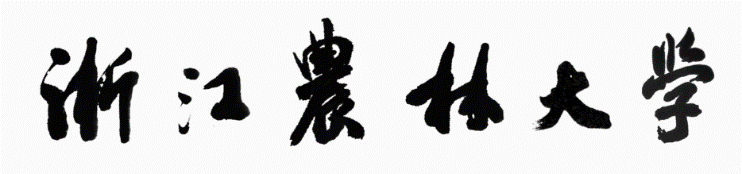
**** 

**《微机系统和接口技术》课程设计报告**

**（2020-2021学年第一学期）**

**学生姓名： 王雅洁、张丽雅**

**学生学号： 201805120605、201805020217**

**学生班级： 计算机183班**

**学院名称： 信息工程学院**

**指导教师： 舒红波**

**目录**

**[一、 程序总体介绍 3](#_Toc17625)**

**[二、 大体功能模块 3](#_Toc25030)**

**[三、 运用到的所学的知识 3](#_Toc26431)**

**[四、 操作实现流程图（操作->代码->贴图） 3](#_Toc12874)**

**[（一） 操作流程图 3](#_Toc11834)**

**[（二）效果图+对应代码 4](#_Toc5351)**

**[五、 心得体会 11](#_Toc15445)**

**[六、 附件程序代码 12](#_Toc13017)**

## 

## 程序总体介绍

音乐播放加图像变换小程序

## **大体功能模块**

1.欢迎界面

2.主界面

3.声音播放

4.图像显示和颜色变换

## **运用到的所学的知识**

1.文本模式

2.图形模式（自定义颜色）

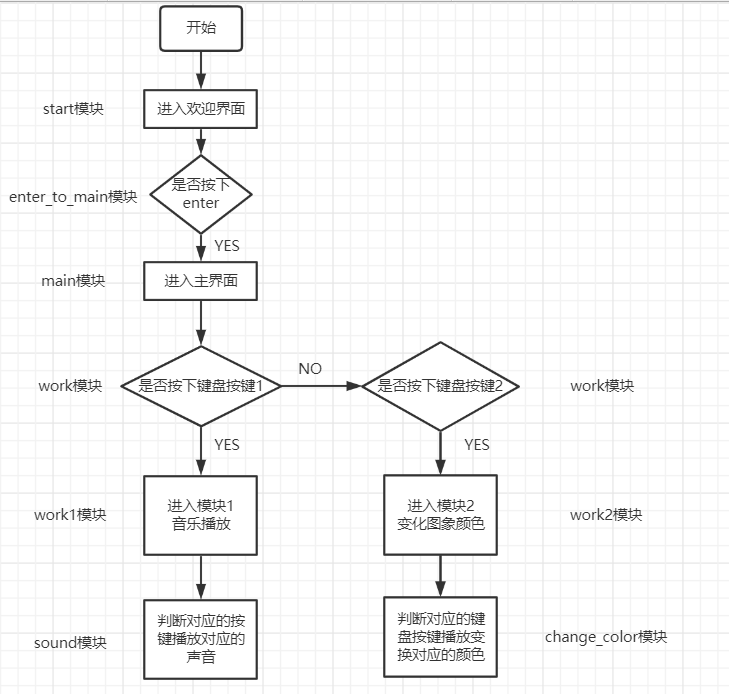
3.时钟中断

4.键盘中断

5.扬声器

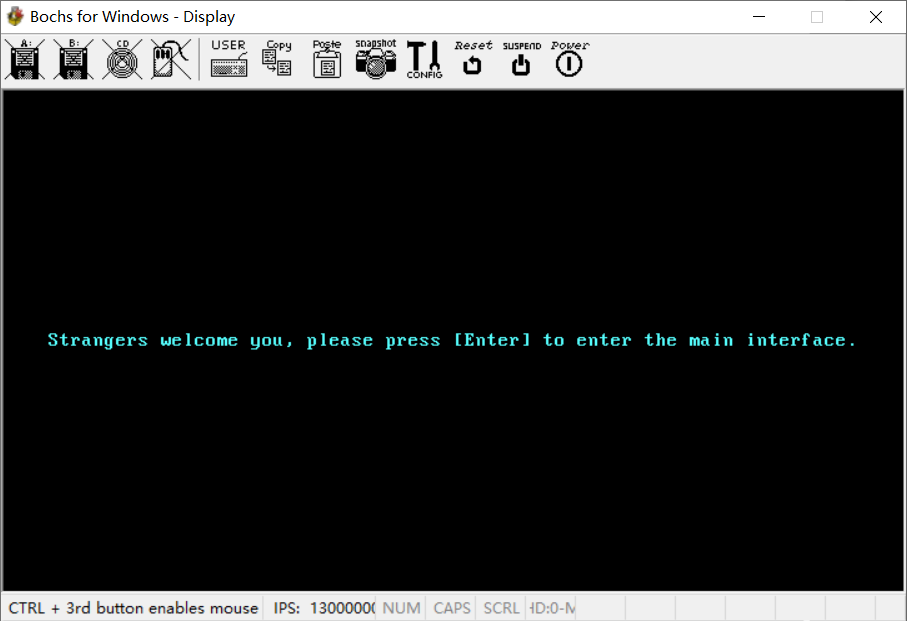
## 操作实现流程图（操作->代码->贴图）

1. 操作流程图



（二）效果图+对应代码

1.欢迎界面（张丽雅负责）



对应代码

start:

mov ax,0xb800

mov es,ax

mov ax,0

mov ds,ax

call clear\_screen ; 清屏

call welcome ; 欢迎界面

sti

mov word[ds:0x24],int\_key ; 键盘中断，是否进入主界面

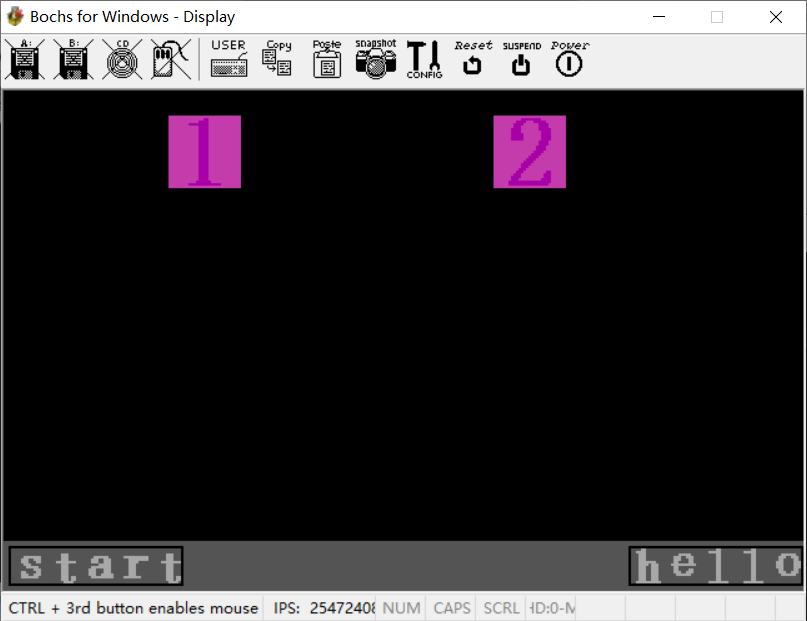
mov word[ds:0x26],0

enter\_to\_main:

cmp byte[ds:flag\_num],0x1c ; 回车

jne enter\_to\_main

2.主界面（王雅洁负责）



对应代码

main:

mov ah,00h

mov al,13h

int 10h

mov ax,0xa000

mov es,ax

mov ax,0

mov ds,ax

mov si,mycolor

call editcolor

call draw\_num

call taskbar

mov ah,7

call draw\_start

mov word[ds:0x20],int\_time ; 时钟中断

mov word[ds:0x22],0

mov word[ds:0x24],int\_key ; 键盘中断，是否进入主界面

mov word[ds:0x26],0

work:

cmp byte[ds:flag\_num],0x82 ; 1，work1

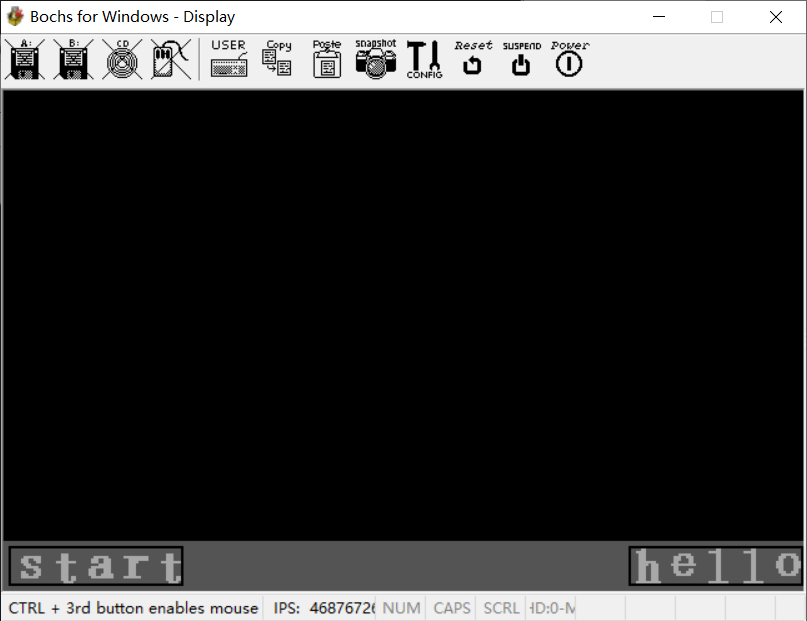
je work1

cmp byte[ds:flag\_num],0x83 ; 2，work2

je work2

jne work

1. 声音模块界面（王雅洁负责）



对应代码

work1:

mov ah,00h

mov al,13h

int 10h

mov ax,0xa000

mov es,ax

mov ax,0

mov ds,ax

mov si,mycolor

call editcolor

call taskbar

mov ah,7

call draw\_start

sound:

cmp byte[ds:flag\_num],0x9e ; 字母a

je key\_a

cmp byte[ds:flag\_num],0x9f ; 字母s

je key\_s

cmp byte[ds:flag\_num],0xa0 ; 字母d

je key\_d

cmp byte[ds:flag\_num],0xa1 ; 字母f

je key\_f

cmp byte[ds:flag\_num],0xa2 ; 字母g

je key\_g

cmp byte[ds:flag\_num],0xa3 ; 字母h

je key\_h

cmp byte[ds:flag\_num],0xa4 ; 字母j

je key\_j

cmp byte[ds:flag\_num],0xa5 ; 字母k

je key\_k

cmp byte[ds:flag\_num],0xa6 ; 字母l

je key\_l

cmp byte[ds:flag\_num],0x81 ; esc

je main

jne sound

jmp $

key\_a:

mov bx,110

jmp play

key\_s:

mov bx,220

jmp play

key\_d:

mov bx,440

jmp play

key\_f:

mov bx,880

jmp play

key\_g:

mov bx,1760

jmp play

key\_h:

mov bx,1864

jmp play

key\_j:

mov bx,1975

jmp play

key\_k:

mov bx,3951

jmp play

key\_l:

mov bx,3520

play:

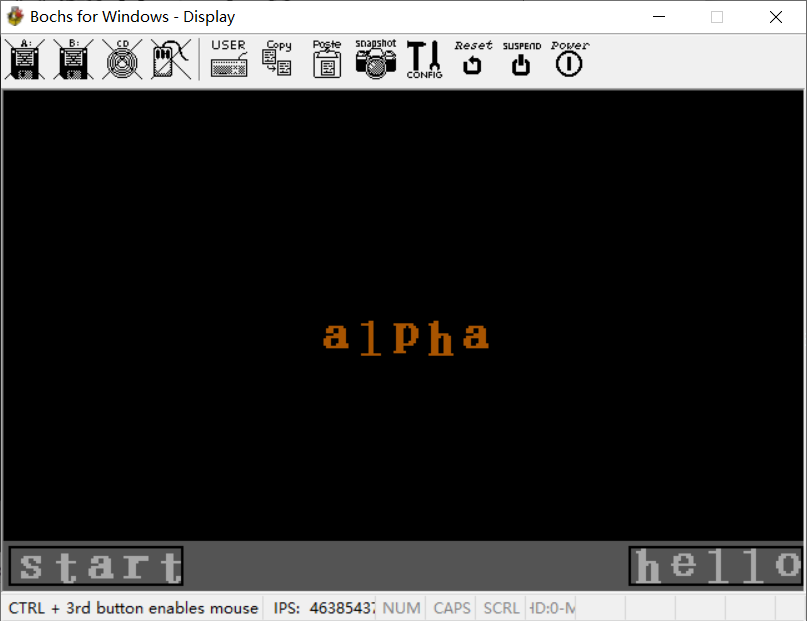
call change\_sound

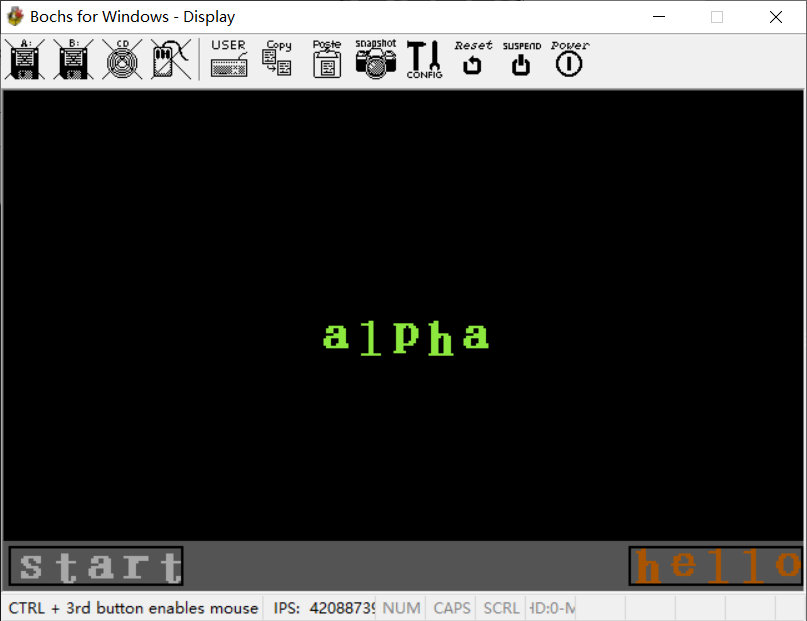
call play\_sound

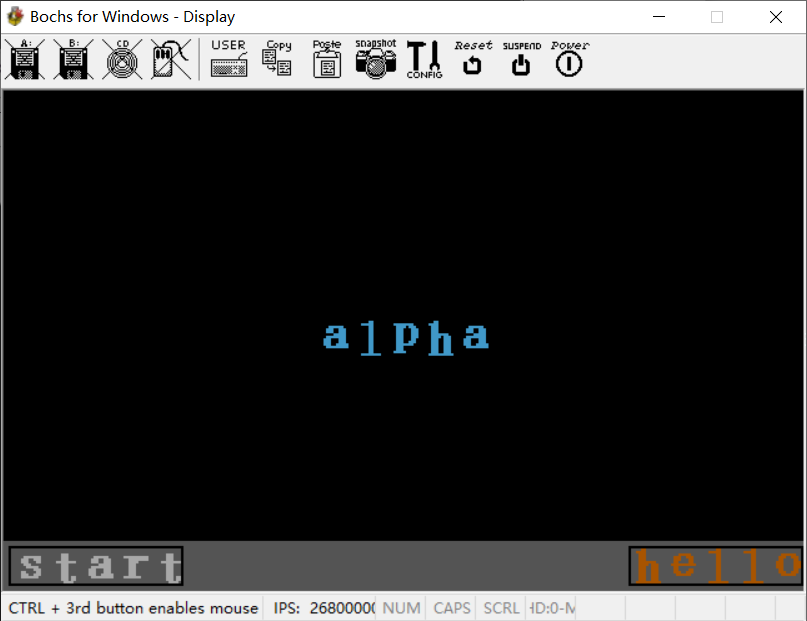
call clear\_flagnum

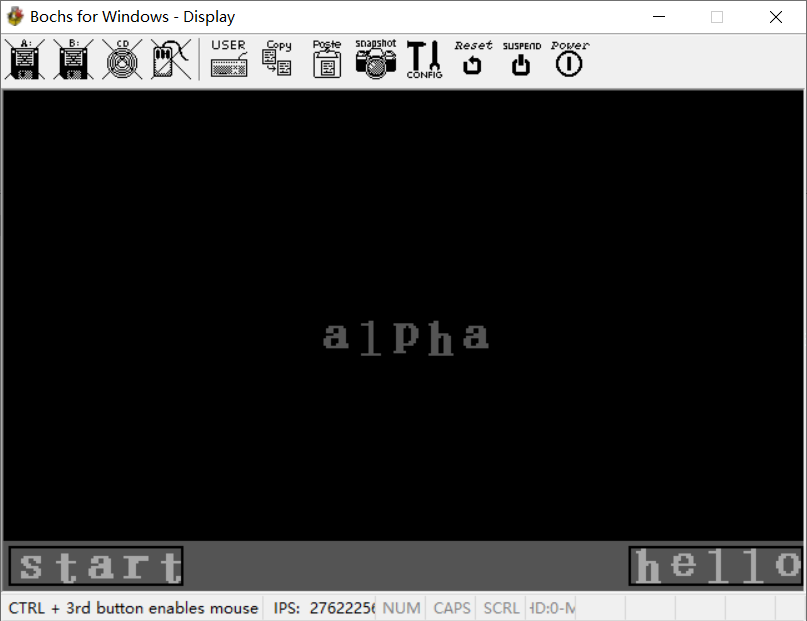
jmp sound

1. 颜色变化（张丽雅负责）









对应代码

work2:

mov ah,00h

mov al,13h

int 10h

mov ax,0xa000

mov es,ax

mov ax,0

