

**课 程 实 验 报 告**

**课程名称： 汇编语言程序设计实验**

**实验名称： 实验二 分支循环程序设计**

**实验时间： 2017-10-17，18：30-21：50 实验地点： 南一楼804室5号实验台**

**指导教师： 许向阳**

**专业班级：信息安全 201601班**

**学 号： U201614804 姓 名： 黄永恒**

**同组学生： 无 报告日期： 2017年 10 月 18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：

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](#_Toc500538722)

[2 实验内容 1](#_Toc500538723)

[3 实验过程 3](#_Toc500538724)

[3.1 任务 3](#_Toc500538725)

[3.1.1 设计思想及存储单元分配 3](#_Toc500538726)

[3.1.2 流程图 4](#_Toc500538727)

[3.1.3 源程序 8](#_Toc500538728)

[3.1.4 实验步骤 11](#_Toc500538729)

[3.1.5 实验记录与分析 12](#_Toc500538730)

[4 总结与体会 15](#_Toc500538731)

[参考文献 16](#_Toc500538732)

# 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. 熟悉常用的DOS功能调用；
4. 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；
5. 加深对转移指令及一些常用的汇编指令的理解。

# 实验内容

**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，即串长最大为9），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、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。

（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。

**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。

**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。**

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”。

**任务1中，重要操作提示：**

|  |  |  |
| --- | --- | --- |
| 功能编号 | 序号 | 学生尝试的操作 |
| 功能一、二 | 1 | 如何让9号功能调用显示的信息放在自己希望的位置？ |
| 2 | 如果在9号功能调用时，带显示字符串的结尾没有“$”结束符会怎样？ |
| 3 | 如果在9号功能调用前，未对DS赋值，或者未对DX给予正确的值，结果会如何？ |
| 4 | 10号功能调用时，输入的字符数超过定义的数量时，它是如何处理的？ |
| 5 | 查找学生姓名时，如何提高查找速度？ |
| 6 | 循环或转移时，是否有多种指令的组合方式实现？ |
| 7 | 注意观察转移指令机器码的编码方法，观察对应标号的偏移地址与该编码之间的关系。 |
| 8 | 检查循环程序，是否可能出现死循环的情况。 |
| 功能三、四 | 1 | 把除以3.5，改成乘2除以7。简化一下计算公式。 |
| 2 | 单步观察计算过程，结果是否正确？ |
| 3 | 用TD直接修改BUF区的成绩值，使得程序对应的每个分支都能得到验证。 |
| 4 | 选取特殊的值，观察计算结果溢出的情况。 |
|  |  |

**6、选做题**

对于上述任务，增加一些难度（可以选做部分功能）

（1）对输入的姓名进行合法性检查，比如：是否存在大小写字母之外的字符，等等；若存在非法的输入情形，则提示之后重新输入。

（2）对BUF中预先定义的课程成绩表中的所有成绩项进行合法性检查，判断是否符合实际（如超过100分等），不符合则提示并退出程序。预先定义的学生姓名中，如果有重名如何处理？

（3）除了显示平均成绩的等级外，还将平均成绩同时显示。

（4）由于有些课程体系中，成绩的满分可能会达到300分，此时，成绩项的定义必须采用字类型，那么，如何改造程序呢？采用数据结构来定义BUF存储区是否会更好？

（5）在功能一种，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。

（6）功能二中，若未找到相应学生，在提示用户该学生不存在后，回到“功能一（1）”的位置，提示并重新输入姓名。

# 实验过程

## 任务

### 设计思想及存储单元分配

（一）总体设计

存储单元分配

BUF：字节变量，存储N个学生的姓名以及成绩，每个学生分配14个字节，其中10个字节存储名字，若名字小于10个字节，则置0,剩下4个字节存储成绩，分别是语文、数学、英语以及平均值。

PR：字节变量，存储字符串‘Please input name:(# = quit) $’，提示输入名字

in\_name：字节变量，作为输入缓冲区，缓冲区分配10个字节

PR2：字节变量，存储字符串‘Error！ Please input again: $’，提示输入名字错误

PR3：字节变量，存储字符串‘NO EXIST! $’，提示不存在该名字

PR4：字节变量，存储字符串‘Grades error! $’，提示成绩预先有错误

PR5：字节变量，存储字符串‘100 $’

