****

浙江农林大学

课程报告

（ 2020-2021 学年第 一 学期）

**课程名称：** 微机系统与接口技术

**姓　　名：** 泮璐洁

**学　　号：** 201805120506

**专 业：** 计算机科学与技术183班

**指导老师：** 舒洪波

**2021年 1月 20号**

目录

[目录 2](#_Toc62241221)

[1课程目的 3](#_Toc62241222)

[2 程序总体介绍 3](#_Toc62241223)

[3 大体功能模块 3](#_Toc62241224)

[3.1修改颜色 3](#_Toc62241225)

[3.2 矩阵输出 6](#_Toc62241226)

[3.3 图形移动 6](#_Toc62241227)

[4 运用到的所学的知识 6](#_Toc62241228)

[5 操作实现流程图 7](#_Toc62241229)

[5.1 操作流程图 7](#_Toc62241230)

[5.2 运行截图 8](#_Toc62241231)

[6 心得体会 12](#_Toc62241232)

[7 附件程序代码 12](#_Toc62241233)

# 1课程目的

课程设计的主要目的是通过本课程设计，将微机课上所学知识转化为实践经验，并藉此进一步掌握计算机汇编语言编程技术，实现基本的功能;在编程的过程中，提高自己的动手能力，以及分析和解决问题的能力，为将来能够开发出更完整的系统打好基础。

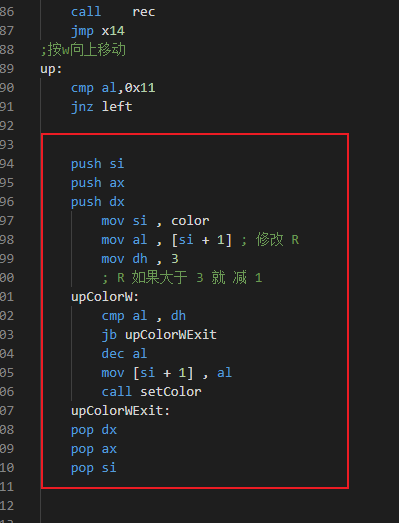
# 2 程序总体介绍

首先，打开bochs，直接进入开始界面，按Q出现操作的主界面，通过方块的下移和右移划线，通过方块的左移和上移擦除，按Q可以清空界面，重新开始划线。按Z结束该程序，出现再见界面。