mov ds,ax

mov si,mycolor

call editcolor

call taskbar

mov ah,7

call draw\_start

mov ah,6

call draw\_alpha

change\_color:

cmp byte[ds:flag\_num],0x93 ; 字母r

je red

cmp byte[ds:flag\_num],0xa2 ; 字母g

je green

cmp byte[ds:flag\_num],0xb0 ; 字母b

je blue

cmp byte[ds:flag\_num],0x95 ; 字母y

je yellow

cmp byte[ds:flag\_num],0x81 ; esc

je main

jne change\_color

jmp $

red:

mov ah,1

call draw\_alpha

jmp change\_color

green:

mov ah,2

call draw\_alpha

jmp change\_color

blue:

mov ah,3

call draw\_alpha

jmp change\_color

yellow:

mov ah,8

call draw\_alpha

jmp change\_color

## **心得体会**

本次课程设计是一个对平时学到的知识进行综合并且实践的过程。通过对这程序的制作，使我们对汇编语言的基本知识的使用更加熟练，同时通过此次课程设计，我理解了舒老师一直强调的理论和实践结合的重要性，只有理论知识是远远不够的，只有把所学的理论知识与实践相结合起来，从理论中得出结论，才能真正掌握专业知识，夯实专业基础，提高相关专业技能。在此次程序设计中，我们也遇到了一些小问题，比如：在运用到键盘中断相关知识去实现音乐播放相关功能时，一开始我们没有设置清空模块，然后音乐会停不下来一直循环播放。不过通过组内的讨论和合作我们顺利解决了这个问题，通过编写clear\_flagnum模块清除相关标记。

同时，我们还了解了在微机中，最基础的语言是汇编语言，汇编语言是处在机器语言和高级语言中间的一种低级语言。由于计算机能够直接识别的数据是由二进制数О和1组成的代码。所以用机器语言编写的程序是计算机惟一能够直接识别并执行的程序，而用其他语言编写的程序必须经过翻译才能变换成机器语言程序，为了克服机器语言的缺点，人们采用助记符表示机器指令的操作码，用变量代替操作数的存放地址等，这样就形成了汇编语言。

课程设计是培养我们综合运用所学知识，发现、提出、分析、解决问题的一个过程，是对我们所学知识及综合能力的一次考察。此次实践让我们对这学期学习的知识有了一个总结和再次的查漏补缺，也学习到了一些新知识。虽然，在前面的一些课程学习时我们会对一些抽象概念的理解有些模糊，不过通过此次实践解决了这个问题。舒老师是一个非常有教学经验的教师，老师一直在锻炼我们的学习能力，无论是在平时的课程中还是在独自完成这次课设下来，我都深刻体会到了老师的良苦用心，为了完成课设，我们会自主学习许多相关知识，也因此也收获了很多。其实计算机专业就是这样，我们的知识每天都先飞速地翻新，我们免不了要接触许多新鲜的科技，全新的知识，所以需要我们具备非常强的自主学习能力，不能够一直依赖老师手把手的教学。这也正是舒老师一直锻炼我们自主学习能力的主要原因，所以非常感谢老师让我们意识到了这一点。

## 附件程序代码

org 0x8400

jmp start

str\_welcome db "Strangers welcome you, please press [Enter] to enter the main interface."

pos dw 28925 ; 90\*320+125

hello\_pos dw 58170

start\_pos dw 58243

flag\_num db 0

flag db 0

mycolor db 1,242,24,16 ; red

db 2,35,186,15 ; green

db 3,16,38,242 ; blue

db 4,241,207,107 ; 图标背景色

db 5,112,113,116 ; 主页字体颜色

db 6,0,0,0

db 7,255,255,255

db 8,221,223,19

a db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,1,1,1,1,1,1,0,0,0,0

db 0,0,0,1,1,1,0,0,1,1,1,0,0,0

db 0,0,0,1,1,1,0,0,1,1,1,0,0,0

db 0,0,0,0,0,0,1,1,1,1,1,0,0,0

db 0,0,0,1,1,1,1,1,1,1,1,0,0,0

db 0,0,1,1,1,0,0,0,1,1,1,0,0,0

db 0,0,1,1,1,0,0,0,1,1,1,0,0,0

db 0,0,1,1,1,0,0,1,1,1,1,1,0,0

db 0,0,0,1,1,1,1,1,1,1,1,1,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

l db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,1,1,0,0,0,0,0,0

db 0,0,0,1,1,1,1,1,0,0,0,0,0,0

db 0,0,0,0,0,0,1,1,0,0,0,0,0,0

db 0,0,0,0,0,0,1,1,0,0,0,0,0,0

db 0,0,0,0,0,0,1,1,0,0,0,0,0,0

db 0,0,0,0,0,0,1,1,0,0,0,0,0,0

db 0,0,0,0,0,0,1,1,0,0,0,0,0,0

db 0,0,0,0,0,0,1,1,0,0,0,0,0,0

db 0,0,0,0,0,0,1,1,0,0,0,0,0,0

db 0,0,0,0,0,0,1,1,0,0,0,0,0,0

db 0,0,0,0,0,0,1,1,0,0,0,0,0,0

db 0,0,0,0,0,0,1,1,0,0,0,0,0,0

db 0,0,0,0,0,0,1,1,0,0,0,0,0,0

db 0,0,0,1,1,1,1,1,1,1,1,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

p db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,1,1,1,1,1,1,1,1,0,0,0,0

