

课 程 设 计 报 告

课程名称： 微机系统与接口技术

班 级： 计算机科学与技术173班

学 号： 201705110628

姓 名： 朱博威

指导老师： 舒洪波

**2019年 12月**

[一、课程设计的背景 3](#_Toc28179948)

[二、设计的要求及流程图 3](#_Toc28179949)

[2.1设计要求 3](#_Toc28179950)

[2.2流程图 3](#_Toc28179951)

[三、功能介绍 4](#_Toc28179952)

[3.1开发环境 4](#_Toc28179953)

[3.2功能模块介绍 4](#_Toc28179954)

[四、源代码 4](#_Toc28179955)

[五、实现过程及运行结果 18](#_Toc28179956)

[六、心得体会 21](#_Toc28179957)

**考查题目：** 元旦快乐

# 一、课程设计的背景

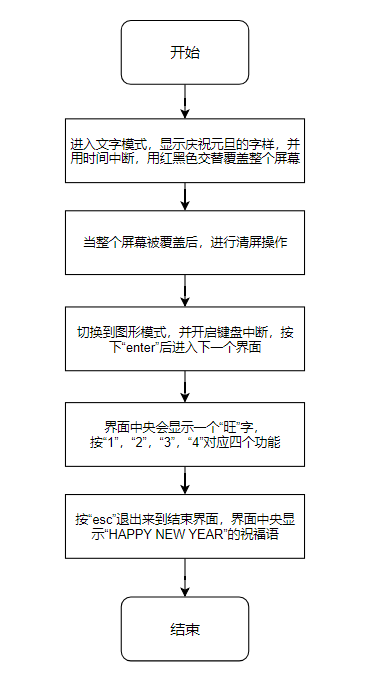
元旦，即公历的1月1日，是世界多数国家通称的“新年”。元，谓“始”，凡数之始称为“元”；旦，谓“日”；“元旦”意即“初始之日”。元旦又称“三元”，即岁之元、月之元、时之元。“元旦”的概念，在不同时代、不同国家，具体所指也不尽相同。世界上大多数国家都采用了国际通行的公历，把每年1月1日作为“元旦”。元旦作为一个很重要的节日，所以在元旦来临之际，我们都会有所庆祝！

# 二、设计的要求及流程图

## 2.1设计要求

本课程的大作业是实现一个微机底层的管理系统，该系统要有多任务管理能力，每个任务要能衔接键盘驱动，响应键盘输入，并能做出相应的图形模式下的图形显示。

## 2.2流程图



# 三、功能介绍

## 3.1开发环境

本系统开发选用Window10操作系统，通过nasm2.8、winhex\_18.2和Bochs-2.4.5三个工具进行开发。

## 3.2功能模块介绍

本系统主要分为三大功能模块，分别为呼吁模块、旺旺旺旺模块、祝福模块。

呼吁功能模块调用weclome显示Let us celebrate the coming of the New Year together的语句呼吁大家一起庆祝元旦，并调用int\_time0、int\_time1实现时间中断，交替写入红黑色的空格文字块覆盖整个屏幕，当整个屏幕被覆盖后，调用qingping清空屏幕。

旺旺旺旺功能模块调用change实现对“1”，“2”，“3”，“4”的键盘响应。按一次“1”，变化一种颜色；按“2”，自动变化颜色，按“esc”返回；按“3”，可以通过“w”，“a”，“s”，“d”控制移动；按“4”，可以通过“w”，“a”，“s”，“d”控制方向，遇到边框自动转弯。