POIN: 字变量，存储查找学生的成绩起始地址

寄存器分配

AX：将DATA段数据传送给DS

（二）功能一：输入姓名

寄存器分配

CX：存放输入名字的字符数

SI: 作为缓冲区in\_name中名字的偏移地址

（三）功能三：判断有无该名字

寄存器分配

CX：计数器，赋值N+1

SI：存储缓冲区in\_name中名字的偏移地址

DI：存储BUF的偏移地址

BX：作为公差，循环一次加一次14

AX：计数器，置为10

DH：存储输入名字的字符

DL：存储BUF中名字的字符

（四）功能四：计算每个学生平均成绩

寄存器分配

CX：计数器，赋值N+1

DI： 存储BUF的偏移地址

BX：作为公差，循环一次加一次14

AX：存储成绩加权和

DL：取每项成绩

（五）功能五：显示学生等级

BX：存储查找学生的成绩存放地址

AX：存放查找学生的加权平均成绩

（六）附加功能一：判断成绩是否合法

CX：计数器，赋值N+1

DI： 存储BUF的偏移地址

BP：作为公差，循环一次加一次14

AX：计数器，置为3

（七）附加功能二：显示学生平均成绩

AH、BH：存放平均成绩的个位数（平均成绩小于100）

BL：存放平均成绩的十位数（平均成绩小于100）

AL：存放平均成绩

### 流程图

图3.1.1是附加功能一的程序流程图。

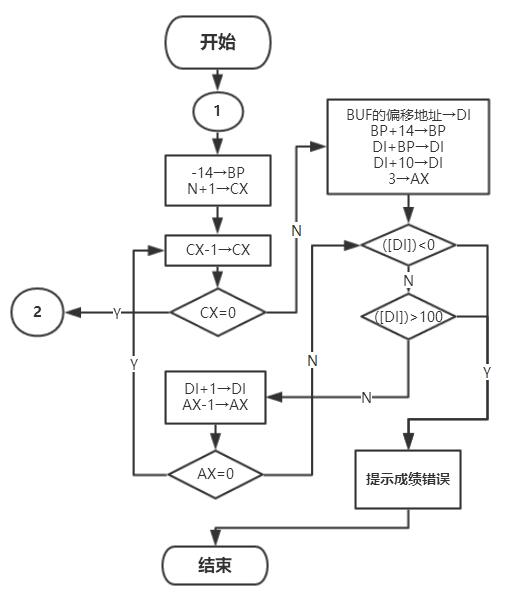


图3.1.1 附加功能一程序流程图

图3.1.2是功能一的程序流程图

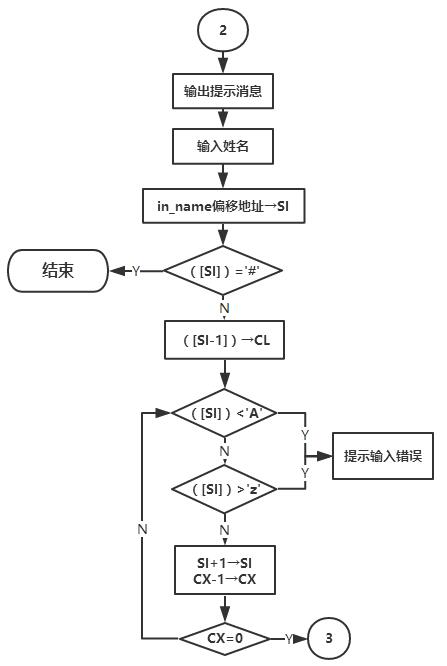


图3.1.2 功能一程序流程图

图3.1.3是功能二的程序流程图

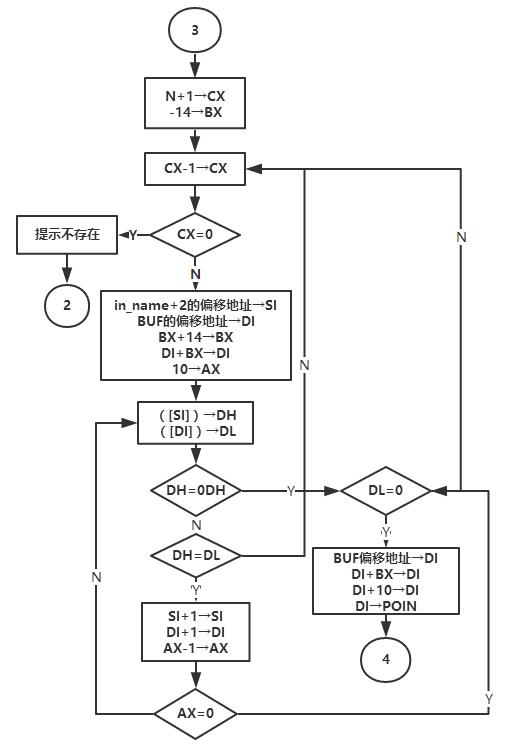


图3.1.3 功能二程序流程图

图3.1.4是功能三的程序流程图

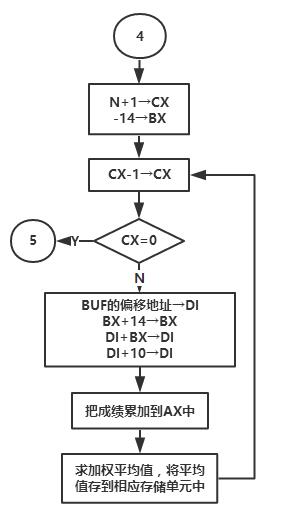


图3.1.4 功能三程序流程图

图3.1.5是功能四的程序流程图

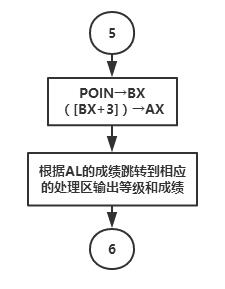


图3.1.5 功能四程序流程图

图3.1.6是附加功能二的程序流程图

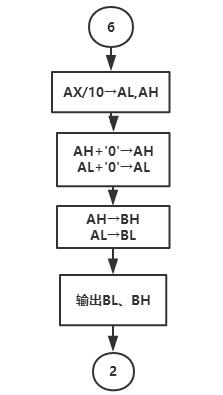


图3.1.6 附加功能二程序流程图

### 源程序

;-------------------------------------------------------------------------

DATA SEGMENT USE16

N EQU 4

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 'yongheng',0,0 ;最后一个必须是自己名字的拼音

