**汇编第三章第一次作业**

20182131055 侯佳耀

1. 编程实现四则运算：（5v+X\*Y-6)/Z --> V，X、Y、Z均为16位有符号数。

代码：cau.asm

DATAS SEGMENT

x dw 2

y dw 3

z dw 4

v dw 1

rss dw ?

DATAS ENDS

Stack segment stack

Stack ends

CODES SEGMENT

ASSUME CS:CODES, DS:DATAS

START:

mov ax,DATAS

mov DS,ax

mov ax,x

imul y

sub ax,6

mov dx,ax

mov ax,v

mov bx,5

imul bx

add ax,dx

div z

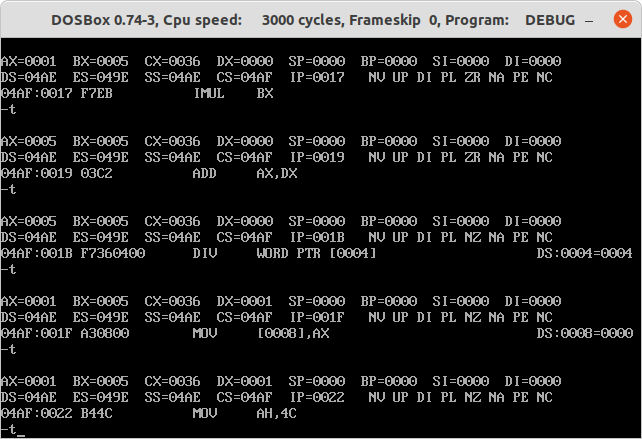
mov rss,ax

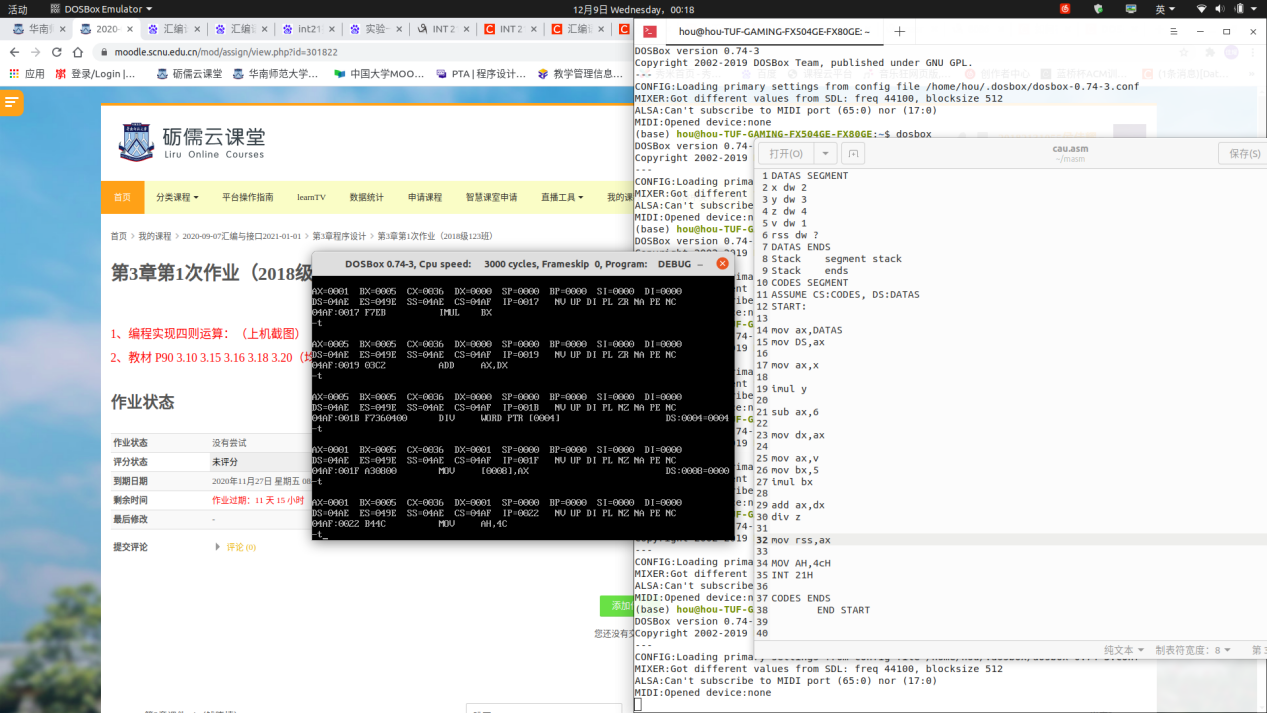
MOV AH,4cH

INT 21H

CODES ENDS

END START

上机结果：



可以发现执行完最后一步除法后ax=0001,dx=0001.结果正确！

1. 课本习题：

（3.10）

代码：

DATAS SEGMENT

byte\_var db 'ABC',10,10h,'EF',3 dup(-1,?,3 dup (4))

DATAS ENDS

Stack segment stack

Stack ends

CODES SEGMENT

ASSUME CS:CODES, DS:DATAS

START:

mov AX,DATAS

mov DS,ax

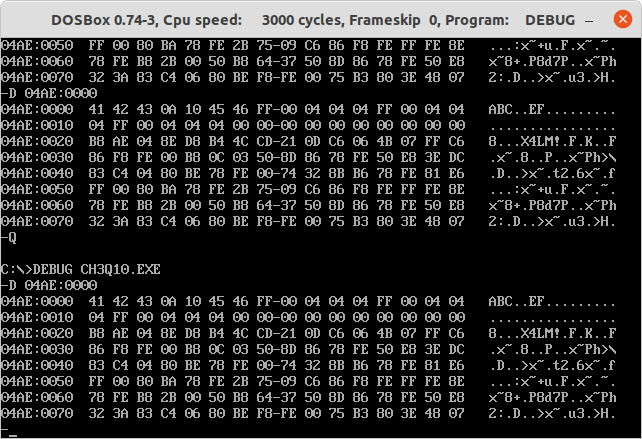