祝福功能模块在图形模式先在界面中央显示HAPPY NEW YEAR祝福语，希望所有看到程序并认可的老师和同学们能够在新的一年里快快乐乐。

# 四、源代码

org 0x8400

jmp start

key db 0

r db 0

g db 0

b db 0

sehao db 0

x dw 0

y dw 0

string db 'Let us celebrate the coming of the New Year together',0x21,'/'

pic1 db '111111111101111111111111'

db '111111111101111111111111'

db '111000011100000011100000'

db '111000011100000011100000'

db '111000011100000011100000'

db '111000011100000011100000'

db '111111111101111111111111'

db '111111111101111111111111'

db '111000011100000011100000'

db '111000011100000011100000'

db '111000011100000011100000'

db '111000011100000011100000'

db '111111111101111111111111'

db '111111111101111111111111'

ph db '110000001100000110000011111111000111111110001100000011000111100001101111111111011001100110001100000011011111111110000011000001111111100'

db '110000001100000110000011000001100110000011001100000011000111100001101111111111011001100110001100000011011111111110000011000001100000110'

db '110000001100001111000011000000110110000001100110000110000110110001101100000000011001100110000110000110011000000000000111100001100000011'

db '110000001100001111000011000001100110000011000110000110000110110001101100000000011001100110000110000110011000000000000111100001100000110'

db '111111111100011001100011111111000111111110000011001100000110011001101111111111011001100110000011001100011111111110001100110001111111100'

db '111111111100011001100011000000000110000000000001111000000110011001101111111111001101101100000001111000011111111110001100110001111100000'

db '110000001100111111110011000000000110000000000000110000000110001101101100000000001101101100000000110000011000000000011111111001100011000'

db '110000001100110000110011000000000110000000000000110000000110001101101100000000001101101100000000110000011000000000011000011001100001100'

db '110000001101100000011011000000000110000000000000110000000110000111101111111111001101101100000000110000011111111110110000001101100000110'

db '110000001101100000011011000000000110000000000000110000000110000111101111111111000110011100000000110000011111111110110000001101100000011'

start:

mov ax,0 ;数据段的段基址

mov ds,ax

mov ax,0x0a000 ;往显存（0xa0000开始）中放入点数据

mov es,ax

call welcome

mov ax,0xb800 ;指向文本模式的显示缓冲区

mov es,ax

mov si,0

sti

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

mov word[ds:0x22],0

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

mov word[ds:0x22],0

cmp si,4000

jae zdjs

jmp zd

zdjs:

mov word[ds:0x20],int\_time\_empty

mov word[ds:0x22],0

call qingping

mov si,0

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

mov word[ds:0x26],0

call enter

mov ax,0 ;数据段的段基址

mov ds,ax

mov ax,0x0a000 ;往显存（0xa0000开始）中放入点数据

mov es,ax

mov al,0x13 ;VGA320x200x8bit显示

mov ah,0x00

int 0x10 ;选择图形模式

mov byte[sehao],1

mov byte[r],0

mov byte[g],0

mov byte[b],0

call color

mov byte[sehao],2

mov byte[r],255

mov byte[g],255

mov byte[b],255

call color

mov byte[sehao],3

mov byte[r],0

mov byte[g],0

mov byte[b],255

call color

mov byte[sehao],2

mov word[x],148

mov word[y],93

call write

call change

call clear

mov word[x],92

mov word[y],95

call write2

jmp $

;-------------------------------------------------------

;时间中断

int\_time0:

mov byte[es:si],' '

mov byte[es:si+1],10

add si,2

mov al,0x20

mov dx,0x20

out dx,al ;EOI=1,中断结束

iret

int\_time1:

mov byte[es:si],' '

mov byte[es:si+1],65

add si,2

mov al,0x20

mov dx,0x20

out dx,al ;EOI=1,中断结束

iret

int\_time\_empty:

push ax

push dx

inc ax

mov al,0x20

mov dx,0x20

out dx,al

pop dx

pop ax

iret

;-------------------------------------------------------

;响应键盘

int\_key:

push ax

push dx

mov dx,0x60

in al,dx

mov [key],al

mov dx,0x20

mov al,0x61

out dx,al

pop dx

pop ax

iret

;-------------------------------------------------------

enter:

cmp byte[key],0x1c

jne enter

mov byte[key],0

ret

;-------------------------------------------------------

change:

cmp byte[key],0x02

je one

cmp byte[key],0x03

je two

cmp byte[key],0x04

je three

cmp byte[key],0x05

je four

cmp byte[key],0x01

je exit

jmp change

;-------------------------------------------------------

;功能一：按一下变一次色

one: push cx

mov byte[key],0

mov ch,byte[sehao]

inc ch

mov byte[sehao],ch

call write

pop cx

jmp change

;-------------------------------------------------------

;功能二：按一下闪烁

two: push cx

mov ch,byte[sehao]

jmp c2

c1: mov ch,2

c2: cmp ch,10

jnb c1

inc ch

mov byte[sehao],ch

call write

cmp byte[key],0x01

je c3

call delay

jmp c2

c3: mov byte[key],0

pop cx

jmp change

;-------------------------------------------------------

;功能三：按WASD四个键控制方向移动

three: push ax

push bx

mov ax,word[x]

mov bx,word[y]

t1: cmp byte[key],0x11

je w

cmp byte[key],0x1e

je a

cmp byte[key],0x1f

je s

cmp byte[key],0x20

je d

cmp byte[key],0x01

je t3

jmp t1

w: call clear

cmp bx,0

je t2

dec bx

jmp t2

s: call clear

cmp bx,186

je t2

inc bx

jmp t2

a: call clear

cmp ax,0

je t2

dec ax

jmp t2

d: call clear

cmp ax,296

je t2

inc ax

jmp t2

t2: mov word[x],ax

mov word[y],bx

call write

call delay

jmp t1

t3: mov byte[key],0

pop bx

pop ax

jmp change

;-------------------------------------------------------

;功能四：按顺时针转动

four: push ax

push bx

push cx

push dx

mov ax,word[x]

mov bx,word[y]

f1: cmp byte[key],0x91

je wm

cmp byte[key],0x9e

je am

cmp byte[key],0x9f

je sm

cmp byte[key],0xa0

je dm

cmp byte[key],0x01

je f3

jmp f1

wm: mov byte[key],0x91

call clear

cmp bx,0

je dm

dec bx

jmp f2

dm: mov byte[key],0xa0

call clear

cmp ax,296

je sm

inc ax

jmp f2

sm: mov byte[key],0x9f

call clear

cmp bx,186

je am

inc bx

jmp f2

am: mov byte[key],0x9e

call clear

cmp ax,0

je wm

dec ax

jmp f2

f2: mov word[x],ax

mov word[y],bx

call write

call delay1

jmp f1

f3: mov byte[key],0

pop dx

pop cx

pop bx

pop ax

jmp change

exit: ret

;-------------------------------------------------------

;延迟（较长）

delay:

push bx

push cx

mov cx,0

d1: mov bx,50

d2: dec bx

cmp bx,0

jne d2

loop d1

pop cx

pop bx

ret

;延迟（较短）

delay1:

push bx

push cx

mov cx,0

d11: mov bx,10

d12: dec bx

cmp bx,0

jne d12

loop d11

pop cx

pop bx

ret

;-------------------------------------------------------

;欢迎

welcome:

push ax

push es

push bx

push dx

push si

mov ax,0xb800 ;指向文本模式的显示缓冲区

mov es,ax

lea si,[0000h]

x2: mov byte[es:si],' '

add si,2h

cmp si,4000

jnz x2

lea si,[1950]

mov bx,0h

mov ds,bx

mov bx,string

x3: mov al,[ds:bx]

cmp al,'/'

je x1

mov byte[es:si],al

add si,2h

inc bx

jmp x3

x1: pop si

pop dx

pop bx

pop es

pop ax

ret

;-------------------------------------------------------

;清屏

qingping:

push bx

mov bx,0

qingping\_loop:

mov byte[es:bx],' '

inc bx

mov byte[es:bx],0

inc bx

cmp bx,4000

jnz qingping\_loop

pop bx

ret

;-------------------------------------------------------

;设置自己的颜色函数

color:

push dx

push ax

mov al,byte[sehao] ;颜色号

mov dx,0x3c8

out dx,al

mov al,byte[r] ;设置R

mov dx,0x3c9

out dx,al

mov al,byte[g] ;设置G

mov dx,0x3c9

out dx,al

mov al,byte[b] ;设置B

mov dx,0x3c9

out dx,al

pop ax

pop dx

ret

;-------------------------------------------------------

;描点画字符

;bx:x

;si:y

;dl:sehao

write:

push ax

push bx

push cx

push dx

push si

mov ax,320

mov si,[y]

mul si

mov bx,[x]

add bx,ax

mov ax,bx

mov si,1

mov ch,1

mov dl,[sehao]

mov di,pic1

wx1: mov cl,[di]

cmp cl,'0'

jnz wx2

mov byte[es:bx],1

jmp wx3

wx2: mov byte[es:bx],dl

wx3: cmp ch,14

ja wx4

inc bx

inc di

inc si

cmp si,24

jng wx1

inc ch

mov si,1

add ax,320

mov bx,ax

jmp wx1

wx4: pop si

pop dx

pop cx

pop bx

pop ax

ret

;-------------------------------------------------------

;描点画字符

;bx:x

;si:y

;dl:sehao

write2:

push ax

push bx

push cx

push dx

push si

mov ax,320

mov si,[y]

mul si

mov bx,[x]

add bx,ax

mov ax,bx

mov si,1

mov ch,1

mov dl,[sehao]

mov di,ph

wx12: mov cl,[di]

cmp cl,'0'

jnz wx22

mov byte[es:bx],1

jmp wx32

wx22: mov byte[es:bx],dl

wx32: cmp ch,10

ja wx42

inc bx

inc di

inc si

cmp si,135

jng wx12

inc ch

mov si,1

add ax,320

mov bx,ax

jmp wx12

wx42: pop si

pop dx

pop cx

pop bx

pop ax

ret

;-------------------------------------------------------

;清除画图

clear:

push ax

push bx

push cx

push si

mov ax,320

mov si,[y]

mul si

mov bx,[x]

