

第 1 次上机

班级	学号	姓名
计试 2201	2223312202	林圣翔

1、熟悉汇编语言实验环境（masm、link、debug）

（1）使用 masm 编译 demoadd.asm，同时生成.lst、.crf 文件，给出运行结果截图。

TODO: 你的截图

```
C:\LEARN>masm demoadd
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.

Object filename [demoadd.OBJ]:
Source listing [NUL.LST]: demoadd
Cross-reference [NUL.CRF]: demoadd

50564 + 465884 Bytes symbol space free

0 Warning Errors
0 Severe Errors

C:\LEARN>dir demoadd.*
Directory of C:\LEARN\
DEMOADD  ASM                507 31-03-2024  0:05
DEMOADD  CRF                194 31-03-2024  8:28
DEMOADD  LST               2,047 31-03-2024  8:28
DEMOADD  OBJ               135 31-03-2024  8:28
4 File(s)                2,883 Bytes.
0 Dir(s)                262,111,744 Bytes free.
```

（2）使用 link 编译 demoadd.asm，同时生成同名的.map 文件，给出运行结果截图。

TODO: 你的截图

```
C:\LEARN>link demoadd
Microsoft (R) Overlay Linker Version 3.60
Copyright (C) Microsoft Corp 1983-1987. All rights reserved.

Run File [DEMOADD.EXE]:
List File [NUL.MAP]: demoadd
Libraries [.LIB]:
LINK : warning L4021: no stack segment

C:\LEARN>dir demoadd.*
Directory of C:\LEARN\
DEMOADD  ASM                507 31-03-2024  0:05
DEMOADD  CRF                194 31-03-2024  8:28
DEMOADD  EXE                548 31-03-2024  8:32
DEMOADD  LST               2,047 31-03-2024  8:28
DEMOADD  MAP                224 31-03-2024  8:32
DEMOADD  OBJ               135 31-03-2024  8:28
6 File(s)                3,655 Bytes.
0 Dir(s)                262,111,744 Bytes free.
```

（3）使用 debug 调试 demoadd.exe，按下面的要求分别给出结果截图。

（a）反汇编指令 U

TODO: 你的截图

```
C:\LEARN>debug demoadd.exe
-u
0777:0000 B87607      MOV     AX,0776
0777:0003 8ED8        MOV     DS,AX
0777:0005 A00000      MOV     AL,[0000]
0777:0008 02060100     ADD     AL,[0001]
0777:000C A20200      MOV     [0002],AL
0777:000F B8004C      MOV     AX,4C00
0777:0012 CD21        INT     21
0777:0014 FF00        INC     WORD PTR [BX+SI]
0777:0016 C682FBFE00     MOV     BYTE PTR [BP+SI+FEFB],00
0777:001B 2BC0        SUB     AX,AX
0777:001D 50          PUSH    AX
0777:001E 8D86FBFE      LEA     AX,[BP+FEFB]
```

(b) 显示寄存器指令 R

TODO: 你的截图

```
-r
AX=FFFF BX=0000 CX=0024 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=0766 ES=0766 SS=0775 CS=0777 IP=0000  NU UP EI PL NZ NA PO NC
0777:0000 B87607      MOV     AX,0776
```

(c) 单步调试指令 T, 执行 2 次 T 指令后的结果

TODO: 你的截图

```
-t
AX=0776 BX=0000 CX=0024 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=0766 ES=0766 SS=0775 CS=0777 IP=0003  NU UP EI PL NZ NA PO NC
0777:0003 8ED8        MOV     DS,AX
-t
AX=0776 BX=0000 CX=0024 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=0776 ES=0766 SS=0775 CS=0777 IP=0005  NU UP EI PL NZ NA PO NC
0777:0005 A00000      MOV     AL,[0000] DS:0000=11
```

(d) 显示内存指令 D, 显示 数据段 前 3 个字节的的内容

TODO: 你的截图

```
-d 0 2
0776:0000 11 22 33
```

(e) 执行程序指令 G, 运行到 add al, y 行

TODO: 你的截图

```
-g 8
AX=0711 BX=0000 CX=0024 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=0776 ES=0766 SS=0775 CS=0777 IP=0008  NU UP EI PL NZ NA PO NC
0777:0008 02060100     ADD     AL,[0001] DS:0001=22
```

(f) 执行程序指令 G

TODO: 你的截图

```
-g
Program terminated normally
```

2、算术指令程序设计。在数据段分别定义 stu, X,Y,Z,W 变量。其中, 将 stu 初始化为自己的学号, X,Y,W 可以任意初始化。截图中 stu 的初始值和学号不相符的将判定为抄袭。(说明: 为简化编程, 除法运算后不考虑余数)

(1) 汇编、连接后的截图

TODO: 你的截图

```
C:\LEARN>link d1t2

Microsoft (R) Overlay Linker Version 3.60
Copyright (C) Microsoft Corp 1983-1987. All rights reserved.

Run File [D1T2.EXE]:
List File [NUL.MAP]: d1t2
Libraries [.LIB]:
LINK : warning L4021: no stack segment

C:\LEARN>dir d1t2.*
Directory of C:\LEARN\
D1T2      ASM                785 31-03-2024   9:09
D1T2      CRF                250 31-03-2024   9:11
D1T2      EXE                563 31-03-2024   9:13
D1T2      LST               2,763 31-03-2024   9:11
D1T2      MAP                224 31-03-2024   9:13
D1T2      OBJ                167 31-03-2024   9:11
        6 File(s)            4,752 Bytes.
        0 Dir(s)             262,111,744 Bytes free.
```

