

**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： 2017-3-16，14：30-17：30 实验地点： 南一楼804室15号实验台**

**指导教师： 金良海**

**专业班级：计算机科学与技术201504班**

**学 号： U201514564 姓 名： 张仕钊**

**同组学生： 无 报告日期： 2017年 3 月 17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：2017.3.17

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 1](#_Toc477989079)

[2 实验内容 1](#_Toc477989080)

[3 实验过程 3](#_Toc477989081)

[3.1 任务1 3](#_Toc477989082)

[3.1.1 实验步骤 3](#_Toc477989083)

[3.1.2 实验记录与分析 4](#_Toc477989084)

[3.2 任务2 5](#_Toc477989085)

[3.2.1 源程序 5](#_Toc477989086)

[3.2.2 实验步骤 6](#_Toc477989087)

[3.2.3 实验记录与分析 6](#_Toc477989088)

[3.3 任务三 9](#_Toc477989089)

[3.3.1 源程序 9](#_Toc477989090)

[3.3.2 实验步骤 10](#_Toc477989091)

[3.3.3 实验记录与分析 10](#_Toc477989092)

[3.4 任务四 11](#_Toc477989093)

[3.4.1 源程序 11](#_Toc477989094)

[3.4.2 实验步骤 15](#_Toc477989095)

[3.4.3 实验记录与分析 15](#_Toc477989096)

[4 总结与体会 16](#_Toc477989097)

[参考文献 17](#_Toc477989098)

# 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。

# 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：**设计实现一个学生成绩查询的程序。

1、实验背景

在以BUF为首址的字节数据存储区中，存放着n个学生的课程成绩表（百分制），每个学生的相关信息包括：姓名（占10个字节，结束符为数值0），语文成绩（1个字节），数学成绩（1个字节），英语成绩（1个字节），平均成绩（1个字节）。

例如：

N EQU 30

BUF DB ‘zhangsan’,0,0 ;学生姓名，不足10个字节的部分用0填充

DB 100, 85, 80,？ ; 平均成绩还未计算

DB ‘lisi’,6 DUP(0)

DB 80, 100, 70,?

DB N-3 DUP( ‘TempValue’,0,80,90,95,?) ;除了3个已经具体定义了学生信息的成绩表以外，其他学生的信息暂时假定为一样的。

DB ‘wangwu’，0，0，0，0 ；最后一个必须是自己名字的拼音

DB 85, 85, 100, ?

2、功能一：提示并输入待查询成绩的学生姓名

（1）使用9号DOS系统功能调用，提示用户输入学生姓名。

（2）使用10号DOS系统功能调用，输入学生姓名。输入的姓名字符串放在以in\_name为首址的存储区中。

（3）若只是输入了回车，则回到“(1)”处重新提示与输入；若仅仅输入字符q，则程序退出,否则，准备进入下一步处理。

3、功能二：以学生姓名查询有无该学生

（1）使用循环程序结构，在成绩表中查找该学生。

（2）若未找到，就提示用户该学生不存在，并回到“功能一（1）”的位置，提示并重新输入姓名。

（3）若找到，则将该学生课程成绩表的起始偏移地址保存到POIN字变量中。

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，则还要去判断成绩表中名字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名字仅仅是数据段中所定义名字的子集的误判情况）。

4、功能三：计算所有学生的平均成绩

使用算数运算相关指令计算并保存每一个学生的平均成绩。

平均成绩计算公式：(A\*2+B+C/2)/3.5，即将语文成绩A乘以权重2、英语成绩C除以权重2后，与数学成绩B一起求和，再计算该生的平均成绩。要求避免溢出。

提示：使用循环程序结构，注意寻址方式的灵活使用。把小数3.5转换成分数后再运算避免使用浮点数指令。

5、功能四：将功能二查到的学生的平均成绩进行等级判断，并显示判断结果。

（1）平均成绩等级显示方式：若平均成绩大于等于90分，显示“A”；大于等于80分，显示“B”；大于等于70分，显示“C”；大于等于60分，显示“D”；小于60分，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。

（2）使用转移指令回到“功能一（1）”处（提示并输入姓名）。

# 实验过程

## 任务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,0110011B；MOV BH,1011010B；ADD AH,BH；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。

