

课 程 设 计 报 告

课程名称： 微机系统与接口技术

题 目： 模拟系统图形界面

班 级：\_计算机科学与技术183

学 号： 201805120603

姓 名： 李娜

**2020年12月**

目录

[一、 程序总体介绍 3](#_Toc62202686)

[二、 大体功能模块 3](#_Toc62202687)

[三、 运用到的所学的知识 3](#_Toc62202688)

[四、 操作实现 3](#_Toc62202689)

[五、 心得体会 12](#_Toc62202690)

[六、 附件程序代码 12](#_Toc62202691)

1. 程序总体介绍

一个模拟系统图形界面的程序，有加载读条，弹出任务框，通过键盘按键能显示图案和文字。

1. 大体功能模块

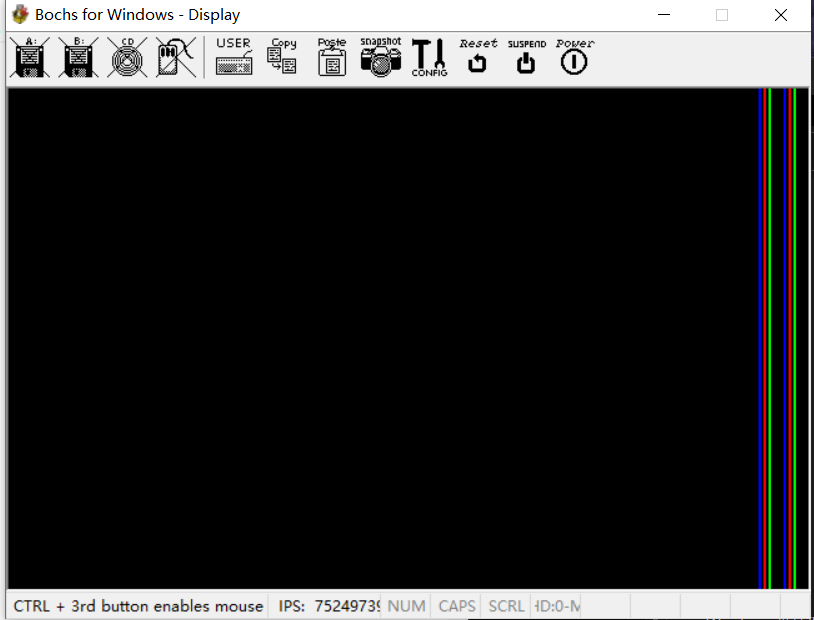
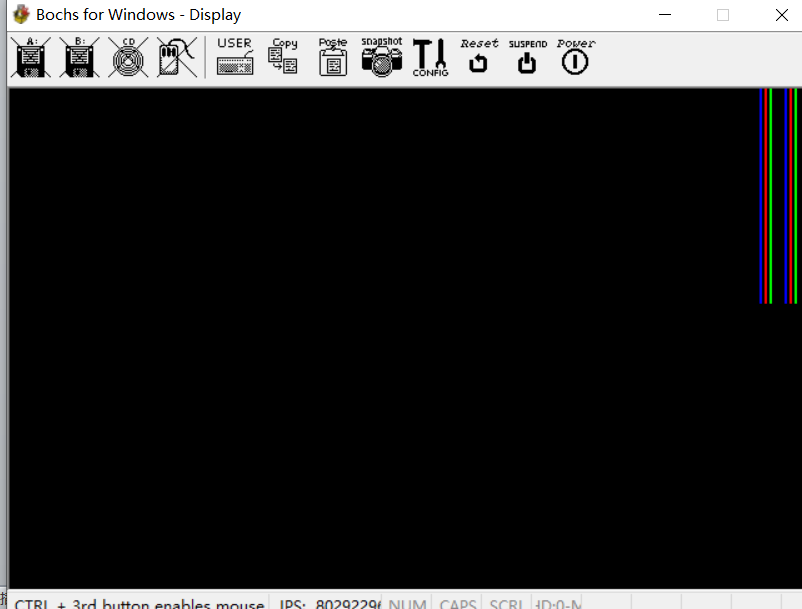
通过键盘按键A和SHIFT+A实现改变背景颜色，按enter键弹出任务框，L和SHIFT+L显示出名字，S和SHIFT+S打印三角形，Z和SHIFT+Z打印正方形