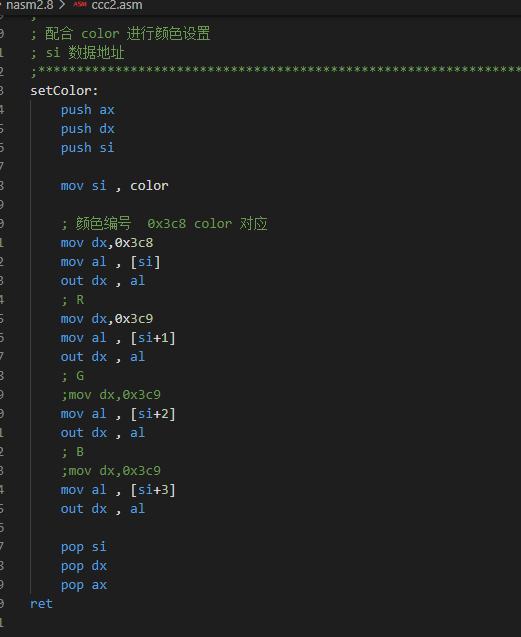
# 3 大体功能模块

## 3.1修改颜色

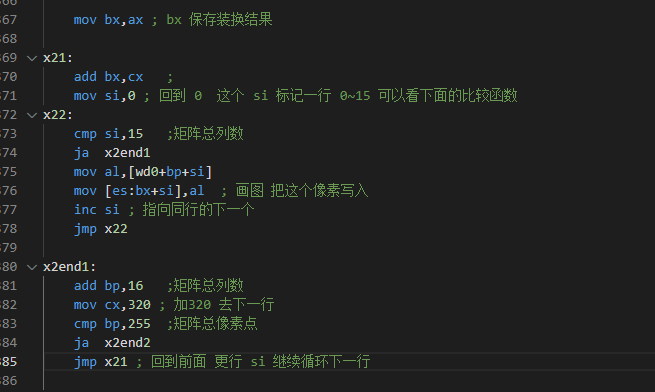
读取并修改内存中的颜色数值



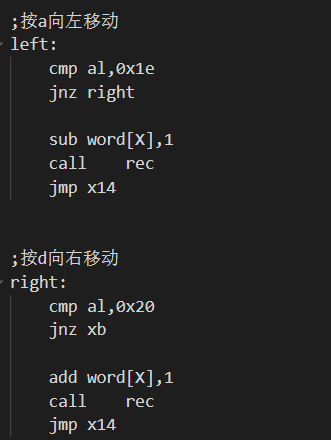
颜色写入函数，读取color内存端的数据



## 3.2 矩阵输出



## 3.3 图形移动

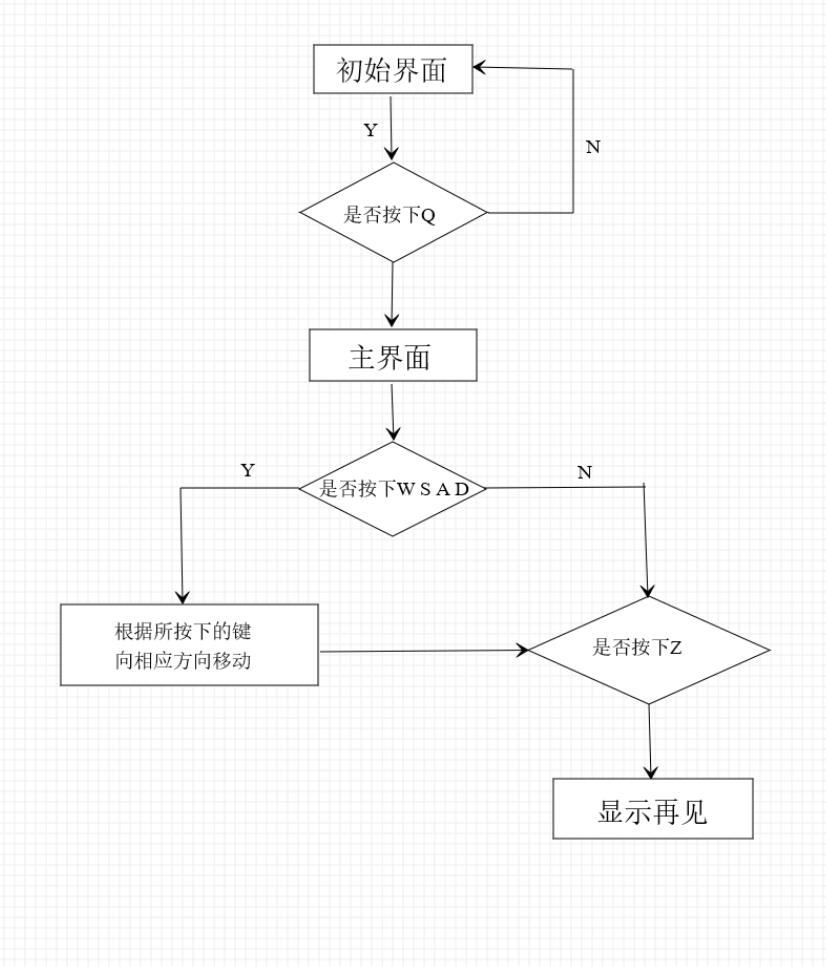


# 4 运用到的所学的知识

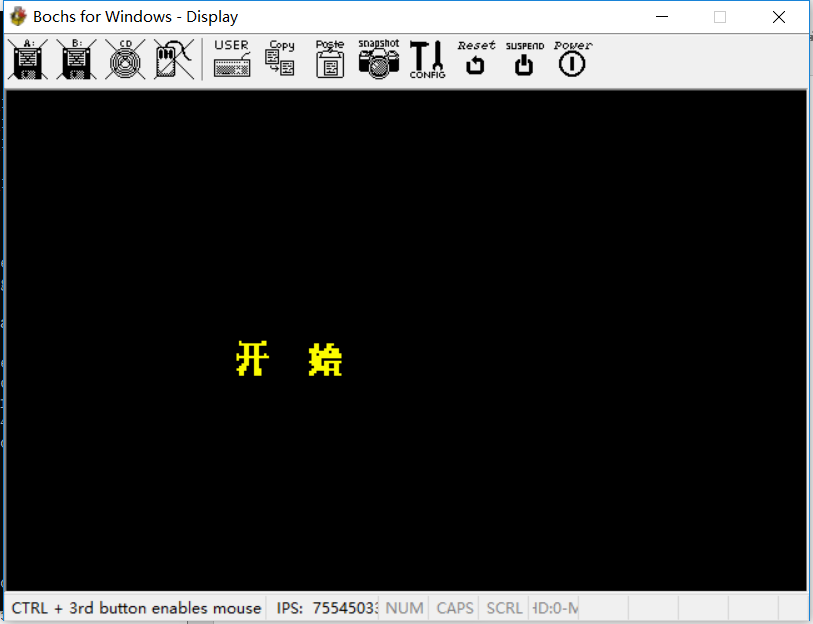
键盘中断，图形界面，颜色改变等。

# 5 操作实现流程图

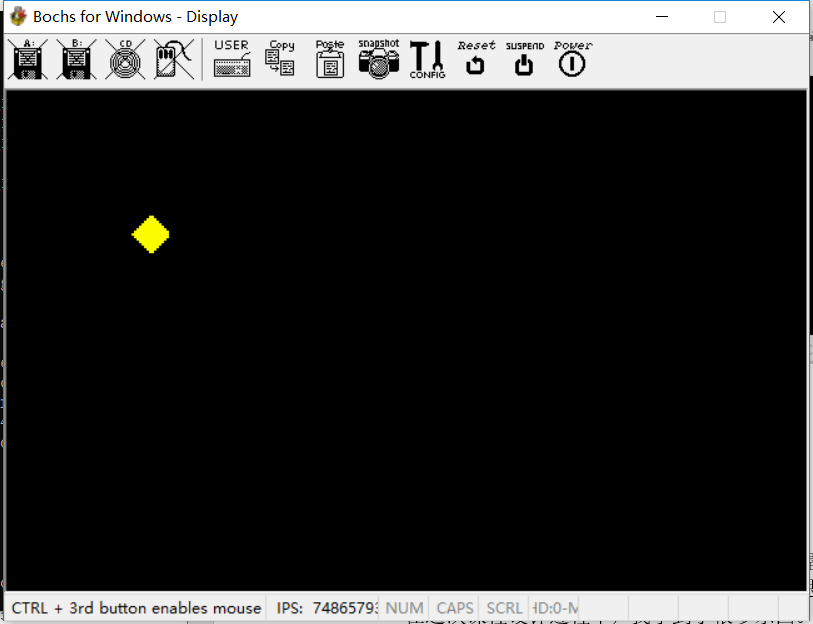
