

Turbo C 2.0 高级应用程序设计

王小华 编写



陕西电子编辑部

前 言

《Turbo C 高级应用程序设计》收集了Superior Editor 编辑软件完整的程序（约四千条语句），讲述了一些常用程序的设计方法。如弹出式菜单、下拉式菜单、覆盖式菜单等多种子程序的设计。Superior Editor 的功能包含Turbo C 编辑器的全部功能，另外可实现自动汇编的格式安排，同时打开多个文件进行编辑，可以通过分割窗口，在一个屏幕上同时编辑两个文件。Superior Editor 有较强的块处理功能，可以实现块复制、块删除、块打印、读块与写块。该软件字符串处理能力很强，能实现字符串查找与替换。通过对源程序的分析，揭示了实现这些功能的程序设计方法。

编辑软件是最基本最常用的文字处理软件，可以单独存在，也可以集成到其它应用软件中。在软件设计上，Superior Editor 留出了许多可供设计者增加功能的接口，以适应软件设计与应用的灵活性。

整个程序由 Turbo C 编写，结构紧凑，风格独特，不失为一份珍贵的软件资料。

注：为便于读者自学，与本书配套发行软盘一张，该软盘包含书中全部源程序和SE 执行文件。有意者请与陕西电子编辑部联系。

目 录

一	几种菜单子程序的设计	1
一、	弹出式菜单的设计	1
二、	亮条移动的设计方法	2
三、	菜单 1	3
四、	菜单 2	10
五、	菜单 3	14
六、	菜单 4	15
二	Superior Editor 功能简介	18
三	Superior Editor 子程序的设计	23
	Barcol ()	23
	Barrow ()	23
	Cmph ()	23
	curx ()	24
	Dir ()	25
	Doscommand ()	25
	Drawframe1 ()	26
	Drawframe2 ()	28
	Findy ()	29
	Framecol ()	29
	Framerow ()	30
	Getfile ()	30
	getkey ()	34
	Gotoxy ()	34
	insrow ()	34
	Klen ()	35
	lastpt ()	35
	nextpt ()	35
	Page ()	36
	Print ()	36

Printfile()	36
Ptinform()	37
Ptline()	37
Putch()	38
Putcur()	38
Putsheet()	38
Putsheet1()	39
Putsheet2()	41
Putsheet3()	42
Putstr()	43
Savefile()	43
Setback()	44
setbottom()	44
Showstr()	45
Strr()	45
tabx()	46
Editor()	47
四 Superior Editor 源程序清单	83
TEST.H	83
FILE0.C	87
FILE1.C	101
FILE2.C	104
FILE3.C	106
FILE4.C	111
FILE5.C	112
FILE6.C	154
FILE7.C	166
附 键值表	177

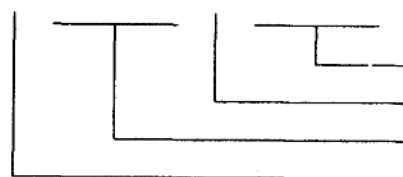
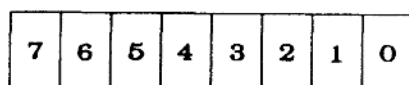
一、几种菜单子程序的设计

一、弹出式菜单的设计:

我们知道,菜单的优点在于为用户提供友好的对话界面,使用户在使用软件的过程中方便自如。随着计算技术的飞速发展,人们思维的不断更新,各种形式的菜单正广泛地应用于高级程序的设计中,在此,结合本人的实践,给出几种菜单的设计新方法。

Turbo c 2.0 版中提供了函数 `gettext(int left, int top, int right, int bottom, void *destin)` 和 `puttext(int left, int top, int right, int bottom, void *source)` 两个基本函数,前者得到屏幕上由 $(left, top)$, $(right, bottom)$ 坐标为对角线所指定的矩形区域上各点的信息,存入由 `destin` 指针为首址的内存中。屏幕上每一点的信息由字符与属性两部分组成,字符以 `ascII` 形式存入内存,属性包括该字符的前景、背景及字符的闪烁状态,由一个字节组成,其八位码各位的意义如下:

(对CGA)



前景颜色
前景强度
背景颜色
闪烁状态

标准颜色值:

符	号	值	名	称
BLACK		0	黑	色
BLUE		1	蓝	色
GREEN		2	绿	色
CYAN		3	青	蓝 色
RED		4	红	色
MAGENTA		5	洋	红 色
BROWN		6	棕	色
LIGHTGRAY		7	灰	色
DARKGRAY		8	紫	色
LIGHTBLUE		9	亮	蓝 色