1. 运用到的所学的知识

定时器中断，键盘中断，颜色设置

1. 操作实现

下图是一开始的运行界面，右边的彩色竖线是缓慢从上到下通过定时器一点点出现直到完全的。两个左边和右边两条线是分别通过int\_time0:和 int\_time1：模块交叉调用实现同时出现和延展完成。每次刷新背景颜色都会从头开始出现彩色竖线。



其模块代码：

int\_time0:

mov word [ds:0x20],int\_time1

mov word [ds:0x22],0

cmp bx,64000

ja x2

add bx,310

mov byte[es:bx],2

add bx,2

mov byte[es:bx],3

add bx,2

mov byte[es:bx],4

sub bx,314

add bx,320

x2: mov al,0x20

mov dx,0x20

out dx,al

iret

int\_time1:

mov word [ds:0x20],int\_time0

mov word [ds:0x22],0

cmp bx,64000

ja x3

add bx,300

mov byte[es:bx],2

add bx,2

mov byte[es:bx],3

add bx,2

mov byte[es:bx],4

sub bx,304

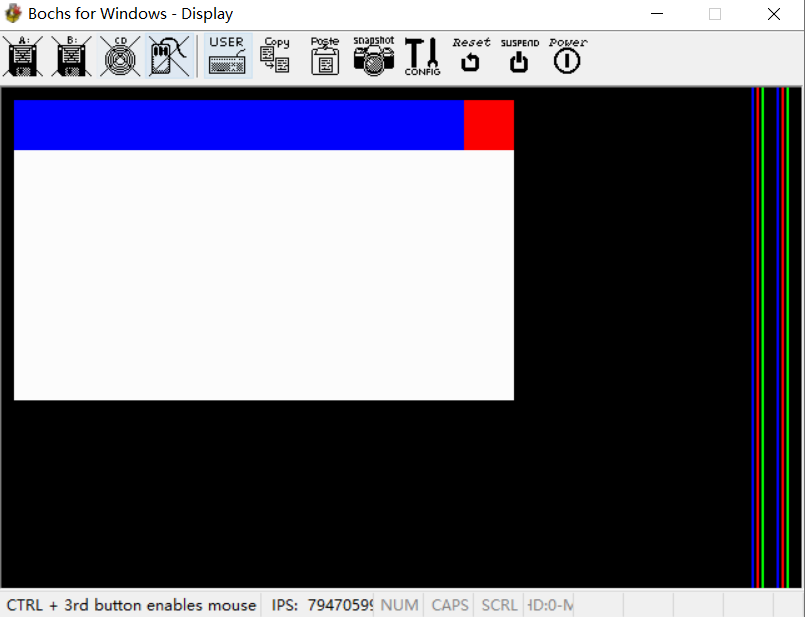
x3: mov al,0x20

mov dx,0x20

out dx,al

iret

然后按enter键出现模拟框：



其模块代码：

rectangle: mov bx,[ds:di]

mov ax,[ds:di+2]

mov dx,320

mul dx

add bx,ax

mov cx,[ds:di+4]

add cx,bx

mov ax,0

mov dl,[ds:si]

r: mov byte[es:bx],dl

inc bx

cmp bx,cx

jnz r

add cx,320 ;新一行

sub bx,[ds:di+4]

add bx,320

inc ax

cmp ax,[ds:di+6]