## 5.1 操作流程图



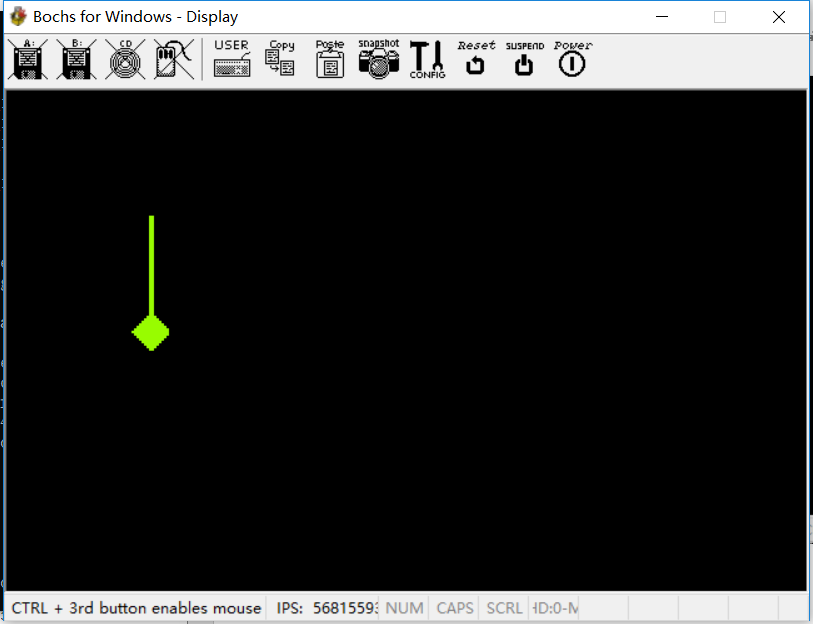
## 5.2 运行截图

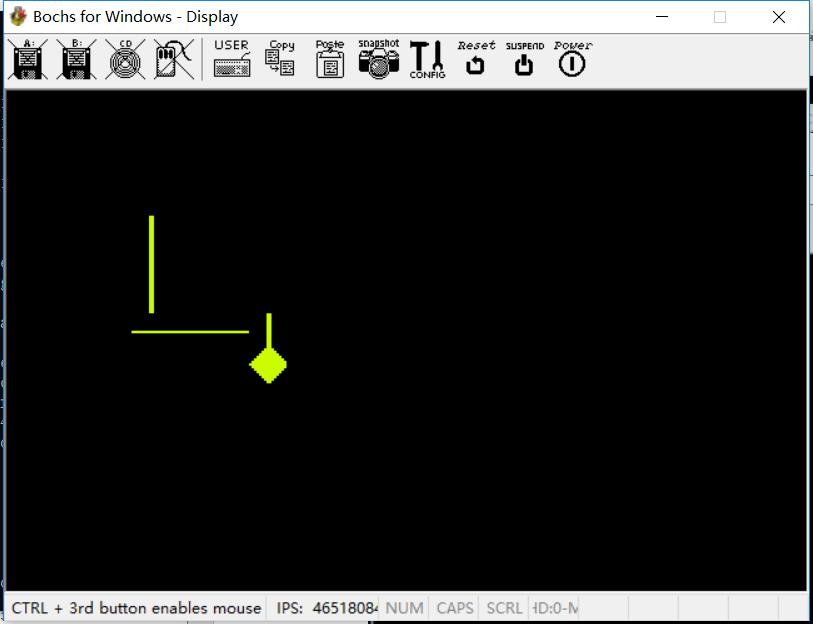


按Q进入主界面

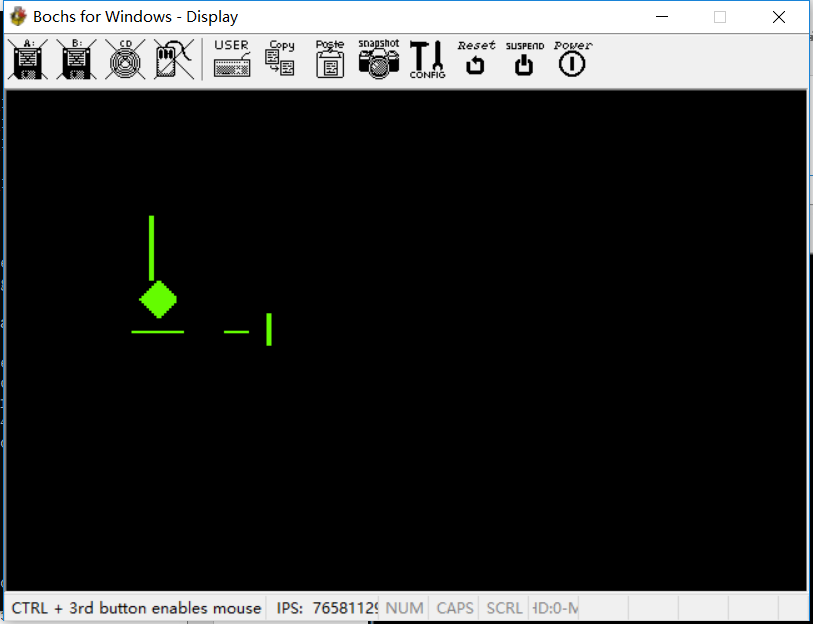


通过W S A D 四个键来控制方块的移动，且边移动边变色。

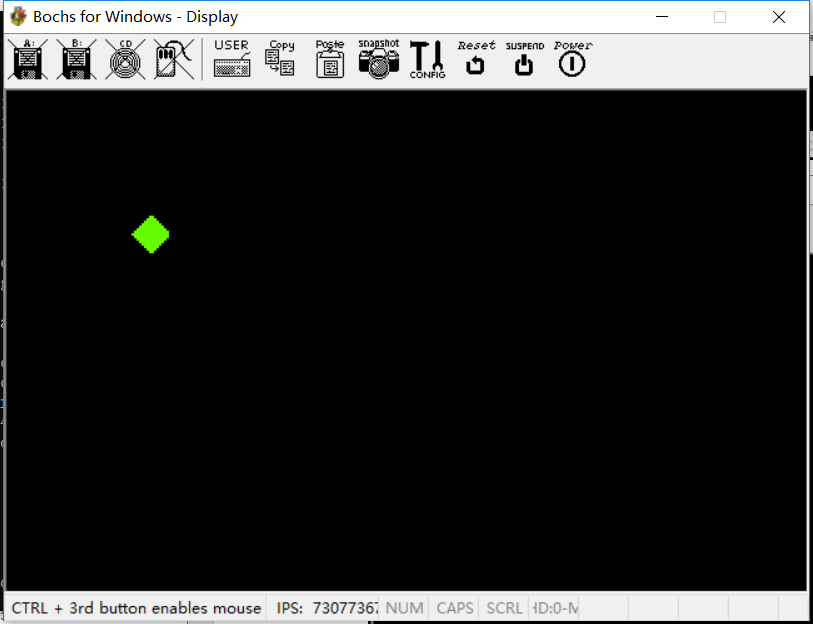
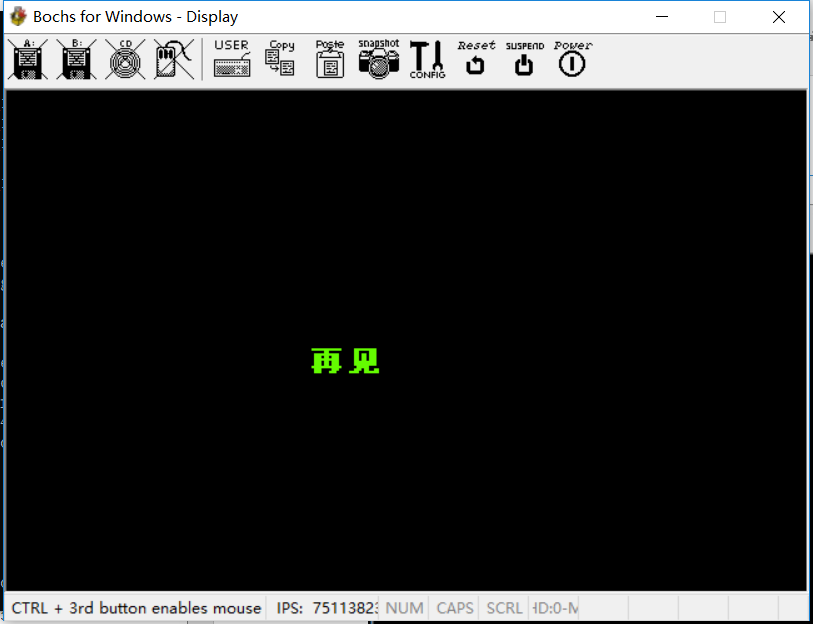




右移下移划线，左移上移擦线



按Q重置清屏

按Z进入再见界面

# 6 心得体会

本次课程设计综合运用了所学习的知识，掌握了调色、中断、绘制图形等过程。从构建颜色块到移动位置，我的编程能力和思维都得到了锻炼。