DB 85, 85, 100, ?

PR DB 'Please input name:(# = quit) $' ;提醒输入，输入'#'退出

in\_name DB 10

DB ?

DB 10 DUP(0)

PR2 DB 'Error！ Please input again: $'

PR3 DB 'NO EXIST! $'

PR4 DB 'Grades error! $'

PR5 DB '100 $' ;若平均成绩为100，则通过字符串输出

POIN DW ?

DATA ENDS

;-------------------------------------------------------------------------

STACK SEGMENT STACK

DB 200 DUP(0)

STACK ENDS

;-------------------------------------------------------------------------

CODE SEGMENT USE16

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

START:

;-------------------------------------------------------------------------;附加功能1：判断成绩是否合法

MOV AX,DATA

MOV DS,AX

MOV BP,-14

MOV CX,N+1

WX2: DEC CX

JZ TSSR

LEA DI, BUF

ADD BP,14

ADD DI,BP

ADD DI,10

MOV AX,3

NX2: CMP BYTE PTR[DI], 0

JB ERR2

CMP BYTE PTR[DI], 100

JA ERR2

INC DI

DEC AX

JNZ NX2

JMP WX2

;-------------------------------------------------------------------------;功能一：输入姓名

TSSR:LEA DX, PR ;若成绩正常，则提示输入姓名

MOV AH, 9

INT 21H

SR: LEA DX, in\_name ;键盘输入名字

MOV AH,10

INT 21H

MOV DL, 0AH ;换行

MOV AH, 2

INT 21H

MOV SI, OFFSET in\_name+2 ;将SI指向缓冲区姓名起始地址

CMP BYTE PTR [SI], '#' ;附加功能：若输入‘#’，则退出程序

JZ EXIT

MOV CL,[SI-1]

LOPA: CMP BYTE PTR [SI],'A' ;附加功能：判断输入合法性

JB ERR

CMP BYTE PTR [SI],'z'