若以16进制输入，则需要在输入的数据后加上H这一16进制标志，如果该16进制的首位为字母，则在开头加一个数字0。

预计ADD执行之后（AH）=8DH 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. 加法：

输入MOV AH,-0101001B；MOV BH,-1011101B；ADD AH,BH；观察标志位特点；

输入MOV AH, 1100101B；MOV BH,-1011101B；ADD AH,BH；观察标志位特点；

有符号数减法：

输入MOV AH, 1011010B；MOV BH, 0110011B；SUB AH,BH；观察标志位特点；

输入MOV AH,-0101001B；MOV BH,-1011101B；ADD AH,BH；观察标志位特点；

输入MOV AH, 1100101B；MOV BH,-1011101B；SUB AH,BH；观察标志位特点；

无符号数减法：

输入MOV AH, 1011010B；MOV BH, 0110011B；SUB AH,BH；观察标志位特点；

输入MOV AH, 1011101B；MOV BH, 0101001B；SUB AH,BH；观察标志位特点；

输入MOV AH, 1100101B；MOV BH, 1011101B；SUB AH,BH；观察标志位特点；

4. 输入指令MOV AH,-128并执行后，AH寄存器的内容为0D8H，查看寄存器的值只需寻找td窗口中对应的寄存器并查看其中的值，修改的话只需执行MOV指令即可。标志寄存器在td窗口的最右侧。

### 实验记录与分析

1. 实验环境条件：i7 5500U 2.4GHz，8G内存；WINDOWS 10下DOSBox0.74；TD.EXE 5.0。

2. 执行三条指令后的结果如图3.1.1所示。可以看出，计算结果在AX的高字节中（8DH）与标志位的状态(CF=0,ZF=0,SF=1,OF=1)与事前预期的是一致的。

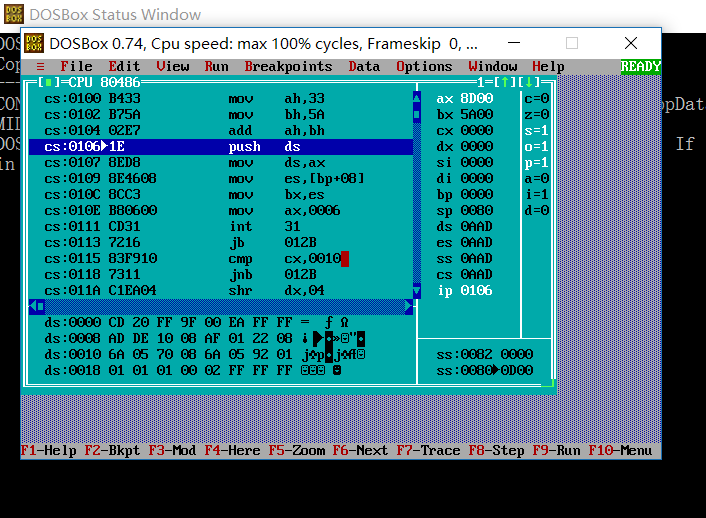


图3.1.1 执行完测试语句后的状态

3.同理，输入其它的指令，其结果与估计均相同，结果见表3.1.2所示。注意：要保持AH大于BH。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | SF | OF | CF | ZF |
| X1=+0110011B  X2=+1011010B | 有符号数减法 | 0 | 0 | 0 | 0 |
| 无符号数减法 | 0 | 0 | 0 | 0 |
| X1=-0101001B  X2=-1011101B | 加法 | 0 | 1 | 1 | 0 |
| 有符号数减法 | 0 | 0 | 0 | 0 |
| 无符号数减法 | 0 | 0 | 0 | 0 |
| X1=+1100101B  X2=-1011101B | 加法 | 0 | 0 | 1 | 0 |
| 有符号数减法 | 1 | 1 | 1 | 0 |
| 无符号数减法 | 0 | 0 | 0 | 0 |

由表可以分析得出加法中只有相同符号的数相加才可能产生进位与溢出，正数与负数相加不会产生进位与溢出。

减法中，只有不同符号的数相减会产生进位或移除，相同符号的数相减不会产生进位和溢出。

## 任务2

### 源程序

.386

STACK SEGMENT USE16 STACK

DB 200 DUP(0)

STACK ENDS

DATA SEGMENT USE16

BUF1 DB 0,1,2,3,4,5,6,7,8,9

BUF2 DB 10 DUP(0)

BUF3 DB 10 DUP(0)

BUF4 DB 10 DUP(0)

DATA ENDS

CODE SEGMENT USE16

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

START: MOV AX,DATA

MOV DS,AX

MOV SI,OFFSET BUF1

MOV DI,OFFSET BUF2

MOV BX,OFFSET BUF3

MOV BP,OFFSET BUF4

MOV CX,10

LOPA: MOV AL,[SI]

