

课 程 实 验 报 告

**课程名称： 汇编语言程序设计实验**

**实验名称： 实验一 编程基础**

**实验时间： 2017-10-10，18：30-21：50 实验地点： 南一楼804室15号实验台**

**指导教师： 许向阳**

**专业班级：信息安全 201601班**

**学 号： U2016 姓 名：**

**同组学生： 无 报告日期： 2017年 10 月 14日**

**原创性声明**

  本人郑重声明：本报告的内容由本人独立完成，有关观点、方法、数据和文献等的引用已经在文中指出。除文中已经注明引用的内容外，本报告不包含任何其他个人或集体已经公开发表的作品或成果，不存在剽窃、抄袭行为。

特此声明！

学生签名：

日期：2017.10.14

成绩评定

|  |  |  |
| --- | --- | --- |
| 实验完成质量得分（70分）（实验步骤清晰详细深入，实验记录真实完整等） | 报告撰写质量得分（30分）（报告规范、完整、通顺、详实等） | 总成绩（100分） |
|  |  |  |

指导教师签字：

                     日 期：

**目录**

1. 实验目的与要求 2
2. 实验内容 2
3. 实验过程 4
   1. 任务1 4
      1. 实验步骤 4
      2. 实验记录与分析 5
   2. 任务2 6
      1. 实验步骤 6
      2. 流程图 6
      3. 源程序 7
      4. 实验步骤 8
      5. 实验记录与分析 9
4. 总结与体会 10

参考文献 11

1. 实验目的与要求

本次实验的主要目的与要求有下面6点，所有的任务都会围绕这6点进行，希望大家事后检查自己是否达到这些目的与要求。

1. 掌握汇编源程序编辑工具、汇编程序、连接程序、调试工具TD的使用；
2. 理解数、符号、寻址方式等在计算机内的表现形式；
3. 理解指令执行与标志位改变之间的关系；
4. 熟悉常用的DOS功能调用；
5. 熟悉分支、循环程序的结构及控制方法，掌握分支、循环程序的调试方法；
6. 加深对转移指令及一些常用的汇编指令的理解。
7. 实验内容

任务1：《80X86汇编语言程序设计》教材中 P31的 1.14题。

要求：(1) 直接在TD中输入指令，完成两个数的求和、求差的功能。求和/差后的结果放在(AH)中。

(2) 请事先指出执行指令后(AH)、标志位 SF、OF、CF、ZF的内容。

(3) 记录上机执行后的结果，与（2）中对应的内容比较。

(4)求差运算中，若将A、B视为有符号数，且A>B, 标志位有何特点？

若将A、B视为无符号数，且A>B, 标志位又有何特点？

任务2. 《80X86汇编语言程序设计》教材中 P45的 2.3题。

要求：（1）分别记录执行到“MOV CX，10”和“INT 21H”之前的(BX), (BP),(SI),(DI)各是多少。

（2）记录程序执行到退出之前数据段开始40个字节的内容，指出 程序运行结果是否与设想的一致。

(3)在标号LOPA前加上一段程序，实现新的功能：先显示提示信息“Press any key to begin!”, 然后，在按了一个键之后继续执行LOPA处的程序。

任务3. 《80X86汇编语言程序设计》教材中 P45的 2.4题的改写。

要求：(1) 实现的功能不变，对数据段中变量访问时所用到的寻址方式中的寄存器改成32位寄存器。

(2) 内存单元中数据的访问采用变址寻址方式。

(3) 记录程序执行到退出之前数据段开始40个字节的内容，检查 程序运行结果是否与设想的一致。

(4)在TD代码窗口中观察并记录机器指令代码在内存中的存放形式，并与TD中提供的反汇编语句及自己编写的源程序语句进行对照，也与任务2做对比。（相似语句记录一条即可，重点理解机器码与汇编语句的对应关系，尤其注意操作数寻址方式的形式）。