JA ERR

INC SI

DEC CX

JNZ LOPA

;------------------------------------------------------------------------;功能二：判断有无该姓名

MOV CX,N+1

MOV BX,-14

WX: MOV SI, OFFSET in\_name+2

MOV DI, OFFSET BUF

DEC CX

JZ NEH ;如果查找完所有仍未找到，则提示不存在

ADD BX, 14

ADD DI, BX

MOV AX, 10

NX: MOV DH, BYTE PTR [SI]

MOV DL, BYTE PTR [DI]

CMP DH, 0DH

JZ PD ;若移动到名字最后，则判断存储区相对应的值是否为'0'

CMP DH,DL

JNE WX ;若两个字母不相等，则跳出内循环

INC SI

INC DI

DEC AX

JNE NX ;若未比较完所有字母，则继续内循环

PD: CMP DL, 0

JNZ WX ;若缓存区相应字符不为0，则转到外循环

MOV DI, OFFSET BUF

ADD DI,BX

ADD DI,10

MOV POIN,DI ;将找到学生的成绩起始地址存到POIN变量中

;-------------------------------------------------------------------------;功能三：计算每个学生平均成绩

MOV CX,N+1

MOV BX,-14

JISUAN:MOV DI, OFFSET BUF

DEC CX

JZ PRINT ;若所有成绩均计算完毕，则跳到显示区

ADD BX,14

ADD DI,BX

ADD DI,10

MOV AX,0

MOV DX,0

MOV DL, BYTE PTR [DI]

MOV AX, DX

ADD AX,AX

MOV DL,BYTE PTR [DI+1]

ADD AX,DX

MOV DL,BYTE PTR [DI+2]

SAR DL,1

ADD AX,DX

SAL AX,1

MOV DL,7

IDIV DL

MOV BYTE PTR [DI+3],AL

JMP JISUAN

;--------------------------------------------------------------------------;功能四：显示学生等级及成绩

PRINT: MOV BX,POIN

MOV AX,[BX+3]

CMP AL,90

JGE LA

CMP AL, 80

JGE LB

CMP AL,70

JGE LC

CMP AL,60

JGE LD

CMP AL,60

JL LF

PRINT2: MOV DL,' ' ;附加功能2：显示学生成绩

MOV AH,2

INT 21H

CMP AL,100

JZ GOOD

MOV AH, 0

MOV DL,10

IDIV DL

ADD AL,'0'

ADD AH,'0'

MOV BL, AL

MOV BH, AH

MOV DL,BL

MOV AH,2

INT 21H

MOV DL,BH

MOV AH,2

INT 21H

MOV DL,0AH

MOV AH,2

INT 21H

JMP TSSR

GOOD: LEA DX,PR5

MOV AH,9

INT 21H

MOV DL,0AH

MOV AH,2

INT 21H

JMP TSSR ;若该同学相应操作处理完毕，重新提示输入名字

;---------------------------------------------------------------------------;结束程序

EXIT: MOV AH,4CH

INT 21H

ERR: LEA DX,PR2 ;提示输入名字错误

MOV AH,9

INT 21H

JMP SR

ERR2: LEA DX,PR4 ;显示成绩错误，并结束程序

MOV AH,9

INT 21H

JMP EXIT

NEH: LEA DX,PR3 ;提示不存在

MOV AH,9

INT 21H

MOV DL,0AH

MOV AH,2

INT 21H

JMP TSSR

LA: MOV DL,'A'

MOV AH,2

INT 21H

JMP PRINT2

LB: MOV DL,'B'

MOV AH,2

INT 21H

JMP PRINT2

LC: MOV DL,'C'

MOV AH,2

INT 21H

JMP PRINT2

LD: MOV DL,'D'

MOV AH,2

INT 21H

JMP PRINT2

LF: MOV DL,'F'

MOV AH,2

INT 21H

JMP PRINT2

CODE ENDS

END START

### 实验步骤

1. 准备上机环境，编辑、汇编、连接文件DEMO1。
2. 输入TD DEMO1，进入调试窗口
3. 输入‘lisi’，看数据段in\_name缓冲区变化
4. 在PRINT: MOV BX,POIN行设置断点，运行到此处，看数据段BUF区平均值是否正确，预期为5C（92）、54（84）、54（84）、57（87）。
5. 继续调试，看成绩等级是否正确。预期输出 B