MOV [DI],AL

INC AL

MOV [BX],AL

ADD AL,3

MOV DS:[BP],AL

INC SI

INC DI

INC BP

INC BX

DEC CX

JNZ LOPA

MOV AH,4CH

INT 21H

CODE ENDS

END START

### 实验步骤

1. 准备上机实验环境，编辑、汇编、连接文件test。

2. 使用TD.EXE观察TEST的执行情况。即 TD TEST.EXE回车

（1）执行到“MOV CX,10”，观察（BX）,(BP),(SI),(DI)各是多少。

（2）执行到“INT 21H”，观察（BX）,(BP),(SI),(DI)各是多少。

（3）观察DS：0开始数据区，记录程序执行到退出之前DS数据段开始40个字节的内容。

（4）编辑文件test.asm,在DATA数据段里定义MSG DB ‘Press any key to begin!$’。

（5）在LOPA标记前加上以下5行代码：

MOV AH,9

MOV DX,OFFSET MSG

INT 21H

MOV AH,1

INT 21H

3. 重新汇编、链接文件test ,然后执行，观察是否实现了要求的功能。

4. 使用TD.EXE观察TEST的执行情况。即 TD TEST.EXE回车

在执行到“MOV AH，4CH”时观察（1）到（3）问的各项数据。

### 实验记录与分析

1. 实验环境条件：i7 5500U 2.4GHz，8G内存；WINDOWS 10下DOSBox0.74；EDIT.EXE 2.0；MASM.EXE 6.0； LINK.EXE 5.2; TD.EXE 5.0。

2. 执行到“MOV CX，10”之前时，(BX)=0014H，（BP）=001EH，（SI）=0000H，（DI）=000AH，如图3.2.1所示。

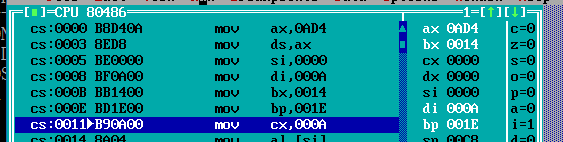


图3.2.1

3. 执行到“INT 21H”之前时，(BX)=001EH，（BP）=0028H，（SI）=000AH，（DI）=0014H，如图3.2.2所示。

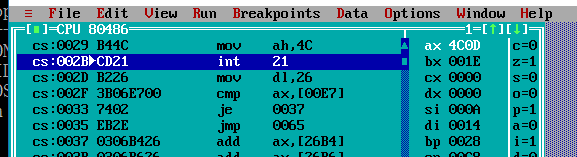


图3.2.2

4. 在执行完之前即INT 21H之前，鼠标点击下边数据区按住ctrl建加G，输入ds：0调出ds的数据，然后查看并记录ds 40个字节的内容，如图3.2.3和3.2.4所示。

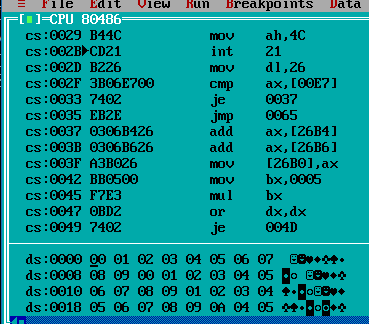


图3.2.3

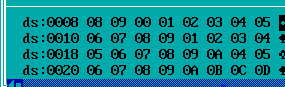


图3.2.4

5. 在修改完代码并汇编、连接后，直接在命令行输入TEST.EXE执行它，其结果如图3.2.5所示，在按下任意键后如图3.2.6，说明该功能得到了实现。

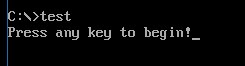


图3.2.5

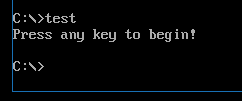


图3.2.6

6. 然后再td它，在执行到MOV AH,4CH时，其各项数据如图3.2.7和3.2.8所示。

其中（BX）=0036H，（BP）=0040H，（SI）=0022H，（DI）=002CH。

DS区的40个字节在图中。

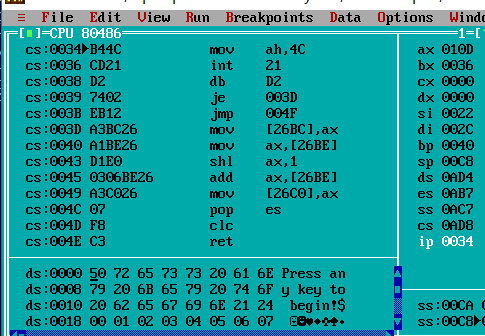


图3.2.7

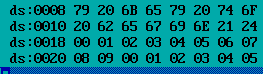


图3.2.8

## 任务三

### 源程序

.386

STACK SEGMENT USE16 STACK

DB 200 DUP(0)