（5）观察连续存放的二进制串在反汇编成汇编语言语句时，从不同字节位置开始反汇编，结果怎样？理解 IP/EIP指明指令起始位置的重要性。

任务4. 内存单元的访问。

以四种不同的内存寻址方式，将自己学号的后四位依次存储到 以 XUEHAO开头的存储区中，要求学号的存放以字符方式存放。

要求：在报告中给出完整的程序；给出运行效果截图；（不需要画流程图）；在程序注释中，明确指出访问存储单元时，用的是什么寻址方式。

任务**5.** 设计实现一个网店商品信息查询的程序。

1、实验背景

有一个老板在网上开了2个网店SHOP1,SHOP2；每个网店有n种商品销售，不同网店之间销售的商品种类相同，但数量和销售价格可以不同。每种商品的信息包括：商品名称（10个字节，名称不足部分补0），进货价(字类型)，销售价（字类型），进货总数（字类型），已售数量（字类型），利润率（%）【=（销售价\*已售数量-进货价\*进货总数）\*100/（进货价\*进货总数），字类型】。老板管理网店信息时需要输入自己的名字（10个字节，不足部分补0）和密码（6个字节，不足部分补0），登录后可查看商品的全部信息；顾客（无需登录）可以查看所有网店中每个商品除了进货价、利润率以外的信息。

例如：

BNAME DB ‘ZHANG SAN’,0 ;老板姓名（必须是自己名字的拼音）

BPASS DB ‘test’，0，0 ；密码

N EQU 30

S1 DB ‘SHOP1’,0 ;网店名称，用0结束

GA1 DB ‘PEN’, 7 DUP(0) ; 商品名称

DW 35，56，70，25，？ ；利润率还未计算

GA2 DB ‘BOOK’, 6 DUP(0) ; 商品名称

DW 12，30，25，5，？ ；利润率还未计算

GAN DB N-2 DUP( ‘Temp-Value’,15，0，20，0，30，0，2，0，？，？) ;除了2个已经具体定义了商品信息以外，其他商品信息暂时假定为一样的。

S2 DB ‘SHOP2’,0 ;网店名称，用0结束

GB1 DB ‘BOOK’, 6 DUP(0) ; 商品名称

DW 12，28，20，15，？ ；利润率还未计算

GB2 DB ‘PEN’, 7 DUP(0) ; 商品名称

DW 35，50，30，24，？ ；利润率还未计算

……

2、功能一：提示并输入登录用户的姓名与密码

（1）使用9号DOS系统功能调用，先后分别提示用户输入姓名和密码。

（2）使用10号DOS系统功能调用，分别输入姓名和密码。输入的姓名字符串放在以in\_name为首址的存储区中，密码放在以in\_pwd为首址的存储区中，进入功能二的处理。

（3）若输入姓名时只是输入了回车，则将0送到AUTH字节变量中，跳过功能二，进入功能三；若在输入姓名时仅仅输入字符q，则程序退出。

3、功能二：登录信息认证

（1）使用循环程序结构，比较姓名是否正确。若不正确，则跳到（3）。

（2）若正确，再比较密码是否相同，若不同，跳到（3）。

（3）若名字或密码不对，则提示登录失败，并回到“功能一（1）”的位置，提示并重新输入姓名与密码。

（4）若名字和密码均正确，则将1送到AUTH变量中，进到功能三。

提示：字符串比较时，当采用输入串的长度作为循环次数时，若因循环次数减为0而终止循环，则还要去判断网店中定义的字符串的下一个字符是否是结束符0，若是，才能确定找到了（这样做是为了避免输入的字符串仅仅是数据段中所定义字符串的子集的误判情况）。

4、功能三：计算指定商品的利润率。

（1）提示用户输入要查询的商品名称。若未能在第一个网店中找到该商品，重新提示输入商品名称。若只输入回车，则回到功能一（1）。

（2）判断登录状态，若是已经登录的状态，转到（3）。否则，转到（4）。

（3）首先计算第一个网店该商品的利润率PR1，然后在第二个网店中寻找到该商品，也计算其利润率PR2。最后求出该商品的平均利润率APR=(PR1+PR2)/2。进入功能四。

（4）若是未登录状态，则只在下一行显示该商品的名称，然后回到功能一（1）。

要求尽量避免溢出。

提示：使用循环程序结构，注意寻址方式的灵活使用。结果只保留整数部分。

5、功能四：将功能三计算的平均利润率进行等级判断，并显示判断结果。

（1）等级显示方式：若平均利润率大于等于90%，显示“A”；大于等于50%，显示“B”；大于等于20%，显示“C”；大于等于0%，显示“D”；小于0%，显示“F”。

提示：使用分支程序结构，采用2号DOS系统功能调用显示结果（注意，“%”是不要出现在计算式子和指令语句中的）。

（2）使用转移指令回到“功能一（1）”处（提示并输入姓名和密码）。

1. 实验过程
   1. 任务1
      1. 实验步骤

1. 准备上机实验环境。

2.在TD的代码窗口中的当前光标下输入第一个运算式对应的两个8位数值对应的指令语句MOV AH,01001101B；ADD 01011010B;观察代码区显示的内容与自己输入字符之间的关系；然后确定CS:IP指向的是自己输入的第一条指令的位置，单步执行三次，观察寄存器内容的变化，记录标志寄存器的结果。

预计ADD执行之后（AH）=08DH SF=1、OF=1、CF=0、ZF=0

重复上述过程，将剩下几个表达式计算完毕，比较结果。

3. 输入MOV AH,10101001B；ADD AH,11011101B；观察标志位特点；

预计ADD执行之后（AH）=86H;SF=1,OF=0,CF=1,ZF=0.

4.输入MOV AH,01100101B;ADD AH,11011101B;观察标志位特点。

预计ADD执行之后(AH)=42H;SF=0,OF=0,CF=1,ZF=0.

表3-1 任务一预计结果表

|  |  |  |
| --- | --- | --- |
| 操作1 | 操作2 | 预计结果 |
| MOV AH,01001101B | ADD 01011010B | （AH）=08DH SF=1、OF=1、CF=0、ZF=0 |
| MOV AH,10101001B | ADD AH,11011101B | （AH=86H;SF=1,OF=0,CF=1,ZF=0 |
| MOV AH,01100101B; | ADD AH,11011101B | (AH)=42H;SF=0,OF=0,CF=1,ZF=0 |

* + 1. 实验记录与分析

1. 实验环境条件：8G内存；mac OS下DOSBox0.72； TD.EXE 5.0。

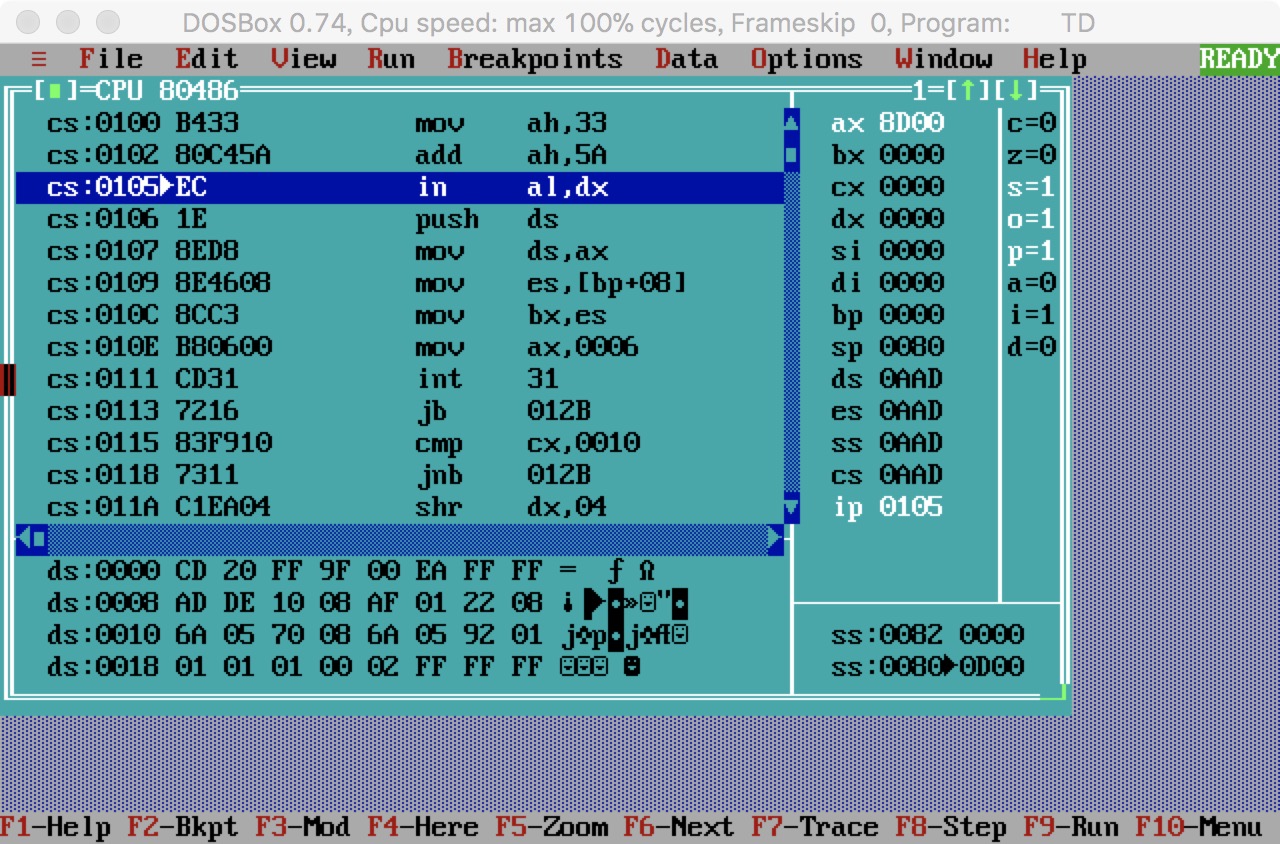
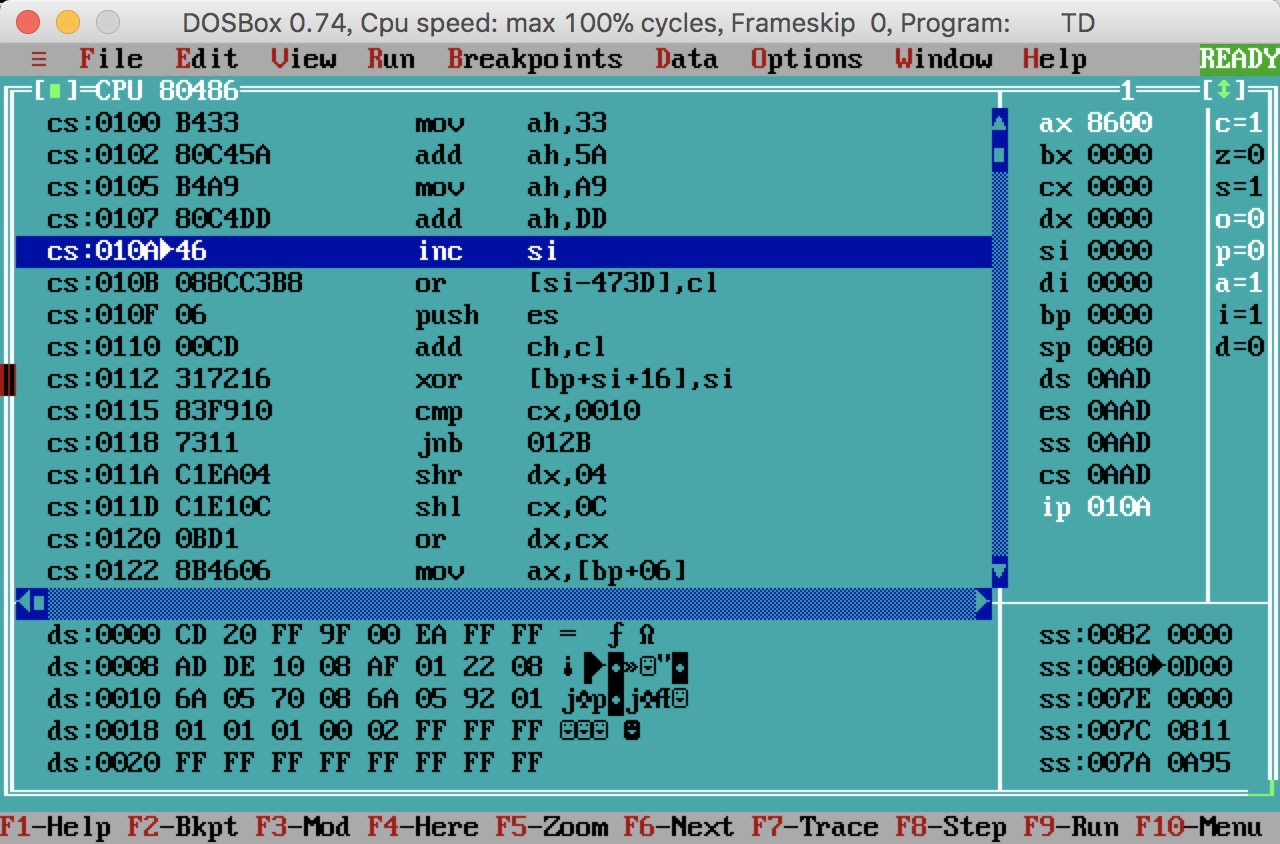
2. 输入第一个指令MOV AH,01001101B;ADD AH,01011010B后，进行单步调试。调试结束后，标志位产生了如图所示变化。