符 号	值	名 称
LIGHTGREEN	10	亮绿色
LIGHTCYAN	11	亮青蓝色
LIGHTRED	12	亮红色
LIGHTMAGENTA	13	亮洋红色
YELLOW	14	黄色
WHITE	15	白色

知道了屏幕上各点与内存的对应关系，我们可以事先运用程序使要显示的内容在内存中按先后次序排队，然后再通过puttext()语句使菜单弹出。一个简单的例子，如：设定字符为W，属性为：前景为红色(4)、背景为青蓝色(3)，在坐标为(20,20)处弹出该字符。

属性值为：背景 $\times 16$ + 前景 = $3 \times 16 + 4 = 52$ 。

简单程序如下：

```
#include <conio.h>
main()
{
    char *text;
    *text=87;      /* W的ASCII值 */
    *(text+1)=52; /* W的属性 */
    puttext(20,20,20,20,text);
}
```

二、亮条移动的设计方法：

亮条移动的基本思想是：根据光标键 upkey, downkey, rightkey, leftkey 的控制，能够使亮条按一定的规律上下左右移动。在此，我们可以认为这是一个改变属性与恢复属性的过程。首先，当亮条在某处时，将这一处字符的属性改变，亮条移至另一处后，恢复先前位置的属性，改变当前位置的属性，这样就能实现亮条的移动变化。

在 Turbo c 2.0 中，并没有提供改变某一处字符属性的函数，在此，我们可以运用 DOS 的功能调用方法，自行设计这一函数：

该函数要求所实现的功能为：在指定的屏幕坐标位置上，给出指定的属性。

```
void Setback(int x, int y, int att)
{
    union REGS reg;
    reg.h.ah=2;      /* 确定光标位置 */
    reg.h.dh=y-1;    /* 光标纵坐标 */
    reg.h.bh=0;      /* 选0显示屏 */
}
```

```

    reg.h.dl=x-1;    /* 光标横坐标 */
    int86(0X10,&reg,&reg);

    reg.h.ah=8;      /* 读光标处的属性和字符 */
    int86(0X10,&reg,&reg);

    reg.h.ah=9;      /* 写光标位置的字符和属性 */
    reg.h.bl=att;     /* 属性值 */
    reg.x.cx=1;       /* 只写一次 */
    int86(0X10,&reg,&reg);
}

```

对于改变横行len 个字符的属性, 运用如下子程序:

```

void Barcol(int x, int y, int att, int len)
{
    int i;
    for (i=0; i<len; i++)
    {
        Setback(x+i, y, att);
    }
}

```

改变纵行len 个字符的属性, 运用如下的子程序:

```

void Barrow(int x, int y, int att, int len)
{
    int i;
    for (i=0; i<len; i++)
    {
        Setback(x, y+i, att);
    }
}

```

在亮条设置好后, 屏幕上还有光标闪烁, 光标的闪烁状态是由硬件引起的, 在字符方式下, 采取最简单的关闭光标的方法是: 将光标移至一个不可显示的位置, 如坐标位置为 $(X, Y) = (0, 25)$ 处。但在 Turbo C 2.0 所提供的函数中, gotoxy(int x, int y) 中 x, y 的取值范围: x 为 (1, 80), y 为 (1, 25), 如果超出这个范围, 光标将停留在 (1, 1) 处。且 (x, y) 为相对坐标, 如果屏幕窗口在屏幕上的坐标为 (10, 10, 50, 12), gotoxy(1, 1) 将使光标移至该窗口顶点 (10, 10) 处。要使光标能在屏幕绝对坐标上变化, 我们可以自行设计一个函数 Gotoxy(int x, int y) (函数设计在下面给出), 运用该函数 Gotoxy(0, 25) 能使光标移出屏幕显示区, 由此达到隐藏光标的效果。

三、菜单 1:

1: 形式如下:

---String---		
Number 1	Number 5	Number 9
Number 2	Number 6	Number 10
Number 3	Number 7	Number 11
Number 4	Number 8	Number 12

2: 功能:

可作为大型程序的主菜单用, 菜单中, 亮条可以在光标键上, 下、左、右的控制下移动到任一指定位置, 按回车键即可转入相应的服务功能。

3: 设计方法及子程:

前面解决了弹出式与亮条的设计方法, 在这里只需将二者有机地结合起来, 设定适当的返回参数, 在每次调用时取得一返回参数值, 根据返回参数值确定相应子程序入口。

aa: 菜单弹出子程:

```
void mainmenu(int bar, char *au)
{
    int i, n, L;
    char *s, *es;
    int x, y;
    Sheet1(11, 7, 68, 17, au, 49); /* 在内存中构造出菜单的主边框 */
    n=68-11; /* 边框的长 */
    /* 给出亮条所对应的横坐标 */
    if (bar<=4)
    { x=15;
      y=10+bar;
    }

    else if ((bar>4) && (bar<=8))
    { x=32;
      y=10+(bar-4);
    }

    else
    { x=49;
      y=10+(bar-8);
    }
}
```



```

es=au+2*(n+18);
s='--- Superior Editor ---';
L=strlen(s);
/* 将字符串S输入到相应的内存中, 给定该串属性值 */

for (i=0; i<L; i++)
{
    *es++=*s++;
    *es++=60; /* 属性值 */
}

es=au+2*(2*n+32);
s='. Xi'an China .';
L=strlen(s);

for (i=0; i<L; i++)
{
    *es++=*s++;
    *es++=58; /* 属性值 */
}

es=au+2*(9*n+36);
s='... Designed by Wang Xiao Hua';
L=strlen(s);

for (i=0; i<L; i++)
{
    *es++=*s++;
    *es++=55; /* 属性值 */
}

es=au+2*(3*n+7);
for (i=0; i<50; i++)
{
    *es++=196;
    es++;
}

es=au+2*(8*n+12);
for (i=0; i<50; i++)
{
    *es++=196;
    es++;
}

```

```

es=au+2*(4*n+7);
s=' Load      -F5 ';Xhstring(es,s);
es=au+2*(5*n+8);
s=' Save        -F6 ';Xhstring(es,s);
es=au+2*(6*n+9);
s=' Print       -F7 ';Xhstring(es,s);
es=au+2*(7*n+10);
s=' Dos command';Xhstring(es,s);
es=au+2*(4*n+24);
s=' New';Xhstring(es,s);
es=au+2*(5*n+25);
s=' Directory';Xhstring(es,s);
es=au+2*(6*n+26);
s=' Link';Xhstring(es,s);
es=au+2*(7*n+27);
s=' Pick';Xhstring(es,s);
es=au+2*(4*n+41);
s=' OS Shell';Xhstring(es,s);
es=au+2*(5*n+42);
s=' Write to';Xhstring(es,s);
es=au+2*(6*n+43);
s=' Tools';Xhstring(es,s);
es=au+2*(7*n+44);
s=' Quit';Xhstring(es,s);
es=au+2*(3*n+6);
*es=218;
es+=34;*es=194;
es+=34;*es=194;
es+=34;*es=191;
es=au+2*(8*n+11);
*es=192;
es+=34;*es=193;
es+=34;*es=193;
es+=34;*es=217;
puttext(11,7,68,17,au);
/* 阴影 */
Barcol(12,18,6,58);
Barrow(69,8,6,10);

```

```

        Barcol(x, y, 6, 16);
        Gotoxy(0, 25);
    }

```

与该子程相关的两个子程：

```

void Xhstring(char *au, char *a)
/* 将串S配以属性后输入到内存中 */
{
    int i, len;
    char *as;
    as=au;
    len=strlen(a);
    *au+=179; au++;
    *(as+5)=63;

    for (i=0; i<len; i++)
    {
        *au++=*a++;
        au++;
    }
    as+=34;
    *as=179;
}

void Sheet1(int x1, int y1, int x2, int y2, char *au, int att)
{
    int n, i, j;
    int lm;
    char *es;

    lm=x2-x1-2;
    n=y2-y1-1;
    es=au;
    *es+=201;
    *es+=att;

    for (i=0; i<=lm; i++)
    {
        *es+=205;
        *es+=att;
    }
}

```

```

    *es++=187; *es++=att;
    for (i=0; i<n; i++)
    {
        *es++=186;
        *es++=att;

        for (j=0; j<=lm; j++)
        {
            *es++=32;
            *es++=att;
        }

        *es++=186;
        *es++=att;
    }

    *es++=200; *es++=att;

    for (i=0; i<=lm; i++)
    {
        *es++=205;
        *es++=att;
    }

    *es++=188; *es++=att;
}

```

b: 亮条移动子程:

```

int Getmainbar (int bar)
{
    int x, y, n, c;
    /* 确定亮条相应的横坐标与纵坐标 */
    do
    {
        if (bar<=4)
        {
            x=15;
            y=10+bar;
        }
        else if ((bar>4)&&(bar<=8))
        {
            x=32;
            y=10+(bar-4);
        }
    }
}

