

**课 程 实 验 报 告**

**课程名称： 汇编语言程序设计实验**

**实验名称： 实验三 模块化程序设计**

**实验时间： 2017-4-5，14：00-17：30 实验地点： 南一楼803室111号实验台**

**指导教师： 张勇**

**专业班级：计算机科学与技术201307班**

**学 号： U201314969 姓 名： 王镇宇**

**同组学生： 无 报告日期： 2017年 4 月 6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](#_Toc479769545)

[2 实验内容 1](#_Toc479769546)

[3 实验过程 4](#_Toc479769547)

[3.1 任务1 4](#_Toc479769548)

[3.1.1 设计思想及存储单元分配 4](#_Toc479769549)

[3.1.2 流程图 5](#_Toc479769550)

[3.1.3 源程序 6](#_Toc479769551)

[3.1.4 实验步骤 13](#_Toc479769552)

[3.1.5 实验记录 14](#_Toc479769553)

[3.2 任务2 15](#_Toc479769554)

[3.2.1 设计思想及存储单元分配 15](#_Toc479769555)

[3.2.2 流程图 16](#_Toc479769556)

[3.2.3 源程序 17](#_Toc479769557)

[3.2.4 实验步骤 20](#_Toc479769558)

[3.2.5 实验记录 20](#_Toc479769559)

[4 体会 21](#_Toc479769560)

[参考文献 22](#_Toc479769561)

# 实验目的与要求

1. 掌握子程序设计的方法与技巧，熟悉子程序的参数传递方法和调用原理；
2. 掌握宏指令、模块化程序的设计方法;
3. 掌握较大规模程序的合作开发与调试方法；
4. 掌握汇编语言程序与C语言程序混合编程的方法；
5. 熟悉C编译器的基本优化方法;
6. 了解C语言编译器的命名方法，主、子程序之间参数传递的机制。

# 实验内容

任务1 宏与子程序设计

进一步修改与增强实验二的学生成绩查询程序的功能，具体要求如下：

1. 程序执行时首先显示一个功能菜单:选择1=录入学生姓名和各科考试成绩，2=计算平均分，3=计算排名，4=输出成绩单，5=程序退出。

**提示：**由于学生姓名和成绩是通过程序录入的，因此，定义学生成绩表缓冲区时，初始值都可以置零。为避免录入成绩的时间过程太长，假定学生人数在5人左右，具体人数自行决定。每个学生成绩表中增加一个**字**用来保存排名。平均分的结果只需保留整数部分。

1. 2人一组，一人负责包括菜单显示、程序退出在内的主程序，以及菜单中的功能1和2；另一人负责菜单中的功能3和4。各自汇编自己的模块，然后连接生成一个程序。

**注意，**在每个模块的开始，注明编写者的名字以及同组同学的名字。

1. 录入学生姓名和各科考试成绩时，首先显示录入的是第几个学生的信息，然后分别在提示之后输入姓名和各科成绩（可以借鉴书上十进制转二进制的子程序F10T2）。所有学生信息录入完毕后回到菜单显示的位置。姓名及考试成绩的存放、平均分的计算，按照实验二的要求。
2. 排名的基本要求是按照平均成绩从高到低计算名次，也可以考虑按照指定课程的成绩排名。相同平均分时排名相同，下一个相邻平均分的排名应该是排名在前的所有人数和的下一个数值。输出成绩单的基本要求是依次显示每个学生的姓名、平均成绩和排名，也可以考虑按照指定课程、指定进制的形式显示（可以借鉴书上二进制转十进制的子程序F2T10）。
3. 将9号和10号DOS系统功能调用定义成宏指令并调用。

