

**课 程 实 验 报 告**

**课程名称： 汇编语言程序设计实验**

**实验名称： 实验四 模块化程序设计**

**实验时间： 2017-10-31/11.7，18：30-21：50 实验地点： 南一楼804室6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： 梁安柱 报告日期： 2017年 11 月 9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](#_Toc500540646)

[2 实验内容 1](#_Toc500540647)

[3 实验过程 2](#_Toc500540648)

[3.1 任务一 2](#_Toc500540649)

[3.1.1 设计思想及存储单元分配 2](#_Toc500540650)

[3.1.2 流程图 5](#_Toc500540651)

[3.1.3 源程序 9](#_Toc500540652)

[3.1.4 实验步骤 15](#_Toc500540653)

[3.1.5 实验记录与分析 15](#_Toc500540654)

[3.2 任务二 21](#_Toc500540655)

[3.2.1 设计思想及存储单元分配 21](#_Toc500540656)

[3.2.2 源程序 21](#_Toc500540657)

[3.2.3 实验步骤 25](#_Toc500540658)

[3.2.4 实验数据记录与分析 25](#_Toc500540659)

[3.3 选做题 28](#_Toc500540660)

[3.3.1 实验数据记录及分析 28](#_Toc500540661)

[4 总结与体会 28](#_Toc500540662)

[参考文献 30](#_Toc500540663)

# 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三的学生成绩查询程序的功能。程序执行时首先显示一个功能菜单: 然后，根据用户选择的菜单项，完成相应的功能。

1=录入学生姓名和各科考试成绩

2=计算平均分

3=按平均分排序从高到低排序，排序结果仍存放在原缓冲区中

4=输出成绩单

5=程序退出。

要求：

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。
2. 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。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，按照实验三的要求。
3. 按平均分排序从高到低排序，排序结果仍存放在原缓冲区中。
4. 输出成绩单时，依次显示每个学生的姓名、各科成绩，平均成绩。可在排序前调用，也可在排序后调用。
5. 将9号和10号DOS系统功能调用定义成宏指令并调用。使用子程序完成各模块的功能。实现串、数转换时，也应使用子程序来完成。

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. 在编程中使用不同的子程序参数传递方法来编写子程序。
  5. 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？
  6. 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。

**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. 观察主、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。
3. 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。

**选做题**

1. 如何使菜单和成绩单显示得更漂亮一点？
2. EXTRN说明语句放在.386之前或者之后有什么区别？
3. 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？
4. 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）。
5. 观察C编译器的优化策略对代码的影响。

# 实验过程

## 任务一

### 设计思想及存储单元分配

1. 主控程序

设计思想：利用9号调用输出菜单，利用1号调用存储输入的选项，然后根据输入的选项跳转到相应的模块，在各个模块里调用相应的子程序，实现各个功能。

存储单元分配：

XINXI：字节与字混合变量，共五组，每组中有14个字节变量，存储学生的姓名和成绩，还有一个字变量存储学生排名。

JISHU：（公共变量）字变量，存储输入学生的数量。

MENU：字节变量，存储字符串，作为菜单名，利用9号调用输出。

TISHI：字节变量，存储字符串，如果输入选项不是1-5，就提示输入错误，利用9号调用输出。

TISHI2：字节变量，存储字符串，如果输入成绩时有错误，提示成绩错误。

TISHI3：字节变量，存储字符串，调用输入子程序后，如果成功输入，提示输入成功。

TISHI4：字节变量，存储字符串，调用计算子程序后，如果成功计算，提示计算成功。

TISHI5：字节变量，存储字符串，调用排序子程序后，如果成功排序，提示排序成功。

TISHI6：字节变量，存储字符串，调用输出子程序后，如果成功输出，提示输出成功。

TISHI7：字节变量，存储字符串，如果在刚开始没有输入学生名字和成绩，就调用其他功能，则提示还没输入信息。

CHJI：（公共变量）字节变量，存储数字串，在数转换串子程序中应用。

寄存器分配：

AL：存储输入选项

1. 功能一：输入学生姓名和成绩

设计思想：利用堆栈方式传递信息存储单元地址以及数字串存储单元地址，在子程序中用一个寄存器指向信息存储单元，利用1号调用，比较输入的字符是不是结束标志‘#’，如果是就退子程序，完成输入；如果不是就比较输入的字符是不是回车，如果是，代表名字已经输完，跳转到输入成绩处，否则将输入的字符存到信息存储单元，寄存器后移，继续1号调用。输入成绩时，利用1号调用，将数字串先存储到相应单元中，如果遇到空格，代表一个成绩已经输入完成，则调用串转换数子程序，以二进制数存储到信息存储单元中，否则，继续存储数字串。存储三门成绩后，跳转到下一个学生姓名存储处。