（附加功能检验）

1. 继续调试，看是否输出平均成绩以及是否正确。预期输出 84
2. 退出程序，重新运行，分别输入 ‘lisik’、‘lisi66’、回车、以及‘#’，检查程序功能是否正确。预期分别输出 NO EXIST! 并提示输入名字

Error Please input again:

Error Please input again:

退出程序

1. 改变代码段成绩，设置一些分数为103、-5等数字，运行程序。预期输出 Grades error！并直接退出程序
2. 改变代码段成绩，将lisi分数设为100、100、100，运行程序。预期输出 A 100

### 实验记录与分析

1. 实验环境条件：i7-5500U 3.0GHz，8G内存；WINDOWS 10下DOSBox0.73；EDIT.EXE 2.0；MASM.EXE 6.0； LINK.EXE 5.2; TD.EXE 5.0。
2. 编译连接正常，如图3.1.7所示

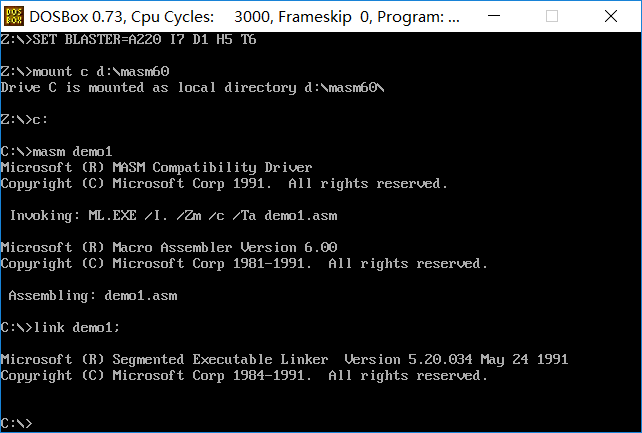


图3.1.7 正常编译连接

1. 输入td demo，进入调试器，输入lisi，观察缓冲区的数据，如图3.1.8、图3.1.9所示

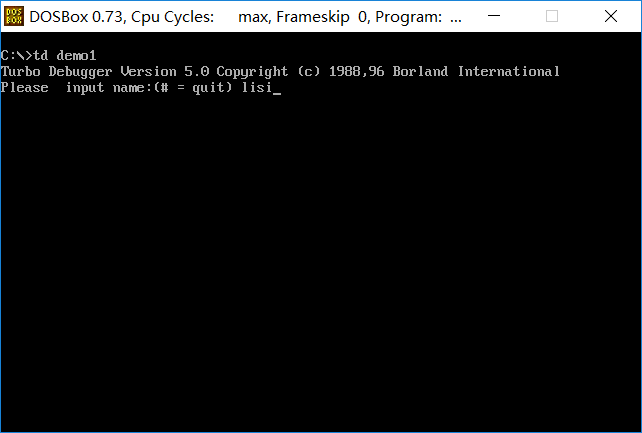


图3.1.8 输入‘lisi’