jnz r

inc si

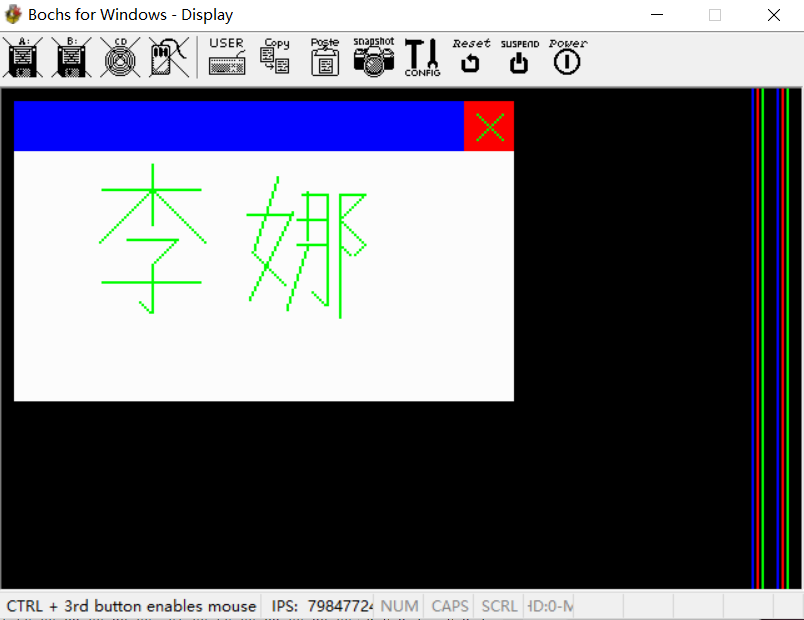
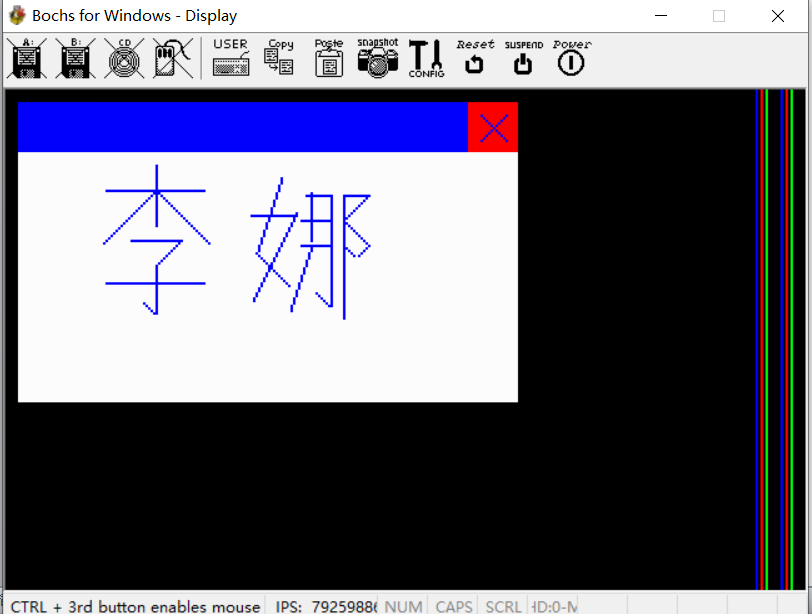
add di,8

cmp di,rect+24

jnz rectangle

ret

然后按L键显示名字(蓝色)，shift+L（绿色）：



其模块代码：

line: mov bx,[ds:di]

mov ax,[ds:di+2]

mov dx,320

mul dx

add bx,ax

mov dx,0

mov cx,[ds:di+4] ;长度

x: mov al,[ds:si]

mov byte[es:bx],al

add bx,[ds:di+6]