图3.1.1 执行完测试语句后的状态

3.输入MOV AH,10101001B；ADD AH,11011101B后，单步调试，观察结果如下。



* 1. 任务2
     1. 实验步骤

1. 用文本编辑器输入 题目中的代码，将其保存为test.asm
2. 在dosbox中输入命令：masm test.asm,生成test.obj
3. 在dosbox中输入：link test.obj,生成test.exe文件
4. 在dosbox中输入：td test.exe调试
5. 单步调试，预计在执行”MOV CX 10”之前，BX=0014H,BP=001EH,SI=0000H,DI=000AH;预计在执行到”INT 21H”之前,BX=
   * 1. 流程图

图3.2.1是任务2求一个数的立方值的程序流程图。

图3.2.1 计算立方值的程序流程图

* + 1. 源程序

.386

STACK SEGMENT USE16 STACK

DB 200 DUP(0)

STACK ENDS

DATA SEGMENT USE16

INPUT DB 'PLEASE INPUT X(0-9):$'

TAB DW 0,1,8,27,64,125,216,343,512,729

X DB ?

XXX DW ?

DATA ENDS

CODE SEGMENT USE16

ASSUME CS:CODE,DS:DATA,SS:STACK

BEGIN: MOV AX, DATA

MOV DS, AX

MOV DX, OFFSET INPUT

MOV AH, 9

INT 21H ；显示PLEASE INPUT X(0-9): //录入错误：“；”写成了中文

MOV AH, 1

INT 21H ；从键盘接受一数字x的ASCII码

AND AL, 0FH //录入错误： 数字“0”敲成了字母“O”

MOV X, AL ；x的真值 → AL → X

MOV EBX，AL ；x的真值 → EBX //设计错误，应改成：XOR EBX, EBX

MOV BL, AL

MOV AX, TAB[EBX\*2] ； （TAB + [2 \* EBX]）→ AX

MOV XXX, AX ； 保存立方值

MOV AH, 4CH

INT 21H

CODE ENDS

END BEGIN

* + 1. 实验步骤

1. 准备上机实验环境。

2. 使用编辑程序EDIT.EXE录入源程序，存盘文件名为CUBE.ASM。使用MASM 6.0汇编源文件。即MASM CUBE；观察提示信息，若出错，则用编辑程序修改错误，存盘后重新汇编，直至不再报错为止。