在这次课程设计过程中，我学到了很多东西。首先是对这一个学期所学的内容进行了大体回顾，在做的过程中，构建设计思路，再结合自己先前在课上打下的代码及一些平时作业，对最终的大作业进行代码编写。在编写代码过程中，也遇到了非常多的问题，包括对一些定义的不理解，对中断概念生疏，逻辑捋不清等等。微机编程语言与 C 语言有极大的不同，于我而言，理解和应用起来稍有难度，我所了解到的内容依旧很浅，仍需自行深入学习。

最后，感谢舒老师在课程上的谆谆教诲，非常耐心的为我们解答疑难问题，也正是日常在老师指导下对各个模块的学习积累，才让我顺利完成本次的课题。

# 7 附件程序代码

org 0x8400

jmp start

X dw 0

Y dw 0

; R G B

color db 49, 127 , 127, 128 ; 画笔颜色

;开

;================================================================================

wd0 db '0001000000001100'

db '0001111111111100'

db '0000011001100000'

db '0000011001100000'

db '0000011001100000'

db '0010011001100100'

db '0011111111111110'

db '0010011001100000'

db '0000011001100000'

db '0000111001100000'

db '0001110001100000'

db '0011110001110000'

db '0001100001110000'

db '0001100001110000'

db '0000000000000000'

db '0000000000000000'

;始

;================================================================================

wd1 db '0000000000000000'

db '0001100011100000'

db '0001100011000000'

db '0111100111111000'

db '0111111110111000'

db '0011011111111100'

db '0011111111111100'

db '0011110000000000'

db '0111110111111100'

db '0011110110011000'

db '0001111110011000'

db '0011111111111000'

db '0111000110011000'

db '0110000110011100'

db '0000000000000000'

db '0000000000000000'

;再见

;================================================================================

wd2 db '00000000000000000000000000000000'

db '00000000000000000000000000000000'

db '00000000000000000000000000000000'

db '00111111111111000001111111110000'

db '00000001100000000001100001110000'

db '00011111111110000001101101110000'

db '00011101101110000001101101110000'

db '00011111111110000001101101110000'

db '00011101101110000001111111110000'

db '00011101101110000001111111110000'

db '00111111111111000000111110000000'

db '00011100001110000001110110011000'

db '00011100011100000111100111111000'

db '00000000000000000000000000000000'

db '00000000000000000000000000000000'

db '00000000000000000000000000000000'

;方块块

;================================================================================

wd3 db '0000000110000000'

db '0000001111000000'

db '0000011111100000'

db '0000111111110000'

db '0001111111111000'

db '0011111111111100'

db '0111111111111110'

db '1111111111111110'

db '0111111111111110'

db '0011111111111100'

db '0001111111111000'

db '0000111111110000'

db '0000011111100000'

db '0000001111000000'

db '0000000110000000'

db '0000000000000000'

;主函数

;================================================================================

start:

sti

mov al,0x13

mov ah,0x00

int 0x10

mov ax,0

mov ds, ax

mov bx,0

call draw ;上色

mov ax,0x0a000;往显存(0xa0000开始)物理地址

mov es,ax

mov cx,0

mov ds,cx

mov word[ds:X],90 ;字符左上角起点x

mov word[ds:Y],100 ;字符左上角起点y

call kai ;打字 “开”

mov word[ds:X],120 ;字符左上角起点x

mov word[ds:Y],100 ;字符左上角起点y

call shi ;打字“始”

mov bx,0

mov word[ds:0x24],int\_key ;中断 进入方块块模式

mov word[ds:0x26],0

jmp $

;================================================================================

;按q出现方块块

;================================================================================

int\_key:

push ax

push si

mov dx,0x60

in al,dx

cmp al,0x10 ;10——Q按下

jnz down

mov si,0

clear:

cmp si,320\*200 ;清屏

ja x

mov al,48

mov [es:si],al

inc si

jmp clear

x:

mov word[ds:X],50 ;方块左上角起点x

mov word[ds:Y],50 ;方块左上角起点y

call rec ;调用方块块

jmp x14

;================================================================================

;上下左右的移动

;按s向下移动

down:

cmp al,0x1f

jnz up

; 屏蔽 si

push si

push ax

push dx

mov si , color

mov al , [si + 1] ; 修改 R

; R 如果小于 200 就 加 1

mov dh , 200

upColorS:

cmp al , dh

ja upColorSExit

inc al

mov [si + 1] , al

call setColor

upColorSExit:

pop dx

pop ax

pop si

add word[Y],1

call rec

jmp x14

;按w向上移动

up:

cmp al,0x11

jnz left

;读取并修改内存中的颜色数值

push si

push ax

push dx

mov si , color

mov al , [si + 1] ; 修改 R

mov dh , 3

; R 如果大于 3 就 减 1

upColorW:

cmp al , dh

jb upColorWExit

dec al ;改变颜色变化速度,或者换成其他数值

mov [si + 1] , al

call setColor ;把颜色数值写入到画笔 （49号里）

upColorWExit:

pop dx

pop ax

pop si

sub word[Y],1

call rec

jmp x14

;按a向左移动

left:

cmp al,0x1e

jnz right

sub word[X],1

call rec

jmp x14

;按d向右移动

right:

cmp al,0x20

jnz xb

add word[X],1

call rec

jmp x14

;================================================================================