inc dx

cmp dx,cx

jnz x

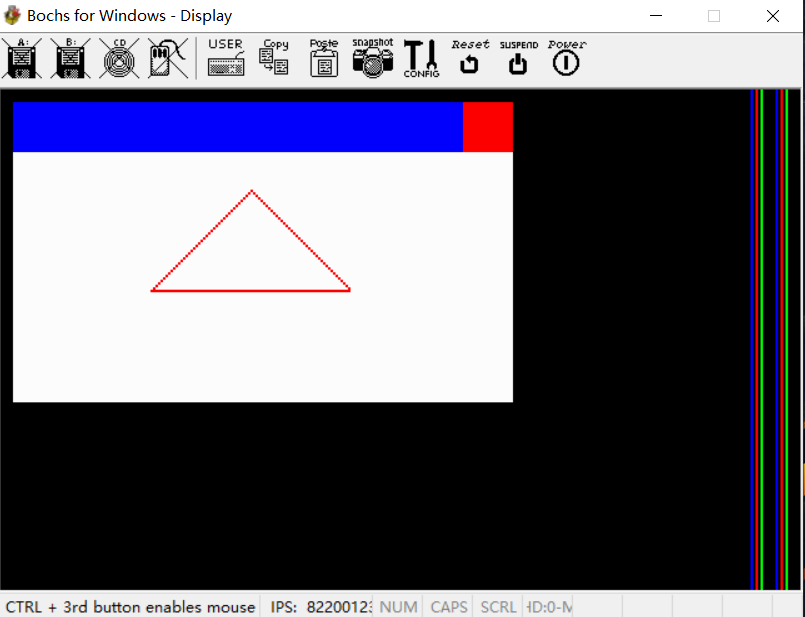
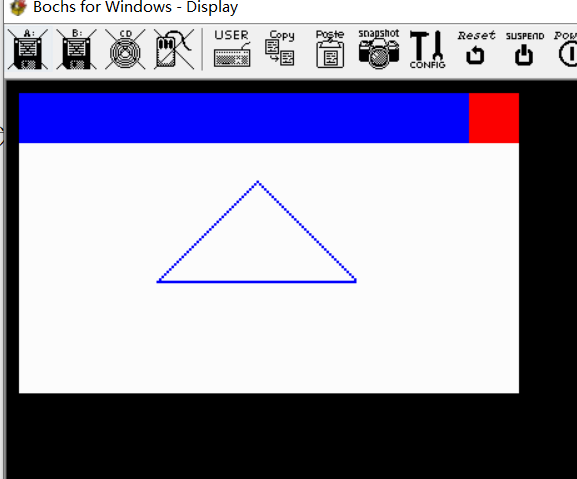
add di,8

cmp di,dot+264

jnz line

ret

按S键出现三角形（红），shift+S变为三角形（蓝）：

其模块代码：

showsjx: mov bx,[ds:di]

mov ax,[ds:di+2]

mov dx,320

mul dx

add bx,ax

mov dx,0

mov cx,[ds:di+4] ;长度

psjx: mov al,[ds:si]

mov byte[es:bx],al

add bx,[ds:di+6]

inc dx

cmp dx,cx

jnz psjx

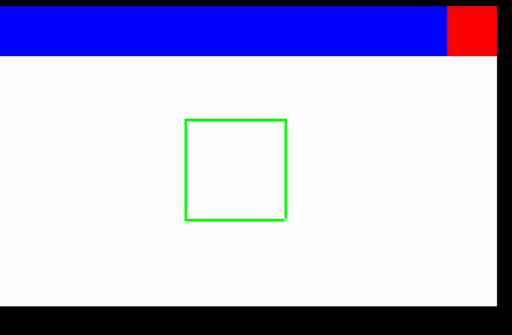
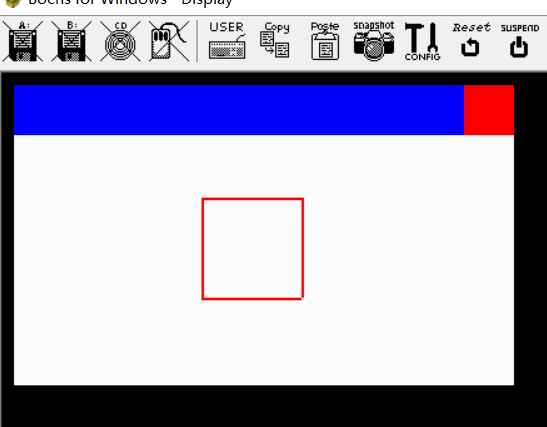
add di,8

