**X86**汇编语言程序设计

实验报告

姓名：魏远卓 学号:71113211

东南大学计算机科学与工程学院、软件学院

School of Computer Science & Engineering

College of Software Engineering

Southeast University

二0一五年四月

实验一汇编语言程序上机过程

一）实验目的

学会安装“16位汇编程序开发软件”的安装，完成将汇编语言源程序录入进计算机、利用ml.exe进行汇编，LINK进行链接，并用DEBUG调试16位程序的全部过程。本实验大家不必了解程序细节，只是为了熟悉开发环境和上机过程。

二）实验内容

1. 写16位汇编程序，从键盘输入一个字符串，然后换行后将该字符串输出到屏幕。

**1**）源程序

.8086

.model small

.stack

.DATA

CR DB 0AH,0DH,'$'

N EQU 20

MAXLEN DB N

ACTLEN DB ?

STRING DB N DUP('$')

.code

HELLO:

MOV AX,@DATA

MOV DS,AX

MOV AH,0AH

MOV DX,OFFSET MAXLEN

INT 21H

MOV DX,OFFSET CR

MOV AH,09H

INT 21H

MOV DX,OFFSET STRING

MOV AH,09H

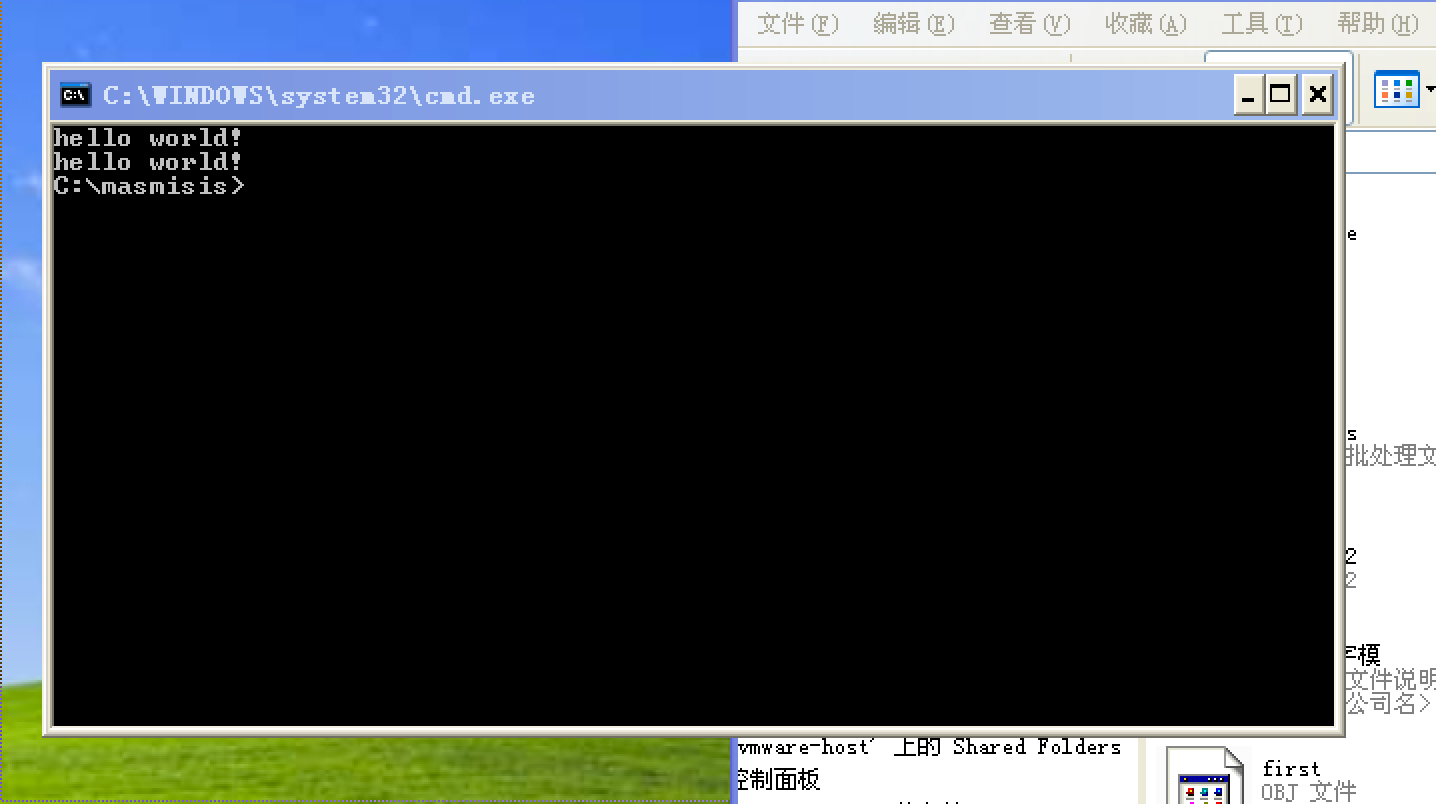
INT 21H

MOV AH,4CH

INT 21H

END HELLO

**2**）运行结果贴图



**3**）编程与调试心得（遇到的问题和解决的办法，以及获得的收获）

1 .DATA区中，内存数据按顺序存放

2 CR DB 0AH，ODH，‘$' （写在.data下）如果不加回车换行，屏幕上显示的输出会覆盖输入，分不清输出是哪些

3 要以mov，@data MOV DS，AX开头

4 一行输入或输出结尾对应一个INT 21H

5 一定要以‘$'因为这才表示真的结束，否则就会输出乱码（机器不知到输出何时终止）

**实验二 顺序程序设计**

一）实验目的

通过这一部分的实验，进一步熟悉汇编过程和DEBUG调试过程；掌握用汇编语言编写顺序程序。

二）实验内容

**2**. 写完整程序16位程序，在内存中从Table开始的10个单元中连续存放0到9的平方值，任给一个0到9的数X，该数存放在内存单元XX中，用XLAT指令查表求X的平方值，并将结果存于内存YY单元中。编写程序，并在DEBUG中进行调试和验证结果。

1）源程序

.8086

.model small

.stack

.DATA

TABLE DB 0,1,4,9,16,25,36,49,64,81

XX DB 9

YY DB ?