(2) 反汇编的截图

TODO: 你的截图

```
-u
0777:0000 B87607      MOV     AX,0776
0777:0003 8ED8      MOV     DS,AX
0777:0005 A00D00      MOV     AL,[000D]
0777:0008 2A060A00     SUB     AL,[000A]
0777:000C F62E0B00     IMUL    BYTE PTR [000B]
0777:0010 BB0500      MOV     BX,0005
0777:0013 F6FB      IDIV    BL
0777:0015 D1E8      SHR     AX,1
0777:0017 A20C00      MOV     [000C],AL
0777:001A B8004C      MOV     AX,4C00
0777:001D CD21      INT     21
0777:001F 86FB      XCHG    BH,BL
```

(3) 在进行计算前，显示变量 `stu`、`X`、`Y`、`Z`、`W` 的内存值的截图（只能显示这 5 个变量的内存值，多显示、少显示均扣分）

TODO: 你的截图

```
-d 0 d
0776:0000 32 32 32 33 33 31 32 32-30 32 22 23 00 27      2223312202"#.'
```

(4) 执行完计算后，显示变量 `stu`、`X`、`Y`、`Z`、`W` 的内存值的截图（只能显示这 5 个变量的内存值，多显示、少显示均扣分）

TODO: 你的截图

```
-d 0 d
0776:0000 32 32 32 33 33 31 32 32-30 32 22 23 11 27      2223312202"#.'
```

(5) 源代码

TODO: 你的源代码

```

D: > asmEnv20200508 > asmEnv > learn > ASM d1t2.asm
1   ; empty asm file
2
3   title I love asm
4
5   data segment
6       stu db "2223312202"
7       x db 22H
8       y db 23H
9       z db ?
10      w db 27H
11  data ends
12
13  code segment
14      assume cs:code, ds:data
15      main    proc
16          ; assign the data segment base address to DS
17          mov  ax, data
18          mov  ds, ax
19
20          ; TODO ...
21          ; | add your code between arrows |
22          ; v ----- v
23          mov al,w
24          sub al,x
25          imul y
26          mov bx,5
27          idiv bl
28          shr ax,1
29          mov z,al
30          ; ^ ----- ^
31          ; |           The END           |
32
33          ; method 2: return to dos
34          mov  ax, 4c00h
35          int  21h
36      main    endp
37  code ends
38  end main

```

3、寄存器使用程序设计。寄存器 BL、CL 的值根据需要进行初始化。（说明：为简化编程，除法运算后不考虑余数）

（1）反汇编的截图

TODO: 你的截图

```

-u
0777:0000 B87607      MOV     AX,0776
0777:0003 8ED8        MOV     DS,AX
0777:0005 B314        MOV     BL,14
0777:0007 B10A        MOV     CL,0A
0777:0009 8AC3        MOV     AL,BL
0777:000B 98          CBW
0777:000C F6F9        IDIV    CL
0777:000E F62E0000     IMUL    BYTE PTR [0000]
0777:0012 8BD0        MOV     DX,AX
0777:0014 B8004C        MOV     AX,4C00
0777:0017 CD21        INT     21
0777:0019 FE00        INC     BYTE PTR [BX+SI]
0777:001B 2BC0        SUB     AX,AX
0777:001D 50          PUSH    AX
0777:001E 8D86FBFE     LEA     AX,[BP+FEFB]

```

(2) 在进行计算前，显示寄存器的值截图

TODO: 你的截图

赋初值前

```

-r
AX=FFFF BX=0000 CX=0029 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=0766 ES=0766 SS=0775 CS=0777 IP=0000 NU UP EI PL NZ NA PO NC
0777:0000 B87607      MOV     AX,0776

```

赋完初值后

```

AX=0014 BX=0014 CX=000A DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=CC76 ES=0766 SS=0775 CS=0777 IP=000C NU UP EI PL NZ NA PO NC
0777:000C F6F9        IDIV    CL

```

(3) 执行完计算后，显示寄存器的值截图

TODO: 你的截图

```

-r
AX=0004 BX=0014 CX=000A DX=0004 SP=0000 BP=0000 SI=0000 DI=0000
DS=0776 ES=0766 SS=0775 CS=0777 IP=0014 NU UP EI PL NZ NA PO NC
0777:0014 B8004C        MOV     AX,4C00

```

(4) 源代码

TODO: 你的源代码

```

1  ; empty asm file
2
3  title I love asm
4
5  data segment
6  | X db 2H
7  data ends
8
9  code segment
10  assume cs:code, ds:data
11  main    proc
12      ; assign the data segment base address to DS
13      mov  ax, data
14      mov  ds, ax
15
16      ; TODO ...
17      ; | add your code between arrows |
18      ; v ----- v
19      mov  bl,20

```

```

20      mov  cl,10
21      mov  al,bl
22      cbw
23      idiv cl
24      imul X
25      mov  dx,ax
26
27      ; ^ ----- ^
28      ; |           The END           |
29
30      ; method 2: return to dos
31      mov  ax, 4c00h
32      int  21h
33  main    endp
34 code ends
35 end main

```