cmp di,sjx+24

jnz showsjx

ret

按z键出现正方形（红色），shift+z（绿色）



其模块代码：

showzfx: mov bx,[ds:di]

mov ax,[ds:di+2]

mov dx,320

mul dx

add bx,ax

mov dx,0

mov cx,[ds:di+4] ;长度

pzfx: mov al,[ds:si]

mov byte[es:bx],al

add bx,[ds:di+6]

inc dx

cmp dx,cx

jnz pzfx

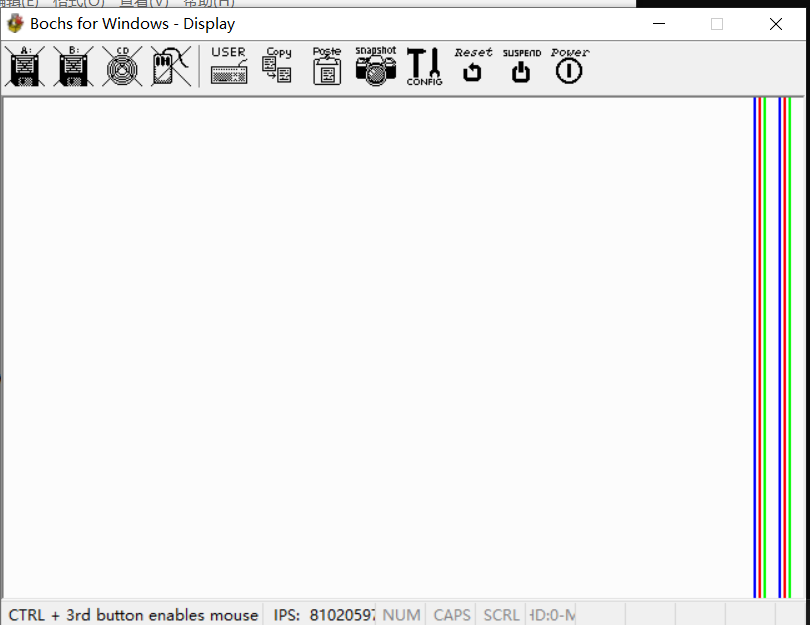
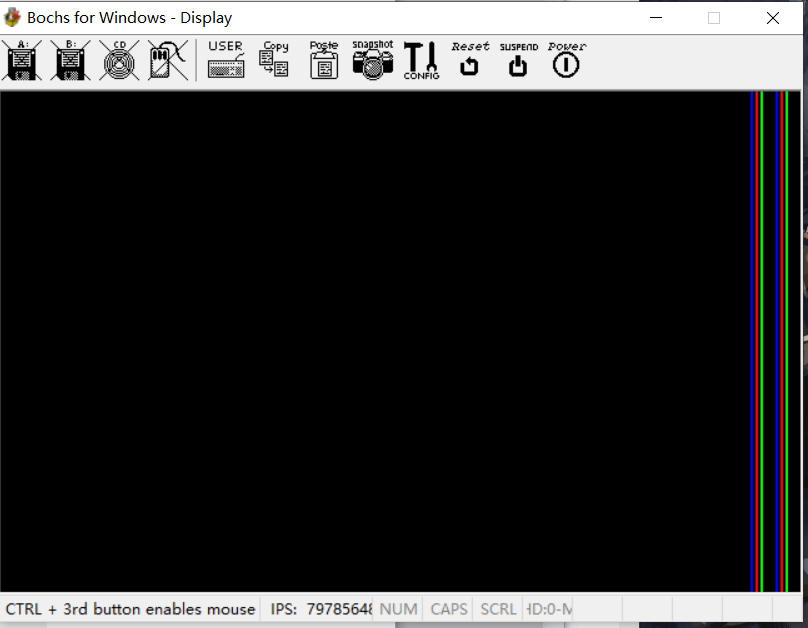
add di,8

cmp di,zfx+32

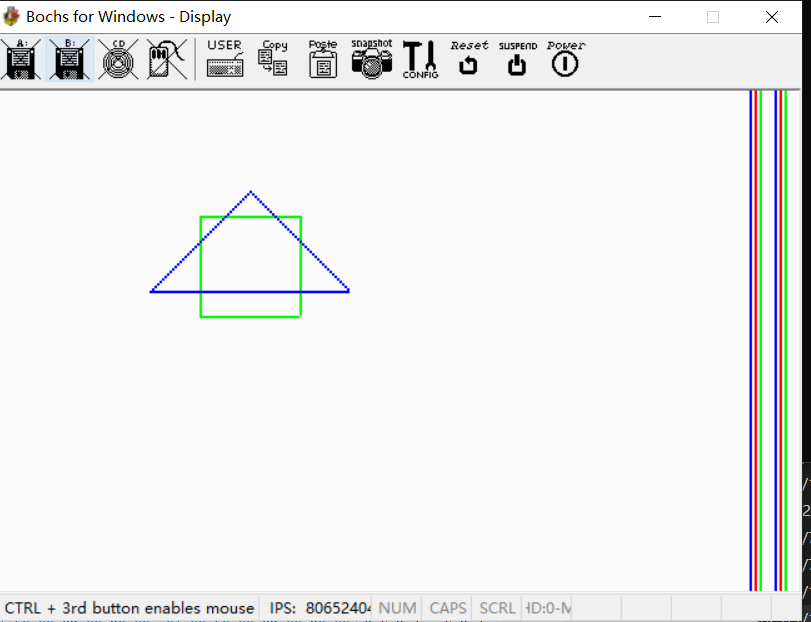
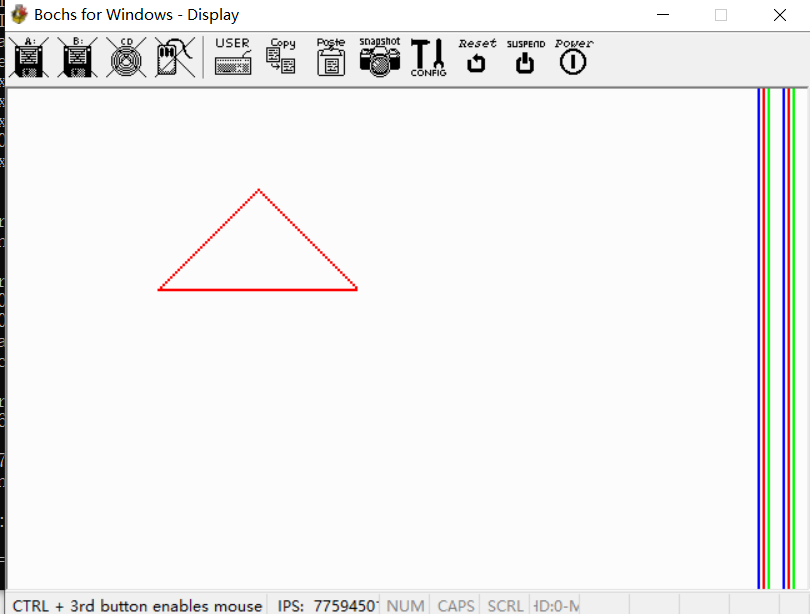
jnz showzfx

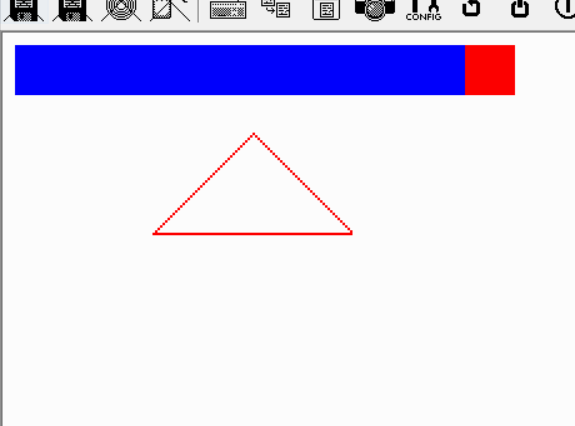
ret

按A键可以把背景变为白色，按shift+A变回黑色（每次变色右边的彩色竖线都会缓慢向下出现）：

然后按S键出现三角形，也可以组合出现



键盘响应代码：

int\_key:

mov dx,0x60

in al,dx ;读出来

cmp al,0x2a

je shiftdown ;L SHIFT键

cmp al,0xaa

je shiftup ;shift弹回

words: cmp al,0x9e

je a

cmp al,0x2c

je z

cmp al,0x1f

je s

cmp al,0x1c

je enter

cmp al,0x26

je l

jmp k1

shiftdown:

mov cl,1 ;此处按下shift

jmp words

shiftup:

mov cl,0 ;此处弹回shift

jmp k1

1. 心得体会

通过这课程学习，我了解了汇编语言的一些基本用法，但是自己综合起来写系统界面的时候我就觉得目前所学是远远不够的，也明白了设计一个系统界面所需要考虑到的东西还有很多。我在写这个课程设计的时候也遇到了很多问题，比如寄存器不够用、中断打印出错等等。后来通过一系列的查找资料和同学帮助，成功解决了。

这门课程让我学到了很多新的知识，拓展了我的知识面，引起了我新的思考，让我对微机系统与接口技术有了新的理解和体会，让我对计算机专业有了更深入的而了解。

1. 附件程序代码

org 0x8400

jmp start

colors db 1,255,255,255,2,0,0,255,3,255,0,0,4,0,255,0,5,0,0,0 ;颜色1白2蓝3红4绿5黑

colordot db 2,3,4

dot dw 40,40,40,1,60,30,25,320,60,40,22,319,60,40,22,321,50,60,20,1,70,60,10,319,60,70,20,320,55,85,5,321,40,77,40,1,110,35,10,959,110,36,10,959,110,37,10,959,100,65,14,321,116,49,18,639,116,50,18,639,98,50,19,1,120,42,10,1,130,42,45,320,118,52,12,1,118,62,12,1,122,41,20,320,122,62,9,959,122,63,9,959,122,64,9,959,124,81,6,321,135,42,50,320,135,42,10,1,145,42,10,319,135,52,10,321,145,62,5,319,135,62,5,321,190,10,11,321,200,10,11,319

zfx dw 80,50,40,1,80,50,40,320,80,90,40,1,120,50,40,320

sjx dw 100,40,40,321,100,40,40,319,60,80,80,1

;x,y,长度，类型321是\，319是/, 320是|, 1是一

colorrect db 1,2,3

rect dw 5,5,200,120,5,5,200,20,185,5,20,20 ;左上角x，y和宽，高

start :

mov ah,00h ;VGA320X200X8bit显示模式

mov al,13h ;进入13号320x200 256色的图形模式

int 10h

mov ax,0

mov ds,ax

mov si,colors

call col

mov ax,0x0a000

mov es,ax

mov cl,0

cli

mov word [ds:0x20],int\_time0

mov word [ds:0x22],0

mov sp,0

mov bx,0

mov word [ds:0x24],int\_key

mov word [ds:0x26],0

sti

jmp $

int\_key:

mov dx,0x60

in al,dx ;读出来

cmp al,0x2a

je shiftdown ;L SHIFT键

cmp al,0xaa

je shiftup ;shift弹回

words: cmp al,0x9e

je a

cmp al,0x2c

je z

cmp al,0x1f

je s

cmp al,0x1c

je enter

cmp al,0x26

je l

jmp k1

shiftdown:

mov cl,1 ;此处按下shift

jmp words

shiftup:

mov cl,0 ;此处弹回shift

jmp k1

enter:

mov di,rect

mov si,colorrect

call rectangle

mov cl,0

jmp k1

a: mov bx,0

mov ch,1

cmp cl,0

je a2

a1: mov byte[es:bx],5

inc bx

cmp bx,64000

jnz a1

jmp k1

a2: mov byte[es:bx],1

inc bx

cmp bx,64000

jnz a2

jmp k1

l: cmp cl,0

je l2

l1: mov si,colordot+2

mov di,dot

call line

mov cl,0

jmp k1

l2: mov si,colordot+0

mov di,dot

call line

mov cl,0

jmp k1

z: cmp cl,0

je z2

z1: mov si,colordot+2

mov di,zfx

call showzfx

mov cl,0

jmp k1

z2: mov si,colordot+1

mov di,zfx

call showzfx

mov cl,0

jmp k1

s: cmp cl,0

je s2

s1: mov si,colordot

mov di,sjx

call showsjx

mov cl,0

jmp k1

s2: mov si,colordot+1

mov di,sjx

call showsjx

mov cl,0

jmp k1

showzfx: mov bx,[ds:di]

mov ax,[ds:di+2]

mov dx,320

mul dx

add bx,ax

mov dx,0

mov cx,[ds:di+4] ;长度

pzfx: mov al,[ds:si]

mov byte[es:bx],al

add bx,[ds:di+6]

inc dx

cmp dx,cx

jnz pzfx

add di,8

cmp di,zfx+32

jnz showzfx

ret

showsjx: mov bx,[ds:di]

mov ax,[ds:di+2]

mov dx,320

mul dx

add bx,ax

mov dx,0

mov cx,[ds:di+4] ;长度

psjx: mov al,[ds:si]

mov byte[es:bx],al

add bx,[ds:di+6]

inc dx

cmp dx,cx

jnz psjx

add di,8

cmp di,sjx+24

jnz showsjx

ret

line: mov bx,[ds:di]

mov ax,[ds:di+2]

mov dx,320

mul dx

add bx,ax

mov dx,0

mov cx,[ds:di+4] ;长度

x: mov al,[ds:si]

mov byte[es:bx],al

add bx,[ds:di+6]

inc dx

cmp dx,cx

jnz x

add di,8

cmp di,dot+264

jnz line

ret

k1:

mov bx,0

mov dx,0x20

mov al,0x61

out dx,al ;中断结束

iret

col: mov dx,0x3c8

mov al,[ds:si]

out dx,al

mov dx,0x3c9

mov al,[ds:si+1]

out dx,al

mov dx,0x3c9

mov al,[ds:si+2]

out dx,al

mov dx,0x3c9

mov al,[ds:si+3]

out dx,al

add si,4

cmp si,colors+20

jnz col

ret

rectangle: mov bx,[ds:di]

mov ax,[ds:di+2]

mov dx,320

mul dx

add bx,ax

mov cx,[ds:di+4]

add cx,bx

mov ax,0

mov dl,[ds:si]

r: mov byte[es:bx],dl

inc bx

cmp bx,cx

jnz r

add cx,320 ;新一行

sub bx,[ds:di+4]

add bx,320

inc ax

cmp ax,[ds:di+6]

jnz r

inc si

add di,8

cmp di,rect+24

jnz rectangle

ret

int\_time0:

mov word [ds:0x20],int\_time1

mov word [ds:0x22],0

cmp bx,64000

ja x2

add bx,310

mov byte[es:bx],2

add bx,2

mov byte[es:bx],3

add bx,2

mov byte[es:bx],4

sub bx,314

add bx,320

x2: mov al,0x20

mov dx,0x20

out dx,al

iret

int\_time1:

mov word [ds:0x20],int\_time0

mov word [ds:0x22],0

cmp bx,64000

ja x3

add bx,300

mov byte[es:bx],2

add bx,2

mov byte[es:bx],3

add bx,2

mov byte[es:bx],4

sub bx,304

x3: mov al,0x20

mov dx,0x20

out dx,al

iret