MOV AH,4cH

INT 21H

CODES ENDS

END START

运行结果：



DATAS SEGMENT

word\_var dw 10h,-5,3 dup(?)

DATAS ENDS

Stack segment stack

Stack ends

CODES SEGMENT

ASSUME CS:CODES, DS:DATAS

START:

mov AX,DATAS

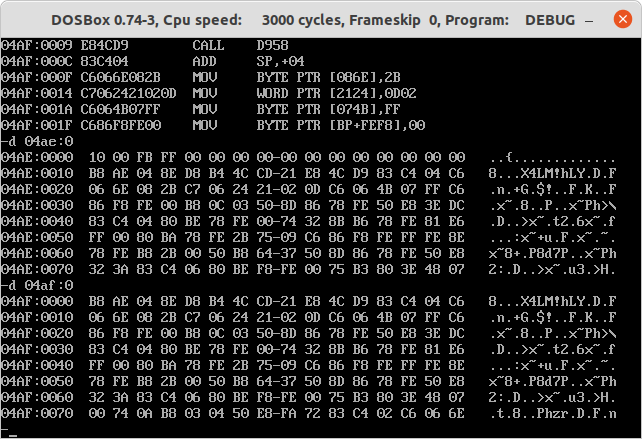
mov DS,ax

MOV AH,4cH

INT 21H

CODES ENDS

END START



（3.15）

(1)byte ptr的操作数为字节类型，而1000大于FFH，故不正确

(2)offset返回偏移地址需要程序执行确定，故参数无法确定

(3)cmp不允许直接比较两个储存单元

(4)同(2)，变量不确定

(5)AL一个字节与字类型不匹配

(6)jnz只支持相对寻址，不支持间接寻址

（3.16）

code:

data segment

mess db 'input error!',0ah,0dh,'$'

data ends

code segment

assume ds:data,cs:code

start:

mov ax,data

mov ds,ax

mov ah,01h

int 21h

cmp al,'A'

jb err

cmp al,'z'

ja err

add al,20h

mov dl,al

mov ah,02h

int 21h

jmp exit

err:

mov dx,offset mess

mov ah,09h

int 21h

exit:

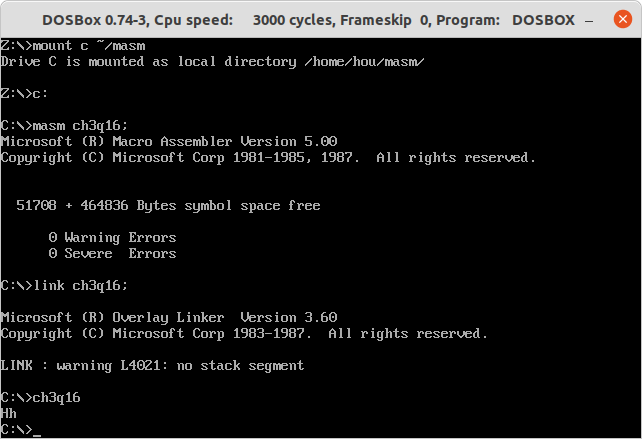
mov ah,4ch

int 21h

code ends

end start

result:



（3.18）

code:

data segment

bufX dw 3

bufY dw 2

bufZ dw ?

data ends

code segment

assume ds:data,cs:code

start:

mov ax,data

mov ds,ax

mov ax,bufX

sub ax,bufY

cmp ax,0

jl gotomov2

jmp gotomov1

gotomov1:

mov ax,bufX

mov bufZ,AX

mov ah,4ch

int 21h

gotomov2:

mov ax,bufY

mov bufZ,AX

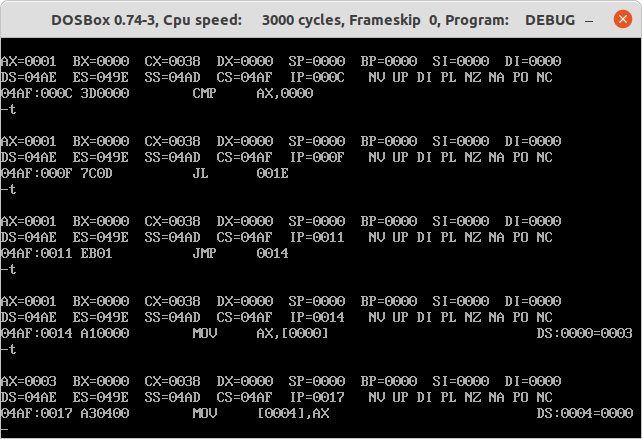
mov ah,4ch

int 21h

code ends

end start

result:



通过debug可以发现，最后程序跳转至gotomov1

结果正确！

（3.20）

code:

data segment

bufX dw 3

bufY dw 2

bufZ dw 2

data ends

code segment

assume ds:data,cs:code

start:

mov ax,data

mov ds,ax

mov ax,bufX

sub ax,bufY

cmp ax,0

jne st2

je st1

st1:

mov dx,1

mov ax,bufY

sub ax,bufZ

cmp ax,0

je st3

jne st4

st2:

mov dx,0

mov ax,bufY

sub ax,bufZ

cmp ax,0

je st3

jne st4

st3:

add dx,1

cmp dx,2

je two

jne one

st4:

cmp dx,1

je one

jne zero

two:

mov dl,32h

mov ah,2

int 21h

mov ah,4ch

int 21h

one:

mov dl,31h

mov ah,2

int 21h

mov ah,4ch

int 21h

zero:

mov dl,30h

mov ah,2

int 21h

mov ah,4ch

int 21h

code ends

end start

result:

