安徽大学人工智能学院

实验报告

课程名称：《计算机组成原理与汇编语言》

专　　业：人工智能

学　　号：WA2214014

姓　　名：杨跃浙

指导老师：杜库

实验项目	实验 9-第九次上机实验			实验次序	09
实验地点	笃行南楼 A104	参与人员	杨跃浙	实验日期	06.05

一、实验目的

汇编语言程序设计

二、实验环境

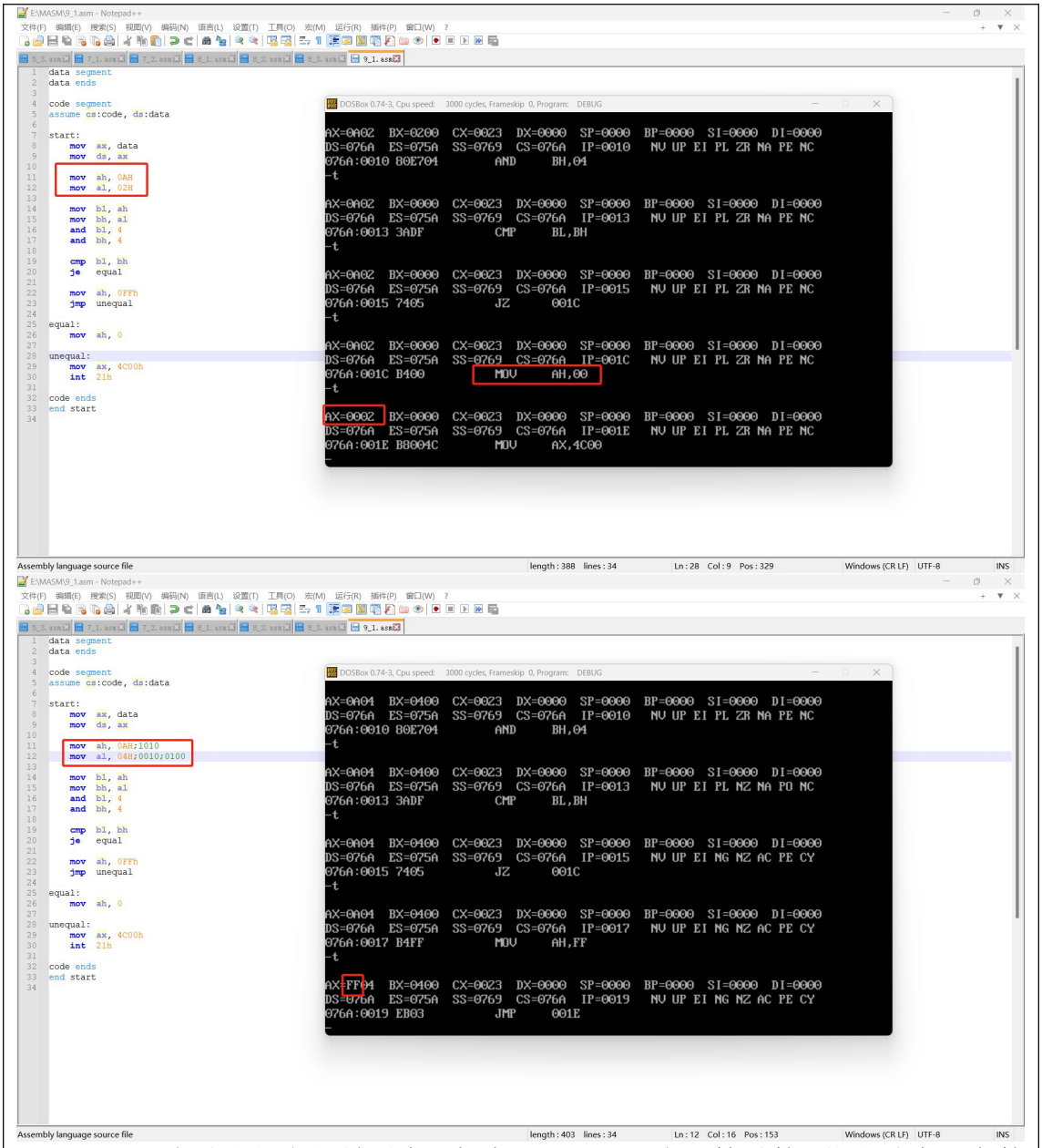
Windows 2011，DOSBox

三、实验内容

1、判断寄存器 AH 和 AL 中第 3 位是否相同，相同，AH 置 0，否则 AH 置全 1。

5_3.asm | 7_1.asm | 7_2.asm | 8_1.asm | 8_2.asm | 8_3.asm | 9_1.asm | 9_

```
data segment
data ends

code segment
assume cs:code, ds:data

start:
    mov  ax, data
    mov  ds, ax

    mov  ah, 0AH;1010
    mov  al, 04H;0010;0100

    mov  bl, ah
    mov  bh, al
    and  bl, 4
    and  bh, 4

    cmp  bl, bh
    je   equal

    mov  ah, 0FFh
    jmp  unequal

equal:
    mov  ah, 0

unequal:
    mov  ax, 4C00h
    int  21h

code ends
end start
```

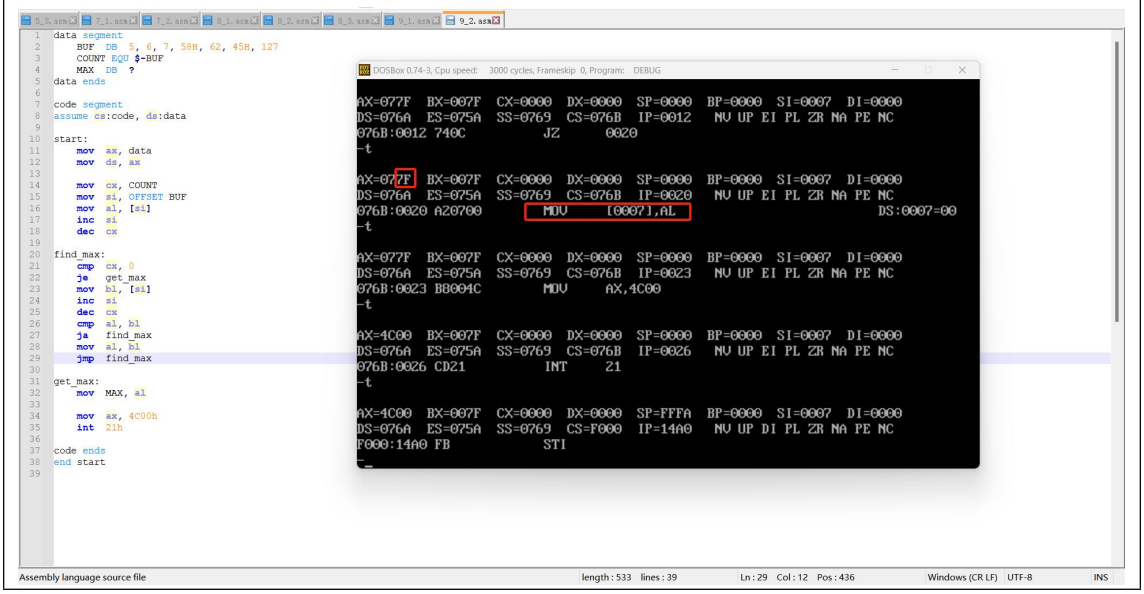


2、以 BUF 为首址的字节单元中，存放了 COUNT 个无符号数，找出其中最大数并送入 MAX 单元中。

```
BUF     DB  5,6,7,58H,62,45H,127, ……
COUNT   EQU     $-BUF
MAX         DB      ?
```

5_3.asm | 7_1.asm | 7_2.asm | 8_1.asm | 8_2.asm | 8_3.asm | 9_1.asm | 9_2.asm

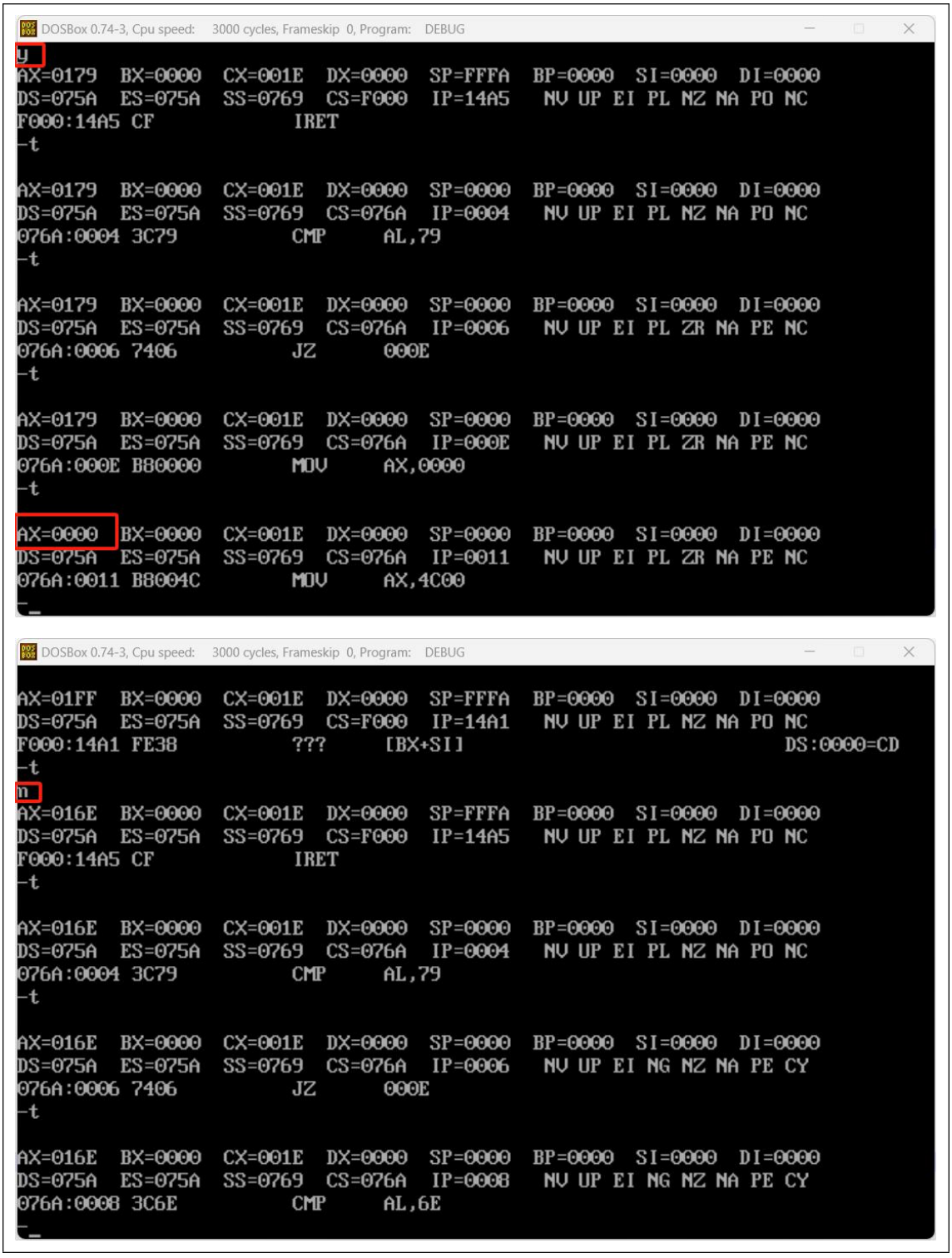
```
data segment
    BUF   DB   5, 6, 7, 58H, 62, 45H, 127
    COUNT EQU $-BUF
    MAX   DB   ?
data ends

code segment
assume cs:code, ds:data

start:
    mov   ax, data
    mov   ds, ax

    mov   cx, COUNT
    mov   si, OFFSET BUF
    mov   al, [si]
    inc   si
    dec   cx

find_max:
    cmp   cx, 0
    je    get_max
    mov   bl, [si]
    inc   si
    dec   cx
    cmp   al, bl
    ja    find_max
    mov   al, bl
    jmp   find_max

get_max:
    mov   MAX, al

    mov   ax, 4C00h
    int   21h

code ends
end start
```