存储单元分配

CHJI2：字节变量，存储数字串，在串转换数子程序中应用。

寄存器分配

SI：指向信息存储单元

DI：指向信息存储单元

BX：指向数字串存储单元

AL：存储输入字符（名字和成绩）

CX：计数，记录输入的成绩是几位数

BP：指向栈顶，传递堆栈区的变量

串转换数（F10T2）子程序中寄存器分配

SI：参数，指向当前输入学生成绩的地址

AX：存储对应的数

DI：指向字节变量CHJI2

DL：取出数字串中的数字字符

1. 功能二：计算所有学生的平均成绩

设计思想：首先判断是否输入学生信息，如果已输入，则继续执行，否则提示输入。用一个寄存器法指向信息存储单元，用另一个寄存器存储学生人数，调用子程序。计算成绩存储地址，用字节寄存器逐个取出成绩，乘以权重，存到另一个寄存器中，最后将计算后的平均值存到相应单元后，开始计算下一个同学，直到所有同学成绩计算完，退出程序。

寄存器分配

SI：参数，指向信息存储单元

CX：参数，存储学生人数

AX：存储成绩加权和

DL：取每项成绩

1. 功能三：按平均成绩从高到低排序

设计思想：首先判断是否输入学生信息，如果已输入，则继续执行，否则提示输入。用一个寄存器指向信息存储单元，调用子程序（远）。在子程序中，将参数赋给另两个寄存器，选择一个寄存器，并在此基础上指向下一个学生信息存储地址。分别比较两个相邻学生的平均值，如果前一个小于后一个，则调用交换子程序，交换信息，否则不交换，两个寄存器分别移动指向下一个学生，再进行比较，直到比到最后两个学生，此时成绩最小的将在最后一个位置，第一轮比较完毕。计数器减一，再进行下一轮比较，直到计数器为0，停止比较，此时所有成绩按从高到低依次排好。

寄存器分配

BX：参数，指向信息存储单元

SI：指向前一个学生的信息

DI：指向后一个学生的信息

AL：取出前一个学生的平均成绩

AH：取出后一个学生的平均成绩

CX：存储学生人数

DX：计数，控制比较次数

交换子程序中寄存器分配

SI：参数，指向需要交换信息的前一个学生

BP：存储参数值

AL：取出前一个学生的信息

AH：取出后一个学生的信息

CX：计数，判断是否交换完毕

1. 功能四：输出学生名字和成绩

设计思想：首先判断是否输入学生信息，如果已输入，则继续执行，否则提示输入。用一个寄存器指向信息存储单元，调用子程序（远）。在子程序中，将参数赋给另一个寄存器。比较输出姓名字符是否为‘0’，若不是，则利用2号调用输出该字符，否则跳转到输出成绩功能处，输入一个后，寄存器指向下一个字符，继续比较。输出成绩时，利用数转换串子程序，将2进制数转换成对应的10进制数字串，利用9号调用，输出成绩。若成绩输出完毕，则寄存器指向下一个学生的存储地址，进行下一轮输出。所有学生输出后，退出子程序。