3. 使用连接程序LINK.EXE将汇编生成的CUBE.OBJ文件连接成执行文件。

即LINK CUBE；

若连接时报错，则依照错误信息修改源程序。之后重新汇编和连接，直至不再报错并生成CUBE.EXE文件。

4. 执行该程序。即在命令行提示符后输入CUBE后回车，观察执行现象。准备输入数字3进行测试，观察执行的结果。

（以上步骤是所有编程实验需要经历的，在后续任务或实验报告中可以简化成一条，如：

“准备上机环境，编辑、汇编、连接文件CUBE。”）

5. 使用TD.EXE观察CUBE的执行情况。即 TD CUBE.EXE回车

（1）观察CS、IP、SP、DS、ES、SS的值。

（2）单步执行开始2条指令，观察DATA的实际值，以及DS的改变情况。

（3）观察SS：0至SS：SP区域的数据值。

（4）观察DS：0开始数据区，找到各变量在数据段中的位置和值。

（5）观察第三条语句中源操作数的值，是否和INPUT变量的偏移地址相同。

（6）执行第3至7条指令，输入数字3。观察AL的值是否为33H。

（7）执行到MOV AX, TAB[EBX\*2]，观察源操作数的具体值。

（8）执行MOV XXX, AX，观察目的操作数的形式。到数据段中观察XXX的值是否是3的立方值。

（9）依次输入0~9，观察结果是否都正确。

（10）输入字母A，逗号标点等，观察运算结果。

6. 将程序重新装入TD中（或将CS：IP重置到MOV AH，9的位置），在执行9号功能调用之前，用TD将数据段中INPUT缓冲区的‘$’（24H）改成其他数值（如00H），再执行9号功能调用，观察现象。

7. 当调用1号功能时，若输入大写字母‘A’，则送到XXX的值是哪个存储单元的值；若输入的是‘K’，则送到XXX的值又是哪个存储单元的值。

（以上3个步骤是针对实验要求、提示的信息和每个人自己在学习中不太理解的内容设计的，是训练学生有计划和目的地做事情，反映学生充分准备了实验的重要内容）

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

* + 1. 实验记录与分析

（注：包括实验条件、输入/输出、错误现象等信息的记录，以及问题分析和修改，实验数据得到的结论等。每条实验记录都会对应到实验步骤，也即每个步骤都有对应的实验记录）

1. 实验环境条件：P3 1GHz，256M内存；WINDOWS XP下DOSBox0.72；EDIT.EXE 2.0；MASM.EXE 6.0； LINK.EXE 5.2; TD.EXE 5.0。

2. 汇编源程序时，汇编程序报了3个错误。它们是：

CUBE.asm(17): error A2044: invalid character in file

CUBE.asm(22): error A2022: instruction operands must be the same size

CUBE.asm(20): error A2006: undefined symbol : OFH

按照提示，对对应行仔细检查，发现了错误原因。见源程序中“//”后的说明（直接在源程序中修改并在注释中说明错误原因或修改过程）。

3. 连接过程没有发生异常。

4. 执行之后在新的一行上显示了字符串PLEASE INPUT X(0-9):（可以截图，下同）

输入3之后在冒号后显示了一个3，程序就退出到命令行提示符。说明程序执行基本正常，但由于结果没有显示，所以需要用TD去观察计算结果是否正确。

5. 用TD调入CUBE.EXE后

（1）(CS)=0B0AH、(IP)=0000H、(SP)=00C8H、(DS)=0AEAH、(ES)=0AEAH、(SS)=0AFAH。可从段首址取值的不同得知代码段、数据段、堆栈段处在不同位置，且前后次序是数据段、堆栈段、代码段。

（2）单步执行开始2条指令，DATA的值=0B07H，(DS)→0B07H。观察到TD的数据显示区被切换到了ES:0,若还希望显示DS:0,就需要用Goto指令重新设定。发现此时数据段才是程序的实际位置，各个段的实际次序为堆栈段、数据段、代码段，这与源程序定义各段的次序一致。

（3）SS：0至SS：SP区域的数据值在程序没有执行时均为0。单步执行一次后靠近栈顶的几个字发生了变化,不知为何?