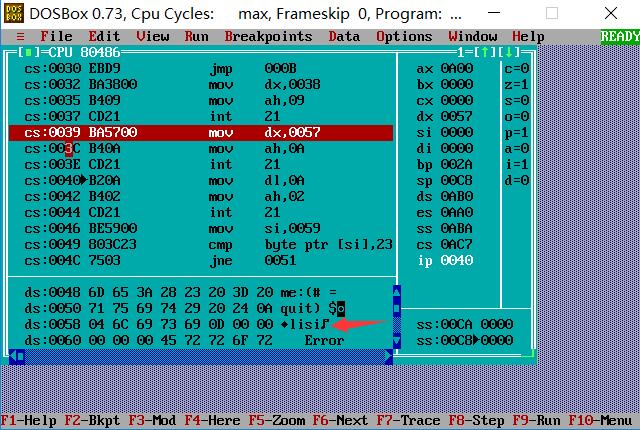


图3.1.9 缓冲区数据

1. 在PRINT: MOV BX,POIN行设置断点，运行到此处，看BUF平均值，如图3.1.10、图3.1.11所示

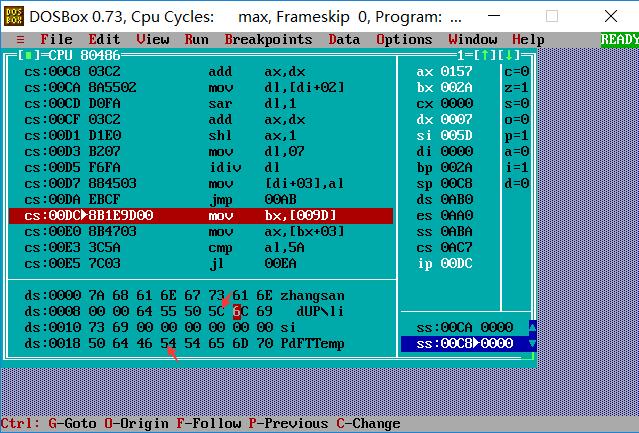


图3.1.10 BUF区数据

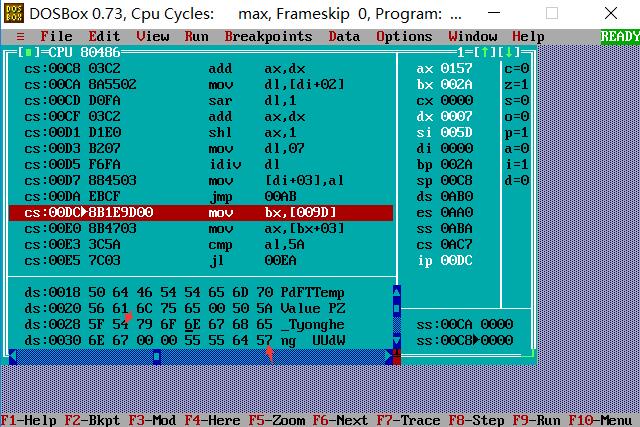


图3.1.11 BUF区数据

1. 继续调试，看成绩等级是否正确。预期输出 B，实际如图3.1.12所示

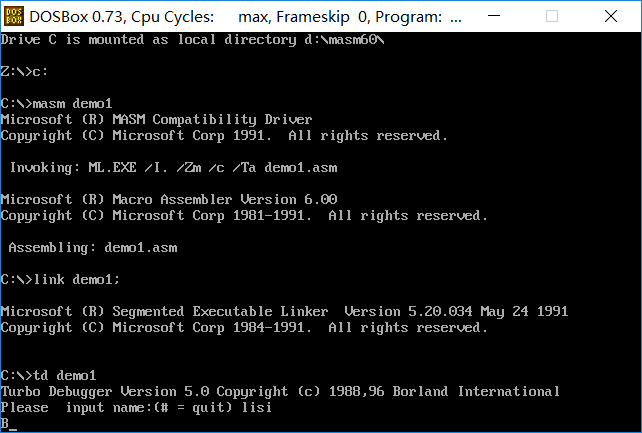


图3.1.12 成绩等级输出

1. 继续调试，看是否输出平均成绩以及是否正确。预期输出 84，实际如图3.1.13所示

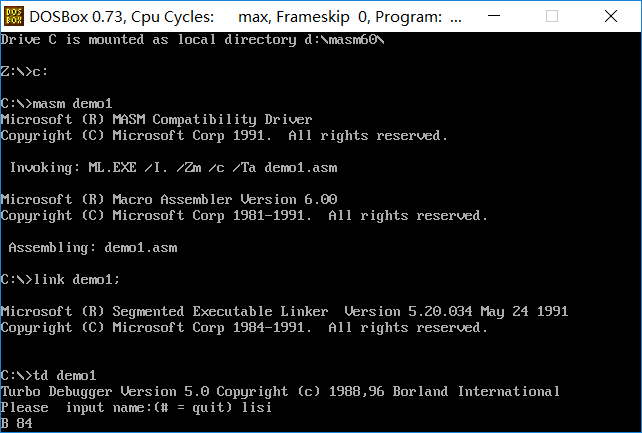


图3.1.12 成绩输出

1. 退出程序，重新运行，分别输入 ‘lisik’、‘lisi66’、回车、以及‘#’，检查程序功能是否正确。实际如图3.1.13所示

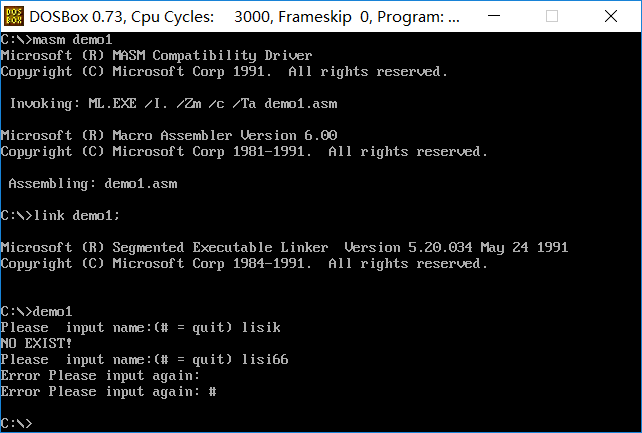


图3.1.13 输出字符正确性检查

1. 改变代码段成绩，设置一些分数为103、-5等数字，运行程序。预期输出 Grades error！并直接退出程序，实际如图3.1.14所示