**上述任务1中，值得思考的问题提示：**

1. 在TD中跟踪到子程序内部有几种方法？在TD中观察子程序调用和返回时堆栈的变化。
2. 注意观察FAR、NEAR类型子程序的RET指令的机器码有何不同？观察FAR类型子程序被调用时堆栈的变化情况。
3. 通过把一个模块拆成多个模块或反之，体会子程序和模块化程序设计的方法，体会模块调用关系图、子程序功能说明、输入/输出说明在程序设计中的作用。
4. 观察不同模块的可合并段合并后变量偏移地址的变化情况。观察不同段在内存里的放置次序。体会模块间段的定义及其对应的装配方法。
5. 在编程中使用不同的子程序参数传递方法来编写子程序。
6. 观察模块间的参数的传递方法，包括公共符号的定义和外部符号的引用，若符号名不一致或类型不一致会有什么现象发生？
7. 通过TD观察宏指令在执行程序中的替换和扩展，解释宏和子程序的调用有何不同。
8. 如何使菜单和成绩单显示得更漂亮一点？
9. EXTRN说明语句放在.386之前或者之后有什么区别？
10. EXTRN说明的变量的段与段寄存器的关联关系（ASSUME伪指令所表达的信息）是否能带入到本模块中？如果不能带入，是否可以通过加段前缀的方法来解决？
11. 如何利用宏功能使汇编语言的程序变得更加直观易读？

例：下面是一个利用宏功能直观化后的完整代码段程序，请写出对应的宏定义，并模仿该方式对自己编写的某段程序进行类似的改写。

StartProgram code，data，stack，start

Initial\_ds

GetStringTo BUF

DisplayStringFrom BUF

ExitToDOS

EndProgram code，start

任务2：在C语言程序中调用 汇编语言实现的函数

对于任务1的程序进行改造，主控程序、以及输入输出等功能用C语言实现，其他功能用独立的汇编语言子程序的方式实现； 在C语言程序中调用汇编语言子程序。

**要求与提示：**

1. 在不同的C语言开发环境中实现与汇编语言程序的混合编程，其操作方法有可能是不同的。请大家选择自己熟悉的C语言开发环境并查找相关的资料完成本实验。
2. 在实验报告中，比较详细的给出你的开发环境及其实现方法。
3. 观察C语言编译器中对各种符号的命名规则（指编译器内部可以识别的命名规则，比如，符号名前面是否加下划线“\_”，等），主、子程序之间参数传递的机制，通过堆栈传递参数后堆栈空间回收的方法。
4. 对混合编程形成的执行程序，用调试工具观察由C语言形成的程序代码与由汇编语言形成的程序代码之间的相互关系，包括段、偏移的值，汇编指令访问C的变量时是如何翻译的，等。
5. 请尝试在C语言源程序中不合理地嵌入汇编语言的指令语句，达到破坏C语言程序的正确性的目的。比如，在连续的几条C语言语句中间加入一条修改AX寄存器（或DS等其他寄存器）的汇编指令语句，而AX的内容在此处本不该被修改，这样就可观察到破坏C语言程序正确性的效果（该项实验表明：在C语言程序中，若不考虑上下语句翻译成怎样的机器码而随意嵌入汇编指令语句时，有可能存在出错的风险）。
6. 观察C编译器的优化策略对代码的影响。
7. 通过调试混合编程的程序，体会与纯粹汇编语言编写的程序的调试过程的差异。
8. 通过本次实验，希望大家明白：不同的编程语言是可以协同解决一个问题的，而且可以利用不同语言的特点来更好地解决问题；利用汇编语言的知识，能够更好地理解高级语言的内部处理原理与策略，为编写更好的C语言程序、用好C编译器提供支持。

# 实验过程

## 任务1

### 设计思想及存储单元分配

任务主要是实现主菜单和功能一、功能二。主菜单很简单，就是重复多次使用9号调用输出菜单栏，然后1号调用输入1、2、3、4、5，选择要实现的功能。

子程序1：寄存器分配：

CL:输入姓名字符串长度

CH:计数：每个学生3次输入成绩

SI:循环录入姓名和成绩时SI每+1实现BUF段14个字节的跳跃

BX:计数：+1在INPUT\_N和INPUT\_S和BUF段中实现后移

AX:存放姓名的字符和成绩的字符

DX:在将输入成绩字符转换为实际成绩时存放成绩

子程序2：寄存器分配：

BX:+1实现寻找成绩所在的字节

SI:每+1实现一个学生到另一个学生的跳跃

AX:存放3门课相加后的成绩

CX: CL:3门课成绩 CX:4个学生

### 流程图

子程序1：



子程序2：



### 源程序

;主程序

;功能：定义数据段，打印功能菜单

PUBLIC BUF

PUBLIC N

EXTRN SORT:NEAR

EXTRN PRINT:NEAR

.386

STACK SEGMENT USE16 STACK

DB 200 DUP(0)

STACK ENDS

DATA SEGMENT USE16 PUBLIC

BUF DB 10 DUP(0)

DB ?,?,?,?