寄存器分配

SI：参数，指向信息存储单元

DI：储存参数值。

BX：计数器，判断学生信息是否全部输出。

CX：计数器，判断成绩是否全部输出。

数转换串（F2T10）子程序中寄存器分配

DI：指向数字串存储单元地址。

DX：赋值为10，作为商。

AL：取出当前学生的平均值

### 流程图

图3.1.1是主程序流程图

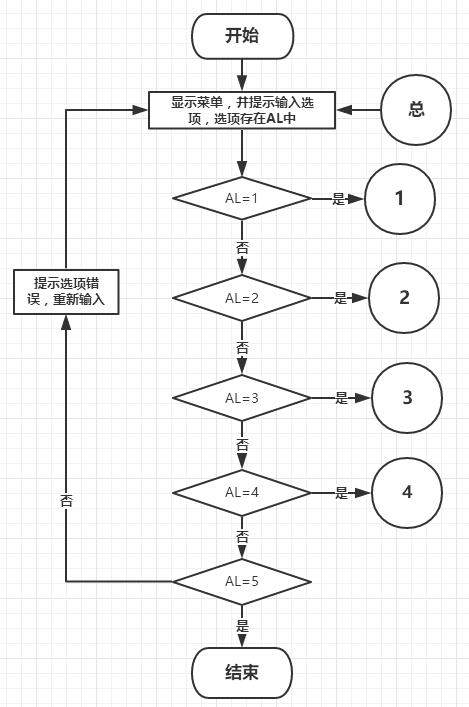


图3.1.1 主程序流程图

图3.1.2是功能一的流程图

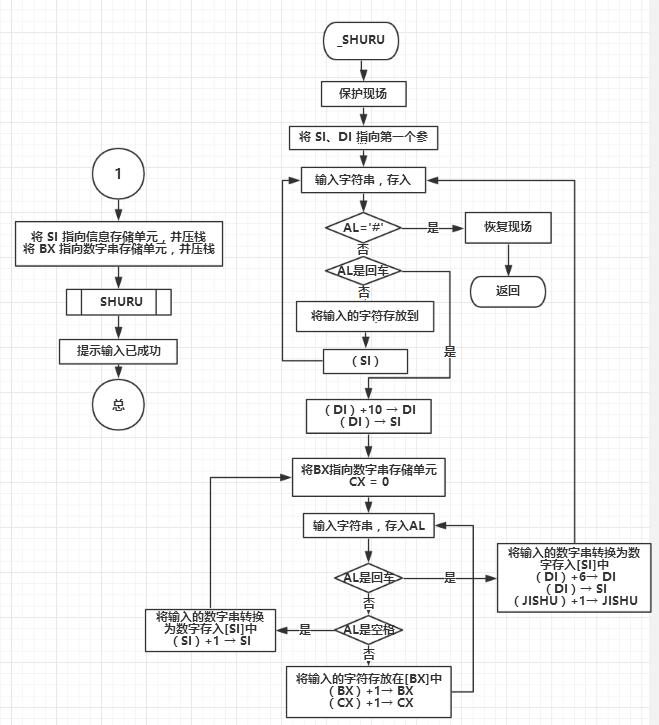


图3.1.2 功能一流程图

图3.1.3是功能二的流程图

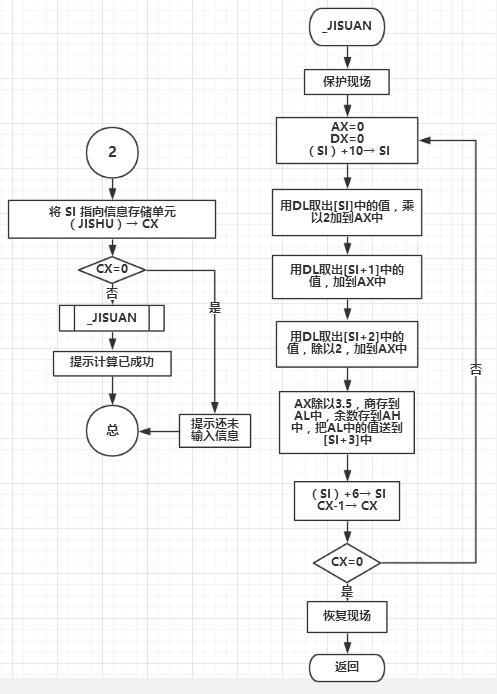


图3.1.3 功能二流程图

图3.1.4是功能三的流程图

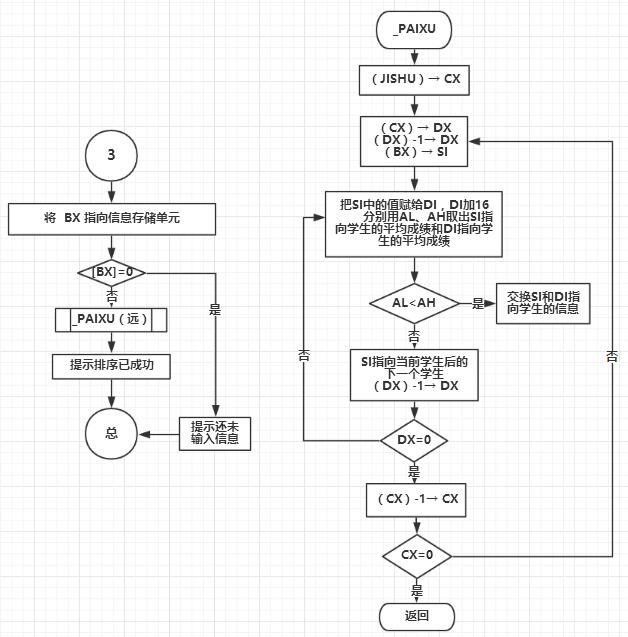


图3.1.4 功能三流程图

图3.1.5是功能四的流程图

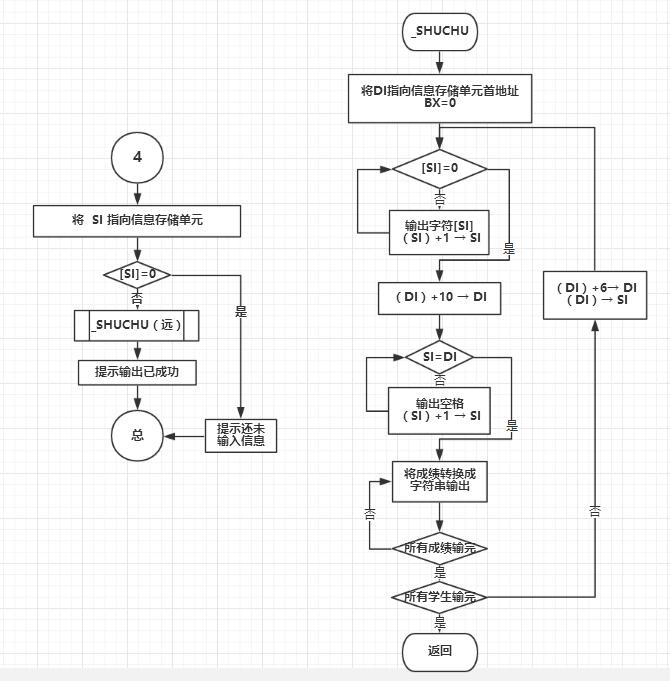


图3.1.5 功能四流程图

### 源程序

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*编写：黄永恒

IO MACRO X, Y

LEA DX, X

MOV AH, Y

INT 21H

ENDM

IC MACRO

MOV AH,1

INT 21H

ENDM

OC MACRO N

MOV DL,N

MOV AH,2

INT 21H

ENDM

HUICHE MACRO

MOV AH,2

MOV DL,0AH

INT 21H

MOV DL,0DH

INT 21H

ENDM

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

EXTRN \_PAIXU: FAR, \_SHUCHU: FAR

PUBLIC JISHU, CHJI

.386

DATA SEGMENT USE16 PUBLIC 'DATA'

XINXI DB 14 DUP (0)

DW 1

DB 14 DUP (0)

DW 2

DB 14 DUP (0)

DW 3

DB 14 DUP (0)

DW 4

DB 14 DUP (0)

DW 5

JISHU DW 0 ;记录输入学生数

CHJI DB ' ', ' ', ' ', '$'

CHJI2 DB 0, 0, 0

COUNT DB 0

MENU DB 'Please choose what you want to do:(1-5)',0AH,0DH,

' 1. Enter name and grades (# = stop)',0AH,0DH,

' 2. Calculate average',0AH,0DH,

' 3. Rank from high to low',0AH,0DH,

' 4. Print report card',0AH,0DH,

' 5. EXIT ',0AH,0DH,'$'

TISHI DB 0AH,0DH,'ERROR! Please choose again: ',0AH,0DH,'$'

TISHI2 DB 0AH,0DH,'GRADES ERROR! Please input again: ',0AH,0DH,'$'

TISHI3 DB 'You have entered successfully! Please choose others:',0AH,0DH,'$'

TISHI4 DB 'You have calculated grades! Please choose others:',0AH,0DH,'$'

TISHI5 DB 'The grades have been ranked! Please choose others:',0AH,0DH,'$'

TISHI6 DB 'The information has been printed! Please choose others:',0AH,0DH,'$'

TISHI7 DB 'You have not entered names and grades!',0AH,0DH,'$'

SHANG DB 7

DATA ENDS

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

STACK SEGMENT USE16 STACK

DB 200 DUP(0)

STACK ENDS

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

CODE SEGMENT USE16 PUBLIC 'CODE'

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

START: MOV AX, DATA

MOV DS, AX

IO MENU,9

XUANZE: IC

CMP AL,'1'

JB ERROR

JZ SHURU

CMP AL, '2'

JZ JISUAN

CMP AL, '3'

JZ PAIXU

CMP AL, '4'

JZ SHUCHU

CMP AL, '5'

JZ TUICHU

JA ERROR

;--------------------------------------------------------------------------功能一：输入学生姓名和成绩

SHURU: HUICHE

LEA SI, XINXI

LEA BX, CHJI2

PUSH SI

PUSH BX

CALL \_SHURU ;子程序：以堆栈法传递参数

IO TISHI3,9

JMP XUANZE

;--------------------------------------------------------------------------功能二：计算平均值