.code

START:

MOV AX,@DATA

MOV DS,AX

MOV BX,OFFSET TABLE

MOV AL,9

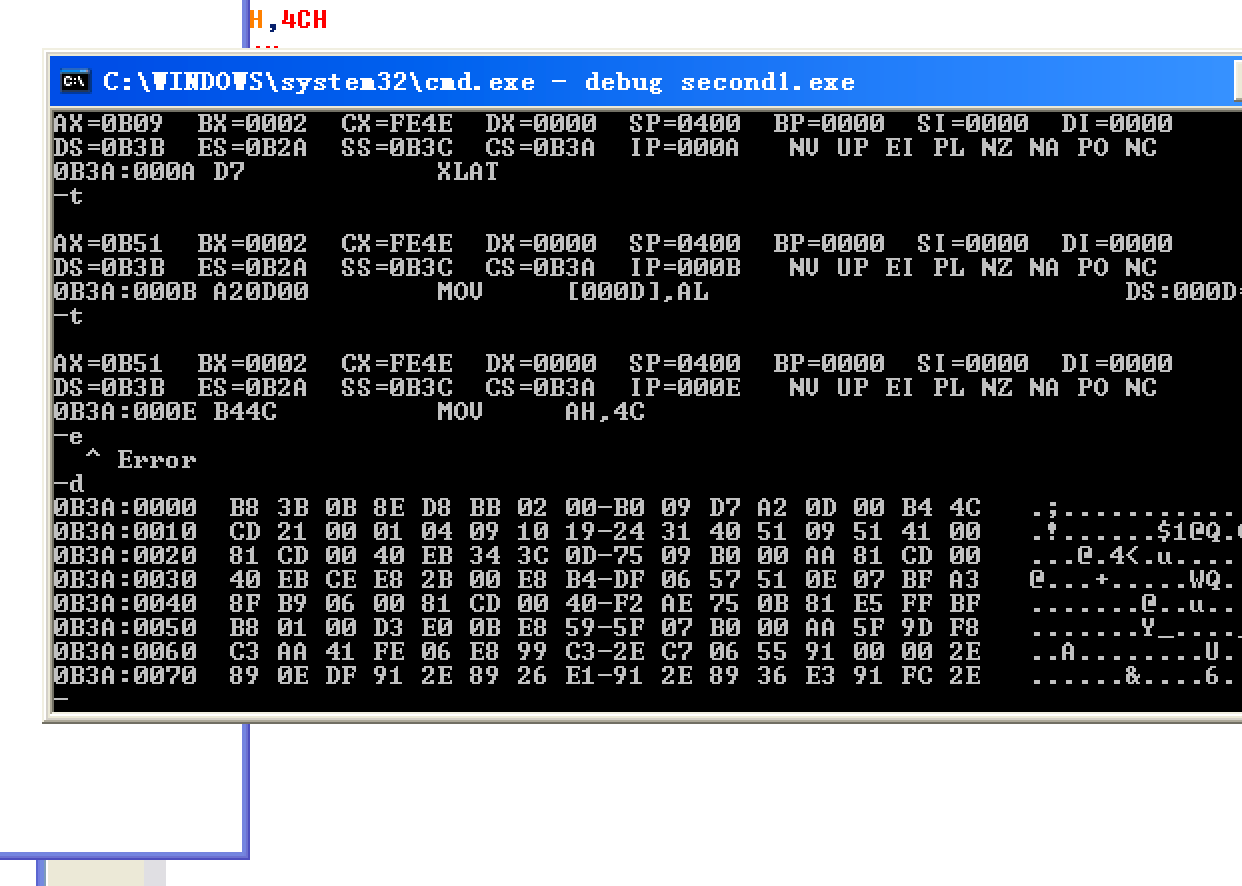
XLAT

MOV YY,AL

MOV AH,4CH

INT 21H

END START

2）运行结果贴图

3）编程与调试心得（遇到的问题和解决的办法，以及获得的收获）

1.首先要理解题意，“任给0到9的数x，该数存在内存单元xx中”的意思是xx为那块内存单元的变量名

2.所有的变量和常量都要在数据段声明

3.要记得写退出和结束

4.调试的时候逐步调试时－t，看内存是－d，看寄存器是－r

**实验三 循环程序设计**

一）实验目的

通过实验，可以掌握循环结构的各种实现方法，进一步了解循环结构中初始化部分、循环体部分、循环控制部分的功能以及他们彼此之间的关系。尤其是多重循环中外层循环和内层循环之间的关系。

二）实验内容

**3**. 所谓回文字符串是指一个字符串正读和倒读都是一样的，例如字符串‘ABCDEFFEDCBA’就是一个回文字符串，而字符串‘ABCFDDCAB’就不是回文字符串。现在编写完整的16位汇编程序，输入一个字符串，判断该字符串是否为回文字符串，并用“It is a palindrome”或“It is NOT a palindrome”作为输出。

1）源程序

.8086

.model small

.stack

.data

n equ 20

maxlen db n

actlen db ?

string db n dup('$')

true db 0ah,0dh,'it is a palindrome’,0ah,0dh,’$’

false db 0ah,0dh,'it is not a palindrome’,0ah,0dh,’$'

cr db 0ah,0dh

.code

start:

mov ax,@data

mov ds,ax

mov ah,0ah

mov dx,offset maxlen

int 21h

mov al,actlen

mov ah,0

mov bl,2

div bl

add al,ah

mov ah,0

mov bl,actlen

mov bh,0

mov si,bx

mov bx,offset string

dec si

lp:cmp si,ax

jna test1

mov dl,[bx+si]

mov dh,[bx]

