C语言画线函数逻辑功能扩充

屈景辉　（总后青藏兵站部）　任京芳　（西安市石油公司）

Turbo C2.0的图形图象函数库中,仅图象函数在处理过程中具有逻辑功能。而图形函数中的画点、线类函数是不具备逻辑功能的,这使程序设计十分不便。如果要使一条线段或一个圆弧在屏幕上移动,按常规方法处理,是将线段或弧所在矩形区域的图象保存在缓冲区内,然后使用图象函数的逻辑功能再行恢复,不断地保存恢复,才使其动作起来。也就是说,制作动画除用页切换或是setwritemode()外,均采取图象的保存与恢复的逻辑功能实现。这种方法占用内存、保存信息量大时,速度也慢。因此,笔者根据VGA图形适配器的设置模式,为Turbo C画点、线类函数增加了逻辑功能。

一、逻辑功能的引入

VGA图形控制器位于显示存储器与系统处理器之间的数据通道上。在缺省状态下,图形控制器是透明的,数据在处理器和显示器之间直接通过,在其它配置下,图形控制器可以通过对数据的逻辑运算,为绘制图形的算法提供硬件支持。

图形控制器的数据循环移位/功能选择寄存器(索引值=3),功能选择允许处理器中的写入数据在软件上与处理器锁存器中的存储数据相结合,在选择写模式0或2时,允许逻辑功能。其功能选择寄存器有关位设定如下:

D4	D3		功能
0	0	(0)	写入数据不修改
0	0	(0X8)	写入数据与处理器锁存器之间作"与"运算
0	1	(0X10)	写入数据与处理器锁存器之间作"或"运算
1	1	(0X18)	写入数据与处理器锁存器之间作"异或"运算

这样,可以通过下列函数序列引入逻辑操作:

```
outp(0x3ce,3);        选择功能寄存器
outp(0x3cf,0x18);     设定为异或(XOR)方式
line(x,y,x,y1);       画线
outp(0x3ce,3);        结束后设定为替换方式
outp(0x3cf,0);
```

二、逻辑功能适用范围及条件

Turbo C提供了一系列的画线函数。画线是广义的说法,实际上包括画点、直线、圆、弧、椭圆、多边形以及有关的颜色设置等等。

1.适用范围

适用下列7个函数:

(1)putpixel(int x,int y,int color);
在指定位置画规定颜色的点
(2)line(int x1,int y1,int x2,int y2);
从(x1,y1)到(x2,y2)处画一条直线
(3)lineto(int dx,int dy);
从现行坐标画到(x,y)
(4)arc(int x,int y,int sang,int eang,int radius)
从现行坐标按相对增量画到(x+dx,y+dy)
(5)arc (int x,int y,int sang,int eang,int radius)
在(x,y)处按半径从sang到eang画一条圆弧
(6)circle(int x,int y,int radius);
在(x,y)处画半径为radius的圆周线
(7)ellipse(int x,int y,int sang,it eang,int xr,int yr);
按指定坐标、起止角度、长短轴画椭圆弧线

2.条件限制

上述七个函数实现逻辑功能的条件是线型为实线格式(SOLID_LINE),线宽为一点宽(NORM_WIDTH),即默认状态。也可用下列函数设定:

setlinestyle(SOLID_LINE,O,NORM_WIDTH)

三、扩充函数的定义与说明

1.在主文件头部或文件中定义逻辑操作码

#define 0 替换

#define 0X8 逻辑与

#define 0X10 逻辑或

#define 0X18 逻辑异或

2.函数的定义和说明

以画点、线函数为例:

```
void far putpixel_l(int x,int y,int color,int op)
void far lineto_l(int x2,int y1,int op)
void far putpixe_l(int x,int y,int color,int op)
{
outp(0x3ce,3);
outp(0x3cf,op);
putpixel(x,y,color);
outp(0x3ce,3);
outp90x3cf,0);
}
void far lineto_l(int x,int y,int op)
{
outp(0x3ce,3);
outp(0x3cf,op);
lineto(int x,int y);
outp(0x3ce,3);
outp(0x3cf,0);
}
```

其它五个函数仍照此办理,不再赘述。

四、扩充函数的组织和运用

根据上述过程形成源文件,然后用各种模式分别进行编译,用 TLIB 建立不同内存模式的库文件,在实际中可根据内存模式分别引用。

下面是一个圆周在屏幕上四个方向移动用本文所述方法异或图形的演示程序。在 dell 486 机 TC 大模式下运行通过。

附源程序清单:

```
/ * 演示源程序清单 TEST.C * /
#include <draphics.h>
#include <dos.h>
#include ESCKEY 27 / * 定义键值 * /
#include UPKEY 328
#include DOWNKEY 336
#include LEFTKEY 331
#include RIGHTKEY 333
#include SUB_KEY 0x2d
#include ADD_KEY 0x2d
#include COPY_OP 0x00 / * 定义逻辑操作码 * /
#include AND_OP 0x08
#include OR_OP 0x10
#include XOR_OP0x18 / * 逻辑异或 * /
int getkey()
{ / * 读取键值函数 * /
int key,lo,hi;
key = bioskey(0);
lo = key&0x00ff;
hi = (key&0xff00) > > 8;
return((lo = = 0)?hi+256:lo);
}
void circle_1(int x,int y,int radius,int op)
{ / * 功能扩充后的画圆函数举例 * /
outp(0x3ce,3);
outp(0x3ce,op);
circle(x,y,radius);
outp(0x3ce,3);
outp(0x3cf,0);
}
void circle_move(int x,int y,int r);
void main(void)
{
int gd = VGA,gm = VGAHI;
initgraph(&gd,&gm, * * );
circle_move(319,239,50);
closegraph();
}
void circle_move(int x,int y,int r)
{ / * 一个半径 r 的圆以起点(x,y)四方移动 * /
int maxx,maxy,key = 0,speed = 8;
maxx = getmaxx();maxy = getmaxy();
setcolor(RED);
circle_l(x,y,r,XOR_OP);
while (key! = exckey){
key = getkdy();
if(key = = SUB_KEY){
speed- = 2; / * 按'-'移动步长减小 * /
speed = (speed < = 0)?1:speed;
continue;}
if(key = = ADD_KEY){
speed- = 2; / * 按'+'移动步长增大 * /
speed = (speed > = 80)?80:speed;
continue;}
circle_1(x,y,r,XOR_OP);
switch(key){ / * 根据键值确定移动位置 * /
case LEFTKEY:
x- = speed;
if(x, = r) x = r;
break;
case RIGHTKEY:
X+ = SPEED;
IF(x > = maxx-r)r = maxx-r;
break;
case UPKEY:
y- = speed;
if(y > = r) y = r;
break;
case DOWNKEY:
y+ = speed;
if(y > = maxy-r)y-maxy-r;
break;
}
circle_1(x,y,r,XOR_OP);
}
}
```

参考文献:

[1]王洪,Turbo C 实用指南,陕西电子编辑部 1990.8

[2]来文占等,Super VGA 高级编程指南,科海培训中心 1991.5