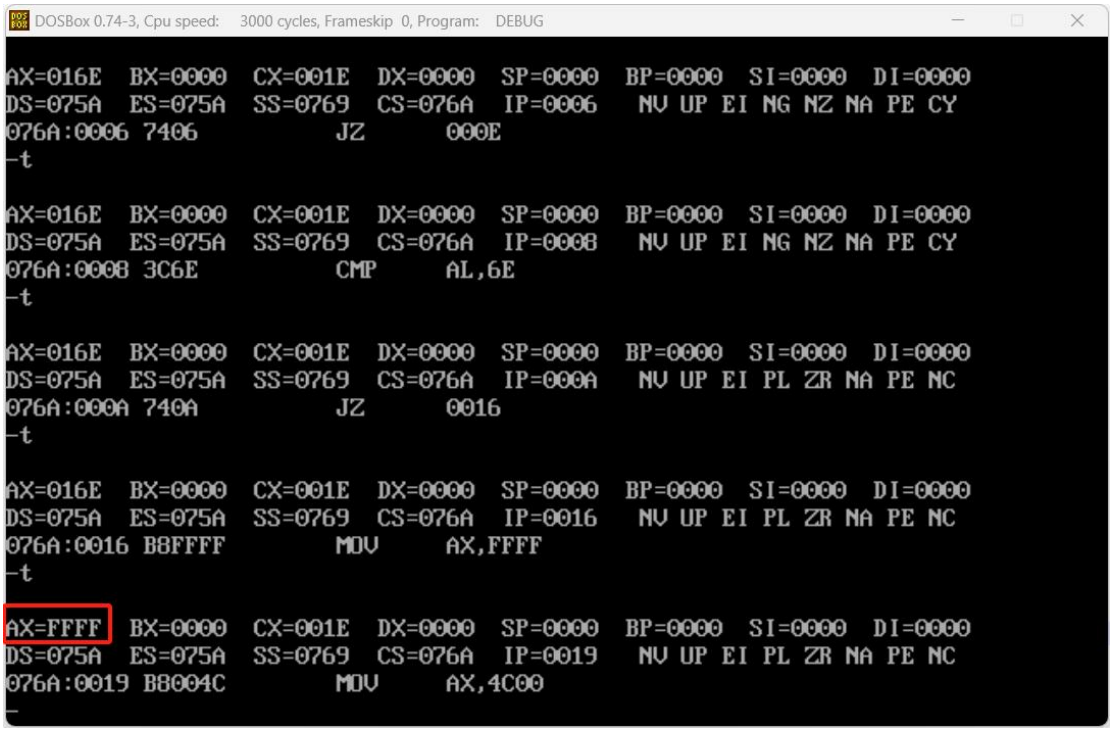
3、 编指程序段，调用系统功能的 1 号子功能输入一个字符，并判断输入的字符。如字符是"y"，则转向 YES 程序段；如字符是"n"，则转向 NO 程序段。(YES 和 NO 分别是两程序段入口处的标号)

5_3.asm | 7_1.asm | 7_2.asm | 8_1.asm | 8_2.asm | 8_3.asm | 9_1.asm | 9_2.asm | 9_3.asm

```
1  data segment
2  data ends
3
4  code segment
5  assume cs:code, ds:data
6
7  start:
8      mov  ah, 01h
9      int  21h
10     cmp  al, 'y'
11     je   YES
12     cmp  al, 'n'
13     je   NO
14     jmp  start
15
16 YES:
17     mov  ax, 0000h
18     mov  ax, 4C00h
19     int  21h
20
21 NO:
22     mov  ax, 0FFFFh
23     mov  ax, 4C00h
24     int  21h
25
26 code ends
27 end start
28
```


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
g
AX=0179 BX=0000 CX=001E DX=0000 SP=FFFA BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=F000 IP=14A5 NV UP EI PL NZ NA PO NC
F000:14A5 CF IRET
-t
AX=0179 BX=0000 CX=001E DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=0004 NV UP EI PL NZ NA PO NC
076A:0004 3C79 CMP AL,79
-t
AX=0179 BX=0000 CX=001E DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=0006 NV UP EI PL ZR NA PE NC
076A:0006 7406 JZ 000E
-t
AX=0179 BX=0000 CX=001E DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=000E NV UP EI PL ZR NA PE NC
076A:000E B80000 MOV AX,0000
-t
AX=0000 BX=0000 CX=001E DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=0011 NV UP EI PL ZR NA PE NC
076A:0011 B8004C MOV AX,4C00
-