cmp dl,dh

jnz test2

inc bx

sub si,2

jmp lp

test1:mov dx,offset true

jmp quit

test2:mov dx,offset false

jmp quit

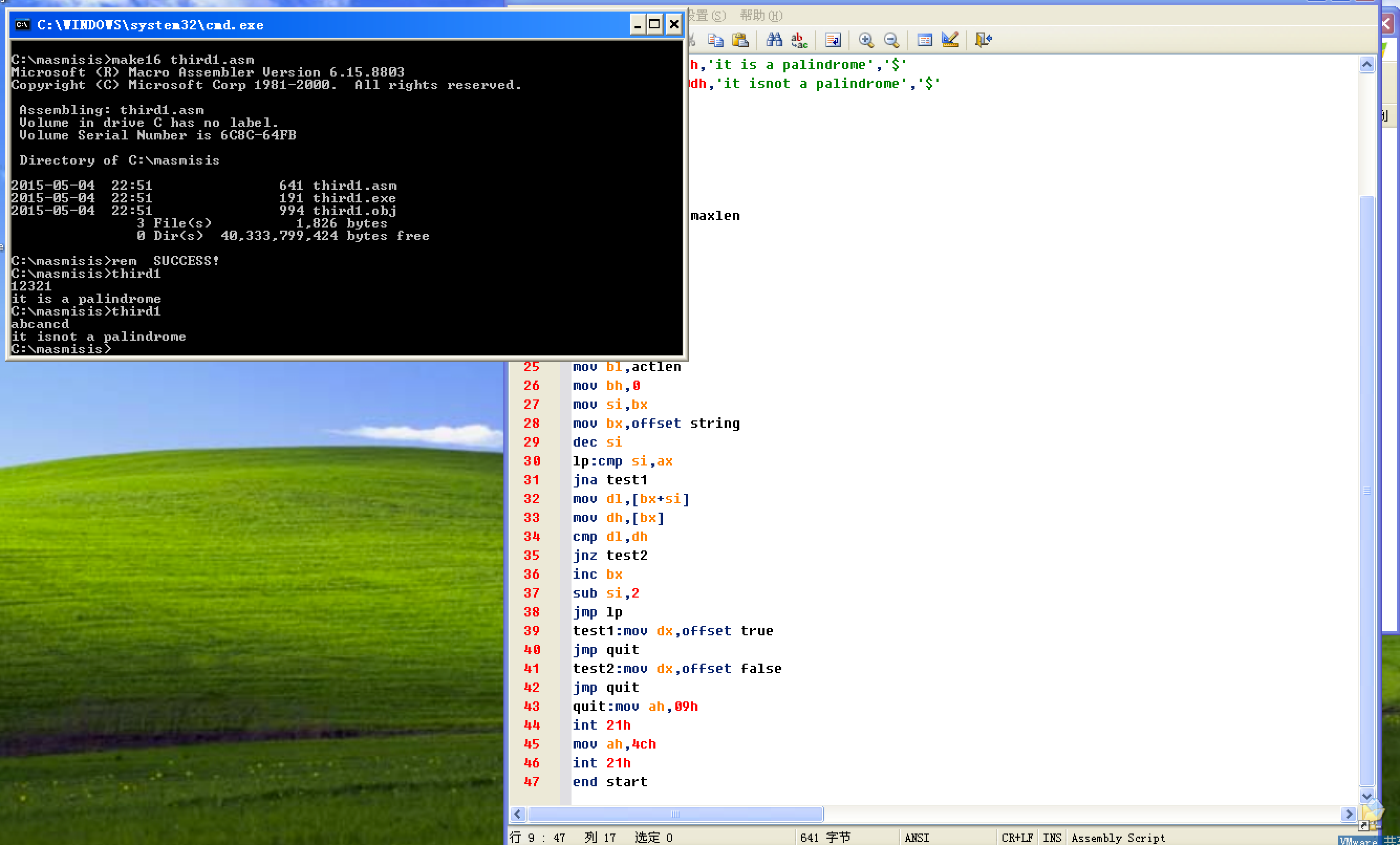
quit:mov ah,09h

int 21h

mov ah,4ch

int 21h

end start

2）运行结果贴图

3）编程与调试心得（遇到的问题和解决的办法，以及获得的收获）

1.注意改变寄存器的值对循环会不会有影响

2.同大小才能比较

3.变量名要前后一致

4.注意不能用loop等保留名作为自己的变量名

5.注意换行回车防止被覆盖

4. 请编写16位完整汇编程序，在一个升序字节数组BUFF中查找数N，找到后将此数从数组中删除，并使得CF=0；没找到返回CF＝1。

1）源程序

.8086

.model small

.stack

.data

buff db 'abcdefg','$'

n db 'f'

.code

start:

mov ax,@data

mov ds,ax

mov al,n

mov ah,'$'

lea di,buff

dec di

lp:inc di

cmp [di],al

jz find

cmp [di],ah

jz nfind

jmp lp

find:inc di

mov al,[di]

cmp [di],ah

jz last

mov [di-1],al

jmp find

last:mov [di-1],al

add ah,0

jmp exit

nfind:mov al,0

sub al,1

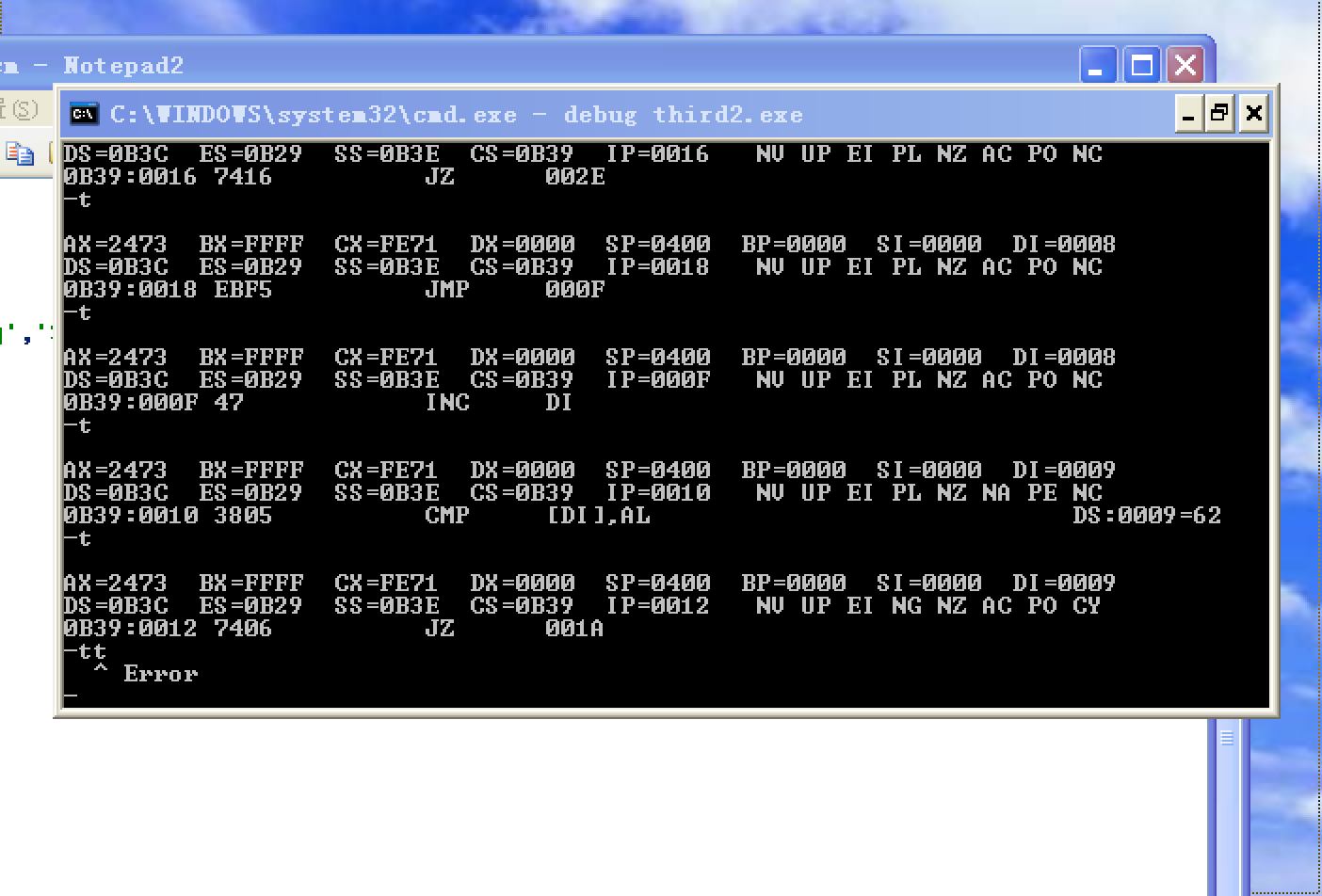
jmp exit

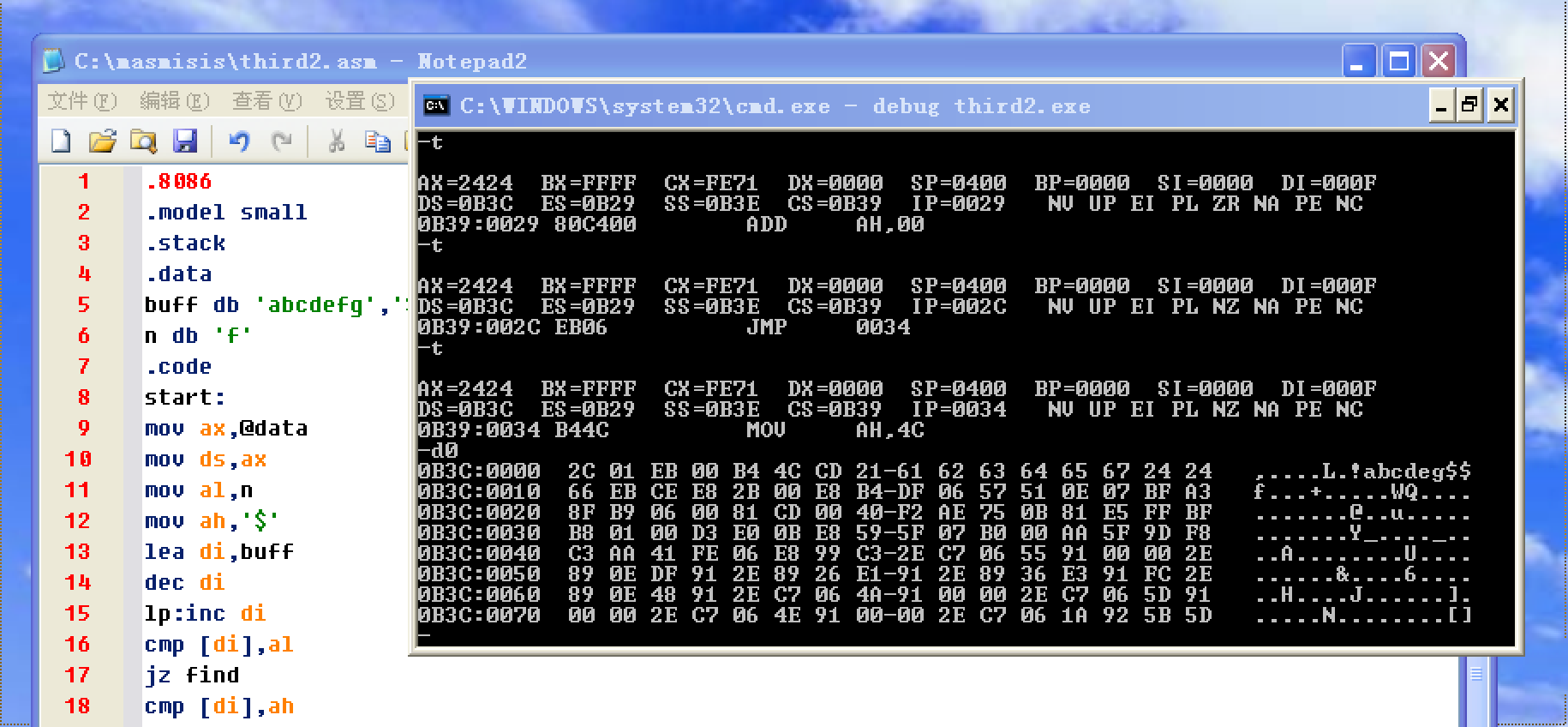
exit:mov ah,4ch

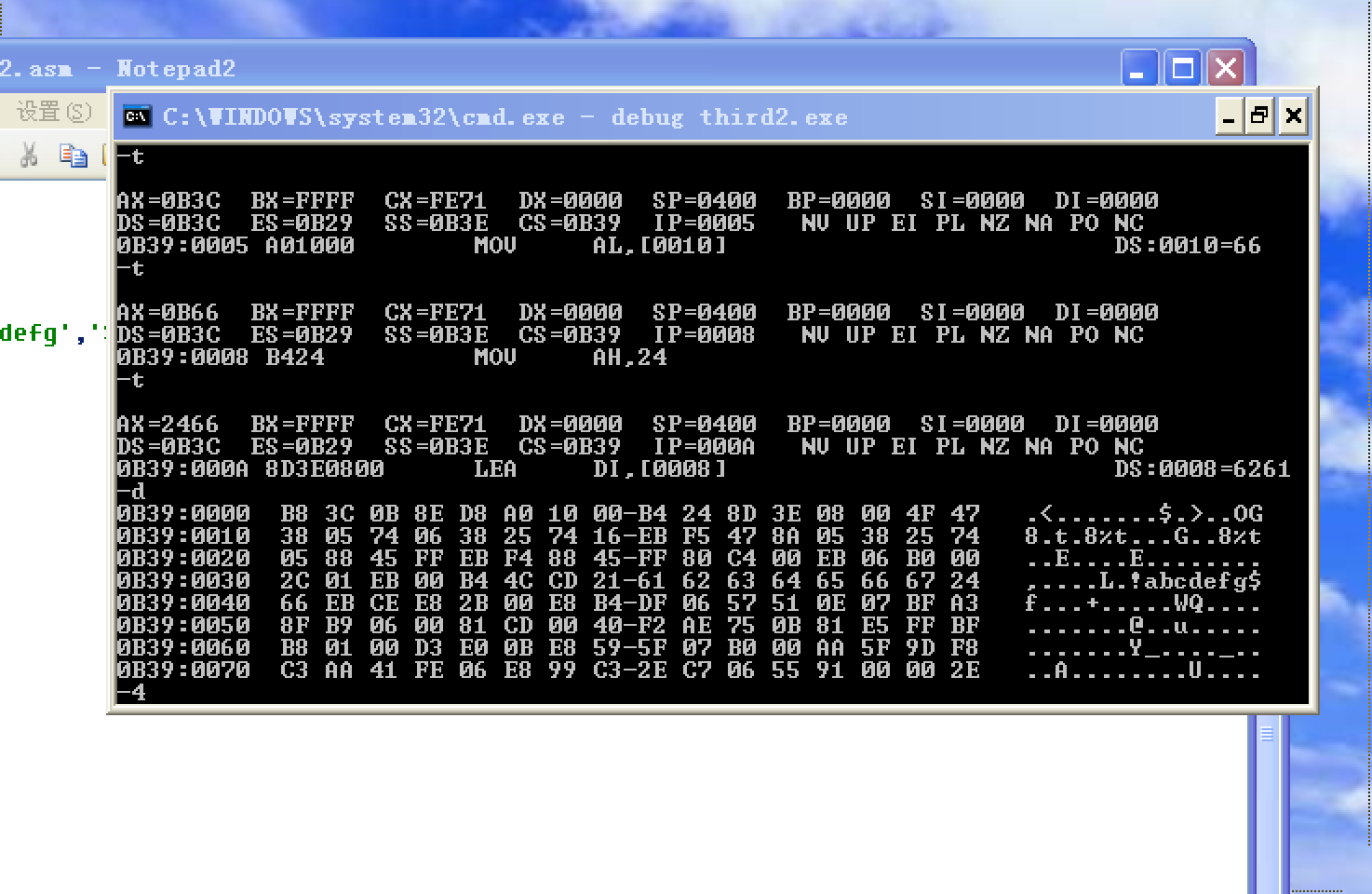
int 21h

end start

2）运行结果贴图







3）编程与调试心得（遇到的问题和解决的办法，以及获得的收获）

1.要确保每一个分支都能执行到后处理语句

**实验四 子程序设计**

一）实验目的

通过本实验，掌握子程序的定义和调用方法。通过程序调试，进一步理解CALL指令和RET指令的功能，掌握子程序调用时参数传递的方法。

二）实验内容

5.请编写完整16位汇编程序从键盘读取字符，如果是十进制的’0’～‘9’则在屏幕上输出该数的8位二进制码，并将数字依次存放到BUF开头的数组中，如果读入的字符是’Q’或者‘q’，则程序退出，其他情况在屏幕上打印“You must input 0~9, or ‘q’ or ‘Q’”。（如输入的字符是‘9’，则输出“00001001”）.提示：输出一个数的2进制形式应该从最高位开始输出。要求打印一个数的2进制形式和输出回车换行分别定义成一个子程序可以将此段程序定义成一个过程。

1）源程序

.8086

.model small

.stack

.data

string db 'You must put a 0~q,or q or Q','$'

.code

next proc

push ax

mov ah,02h

mov dl,0ah

int 21h

mov dl,0dh

int 21h

pop ax

ret

next endp

print proc

cmp al,'0'

jl no

cmp al,'9'

jg no

mov ah,02h

mov cx,8

sub al,30h

lp:mov dl,0

sal al,1

adc dl,30h

push ax

int 21h

pop ax

loop lp

jmp e

no:lea dx,string

mov ah,09h

int 21h

e:ret

print endp

main:mov dx,@data

mov ds,dx

mov cx,0

circle:mov ah,01h

int 21h

cmp al,'q'

je exit

cmp al,'Q'

je exit

call next

call print

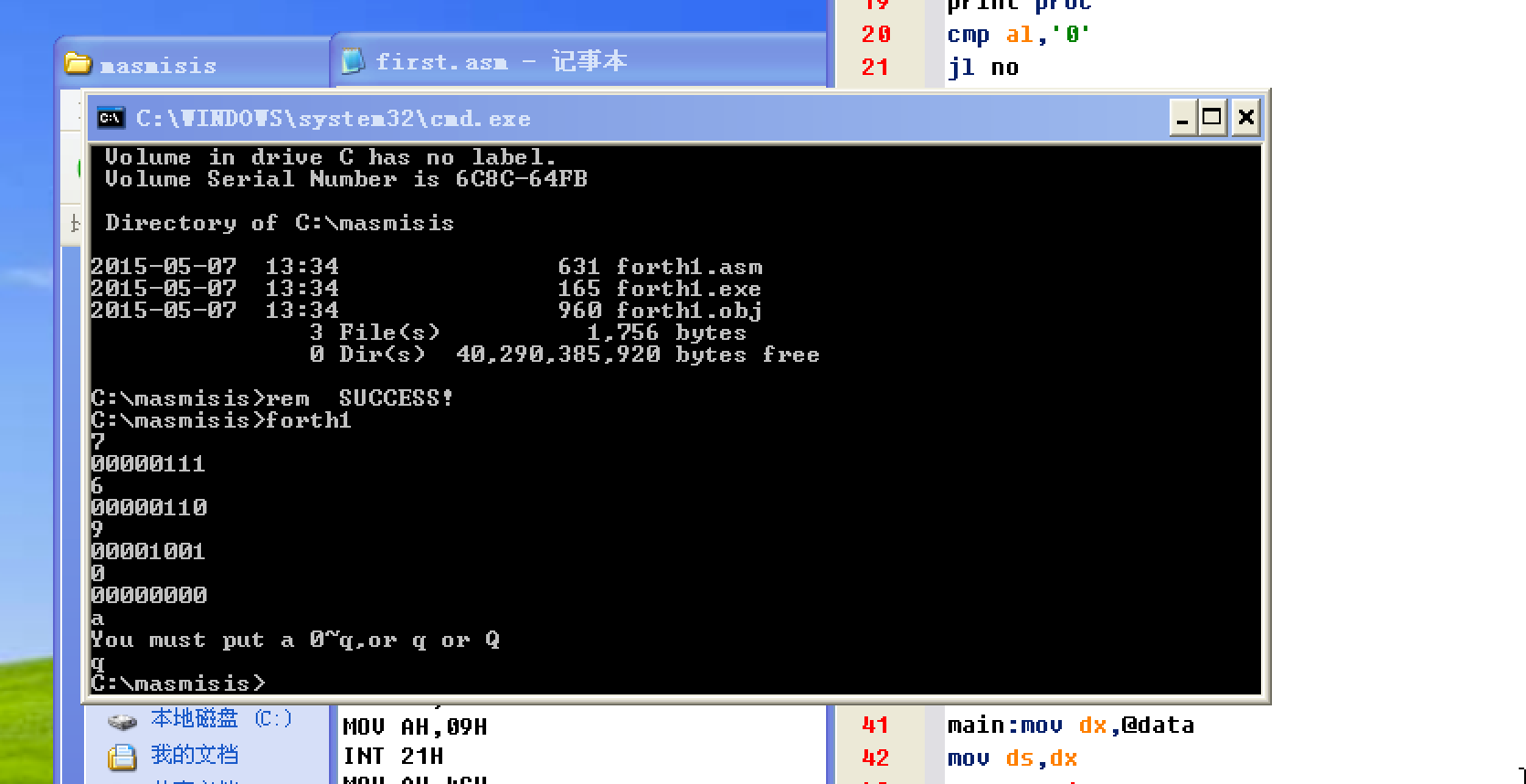
call next

jnz circle

exit:mov ah,4ch

int 21h

end main

2）运行结果贴图

3）编程与调试心得（遇到的问题和解决的办法，以及获得的收获）

1.adc的用法

2.mov ah，01h和mov ah，02h的作用，以及他们对al的影响。后者的返回会存在al中，所以要对ax进行push和pop

3.子程序调用的时候注意push和pop

4.变量和寄存器名注意拼写

6.Programming a sub routine to calculate N!. Specific requirements：Read a number N(1~6) from keyboard，programming a sub routine named DAC to calcute N！, then print the result to screen in decimal form.

1）源程序

.8086

.model small

.stack

.data

string db 0ah,0dh,'You must input a 0~6.',0ah,0dh,'$'

cr db 0ah,0dh,'$'

;number word ?,'$'

a db ?,?,?,?,'$'

.code

print proc

lea dx,a

lea di,a

push dx

mov dx,0

cwd

mov bx,1000

idiv bx

add ax,30h

mov word ptr[di],ax

mov ax,dx

mov dx,0

mov bx,100

idiv bx

add ax,30h

mov word ptr[di+1],ax

mov ax,dx

mov dx,0

mov bx,10

idiv bx

add ax,30h

mov word ptr[di+2],ax

add dx,30h

mov word ptr[di+3],dx

pop dx

mov ah,09h

int 21h

ret

print endp

start:

mov dx,@data

mov ds,dx

mov ah,01h

int 21h

push ax

mov dx,offset cr

mov ah,09h

int 21h

pop ax

cmp al,30h

jng wrong

cmp al,37h

jnl wrong

sub al,30h

mov cl,al

mov ch,0

mov ax,1

lp:imul cx

loopnz lp

call print

jmp exit

wrong:lea dx,string

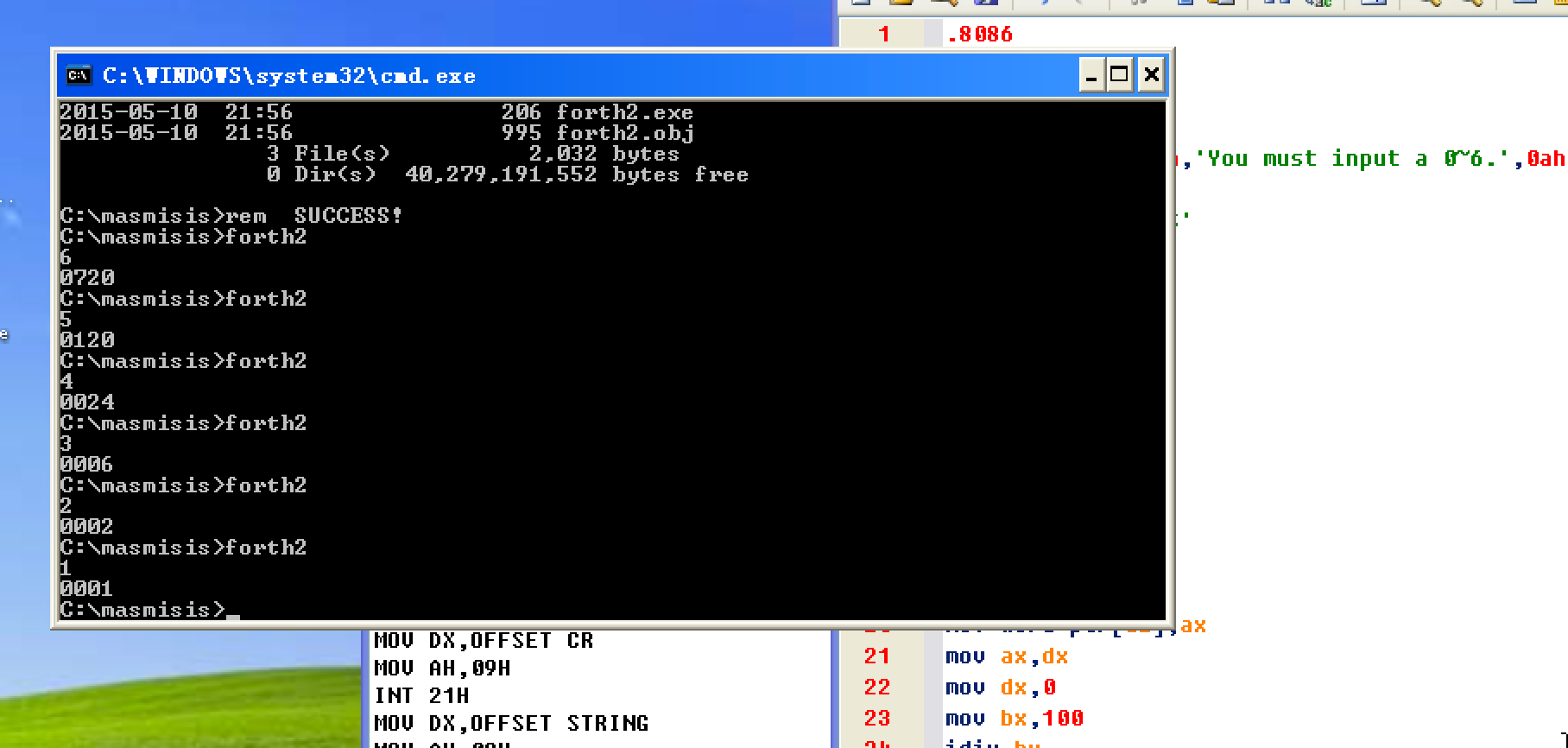
mov ah,09h

int 21h

exit:mov ah,4ch

int 21h

end start

2）运行结果贴图

3）编程与调试心得（遇到的问题和解决的办法，以及获得的收获）

1.注意cwd操作对寄存器的影响，dx：ax被绑定到一起了

2.跳转到某处后如果没有跳转就会一直执行下去

3.注意8位和16位的命令不一样

4.子程序记住结束

5.保存以后还要用的寄存器值

**实验五**

一）实验目的

使用汇编写一个二分查找函数，锻炼编程逻辑与设计思路。

二）实验内容

7.在附加段中有一个从小到大排序的无符号数字数组，其首地址在DI中，数组的第一个单元存放数组长度。要求用折半查找法在数组中查找数N，假设该数已在AX中，如找到，CF=0，并在SI中给出该元素在数组中的偏移地址；如未找到，CF=1。

1）源程序

.8086

.model small

extra segment

array db 10,1,2,3,4,5,6,7,8,9,10

extra ends

code segment

assume cs:code,es:extra

start:

mov bx,extra

mov es,bx

lea di,es:array

mov al,7

mov cl,es:[di]

mov ch,0

inc di

mov dx,di

add dx,cx

mov si,dx

;mov dx,0

cmp al,es:[di]

jb notfind

je find

cmp al,es:[si-1]

ja notfind

je find

work:mov bx,di

add bx,si

shr bx,1

cmp al,es:[bx]

je find

pushf

cmp bx,di

je notfind

popf

jb less

mov di,bx

jmp work

less:mov si,bx

jmp work

notfind:stc

jmp exit

find:clc

mov si,bx

jmp exit

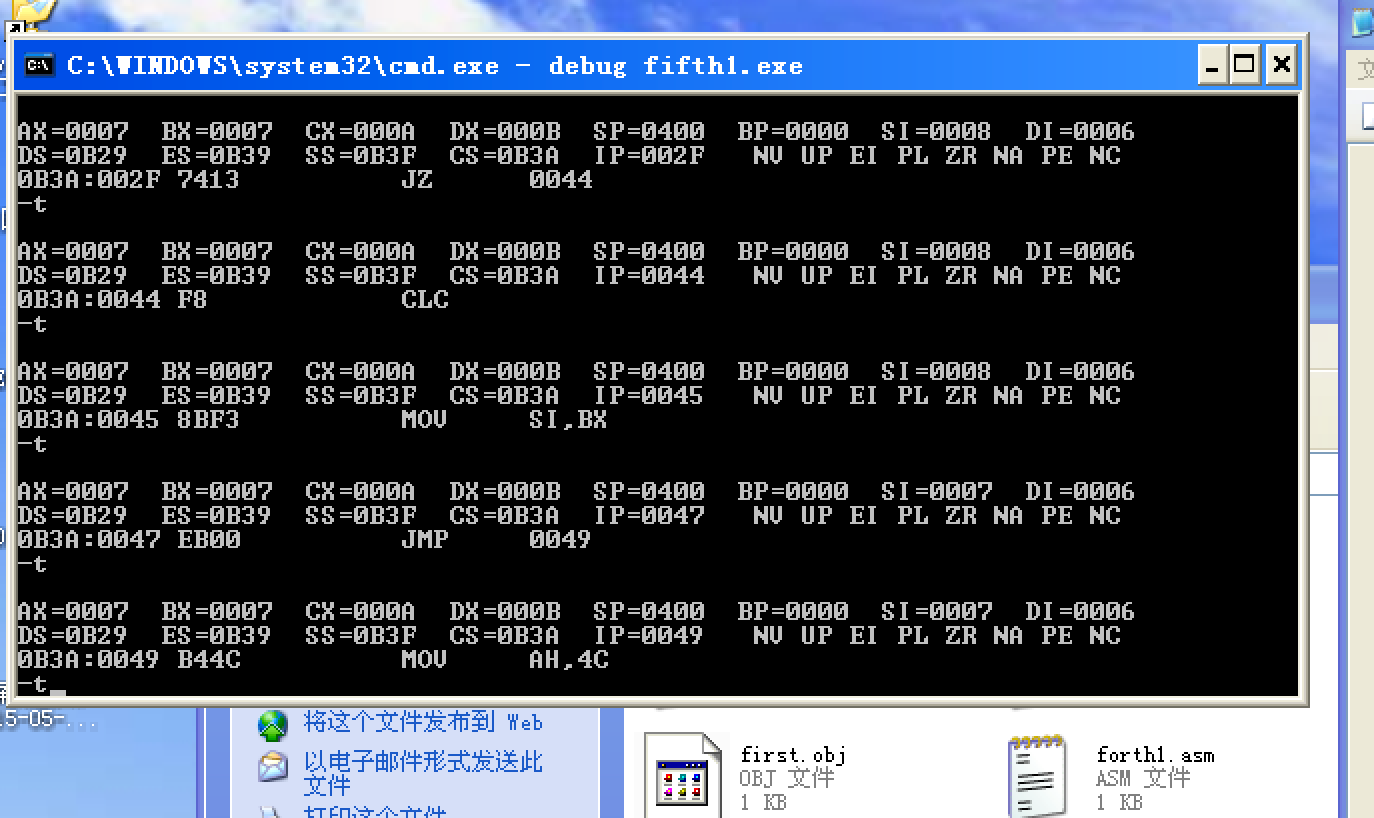
exit:mov ah,4ch

int 21h

code ends

end start

2）运行结果贴图



3）编程与调试心得（遇到的问题和解决的办法，以及获得的收获）

二分查找的关键是左边界、右边界和轴，通过比较、判断不断将左右边界的值赋给轴，并且跳出循环的条件不是左右边界相等，而是左边界和轴相等，否则当只剩下两个数的时候，会造成死循环。并且得到轴的方法可以直接是首位地址相加除以二（除了首地址加上相对的偏移量之外），因为内存空间是连续的。还要注意每一个分支都得到处理。

一）实验目的

用汇编实现冒泡排序，对固定大小的数组进行排序。使用循环方法。

二）实验内容

8.在内存中有一个数组，里面是放着10个学生的某科的成绩，分别是：85,73,92,66,91,98,52,87,83,68， 请用冒泡排序法将这10个数从大到小排序，并将排序的结果在屏幕上打印出来，要求一个数一行的格式输出。

1）源程序

.8086

.model small

.stack

.data

array db 85,73,92,66,91,98,52,87,83,68

cr db 0ah,0dh,'$'

.code

start:

mov ax,@data

mov ds,ax

lea di,array

mov ax,0

mov bx,0

mov cx,9

mov dx,0

l1:mov dx,cx

lea di,array

l2:mov al,[di]

mov bl,[di+1]

cmp al,bl

jb swap

jmp next

swap:mov [di],bl

mov [di+1],al

next:inc di

loop l2

mov cx,dx

loop l1

mov cx,10

lea di,array

print:mov ah,0

mov al,[di]

mov bl,10

div bl

mov dx,ax

add dx,3030h

mov ah,02h

int 21h

mov dl,dh

int 21h

lea dx,cr

mov ah,09h

int 21h

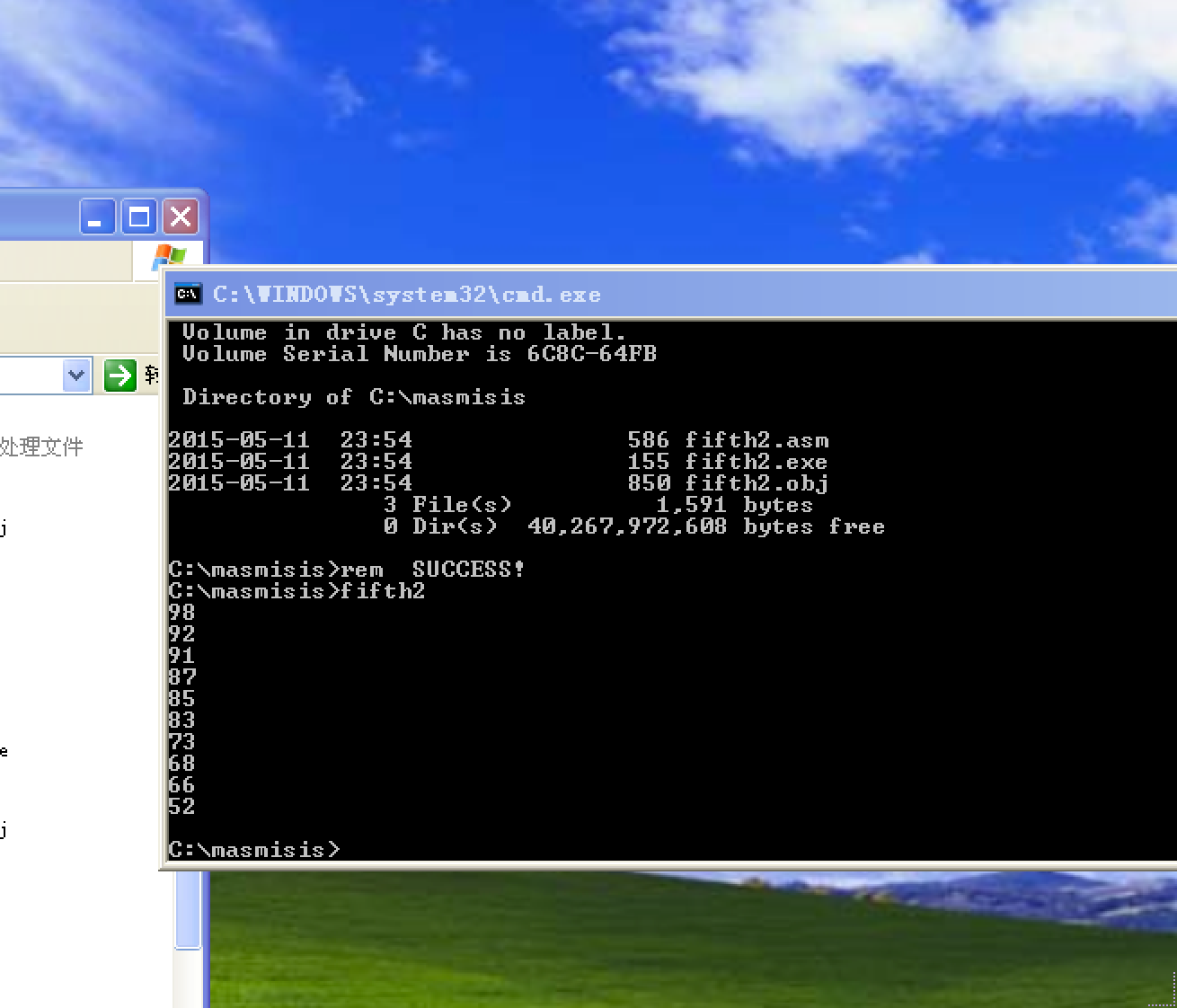
inc di

loop print

mov ah,4ch

int 21h

end start

2）运行结果贴图

3）编程与调试心得（遇到的问题和解决的办法，以及获得的收获）

1.保存cx的值，为了冒泡排序时的外层循环

2.冒泡排序循环n-1次就可以了

3.为了输出十进制要加上30h

4.注意打印回车换行