add bx,ax

mov ax,bx

mov si,1

mov ch,1

cx1: mov byte[es:bx],1

cmp ch,14

ja cx2

inc bx

inc si

cmp si,24

jng cx1

inc ch

mov si,1

add ax,320

mov bx,ax

jmp cx1

cx2: pop si

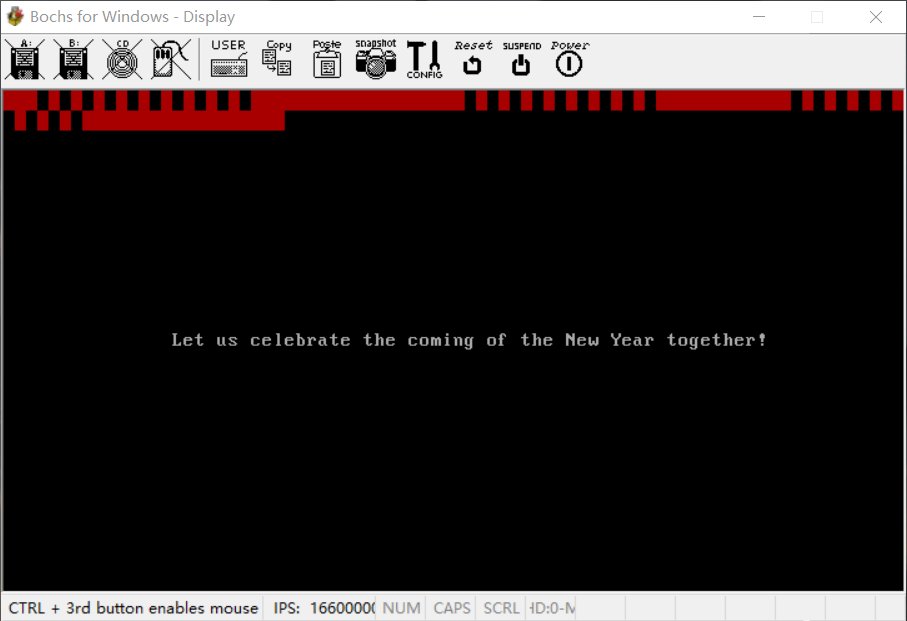
pop cx

pop bx

pop ax

ret

# 五、实现过程及运行结果



**呼吁模块的逻辑：**

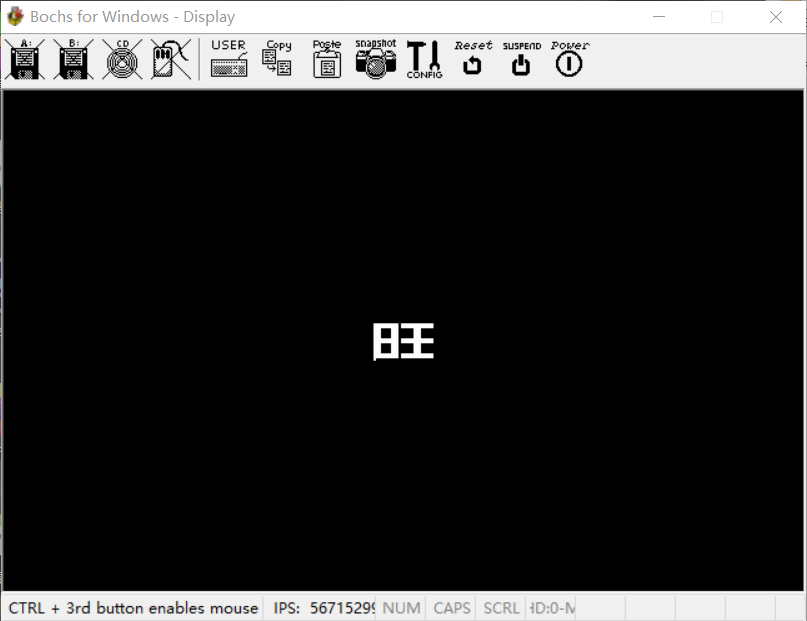
显示模式：文字模式

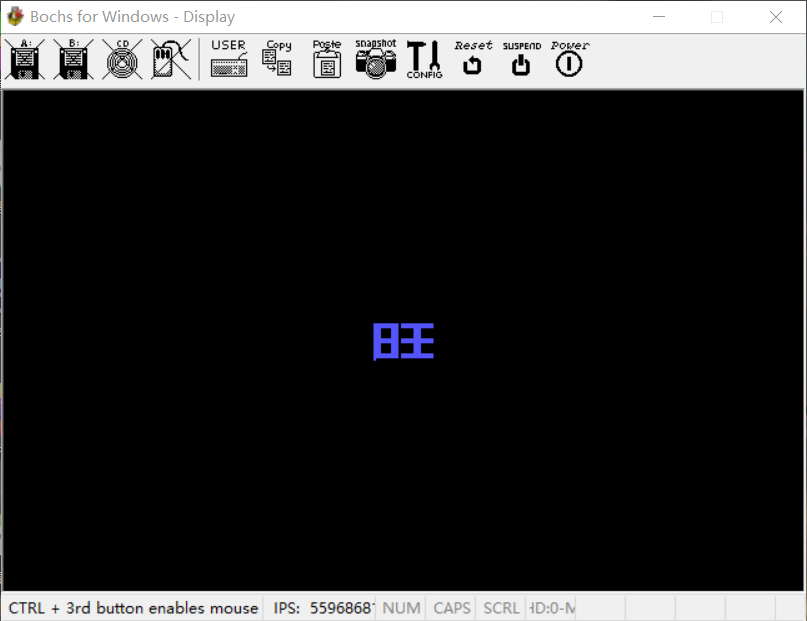
welcome：bx寄存器用来记录显示数据的地址，si寄存器用来记录显示文字的位置，al寄存器用来记录具体数据。

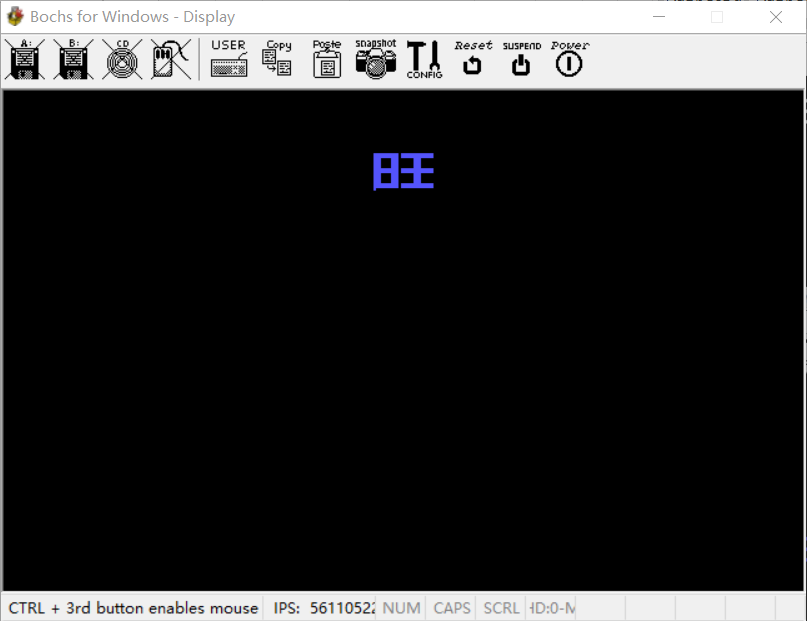
int\_time0、int\_time1:开启时间中断后，int\_time0显示黑色空格文字块，int\_time1显示红色空格文字块，si寄存器用来记录显示位置。

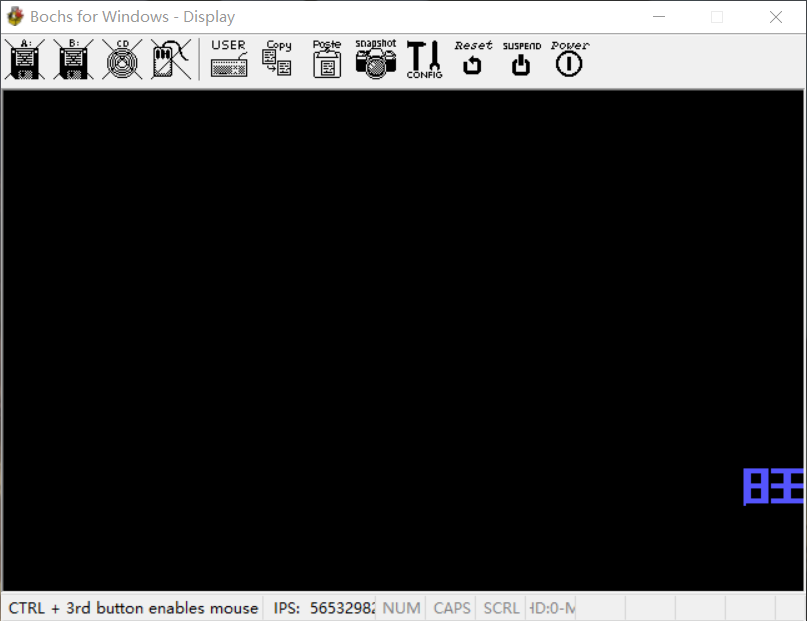
int\_key:开启键盘中断后，al寄存器记录键盘的输入，再将al的数据赋值到key中存储。