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
AX=01FF BX=0000 CX=001E DX=0000 SP=FFFA BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=F000 IP=14A1 NV UP EI PL NZ NA PO NC
F000:14A1 FE38 ??? [BX+SI] DS:0000=CD
-t
g
AX=016E BX=0000 CX=001E DX=0000 SP=FFFA BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=F000 IP=14A5 NV UP EI PL NZ NA PO NC
F000:14A5 CF IRET
-t
AX=016E BX=0000 CX=001E DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=0004 NV UP EI PL NZ NA PO NC
076A:0004 3C79 CMP AL,79
-t
AX=016E BX=0000 CX=001E DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=0006 NV UP EI NG NZ NA PE CY
076A:0006 7406 JZ 000E
-t
AX=016E BX=0000 CX=001E DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=0008 NV UP EI NG NZ NA PE CY
076A:0008 3C6E CMP AL,6E
-



4、在 BUF1 和 BUF2 两个数据区中，各定义有 8 个带符号字数据，试编制一完整的源程序，求它们对应项之和(即 BUF1 的第 0 项对 BUF2 的第 0 项，BUF1 的第 1 项对 BUF2 的第 1 项)，并将和数存入以 S 为首址的数据区中。

```
DATA SEGMENT
   BUF1   DW   -5,2, 4,-10, 9,-8,10,20
   BUF2   DW   4,-2, 24,45, -25,20,30,10
   S      DW   8   DUP (0)
DATA   ENDS
```

5_3.asm | 7_1.asm | 7_2.asm | 8_1.asm | 8_2.asm | 8_3.asm | 9_1.asm | 9_2.asm | 9_3.asm | 9_4.asm

```
data segment
  BUF1  DW  -5, 2, 4, -10, 9, -8, 10, 20
  BUF2  DW  4, -2, 24, 45, -25, 20, 30, 10
  S     DW  8 DUP (0)
data ends

code segment
assume cs:code, ds:data

start:
    mov  ax, data
    mov  ds, ax
    mov  es, ax

    mov  cx, 8
    mov  si, OFFSET BUF1
    mov  di, OFFSET BUF2
    mov  bx, OFFSET S

sum_loop:
    mov  ax, [si]
    add  ax, [di]
    mov  [bx], ax

    add  si, 2
    add  di, 2
    add  bx, 2
    loop sum_loop

    mov  ax, 4C00h
    int  21h

code ends
end start
```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG

```
-t

AX=001E  BX=002E  CX=0001  DX=0000  SP=0000  BP=0000  SI=0010  DI=0020
DS=076A  ES=076A  SS=0769  CS=076D  IP=001F   NV UP EI PL NZ AC PO NC
076D:001F 83C302        ADD     BX,+02
-t

AX=001E  BX=0030  CX=0001  DX=0000  SP=0000  BP=0000  SI=0010  DI=0020
DS=076A  ES=076A  SS=0769  CS=076D  IP=0022   NV UP EI PL NZ AC PE NC
076D:0022 E2EF          LOOP    0013
-t

AX=001E  BX=0030  CX=0000  DX=0000  SP=0000  BP=0000  SI=0010  DI=0020
DS=076A  ES=076A  SS=0769  CS=076D  IP=0024   NV UP EI PL NZ AC PE NC
076D:0024 B8004C        MOV     AX,4C00
-d 0020
076A:0020  FF FF 00 00 1C 00 23 00-F0 FF 0C 00 28 00 1E 00   ......#.....(...
076A:0030  B8 6A 07 8E D8 8E C0 B9-08 00 BE 00 00 BF 10 00   .j..............
076A:0040  BB 20 00 8B 04 03 05 89-07 83 C6 02 83 C7 02 83   . ..............
076A:0050  C3 02 E2 EF B8 00 4C CD-21 5E FC 26 8A 47 0C 2A   ......L.!^.&.G.*
076A:0060  E4 40 50 8B C3 8C C2 05-0C 00 52 50 E8 C1 48 83   .@P.......RP..H.
076A:0070  C4 04 50 8D 86 FA FE 50-E8 17 73 83 C4 06 8B B6   ..P....P..s.....
076A:0080  FA FE 81 E6 FF 00 C6 82-FB FE 00 2B C0 50 8D 86   ...........+.P..
076A:0090  FB FE 50 E8 08 6A 83 C4-04 0B C0 75 03 E9 A5 00   ..P..j.....u....
-_
```