JISUAN: HUICHE

LEA SI, XINXI

MOV CX, JISHU

CMP CX,0

JNZ T

IO TISHI7,9

JMP \_T

T: CALL \_JISUAN ;子程序：以寄存器法传递参数

IO TISHI4,9

\_T: JMP XUANZE

;---------------------------------------------------------------------------功能三：排序（由大到小）

PAIXU: HUICHE

MOV BX, OFFSET XINXI

CMP BYTE PTR[BX],0

JNZ R

IO TISHI7,9

JMP \_R

R: CALL \_PAIXU

IO TISHI5,9

\_R: JMP XUANZE

;---------------------------------------------------------------------------功能四：输出

SHUCHU: HUICHE

LEA SI, XINXI

CMP BYTE PTR[SI],0

JNZ S

IO TISHI7,9

JMP \_S

S: CALL \_SHUCHU

IO TISHI6,9

\_S: JMP XUANZE

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

TUICHU: MOV AH,4CH

INT 21H

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

ERROR: IO TISHI, 9

JMP XUANZE

ERROR2: IO TISHI2, 9

JMP NEXT3

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\_SHURU PROC

PUSH SI

PUSH DI

PUSH AX

PUSH BX

PUSH CX

PUSH BP

MOV BP, SP

MOV SI, 16[BP]

MOV DI, SI

JIXU: IC

CMP AL, '#'

JZ JIESHU

CMP AL, 0DH

JZ NEXT

MOV [SI], AL

INC SI

JMP JIXU

NEXT: ADD DI, 10

MOV SI, DI

NEXT2: MOV BX, 14[BP]

XOR CX, CX

NEXT3: IC

CMP AL, 0DH

JZ M

CMP AL, ' '

JZ P

CMP AL, '0'

JB ERROR2

CMP AL, '9'

JA ERROR2

MOV [BX], AL

INC BX

INC CX

JMP NEXT3

P: CALL F10T2

INC SI

JMP NEXT2

M: CALL F10T2

ADD DI, 6

MOV SI, DI

INC JISHU

JMP JIXU

JIESHU: IC

POP BP

POP CX

POP BX

POP AX

POP DI

POP SI

RET 4

\_SHURU ENDP

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\_JISUAN PROC

PUSH AX

PUSH DX

\_NEXT: XOR AX, AX

XOR DX, DX

ADD SI, 10

MOV DL, [SI]

SAL DX, 2

ADD AX, DX

XOR DX, DX

MOV DL, 1[SI]

SAL DX, 1

ADD AX, DX

XOR DX, DX

MOV DL, 2[SI]

ADD AX, DX

IDIV SHANG

MOV 3[SI], AL

ADD SI, 6

DEC CX

JNZ \_NEXT

POP DX

POP AX

RET

\_JISUAN ENDP

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

F10T2 PROC

PUSH DX

PUSH AX

PUSH DI

XOR AX, AX

MOV DI, 14[BP]

N: MOV DL, [DI]

SUB DL, 30H

IMUL AX,10

ADD AL, DL

INC DI

DEC CX

JNZ N

MOV [SI], AL

POP DI

POP AX

POP DX

RET

F10T2 ENDP

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CODE ENDS

END START

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*编写：梁安柱

IO MACRO X, Y

LEA DX, X

MOV AH, Y

INT 21H

ENDM

IC MACRO

MOV AH,1

INT 21H

ENDM

OC MACRO N

MOV DL, N

MOV AH,2

INT 21H

ENDM

HUICHE MACRO

MOV AH,2

MOV DL,0AH

INT 21H

MOV DL,0DH

INT 21H

ENDM

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

EXTRN JISHU: WORD, CHJI: BYTE

PUBLIC \_PAIXU, \_SHUCHU

.386

STACK SEGMENT USE16

DB 200 DUP (0)

STACK ENDS

CODE1 SEGMENT USE16 PUBLIC 'CODE'

ASSUME CS: CODE1, SS: STACK

\_PAIXU PROC FAR ;排序子程序

MOV CX, JISHU

BIG: MOV DX, CX

DEC DX

MOV SI, BX

\_SMALL: MOV DI,SI

ADD DI,16

MOV AL,13[SI]

MOV AH,13[DI]

CMP AL,AH

JNB CONTU

CALL JIAOHUAN

CONTU: ADD SI,16

DEC DX

CMP DX,0

JG \_SMALL ;如果ax！=0，则比较下一对数

DEC CX

JNZ BIG ;如果cx!=0，则进行下一轮冒泡

RET

\_PAIXU ENDP

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\_SHUCHU PROC FAR

MOV DI, SI

XOR BX, BX

\_NET: CMP BYTE PTR [SI],0

JZ \_JIXU

OC [SI]

INC SI

JMP \_NET

\_JIXU: ADD DI, 10

\_JIXU1: CMP SI,DI

JZ P

OC ' '

INC SI

JMP \_JIXU1

P: MOV CX,4

\_P: CALL F2T10

IO CHJI,9

MOV CHJI,' '

INC SI

DEC CX

JZ M

JMP \_P

\_M: ADD DI, 6

MOV SI, DI

HUICHE

INC BX

CMP BX, JISHU

JNZ \_NET

RET

\_SHUCHU ENDP

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

JIAOHUAN PROC

PUSH AX ;把两个学生的内容交换一下

PUSH BX

PUSH BP

MOV BP, SI

MOV BX,0

LOOPA: MOV AL, DS: [BP]

MOV AH, DS: 16[BP]

MOV DS: [BP], AH

MOV DS: 16[BP], AL

INC BP

INC BX

CMP BX,14

JB LOOPA

POP BP

POP BX

POP AX

RET

JIAOHUAN ENDP

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

F2T10 PROC

PUSH AX

PUSH DX

PUSH DI

LEA DI, CHJI+2

MOV DX, 10

XOR AX, AX

MOV AL, [SI]

\_Q: IDIV DL

MOV [DI],AH

ADD BYTE PTR [DI], '0'

CMP AL,0

JZ \_S

MOV AH,0

DEC DI

JMP \_Q

\_S: POP DI

POP DX

POP AX

RET

F2T10 ENDP

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CODE1 ENDS

END

### 实验步骤

1. 准备上机环境，编辑、汇编、连接文件sy、sy2。
2. 输入sy，运行程序,检验各个功能是否可以正常使用。
3. TD观察宏指令在执行程序中的替换和扩展。
4. 在TD中观察子程序调用和返回时堆栈的变化。
5. 观察FAR、NEAR类型子程序的RET指令的机器码，观察FAR类型子程序被调用时堆栈的变化情况。
6. 修改公共变量或参数（符号名不一致或类型不一致），观察有什么现象。