（4）DS：0开始数据区存放了INPUT变量为首址定义的字符串。EA=15H开始存放TAB立方值表。EA=29H存放X（当前值为0）；EA=30H存放XXX（当前值为0）。

（5）TD中显示的第三条语句为MOV DX，0000，源操作数的值和INPUT变量的偏移地址相同（均为0）。

（6）输入数字3。AL的值从24H变成了33H。

（7）MOV AX, TAB[EBX\*2]在TD显示的形式为MOV AX，[2\*EBX+00000015]说明TAB代表的EA=00000015H，且是按照双字处理的。

（8）MOV XXX, AX在TD显示的形式为MOV [002A]，AX。执行后DS：（002A）=001BH（即27）是3的立方值。

（9）略（留给同学们观察后记录。）

（10）略（留给同学们观察后记录。）

6. 略（留给同学们观察后记录。）

7. 略（留给同学们观察后记录。）

1. 总结与体会

（主要写3个方面的内容：1.整个实验做下来你对实验所涉及的知识点的主要认识与收获；2.回答一下思考题（若有）；3.你自己在实验操作过程中的经验教训与存在的问题。可以分小标题，也可以直接写几段内容）

通过任务1的实验，让我知道了TD不仅可以调入现成的执行程序进行调试，而且能随时输入和测试单一一条指令是否正确，执行效果如何，这将方便未来的学习过程。另外，通过观察，计算机内的加减运算，无论是否有符号数，对应的标志位都是设定好了的，如何使用这些标志，完全由程序员选择的指令来决定，这就要求我们编写程序时要理解好题目的语义，选择合适的指令语句，而不是我写了个负数交给汇编程序，系统就会自动选择有符号指令的。

任务2的实验表明，源程序定义的段的次序，与调到内存中段的次序是有对应关系的。但数据段开始没有被赋值到实际的数据段首址中，需要执行了我们的相关赋值语句后才变得实际有效。另外，各种寻址方式在内存中确实不如源程序直观了，但可以看到变量的偏移值都被很好的计算出来了，我们可以通过源程序中变量与内存中对应位置的信息获得该变量的偏移地址，方便我们在数据显示区输入该地址，查看对应的变量内容。

任务2中若输入0~9之外的字符，结果其实是没有意义的。要解决这个问题，可以增加对输入的值的范围进行判断的语句，超出范围时报错，并要求重新输入。（这是思考题的回答）

本次上机不仅提高了编程水平，熟悉了工具的使用，而且加深了对一些知识的理解。主要的经验教训如下：

首先，更加感受到实验前准备的意义。例如：上机前准备越充分（如先编好源程序，制定好准备做的一些步骤），上机的时候目的越明确，可以解决较多的问题。

其次，录入程序时要注意一些细节，比如中文分号、字母O等问题，虽然汇编程序指出其所在行有错，但很难发现具体是哪个符号错了，耽误了不少时间。TD在程序细节的观察、动态修改方面有很大的作用，要主动用TD的调试功能帮助自己发现、理解与解决问题。

最后，由于操作不够熟练，时间比较紧张等原因，还有些问题需要以后进一步解决，如堆栈中数据变化的原因、各个段在内存中存放的关系、是否可跟踪到INT 21H中去、多次调入程序时初始的段值是否相同等等。

参考文献

[1] 作者.书名.版本(第×版).译者.出版地:出版者,出版年：起页-止页

[序号] URL: 网络地址，如URL:http://www.cbs.dtu.dk/services

------------------------------------------------------

**备注：**本模板任务1属于验证性实验，任务2是设计型的实验。由于验证性、设计型、研究性、综合性等实验的要求各有差异，因此，标题下的内容是可以调整的，具体调整要求见每次布置题目时老师给出的说明。

排版总体要求：

A4纸版面（若打印，则双面打印），除了每次实验的封面和标题之外的正文用5号宋体1.25倍行距；图表用小5号宋体；源程序用小5号宋体、单倍行距。

一级标题居中，小节（二级、三级）标题顶头，其他序号及段落开始空2个汉字。源程序整体空两个汉字。公式居中。参考文献列表全顶头。图表都要居中，按照任务对应的小节编号（节内从1开始依次编号），并给出图名（放在图下）、表名（放在表上）。