其中每一个色对控制一个显示属性的色彩,其具体控制项如表 2,例 :set color of sche 16 to w+/ bg, GR+/ b *, gr+ / W, GR+ / W, GR+ / bg, W+ / n, W+ / N, w+ / n, W+ / n, w+ / bg, -,其中加号(+)使前景色变得明亮,星号(*)使背景色变明亮或闪烁,阴影控制符号“-”表示边框阴影不显示,若为“+”或缺省时则表示边框阴影显示。

表 2

色对	英文字符名称	说明	备注
1	ST	一般的标准显示	如?,??,SAY..等命令
2	ET	加强型的显示	如 GET,ENIT..等命令
3	BD	WINDOW 或 POPUP 的外框	如 DEFINE WINDOW..等
4	TH	WINDOW 或 POPUP 的标题	如 DEFINE WINDOW..等
5	MG	提示信息	如 DEFINE WINDOW..等
6	SL	被选择的项目	
7	SH	状态行,时间显示或热键	状态行及时间仅能由第一个调色盘的第 7 个色对控制
8	SD	WINDOW 或 POPUP 外框的阴影	
9	EC	可以选择的项目	
10	DC	不可以选择的项目	

色组:由 24 个调色盘组成,安装时系统自动建立七个缺省色组,分别为:COMPQ_COLO、COMPQ_MONO、DEFAULT、MONOCHROME、T5200_COLO、T5200_MONO、TOSH_3200,一般这些已是可以满足用户对色彩的要求了,若系统的色组还不能满足用户的要求,则可自行定义词组。方法是改变相应的调色盘或在已设置完的色彩环境下用 CREATE COLOR SET 命令建立一新色组。

三、调色盘的使用

调色盘是色对和色组的联系纽带,其使用得适当与否将直接影响界面色彩的好坏。首先让我们来看表 3,

了解一下色组中各调色盘内每个色对所控制的位置名称。

表 3 系统自动配置的颜色组成

Object Name (调色盘)	色对序号									
	1	2	3	4	5	6	7	8	9	10
	ST	ET	BD	TH	MG	SL	SH	SD	EC	DC
	★	★	★	★	★	★	★	★	★	★
User Winds(1)	√	√	√	√	√	√	√	√		
User Menus(2)	√	√	√	√	√	√		√		
Menu Bar(3)	√	√				√	√			
Menu Pops(4)	√	√	√			√	√	√		
Dialogs(5)	√	√	√			√	√	√	√	√
Dialog Pop(6)	√	√	√			√		√		
Alert(7)	√	√	√			√	√	√	√	√
Windows(8)	√	√	√	√	√	√	√	√	√	√
Wind Pops(9)	√	√	√			√		√		
Browse (10)	√	√	√	√	√	√	√	√		
Report(11)	√	√	√	√	√	√	√	√	√	√
Alert Pops(12)	√	√	√			√		√		

注:1.英文为目标名,(1)~(2)表示系统缺省时的调色盘代号。

2.Scheme 13~24 的所有色对均相同,系统缺省时留给用户自行设置。

3."√"表示该色对有效。

4.标注"★"符号的意义说明见表二。

当你对上表有一全面认识之后,你可以根据自己需要,改变相应的色对,创造出自己的调色盘了。

四、编辑时的应用及注意事项

编程时可由用户自行设置调色盘编号或色对序列的命令主要有:

@.....EDIT:建立一编辑区

@.....SAY|GET:显示输出,接受输入

@.....GET_CHECK BOXES:建立检查框

@.....GET_INVISIBLE BUTTONS:建立不可见按钮

@.....GET_LISTS:建表

@.....GET_POPUPS:建立弹出式菜单

@.....GET_PUSH BOTTONS:建立 PUSH 按钮

@.....GET_RADIO BOTTONS:建立 RADIO 按钮

@.....GET_SPINNER:建立一 SPINNER

@ < row1, column1 > FILL TO < row2, column 2 > : 填充屏幕区域

@row1, column1 > TO < row2, column2 > : 画框和对象

BROWSE:为访问数据库打开一窗口

CHANGE:修改数据库记录

EDIT:编辑数据库记录

DEFINE BAR:定义菜单选项

DEFINE MENU:定义菜单系统

DEFINE PAD:定义菜单点

DEFINE POPUP:定义弹出式菜单

DEFINE WINDOW:定义用户窗口

MODIFY WINDOW:改变窗口属性

READ:激活对象

SHOW GET:再现一 GET 对象

SHOW GETS:再现所有 GET 对象

SHOW OBJECT:显示已定义的弹出菜单

对于以上命令的色彩控制,FOXPRO 为我们提供了两种方法:

1.用户先定义一个自己的调色盘,其命令语句是 SET COLOR OF SCHEME < expN >,然后在所用命令中使用[Color Schem< expN >参数。例如:欲定义一黄色标题、灰色边框,内部是白字暗青蓝色底的窗口时,程序如下:

```
set color of sche 16 to w+ / bg, GR+f / b * , gr+ / W, GR+ / W,
```

```
GR+ / bg, W+ / n, W+ / N, w+ / N, W+ / N, w+ / bg, -
```

```
DEFINE WINDOW WINCS FROM 0,0;
```

```
TO 12,40;
```

```
COLOR SCHEME 16;&&&&使用第 16 号调色板
```

```
SYSTEM;
```

```
NOSHADOW;
```

```
float;
```

```
TITLE"测试颜色窗口";
```

```
FOOTER"确认颜色正确后按任意键退出"
```

```
ACTIVATE WIND WINCS
```

```
MOVE WIND WINCS CENTER
```

```
@3,1 SAY "本窗口是黄色标题、灰色边框、白字暗青蓝色底"
```

```
READ
```

```
RELE WIND WINCS
```

读者可结合表二、表三分析 SCHEME 16,其中各色对依次控制:①SAY 色;②菜单项或 GET 的色;③窗口边框色;④标题色;⑤信息条的色;⑥选中菜单项色;⑦热键色;⑧阴影色;⑨在窗口命令中不起作用;⑩"- "表示边框阴影不显示。此方法定义一次后可以反复使用,但具体控制的色彩不直观。

2.直接在所用命令中使用[Color < color pair list >]参

数,针对本命选项进行颜色设置。例如,上面的白字暗青蓝色底就可用命令语句: @3, 1 SAY" 白字暗青蓝色底"COLOR W+ / BG。本方法直观,但当需要控制多个色对时,编程太繁琐,且应该注意色对的覆盖。

有了上述方法后,用户将能很自由地设计出自己的界面,但也可能会出现这样一问题:当几个程序员编写同一个系统时,同时定义了同一个调色盘,如都用 Scheme 13,但是其设置又可能不一样,这样先调用一方的颜色将影响后调用一方的颜色配置,从而使系统颜色混乱;有时也可能设置的同一界面(如警告提示框)不同人的颜色配置不一样,使程序颜色显得杂乱无章,不适合用户习惯等等。针对这些问题,就要求整个系统颜色保持一致,方法很多,下面介绍一下我们的经验。

首先是对系统总体设计人员,应知道本系统需要多少个调色盘,然后根据需要先以系统给定的 1-12 号盘为基础,至于其缺省时的具体分配情况可由文章后所附的一段小程序,对照表三获得。当前 12 号盘不能满足要求时,还可根据需要定义 13-20 号盘,有了这 20 号调色盘后,一般已能解决问题,若仍不能解决,则需另定义一个色组了,因后 4 号盘(21-24)一般需留给模块设计人员。

对模块设计人员,则应按用到的项目分别在 1-20 号调色盘(板)中采用相应的色板,决不允许对之进行修改或重新定义。当不能满足自己的要求时,可自行定义 21-24 号盘,但必须每次在调用的语句前定义一次,以免调用了别人的程序后受影响。

附:

```
set talk off
set textmerge on to color.dat
dime colset[7]
colset[1]="COMPQ__COLO"
COLSET[2]="COMPQ__MONO"
COLSET[3]="DEFAULT"
COLSET[4]="MONOCHROME"
COLSET[5]="T5200__COLO"
COLSET[6]="T5200__MONO"
COLSET[7]="TOSH__3200"
FOR J=1 TO 7
  SET COLOR SET TO &COLSET[J]
  \
  \ 系统默认色组 < <J>> 的颜色
  \ * * * 1-24 号调色盘的色对分配情况 * * * * *
  \
  \
  \
FOR I=1 TO 24
```

```
\ 调色盘 < <STR(1,2)>> : < <SCHEME(I)>>
NEXT
\
\
\ * * * * *
ENDFOR
SET TALK ON
SET COLOR SET TO DEFAULT
CLOSE ALL
```

基于 FoxPro 的通用数据库记录全屏幕编辑程序

陈 越 刘育楠 (解放军电子技术学院)

摘要:本文介绍了采用表驱动程序生成技术思想编制的一个通用的、能实现对数据库记录的字段进行全屏幕编辑的通用程序。

一、引言

通常对数据库记录的录入与修改的方法是,用多个 @...SAY...GET 语句与 READ 语句配合,以实现数据库记录的全屏幕编辑,这种全屏幕方式有以下几个缺点:

1.各字段必须逐个编辑,发现某字段有误时,也必须逐字段返回,即当前编辑的字段不能随意改变,给应用程序的使用造成不便。

2.对不同的数据库记录进行编辑时,需编写直接对具体数据库字段进行操作的程序,编程量大,开发周期长,难以调整编辑格式和进行维护。

因而,我们在开发应用系统的实践中,编制了一个通用的数据库记录全屏幕编辑程序。

二、设计思想及数据结构说明

本程序采用表驱动程序生成技术的思想,即把程序中的控制信息从程序中分离出来,形成抽象程序与控制信息表两部分。该抽象程序是可重用的,在具体应用中不必改变,而控制信息库则是可变化,可根据具体管理信息填写生成。

例如,为对学生档案库 student.dbf 中记录的全屏幕编辑,我们建立了一个辅助控制库,fstudent.dbf,它们的内容如下:

```
USE STUDENTS INDEX STUDENT
Record#  XH  XM  XB  CSRQ  JG  RXCJ
```