### 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3.1.6所示

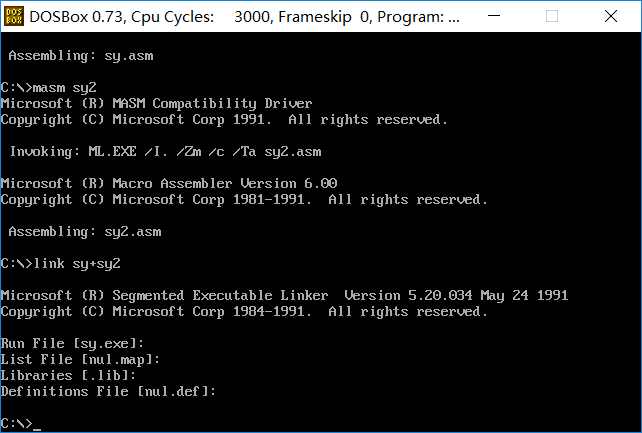


图3.1.6 正常编译连接

1. 输入sy，运行程序，检验各个功能，如图3.1.7、图3.1.8、图3.1.9所示

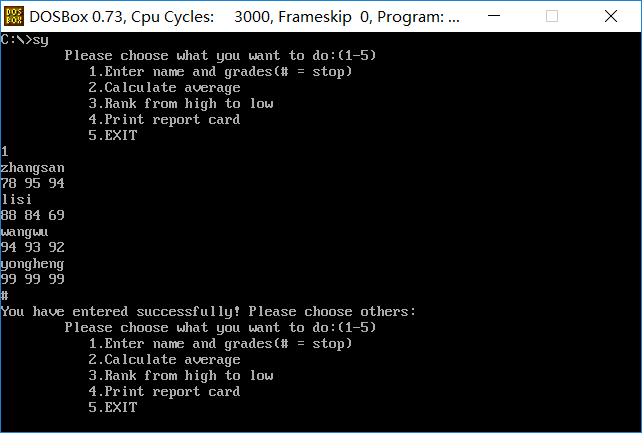


图3.1.7 功能一检验，输入名字和成绩，以‘#’结束

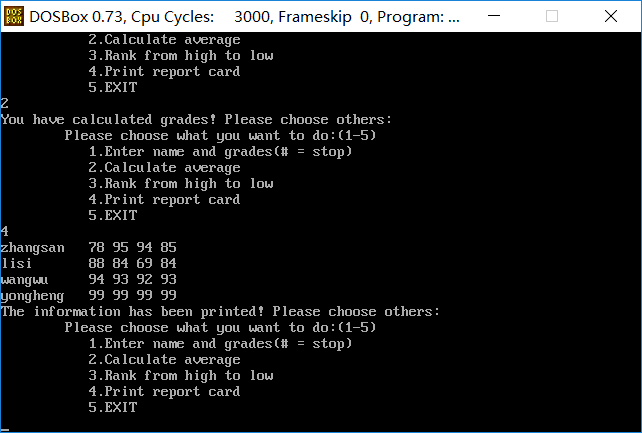


图3.1.8 功能二、四检验，未排序前输出

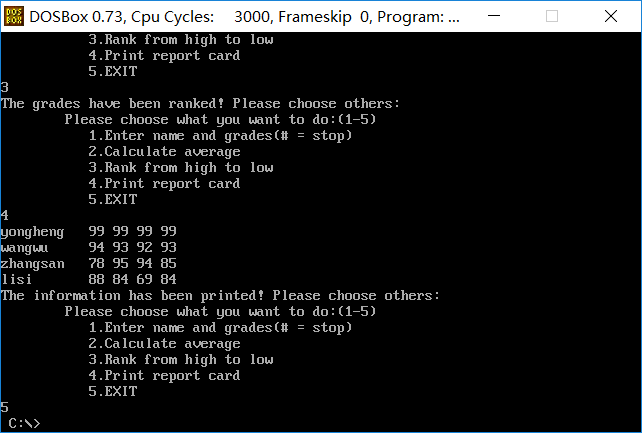


图3.1.9 功能三、五检验，排序后输出

1. TD观察宏指令在执行程序中的替换和扩展，如图3.1.10所示

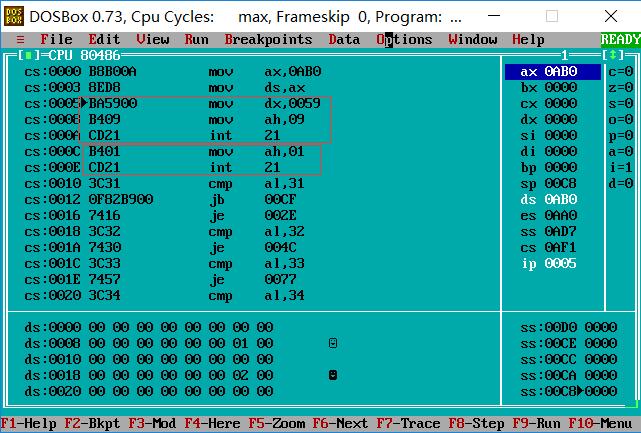


图3.1.10 9号、1号宏指令扩展

1. 在TD中观察子程序调用和返回时堆栈的变化，如图3.1.11、3.1.12、3.1.13、3.1.14所示