DB 10 DUP(0)

DB ?,?,?,?

DB 10 DUP(0)

DB ?,?,?,?

DB 10 DUP(0)

DB ?,?,?,?

N DW 0

S DW 0

TIP1 DB 'Please make the choice: $'

TIP2 DB '1 input name$'

TIP3 DB '2 calculate score$'

TIP4 DB '3 sorting order$'

TIP5 DB '4 print$'

TIP6 DB '5 exit$'

INPUT\_N DB 12

DB ?

DB 10 DUP(0)

INPUT\_S DB 6

DB ?

DB 4 DUP(0)

DATA ENDS

CODE SEGMENT USE16 PUBLIC

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

START: MOV AX, DATA

MOV DS, AX

NINE MACRO A ;宏定义9号调用

LEA DX,A

MOV AH,9

INT 21H

ENDM

MENU: NINE TIP1 ;输出菜单

MOV DL,0AH

MOV AH,2

INT 21H

MOV DL,0DH

MOV AH,2

INT 21H

NINE TIP2

MOV DL,0AH

MOV AH,2

INT 21H

MOV DL,0DH

MOV AH,2

INT 21H

NINE TIP3

MOV DL,0AH

MOV AH,2

INT 21H

MOV DL,0DH

MOV AH,2

INT 21H

NINE TIP4

MOV DL,0AH

MOV AH,2

INT 21H

MOV DL,0DH

MOV AH,2

INT 21H

NINE TIP5

MOV DL,0AH

MOV AH,2

INT 21H

MOV DL,0DH

MOV AH,2

INT 21H

NINE TIP6

MOV DL,0AH

MOV AH,2

INT 21H

MOV DL,0DH

MOV AH,2

INT 21H

MOV AH,1

INT 21H

;MOV DL,0AH

;MOV AH,2

;INT 21H

;MOV DL,0DH

;MOV AH,2

;INT 21H

CMP AL,31H

JZ L1

CMP AL,32H

JZ L2

CMP AL,33H

JZ L3

CMP AL,34H

JZ L4

CMP AL,35H

JZ OVER

JMP MENU

OVER:

MOV AH, 4CH

INT 21H

L1: CALL LOPA1

JMP MENU

L2: CALL LOPA2

JMP MENU

L3: CALL SORT

JMP MENU

L4: CALL PRINT

JMP MENU

;子程序1

;功能：输入4个学生的姓名和成绩

;寄存器分配： CL:输入姓名字符串长度

; CH:计数：每个学生3次输入成绩

; SI:循环录入姓名和成绩时SI每+1实现BUF段14个字节的跳跃

; BX:计数：+1在INPUT\_N和INPUT\_S和BUF段中实现后移

; AX:存放姓名的字符和成绩的字符

; DX:在将输入成绩字符转换为实际成绩时存放成绩

LOPA1 PROC NEAR

TEN MACRO B ;宏定义10号调用

LEA DX,B

MOV AH,10

INT 21H

ENDM

MOV SI,0

LOOP1A:

MOV BX,0

TEN INPUT\_N

MOV DL,0AH

MOV AH,2

INT 21H

MOV DL,0DH

MOV AH,2

INT 21H

MOV CL,INPUT\_N[1]

NEXT\_N:

PUSH AX

MOV AX,N

IMUL AX,14

MOV SI,AX

POP AX

MOV AL,INPUT\_N[2+BX]

MOV [BX+SI],AL

INC BX

DEC CL ;把每个字符放入BUF段

JNZ NEXT\_N

MOV CH,0 ;三门课的成绩

SCORE:

MOV BX,0

TEN INPUT\_S

MOV DL,0AH

MOV AH,2

INT 21H

MOV DL,0DH

MOV AH,2

INT 21H

MOV DL,100

MOV DH,0

MOV CL,INPUT\_S[1] ;输入成绩长度，输入3位数进行转换

NEXT\_S: ;把输入的字符串转为10进制成绩

PUSH AX

MOV AX,N

IMUL AX,14

MOV SI,AX

POP AX

MOV AL,INPUT\_S[2+BX]