db 0,0,0,1,1,1,1,0,1,1,1,0,0,0

db 0,0,0,1,1,1,0,0,0,1,1,1,0,0

db 0,0,0,1,1,1,0,0,0,1,1,1,0,0

db 0,0,0,1,1,1,0,0,0,1,1,1,0,0

db 0,0,0,1,1,1,0,0,0,1,1,1,0,0

db 0,0,0,1,1,1,0,0,0,1,1,0,0,0

db 0,0,0,1,1,1,0,0,1,1,1,0,0,0

db 0,0,0,1,1,1,1,1,1,1,0,0,0,0

db 0,0,0,1,1,1,0,0,0,0,0,0,0,0

db 0,0,0,1,1,1,0,0,0,0,0,0,0,0

db 0,0,1,1,1,1,1,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

h db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,1,1,0,0,0,0,0,0,0,0

db 0,0,1,1,1,1,0,0,0,0,0,0,0,0

db 0,0,0,1,1,1,0,0,0,0,0,0,0,0

db 0,0,0,1,1,1,0,0,0,0,0,0,0,0

db 0,0,0,1,1,1,0,0,0,0,0,0,0,0

db 0,0,0,1,1,1,1,1,1,1,0,0,0,0

db 0,0,0,1,1,1,1,0,1,1,1,0,0,0

db 0,0,0,1,1,1,0,0,1,1,1,0,0,0

db 0,0,0,1,1,1,0,0,1,1,1,0,0,0

db 0,0,0,1,1,1,0,0,1,1,1,0,0,0

db 0,0,0,1,1,1,0,0,1,1,1,0,0,0

db 0,0,0,1,1,1,0,0,1,1,1,0,0,0

db 0,0,0,1,1,1,0,0,1,1,1,0,0,0

db 0,0,1,1,1,1,1,1,1,1,1,1,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

e db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,1,1,1,1,1,0,0,0,0

db 0,0,0,1,1,1,0,0,1,1,1,0,0,0

db 0,0,0,1,1,1,0,0,0,1,1,0,0,0

db 0,0,0,1,1,0,0,0,0,1,1,0,0,0

db 0,0,1,1,1,1,1,1,1,1,1,1,0,0

db 0,0,0,1,1,0,0,0,0,0,0,0,0,0

db 0,0,0,1,1,0,0,0,0,1,1,0,0,0

db 0,0,0,1,1,1,0,0,1,1,1,0,0,0

db 0,0,0,0,1,1,1,1,1,1,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

o db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,1,1,1,1,1,1,0,0,0,0

db 0,0,0,1,1,1,0,0,1,1,1,0,0,0

db 0,0,0,1,1,0,0,0,0,1,1,0,0,0

db 0,0,1,1,1,0,0,0,0,1,1,1,0,0

db 0,0,1,1,1,0,0,0,0,1,1,1,0,0

db 0,0,1,1,1,0,0,0,0,1,1,1,0,0

db 0,0,0,1,1,0,0,0,0,1,1,0,0,0

db 0,0,0,1,1,1,0,0,1,1,1,0,0,0

db 0,0,0,0,1,1,1,1,1,1,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

s db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,1,1,1,1,1,1,1,0,0,0

db 0,0,0,1,1,1,0,0,1,1,1,0,0,0

db 0,0,0,1,1,1,0,0,0,1,1,0,0,0

db 0,0,0,0,1,1,1,0,0,0,0,0,0,0

db 0,0,0,0,0,1,1,1,1,1,0,0,0,0

db 0,0,0,0,0,0,0,0,1,1,1,0,0,0

db 0,0,0,1,1,0,0,0,0,1,1,0,0,0

db 0,0,0,1,1,1,0,0,1,1,1,0,0,0

db 0,0,0,1,1,1,1,1,1,1,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

t db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,1,1,1,0,0,0,0,0,0

db 0,0,0,0,0,1,1,1,0,0,0,0,0,0

db 0,0,0,0,0,1,1,1,0,0,0,0,0,0

db 0,0,0,1,1,1,1,1,1,1,1,0,0,0

db 0,0,0,0,0,1,1,1,0,0,0,0,0,0

db 0,0,0,0,0,1,1,1,0,0,0,0,0,0

db 0,0,0,0,0,1,1,1,0,0,0,0,0,0

db 0,0,0,0,0,1,1,1,0,0,0,0,0,0

db 0,0,0,0,0,1,1,1,0,0,0,0,0,0

db 0,0,0,0,0,1,1,1,0,1,1,0,0,0

db 0,0,0,0,0,1,1,1,0,1,1,0,0,0

db 0,0,0,0,0,0,1,1,1,1,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

r db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,1,1,1,1,1,1,1,1,1,1,0,0

db 0,0,0,0,1,1,1,1,1,1,1,1,0,0

db 0,0,0,0,1,1,1,1,0,0,0,0,0,0

db 0,0,0,0,1,1,1,0,0,0,0,0,0,0

db 0,0,0,0,1,1,1,0,0,0,0,0,0,0

db 0,0,0,0,1,1,1,0,0,0,0,0,0,0

db 0,0,0,0,1,1,1,0,0,0,0,0,0,0

db 0,0,0,0,1,1,1,0,0,0,0,0,0,0

db 0,0,1,1,1,1,1,1,1,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0

number1 db 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

number2 db 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,1,1,1,1,1,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,1,1,1,1,1,0,0,0,0,0,0

db 0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0

db 0,0,0,0,0,0,0,1,1,1,1,1,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0

db 0,0,0,0,0,0,0,1,1,1,1,1,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0

db 0,0,0,0,0,0,0,1,1,1,1,1,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0

db 0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0

db 0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0

db 0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0

db 0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0

db 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

start:

mov ax,0xb800

mov es,ax

mov ax,0

mov ds,ax

call clear\_screen ; 清屏

call welcome ; 欢迎界面

sti

mov word[ds:0x24],int\_key ; 键盘中断，是否进入主界面

mov word[ds:0x26],0

enter\_to\_main:

cmp byte[ds:flag\_num],0x1c ; 回车

jne enter\_to\_main

main:

mov ah,00h

mov al,13h

int 10h

mov ax,0xa000

mov es,ax

mov ax,0

mov ds,ax

mov si,mycolor

call editcolor

call draw\_num

call taskbar

mov ah,7

call draw\_start

mov word[ds:0x20],int\_time ; 时钟中断

mov word[ds:0x22],0

mov word[ds:0x24],int\_key ; 键盘中断，是否进入主界面

mov word[ds:0x26],0

work:

cmp byte[ds:flag\_num],0x82 ; 1，work1

je work1

cmp byte[ds:flag\_num],0x83 ; 2，work2

je work2

jne work

work1:

mov ah,00h

mov al,13h

int 10h

mov ax,0xa000

mov es,ax

mov ax,0

mov ds,ax

mov si,mycolor

call editcolor

call taskbar

mov ah,7

call draw\_start

sound:

cmp byte[ds:flag\_num],0x9e ; 字母a

je key\_a

cmp byte[ds:flag\_num],0x9f ; 字母s

je key\_s

cmp byte[ds:flag\_num],0xa0 ; 字母d

je key\_d

cmp byte[ds:flag\_num],0xa1 ; 字母f

je key\_f

cmp byte[ds:flag\_num],0xa2 ; 字母g

je key\_g

cmp byte[ds:flag\_num],0xa3 ; 字母h

je key\_h

cmp byte[ds:flag\_num],0xa4 ; 字母j

je key\_j

cmp byte[ds:flag\_num],0xa5 ; 字母k

je key\_k

cmp byte[ds:flag\_num],0xa6 ; 字母l

je key\_l

cmp byte[ds:flag\_num],0x81 ; esc

je main

jne sound

jmp $

key\_a:

mov bx,110

jmp play

key\_s:

mov bx,220

jmp play

key\_d:

mov bx,440

jmp play

key\_f:

mov bx,880

jmp play

key\_g:

mov bx,1760

jmp play

key\_h:

mov bx,1864

jmp play

key\_j:

mov bx,1975

jmp play

key\_k:

mov bx,3951

jmp play

key\_l:

mov bx,3520

play:

call change\_sound

call play\_sound

call clear\_flagnum

jmp sound

work2:

mov ah,00h

mov al,13h

int 10h

mov ax,0xa000

mov es,ax

mov ax,0

mov ds,ax

mov si,mycolor

call editcolor

call taskbar

mov ah,7

call draw\_start

mov ah,6

call draw\_alpha

change\_color:

cmp byte[ds:flag\_num],0x93 ; 字母r

je red

cmp byte[ds:flag\_num],0xa2 ; 字母g

je green

cmp byte[ds:flag\_num],0xb0 ; 字母b

je blue

cmp byte[ds:flag\_num],0x95 ; 字母y

je yellow

cmp byte[ds:flag\_num],0x81 ; esc

je main

jne change\_color

jmp $

red:

mov ah,1

call draw\_alpha

jmp change\_color

green:

mov ah,2

call draw\_alpha

jmp change\_color

blue:

mov ah,3

call draw\_alpha

jmp change\_color

yellow:

mov ah,8

call draw\_alpha

jmp change\_color

int\_key:

push ax

push dx

mov dx,0x60

in al,dx

mov byte[ds:flag\_num],al ; 按键信息存放在flag\_num中

mov dx,0x20

mov al,0x61

out dx,al

pop dx

pop ax

iret

int\_time:

push ax

push dx

push cx

mov cl,[ds:flag]

cmp cl,1

je hello1

hello0:

mov ah,6

call draw\_hello

jmp t1

hello1:

mov ah,7

call draw\_hello

t1:

call changeFlag

mov al,0x20

mov dx,0x20

out dx,al

pop cx

pop dx

pop ax

iret

; 清空flag\_num的值

clear\_flagnum:

push cx

mov cx,[ds:flag\_num]

mov cx,0

mov [ds:flag\_num],cx

pop cx

ret

; 修改Flag值

changeFlag:

push ax

mov al,[ds:flag]

cmp al,1

je change0

mov al,1

mov byte[ds:flag],al

jmp back

change0:

mov al,0

mov byte[ds:flag],al

back:

pop ax

ret

; bx放需要修改成的频率,计数初值在ax中

change\_sound:

push di

mov al,10110110b

out 43h,al

mov dx,0012h

mov ax,34deh

mov di,bx

div di

out 42h,al

mov al,ah

out 42h,al

pop di

ret

play\_sound:

push ax

; 开启扬声器

in al,61h

or al,00000011b

out 61h,al

call sleep

call sleep

call sleep

call sleep

call sleep

; 关闭扬声器

in al,61h

and al,11111101b

out 61h,al

pop ax

ret

; 定时时间修改

changeTime:

mov al,0x34

mov dx,0x43

out dx,al

mov al,0x00 ; 高八位

mov dx,0x40

out dx,al

mov al,0x11 ; 低八位

mov dx,0x40

out dx,al

ret

; 1、2绘制

draw\_num:

push bx

push si

mov bx,number1

mov si,3265

call draw

mov bx,number2

mov si,3395

call draw

pop si

pop bx

ret

; 29\*29

draw:

push di

push cx

push si

push ax

mov cx,0

mov di,0

for:

mov al,[bx+di]

inc di

inc cx

inc si

cmp al,1 ; 为1则涂色

je draw\_color

mov byte[es:si],4

cmp cx,29 ; 每行29个像素点

je nextline

ifendline: ; 判断是否结束一个字符

cmp di,841 ; 29\*29个像素点，小于则继续遍历矩阵

jb for

pop ax

pop si

pop cx

pop di

ret

draw\_color:

mov byte[es:si],5

jmp for

nextline:

add si,291 ; 到下一行的开头，320-29

mov cx,0

jmp ifendline

; 清屏模块

clear\_screen:

push cx

push bx

mov cx,0x0001

mov bx,0

clear:

cmp cx,0x07d0 ; 80\*25，是否到屏幕末端

ja jmpclear

mov byte[es:bx],' '

inc bx

mov byte[es:bx],0x0c

inc bx

inc cx

jmp clear

jmpclear:

pop bx

pop cx

ret

; 欢迎界面模块

welcome:

push bx

push di

mov bx,0

mov di,0

print:

mov al,[ds:str\_welcome+di]

mov byte[es:bx+1928],al

mov byte[es:bx+1929],0x0b

call sleep

call sleep

add bx,2

inc di

cmp di,72

jb print

pop di

pop bx

ret

; bx放点阵图，si放起始位置,ah放色号

draw\_small:

push di

push cx

push si

mov cx,0

mov di,0

for\_small:

mov al,[bx+di]

inc di

inc cx

inc si

cmp al,1 ; 为1则涂色

je draw\_color\_small

cmp cx,14 ; 每行14个像素点

je nextline\_small

ifendline\_small: ; 判断是否结束一个字符

cmp di,252 ; 14\*18个像素点，小于则继续遍历矩阵

jb for\_small

pop si

pop cx

pop di

ret

draw\_color\_small:

mov byte[es:si],ah

jmp for\_small

nextline\_small:

add si,306 ; 到下一行的开头，320-14

mov cx,0

jmp ifendline\_small

; 绘制hello单词

draw\_hello:

push bx

push si

mov bx,h

mov si,[ds:hello\_pos]

call draw\_small

add si,14

mov bx,e

call draw\_small

add si,14

mov bx,l

call draw\_small

add si,14

call draw\_small

add si,14

mov bx,o

call draw\_small

pop si

pop bx

ret

; 绘制start单词

draw\_start:

push bx

push si

mov bx,s

mov si,[ds:start\_pos]

call draw\_small

add si,14

mov bx,t

call draw\_small

add si,14

mov bx,a

call draw\_small

mov bx,r

add si,14

call draw\_small

add si,14

mov bx,t

call draw\_small

pop si

pop bx

ret

; 绘制alpha单词

draw\_alpha:

push bx

push si

mov bx,a

mov si,[ds:pos]

call draw\_small

add si,14

mov bx,l

call draw\_small

add si,14

mov bx,p

call draw\_small

add si,14

mov bx,h

call draw\_small

add si,14

mov bx,a

call draw\_small

pop si

pop bx

ret

;编制自定义颜色，色号+RGB事先放入si中

editcolor:

push dx

push ax

push cx

mov cx,0

;设置颜色号接口0x3c8,RGB分量接口0x3c9

set\_color:

mov dx,0x3c8

mov al,[si]

out dx,al

mov dx,0x3c9

mov al,[si+1]

out dx,al

mov al,[si+2]

out dx,al

mov al,[si+3]

out dx,al

add si,4

inc cx

cmp cx,4

jb set\_color

pop cx

pop ax

pop dx

ret

; 延迟模块

sleep:

push ax

mov ax,0

next:

inc ax

cmp ax,0xffff

jb next

pop ax

ret

taskbar:

mov ax,180 ;起点y

mov bx,0 ;起点x

mov cx,320 ;直线 长度

mov dl,8;

call draw\_0

mov cx,20 ; 重复次数cx

taskbar\_bg:

push cx

inc ax

mov bx,0 ;起点x

mov cx,320 ;直线 长度

mov dl,8;

call draw\_0

pop cx

loop taskbar\_bg ; 自动-1

taskbar\_1:

mov ax,182 ;起点y

mov bx,2 ;起点x

mov cx,70 ;直线 长度

mov dl,0;

call draw\_0

mov ax,183 ;起点y

mov bx,2 ;起点x

mov cx,15 ;直线 长度

mov dl,0;

call draw\_1

mov ax,197 ;起点y

mov bx,2 ;起点x

mov cx,70 ;直线 长度

mov dl,0;

call draw\_0

mov ax,182 ;起点y

mov bx,71 ;起点x

mov cx,15 ;直线 长度

mov dl,0;

call draw\_1

taskbar\_2:

mov ax,182 ;起点y

mov bx,250 ;起点x

mov cx,70 ;直线 长度

mov dl,0

call draw\_0

mov ax,183 ;起点y

mov bx,250 ;起点x

mov cx,15 ;直线 长度

mov dl,0

call draw\_1

mov ax,197 ;起点y

mov bx,250 ;起点x

mov cx,70 ;直线 长度

mov dl,0

call draw\_0

mov ax,183 ;起点y

mov bx,319 ;起点x

mov cx,15 ;直线 长度

mov dl,0

call draw\_1

ret

; 横线，ax列y，bx行x，cx长度，dl色号

draw\_0:

push ax

push bx

push cx

push dx

mov dx,320

mul dx

add bx,ax

pop dx

x1: mov [es:bx],dl

inc bx

dec cx

jnz x1

pop cx

pop bx

pop ax

ret

; 竖线

draw\_1:

push ax

push bx

push cx

push dx

mov dx,320

mul dx

add bx,ax

pop dx

x2: mov [es:bx],dl

add bx,320

dec cx

jnz x2

pop cx

pop bx

pop ax

ret