```

```

else
{
    x=49;
    y=10+(bar-8);
}

Barcol(x, y, 4, 16); /* 置亮条 */
Gotoxy(0, 25);      /* 隐藏光标 */

switch(c=getkey())
/* 对光标键进行扫描 */
{
    case UPKEY: /* 上光标键 */
        Barcol(x, y, 49, 16); /* 恢复亮条处属性 */
        Setback(x+1, y, 63); /* 同上 */
        if(bar>1)
            bar--; /* 亮条向上移动一位置 */

        else
            bar=12;
        break;

    case DOWNKEY: /* 下光标键 */
        Barcol(x, y, 49, 16);
        Setback(x+1, y, 63);

        if(bar<12)
            bar++;

        else
            bar=1;
        break;

    case RIGHTKEY: /* 右光标键 */
        Barcol(x, y, 49, 16);
        Setback(x+1, y, 63);
        if(bar<9)
            bar+=4;
        break;

```

```

        case LEFTKEY: /* 左光标键 */
            Barcol(x, y, 49, 16);
            Setback(x+1, y, 63);

            if (bar>4)
                bar-=4;
            break;

        default:
            break;
    }
}

while ((c!=CR) && (c!=ESC));
if (c==ESC)
    bar*=100;
return (bar);
}

```

子程中, bar为亮条所处的位置。

au 为已经分配的内存首址。

四、菜单2:

1: 形式如下:

Number	one
Number	two
Number	three

2: 作用:

可作为主菜单的子菜单, 光标键的上、下键可控制亮条的上下移动, 按回车即转入相应的服务功能。

3: 子程序清单:

a: 菜单弹出子程:

```

void Sheet(int x1, int y1, int x2, int y2, char *au, char *s, int f, int b, int
            bar)
{
    int n, i, j, att;
    int lm, l, matt;
    char *es;

```

```

lm=x2-x1-2;
l=strlen(a);
n=y2-y1-1;
att=b*16+f;
matt=b*16+4;

es=au;
*es+=218; *es+=att;

for (i=0; i<=lm; i++)
{
    *es+=196;
    *es+=att;
}

*es+=191; *es+=att;
for (i=0; i<n; i++)
{
    *es+=179;
    *es+=att;

    for (j=0; j<=lm; j++)
    {
        *es+=32;
        *es+=att;
    }

    *es+=179;
    *es+=att;
}

*es+=192; *es+=att;
for (i=0; i<=lm; i++)
{
    *es+=196;
    *es+=att;
}
*es+=217; *es+=att;
j=0;

es=au+2*(x2-x1+3);
*(es+1)=matt;

```

```

for (i=0; i<l; i++)
{   if (*s! = '^')
    {   *es++=*s++;
        *es++;
    }

    else
    {   j+=2;
        s++;
        es=au+(2+j)*(x2-x1+1)+4;
        *(es+1)=matt;
    }
}

puttext(x1,y1,x2,y2,au);

Barcol(x1+1,y2+1,6,x2-x1+1); /* 阴影部分 */
Barrow(x2+1,y1+1,6,y2-y1);   /* 阴影部分 */

Barcol(x1+1,y1+bar,6,x2-x1-1); /* 亮条所在位置 */

Gotoxy(0,25);                  /* 隐藏光标 */
}

```

x1, y1, x2, y2: 为屏幕上坐标所对应的菜单位置,
au: 已分配的内存,
s: 所写字符串, 以^作为行分隔符,
f: 前景属性,
b: 背景属性,
bar: 亮条对应的位置.

b: 亮条移动子程:

```

int Getbar(int x, int y, int L, int n, int bar)
{   int c;
    do
    {
        Barcol(x+1, y+bar, 4, L+1);
        Gotoxy(0, 25);
    }
}

```



```

switch(c=getkey())
{
    case UPKEY:
        Barcol(x+1, y+bar, 112, L+1);
        Setback(x+2, y+bar, 116);

        if(bar>1)
            bar--;

        else
            bar=n;

        break;

    case DOWNKEY:
        Barcol(x+1, y+bar, 112, L+1);
        Setback(x+2, y+bar, 116);

        if(bar<n)
            bar++;

        else
            bar=1;

        break;

    case RIGHTKEY:
        break;

    case LEFTKEY:
        break;

    default:
        break;

}

}

while((c!=CR) && (c!=ESC));

```