SUB AL,30

IMUL DL ;(AL)\*DL->AX

ADD DH,AL

INC BX

PUSH AX

PUSH BX

XOR AX, AX

MOV AL,DL ;(DL)->AL

MOV BL,10 ;

IDIV BL ;(AX)/10->AL

MOV DL,AL ;(AL)->DL

POP BX

POP AX

DEC CL

JNZ NEXT\_S ;成绩的3个字符进行转换

ADD BL,CH

MOV [BX+SI+7],DH ;把成绩放入BUF[10]

INC CH

CMP CH,3

JNZ SCORE ;循环3次输入成绩

INC N

CMP N,4 ;循环输入4个人的信息

JNZ LOOP1A

RET

LOPA1 ENDP

;子程序2

;功能：计算4个学生的平均成绩

;寄存器分配： BX:+1实现寻找成绩所在的字节

; SI:每+1实现一个学生到另一个学生的跳跃

; AX:存放3门课相加后的成绩

; CX: CL:3门课成绩 CX:4个学生

LOPA2 PROC NEAR

MOV SI,0

MOV CH,4

NEXT2:

MOV AX,0

MOV BX,0

MOV CL,3

PUSH AX

MOV AX,S

IMUL AX,14

MOV SI,AX

POP AX

NEXT1:

MOV DL,BUF[BX+SI+10]

ADD AL,DL

INC BX

DEC CL

JNZ NEXT1 ;三门课成绩相加

MOV CL,3

IDIV CL

MOV BUF[SI+13],AL

INC S

DEC CH

JNZ NEXT2

RET

LOPA2 ENDP

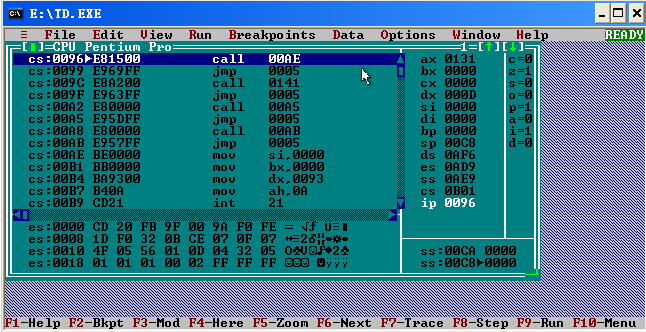
CODE ENDS

END START

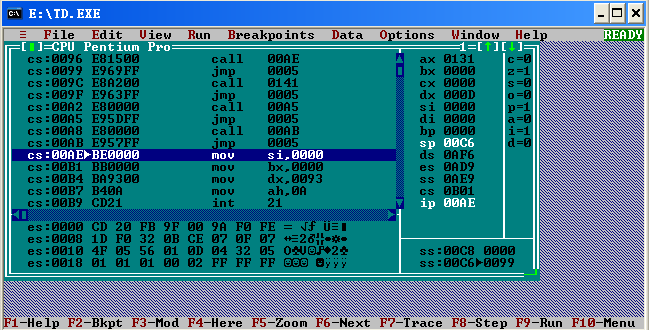
### 实验步骤

1. 在TD中跟踪到子程序内部我使用的方法是当程序运行至子程序入口时，直接按f7进入到子程序。如要进入子程序1：

当程序运行至如图入口时，按下f7



如图，按下f7后进入子程序1执行语句MOV SI,0



2. 公共符号的定义和外部符号的引用，若符号名不一致或类型不一致，会发生报错，当我程序中的公共符号和同组同学的外部符号符号名不一致或类型不一致时，会发生报错

3. 如图是第一次使用宏定义的9号调用在TD中的语句，可见宏指令的展开只是按宏定义的语句替换和展开的。说明宏调用是在汇编时展开的，而子程序是在程序执行时由CALL指令调用的，换言之，在汇编后生成的obj文件中，子程序的定义依然存在，而宏定义却不复存在。



4. 为了程序的美观，在输出菜单时，采取多次输出的办法，另外在每次10号调用输入后都输出一个回车换行符，这样可以使程序尽量美观。

### 实验记录

1. 实验环境条件：P3 1GHz，256M内存；WINDOWS 2000命令行窗口；EDIT.EXE 2.0；MASM.EXE 6.0； LINK.EXE 5.2; TD.EXE 5.0。

2、一开始10号功能调用后面忘了加上回车换行，导致输入的总是会被不断覆盖，也就不能正常显示，在后面加上

MOV DL,0AH

MOV AH,2

INT 21H

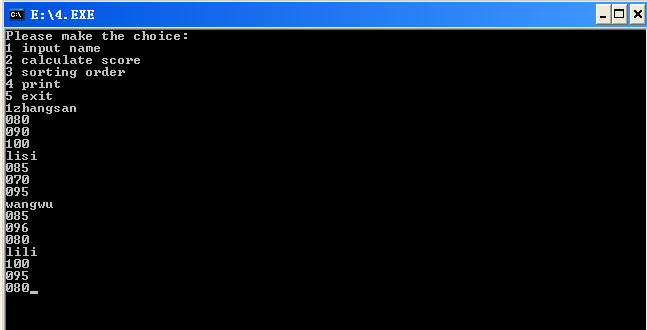
MOV DL,0DH

MOV AH,2

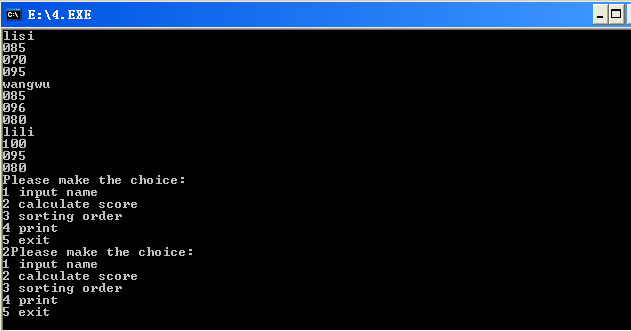
INT 21H

就能够正常显示了

3、寻址方式的编写有一些问题。最后运行的结果如下（由于算法：输入3个字符的ascll码减去30再乘相应的位数，输入成绩都要输3位）



再输入2实现计算平均成绩的功能，并回到主程序显示MENU菜单提示下一步操作



在TD数据段中观察实验结果与输入的姓名成绩、计算的平均成绩相符合

## 任务2

### 设计思想及存储单元分配

任务：在C语言程序中调用 汇编语言实现的函数

对于实验二的程序进行改造，主控程序、以及输入输出等功能用C语言实现，学生姓名搜索和成绩计算用独立的汇编语言子程序的方式实现； 在C语言程序中调用汇编语言子程序。

整体结构功能通过一个do-while循环来实现，而循环的条件根据使用者输入“Y”或“N”来判断。在整个循环的过程中，C语言通过调用汇编语言子程序，来达到学生姓名搜索和成绩计算的功能。而这两个功能则独立利用汇编语言子程序的方式来实现。

1.结构设计：

struct Stu{

char name[12]; //储存姓名

short score[4]; //储存成绩

};

2.存储单元分配

Name:分配12个字节来存储名字

Score:分配1个短整型数组，数组包含4个元素，分别为语文、数学、英语和平均成绩。

### 流程图



### 源程序

汇编语言子程序

.386

.model flat, c

.code

public dif

dif proc par1:dword,par2:dword

mov ecx, 1000

mov edi, par1;

LOAPI:

mov esi, par2

mov edx, 12

LOAPJ:

mov al, byte ptr[esi]

mov bl, byte ptr[edi]

cmp al, bl

jnz NEXT

cmp al, 0

jz FIGURE

dec edx

inc edi

inc esi

jmp LOAPJ

NEXT:

dec ecx

jz REINPUT

add edx, 8

add edi, edx

jmp LOAPI

FIGURE:

add edi, edx

mov al, byte ptr[edi]

mov ah, 0

add ax, ax

add ax, ax

mov dl, byte ptr[edi + 2]

mov dh, 0

add ax, dx

add ax, dx

mov dl, byte ptr[edi + 4]

mov dh, 0

add ax, dx

mov bl, 7

div bl

mov ah, 0

mov word ptr[edi + 6], ax

REINPUT:

ret

dif endp

end

C语言主程序

#include<stdlib.h>

#include<stdio.h>

#include<string.h>

struct Stu{

char name[12]; //储存姓名

short score[4]; //储存成绩

};

extern void dif(struct stuinfo \*, char \*);

int main() {

struct Stu info[3];

char name[12]; //输入的姓名

int i, j; //计数器

char op;