;按c跳出再见

;================================================================================

xb:

cmp al,0x2c ;按下Z——0x2c

jnz x14

mov si,0

xcf:

cmp si,320\*200

ja x01

mov al,48

mov [es:si],al

inc si

jmp xcf

x01:

mov word[ds:X],120 ;汉字左上角起点x

mov word[ds:Y],100 ;汉字左上角起点y

call bye

x14:

mov dx,0x20

mov al,0x61

out dx,al

pop si

pop ax

iret

;================================================================================

;调节颜色板块

;================================================================================

draw:

;调节背景颜色（R B G）

mov al,48

mov dx,0x3c8

out dx,al

mov al,0

mov dx,0x3c9

out dx,al

mov al,0

mov dx,0x3c9

out dx,al

mov al,0

mov dx,0x3c9

out dx,al

;================================================================================

;调节字体颜色（R B G）

mov al,49

mov dx,0x3c8

out dx,al

mov al,127

mov dx,0x3c9

out dx,al

mov al,127

mov dx,0x3c9

out dx,al

mov al,128

mov dx,0x3c9

out dx,al

ret

;再见

;================================================================================

bye:

push ax

push si

push bx

push bp

push cx

mov bp,0

mov cx,0

mov ax,[Y]

mov bx,320

mul bx

add ax,[X]

mov bx,ax

x11:

add bx,cx

mov si,0

x12:

cmp si,31 ;矩阵总列数

ja x1end1

mov al,[wd2+bp+si]

mov [es:bx+si],al

inc si

jmp x12

x1end1:

add bp,32 ;矩阵总列数

mov cx,320

cmp bp,511 ;矩阵总像素点

ja x1end2

jmp x11

x1end2:

pop cx

pop bp

pop bx

pop si

pop ax

ret

;开

;================================================================================

kai:

push ax

push si

push bx

push bp

push cx

mov bp,0

mov cx,0

mov ax,[Y]

mov bx,320

mul bx

add ax,[X]

mov bx,ax ; bx 保存装换结果

x21:

add bx,cx

mov si,0 ; 回到 0 这个 si 标记一行 0~15 可以看下面的比较函数

x22:

cmp si,15 ;矩阵总列数

ja x2end1

mov al,[wd0+bp+si]

mov [es:bx+si],al ; 画图 把这个像素写入

inc si ; 指向同行的下一个

jmp x22

x2end1:

add bp,16 ;矩阵总列数

mov cx,320 ; 加320 去下一行

cmp bp,255 ;矩阵总像素点

ja x2end2

jmp x21 ; 回到前面 更行 si 继续循环下一行

x2end2:

pop cx

pop bp

pop bx

pop si

pop ax

ret

;始

;================================================================================

shi:

push ax

push si

push bx

push bp

push cx

mov bp,0

mov cx,0

mov ax,[Y]

mov bx,320

mul bx

add ax,[X]

mov bx,ax

x31:

add bx,cx

mov si,0

x32: cmp si,15 ;矩阵总列数减一

ja x3end1

mov al,[wd1+bp+si]

mov [es:bx+si],al

inc si

jmp x32

x3end1: add bp,16 ;矩阵总列数

mov cx,320

cmp bp,255 ;矩阵总像素点数减一

ja x3end2

jmp x31

x3end2:

pop cx

pop bp

pop bx

pop si

pop ax

ret

;方块块

;================================================================================

rec:

push ax

push si

push bx

push bp

push cx

mov bp,0

mov cx,0

mov ax,[Y]

mov bx,320

mul bx

add ax,[X]

mov bx,ax

x41:

add bx,cx

mov si,0

x42:

cmp si,15 ;矩阵总列数减一

ja x4end1

mov al,[wd3+bp+si]

mov [es:bx+si],al

inc si

jmp x42

x4end1:

add bp,16 ;矩阵总列数

mov cx,320

cmp bp,255 ;矩阵总像素点减一

ja x4end2

jmp x41

x4end2:

pop cx

pop bp

pop bx

pop si

pop ax

ret

;================================================================================

;================================================================================

; 配合 color 进行颜色设置

; si 数据地址

;================================================================================

setColor:

push ax

push dx

push si

mov si , color

; 颜色编号 0x3c8 color 对应

mov dx,0x3c8

mov al , [si]

out dx , al

; R

mov dx,0x3c9

mov al , [si+1]

out dx , al

; G

;mov dx,0x3c9

mov al , [si+2]

out dx , al

; B

;mov dx,0x3c9

mov al , [si+3]

out dx , al

pop si

pop dx

pop ax

ret