enter:当键盘输入不为0x1c，一直在enter中循环，当键盘输入为0x1c时，跳出循环执行下面代码。









**旺旺旺旺功能模块：**

显示模式：图形模式

color：通过输入R、G、B三个变量的值创建自己的颜色。

write：通过点阵图画“旺”字，传入x，y，sehao变量确定第一个点的起始位置和显示的颜色，si寄存器记录列数和位置的右移，ch寄存器记录行数，遍历整个点阵图后结束。

change：比较key中的数据选择对应的功能。

one：响应键盘，变化“旺”字颜色。

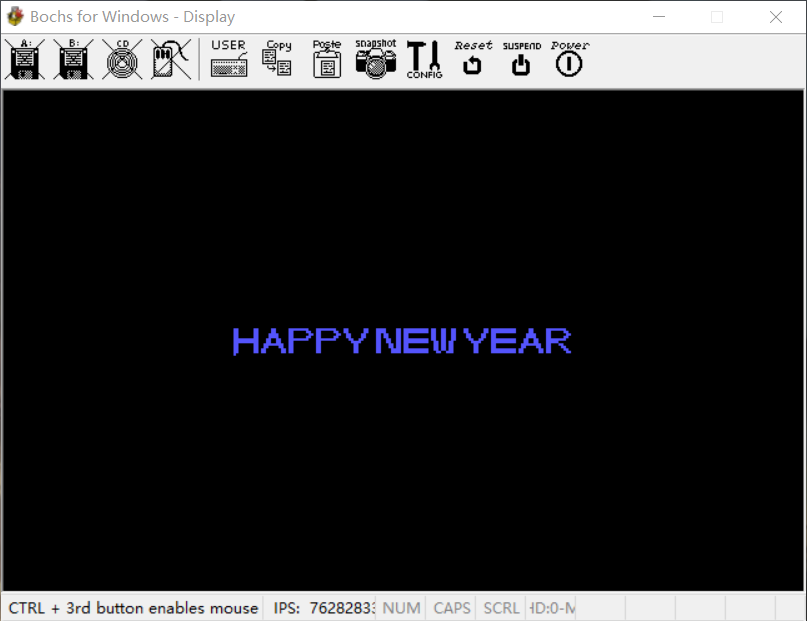
two：响应键盘，自动变化“旺”字颜色。

three：响应键盘，实现上下左右移动。

four：响应键盘，自动上下左右移动，遇到边框自动转弯。

clear：每次重新画“旺”前，先清除之前的“旺”字。

delay：控制变化的时间。



**祝福功能模块：**

显示模式：图形模式

write2：通过点阵图画“HAPPY NEW YEAR”字，传入x，y，sehao变量确定第一个点的起始位置和显示的颜色，si寄存器记录列数和位置的右移，ch寄存器记录行数，遍历整个点阵图后结束。

# 六、心得体会

本次实验中我将老师在本学期中所讲的文本模式文字显示、图形模式文字显示、键盘中断、时间中断、设置颜色等等都复习了一遍，也都融入到了我的课程设计中去，还自主实现了各种强大的移动功能，自动手动移动，碰撞判断，顺时针自己变化方向等等。做课程设计的时候也正好在临近元旦这个节日，我也就将对新的一年的美好祝福融入进了我的设计。汇编语言和c语言最大不同就是很多函数模块都要自己来写，是一个从零开始，通过自己慢慢搭建的一个程序，这个虽然可能比较麻烦，但是其中充满着别样的乐趣。这样的开发过程中，对于尽量减少代码的冗余，是整个编写的重点，通过函数模块的合理有效的编写，能够很大程度上减少代码的行数，也使代码逻辑更加清晰明了。通过微机课程的学习，我意识到了汇编语言的神奇性和无限制性，只要有想法，汇编语言就可以用代码的形式来实现它。最后感谢舒老师在平时课程上给予我们的指导、提供给我们的支持和帮助，老师课上的遵遵教诲是我本次课程设计成功的重要因素。