//初始化三个学生信息用于测试

strcpy(info[0].name,"zhangsan\_a");

info[0].score[0] = 86;

info[0].score[1] = 82;

info[0].score[2] = 88;

info[0].score[3] = 0;

strcpy(info[1].name, "zhangsan\_b");

info[1].score[0] = 48;

info[1].score[1] = 79;

info[1].score[2] = 46;

info[1].score[3] = 0;

strcpy(info[2].name, "zhangsan\_c");

info[2].score[0] = 96;

info[2].score[1] = 92;

info[2].score[2] = 98;

info[2].score[3] = 0;

do{

printf("Inlut the name of the student:");

gets\_s(name, 12);

dif(info, name);

for (i = 0; i < 3; i++){

if (!strcmp(info[i].name, name)){

printf("%s\t", info[i].name);

for (j = 0; j < 4; j++){

printf("%d\t", info[i].score[j]);

}

printf("\n");

}

}

printf("Input Y to exit,N continue...");

op = getchar(); getchar();

} while (op == 'Y' || op == 'y');

for (i = 0; i < 3; i++) {

printf("%s\t", info[i].name);

for (j = 0; j < 4; j++){

printf("%d\t", info[i].score[j]);

}

printf("\n");

}

getchar();

return 0;

}

### 实验步骤

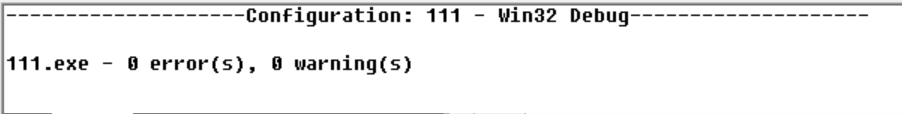
1. 使用Visual C++录入源程序，编译，链接，运行。

2. 观察提示信息，若出错，则用编辑程序修改错误，修改后再重新编译

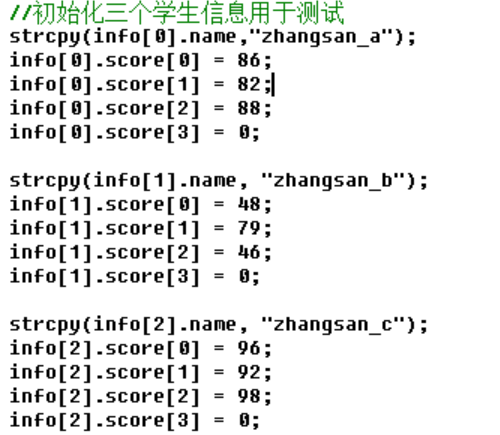
3. 执行该程序。观察程序运行结果。

### 实验记录

1. 实验环境条件：Visual C++；Visual C++窗口；
2. 编译源程序时没有发生异常。
3. 连接过程没有发生异常。



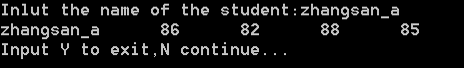
1. 输入提前录入的三个学生姓名。



1. 运行目标程序



1. 根据程序提示输入查找的学生姓名

  
 输入提前录入的学生姓名，输出该学生的各项成绩，并提示是否继续。

# 体会

通过这次实验我掌握子程序设计的方法与技巧，熟悉子程序的参数传递方法和调用原理；

掌握了宏指令、模块化程序的设计方法，掌握了较大规模程序的合作开发与调试方法，掌握汇编语言程序与C语言程序混合编程的方法，熟悉C编译器的基本优化方法，了解C语言编译器的命名方法，主、子程序之间参数传递的机制。

在实验过程中不断发现错误，解决问题的过程中进一步锻炼了自己使用TD调试错误的能力，也加强了汇编语言编写的技巧。更进一步理解了汇编作为一门底层语言与C语言这种高级编程语言的之间的联系与相同通之处，对之后的编程及学习有很重要的意义。

# 参考文献

[1] 王元珍等.80x86汇编语言程序设计.版本(第1版)

[2] 王晓虹等.汇编语言程序设计教程.版本(第1版)