STACK ENDS

DATA SEGMENT USE16

BUF1 DB 0,1,2,3,4,5,6,7,8,9

BUF2 DB 10 DUP(0)

BUF3 DB 10 DUP(0)

BUF4 DB 10 DUP(0)

DATA ENDS

CODE SEGMENT USE16

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

START: MOV AX,DATA

MOV DS,AX

MOV CX,10

MOV ESI,0

LOPA: MOV AL,BUF1[ESI]

MOV BUF2[ESI],AL

INC AL

MOV BUF3[ESI],AL

ADD AL,3

MOV BUF4[ESI],AL

INC ESI

DEC CX

JNZ LOPA

MOV AH,4CH

INT 21H

CODE ENDS

END START

### 实验步骤

1. 准备上机实验环境，编辑、汇编、连接文件test1。

2. 使用TD.EXE观察TEST的执行情况。即 TD TEST.EXE回车

观察DS：0开始数据区，记录程序执行到退出之前DS数据段开始40个字节的内容。

### 实验记录与分析

1. 实验环境条件：i7 5500U 2.4GHz，8G内存；WINDOWS 10下DOSBox0.74；EDIT.EXE 2.0；MASM.EXE 6.0； LINK.EXE 5.2; TD.EXE 5.0。

2. 在执行完之前即INT 21H之前，鼠标点击下边数据区按住ctrl建加G，输入ds：0调出ds的数据，然后查看并记录ds 40个字节的内容，如图3.3.1和3.3.2所示。

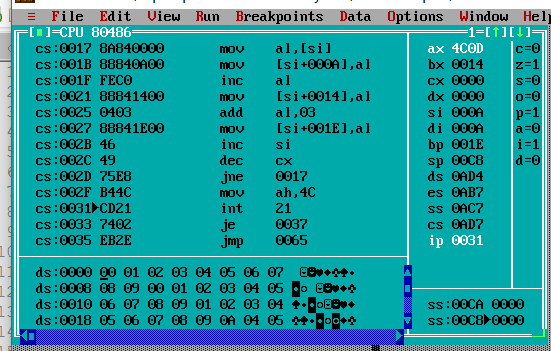


图3.3.1

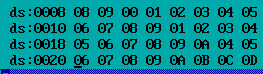


图3.3.2

经观察发现与任务2的数据相同，故任务三成功。

## 任务四

### 源程序

.386

DATA SEGMENT USE16

N EQU 3

BUF DB 'zhangsan',0,0

DB 87,85,80,?

DB 'lisi',6 DUP(0)

DB 65,70,70,?

DB 'shizhao',0,0,0

DB 94,90,100,?

in\_name DB 11

DB ?

DB 11 DUP(0)

print1 DB 0AH,0DH,'Please input the name:$'

print2 DB 0AH,0DH,'This student do not exist!$'

print3 DB 0AH,0DH,'This student have found!$'

POIN DW 0

ascore DB 0AH,0DH,'The score is A.$'

bscore DB 0AH,0DH,'The score is B.$'

cscore DB 0AH,0DH,'The score is C.$'

dscore DB 0AH,0DH,'The score is D.$'

fscore DB 0AH,0DH,'The score is F.$'

DATA ENDS

STACK SEGMENT USE16 STACK

DB 200 DUP(0)

STACK ENDS

CODE SEGMENT USE16

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

BEGIN: MOV AX,DATA

MOV DS,AX

FIRST: LEA DX,print1

MOV AH,9

INT 21H

LEA DX,in\_name

MOV AH,10

INT 21H

MOV BL,in\_name+1

MOV BH,0

CMP BX,0

JE FIRST

CMP BX,1

JE QUIT

MOV CX,0

MOV BX,0

MOV DL,0

MOV AL,BYTE PTR in\_name+1

LOOPA:

MOV BP,2

MOV SI,0

LOOPB : MOV AH,BYTE PTR BUF[BX+SI]

CMP AH,BYTE PTR in\_name[BP+SI]

JNE NEXT

INC SI

INC DL

CMP DL,AL

JNE LOOPB

MOV AH,BYTE PTR BUF[BX+SI]