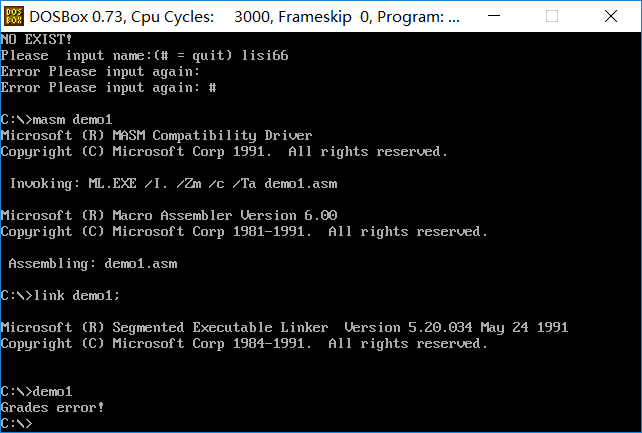


图3.1.14 输入成绩正确性检查

1. 改变代码段成绩，将zhangsan分数设为100、100、100，运行程序。预期输出 A 100，实际如图3.1.15所示

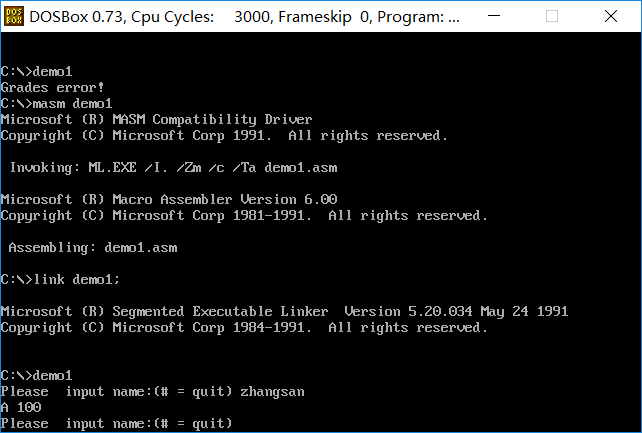


图3.1.15 100分检查

由以上各图可知，程序执行结果与预期一致。

# 总结与体会

本次实验相比第一次，难度上有了一部分增加，而且代码量也比较庞大，但是最终把所以任务完成后，并且代码通过后，内心是很骄傲的。

通过本次实验，我更加熟悉了TD的调用，通过TD的调用，我把许多初次编写的代码错误都找了出来。另外，本次实验中一大亮点是循环和分支，通过这次实验，我对转移指令有了基本的了解，并且可以通过这些质量完成基本循环和分支功能。初次之外，我还了解了DOS调用功能，可以进行简单的人机交互。

在本次实验中，我对汇编程序语言有了新的认识，没想到汇编程序语言也可以像C语言一样，实现这种功能，对汇编语言的兴趣也更加提高了。

总之，我相信自己会越来越熟练，越来越顺利！

# 参考文献

[1] 王元珍 曹忠升 韩宗芬.《80x86汇编语言程序设计》“第三章 宏汇编语言”，“第四章 程序设计的基本方法”.华中科技大学出版社