CMP AH,0

JE EXIST

STC

JC NEXIST

NEXT: ADD BX,14

INC CX

CMP CX,N

JNE LOOPA

STC

JC NEXIST

EXIST: MOV AX,OFFSET BUF

ADD AX,BX

ADD AX,10

MOV POIN,AX

LEA DX,print3

MOV AH,9

INT 21H

STC

JC THIRD

NEXIST: LEA DX,print2

MOV AH,9

INT 21H

STC

JC FIRST

THIRD: MOV SI,10

MOV DI,1

LOOPC: MOV AX,0

MOV BX,0

MOV CX,0

MOV CL,BYTE PTR BUF[SI] ;A

MOV BL,BYTE PTR BUF[SI+1] ;B

MOV AL,BYTE PTR BUF[SI+2] ;C

IMUL CX,2

MOV BP,2

MOV DX,0

IDIV WORD PTR BP

ADD AX,CX

ADD AX,BX

IMUL AX,2

MOV BP,7

MOV DX,0

IDIV WORD PTR BP

MOV BYTE PTR BUF[SI+3],AL

INC DI

ADD SI,14

CMP DI,4

JNE LOOPC

STC

JC FOURTH

FOURTH: MOV BX,POIN

MOV AX,0

MOV AL,BYTE PTR BUF[BX+3]

CMP AL,90

JGE LA

CMP AL,80

JGE LB

CMP AL,70

JGE LC

CMP AL,60

JGE LD

STC

JC LF

LA: LEA DX,ascore

MOV AH,9

INT 21H

STC

JC FIRST

LB: LEA DX,bscore

MOV AH,9

INT 21H

STC

JC FIRST

LC: LEA DX,cscore

MOV AH,9

INT 21H

STC

JC FIRST

LD: LEA DX,dscore

MOV AH,9

INT 21H

STC

JC FIRST

LF: LEA DX,fscore

MOV AH,9

INT 21H

STC

JC FIRST

QUIT: MOV AH,4CH

INT 21H

CODE ENDS

END BEGIN

### 实验步骤

1. 准备上机实验环境，编辑、汇编、连接文件student。

2. 使用TD.EXE观察TEST的执行情况。即 TD TEST.EXE回车

### 实验记录与分析

1. 实验环境条件：i7 5500U 2.4GHz，8G内存；WINDOWS 10下DOSBox0.74；EDIT.EXE 2.0；MASM.EXE 6.0； LINK.EXE 5.2; TD.EXE 5.0。
2. 在编辑的过程中，4个功能是一步一步实现的。
3. 在实现第一个功能中，只出现了一次错误，那就是在文件尾漏掉了MOV AH,4CH这个命令，导致程序无法结束，经调试后发现了这一问题并解决。
4. 第二个功能是这四个功能中最难的，因为我对字符串进行查找时是一个字母一个字母进行比较来实现的，所以我用了一个双重循环，而用汇编语言来实现双重循环特别考验逻辑能力，用了很多次调试才成功。
5. 第三个功能中需要注意的是乘除法不能溢出，其中除法用的是字类型的除，因此要把DX用AX的符号位来进行填充，否则程序会在执行除法的时候中断。
6. 第四个功能只需要多设置几个标志即分支，判断不同的条件调到不同的标志处输出相应的信息即可，比较简单。

7. 其功能实现如图3.4.1所示。

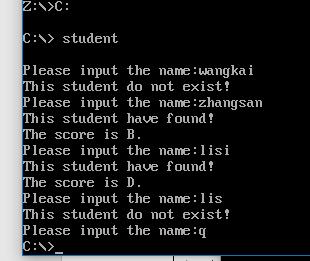


图3.4.1

# 总结与体会

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的实验主要是对td调试器的使用，包括断点设置，查看ds段中的数据，查看标志位等等，这让我对td的使用有了一个很好的锻炼，同时，该程序中运用的循环结构，这让我对循环结构有了一个初步的了解。任务3是在任务2的基础上进行了改进，要求使用变址寻址，同时要用到32位寄存器，这些都比较全面的覆盖了汇编的一些基础语法，也让我对汇编语言的变成越来越熟悉。

而任务4为设计性实验，是此次实验中最难的一个，也是我花费时间最长的一个任务。但是这个任务几乎涵盖了汇编的大部分基础知识。所以通过这个任务，我对汇编的各种指令，条件判断，循环结构，系统指令等都有了比较深刻的了解，相信在以后的实验中运用起来会越来越得心应手。

# 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