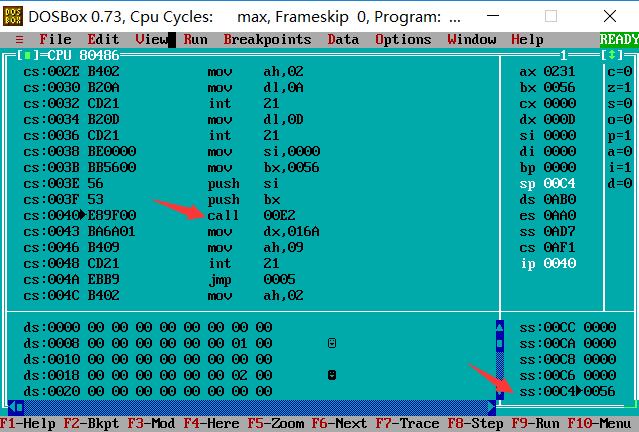


图3.1.11 调用前，参数入栈

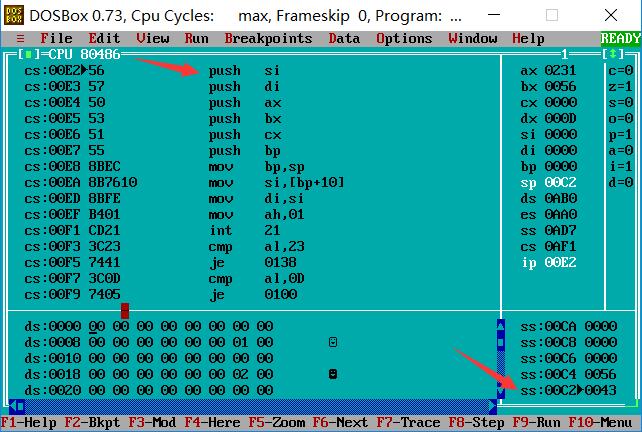


图3.1.12 进入子程序后，堆栈区存放返回IP值

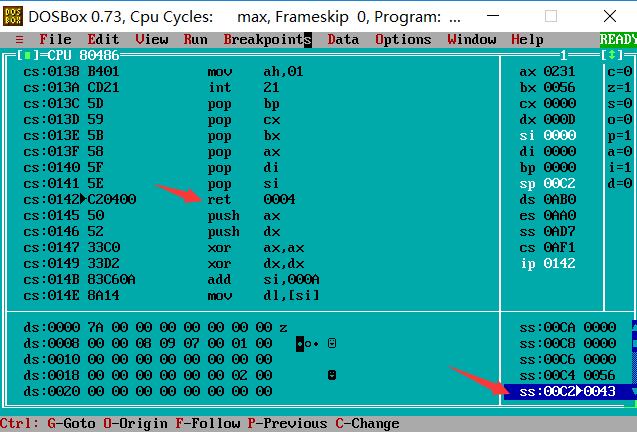


图3.1.13 退出时，堆栈顶存放返回IP值

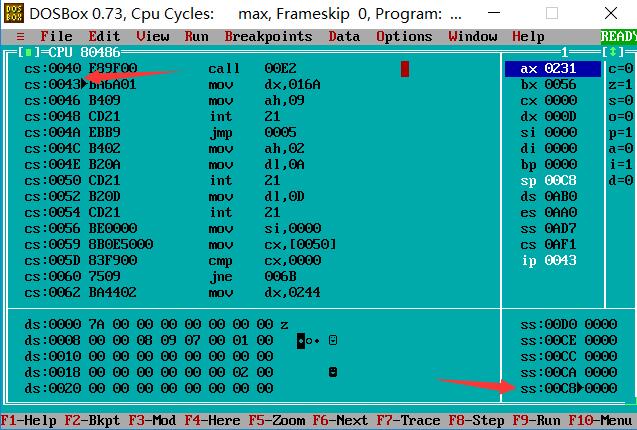


图3.1.14 退出后，程序运行到子程序下一行，参数清除

1. 观察FAR类型子程序被调用时堆栈的变化情况，如图3.1.15、3.1.16、3.1.17、3.1.18所示

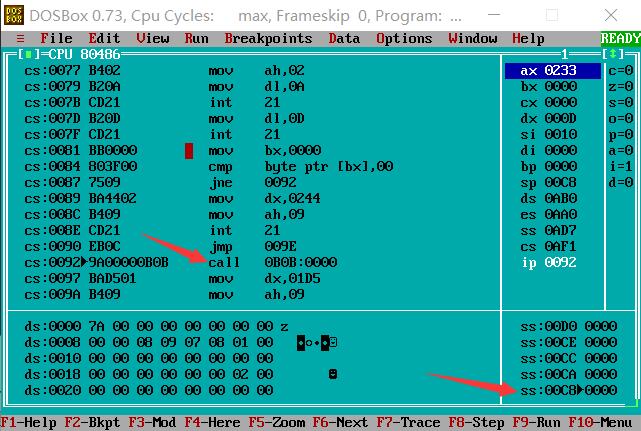


图3.1.15 调用FAR类型子程序前

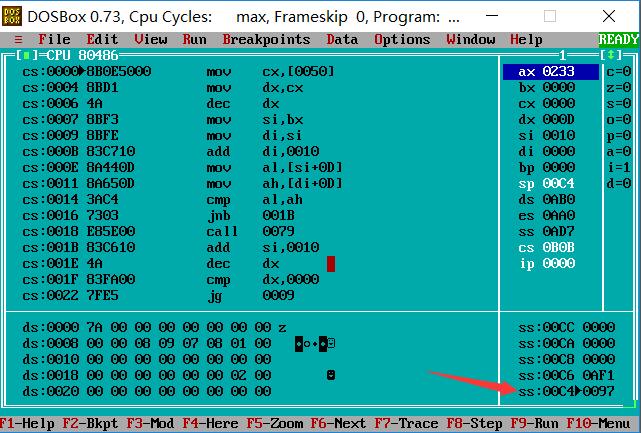


图3.1.16 进入FAR类型子程序后，堆栈区先后存储CS、IP值

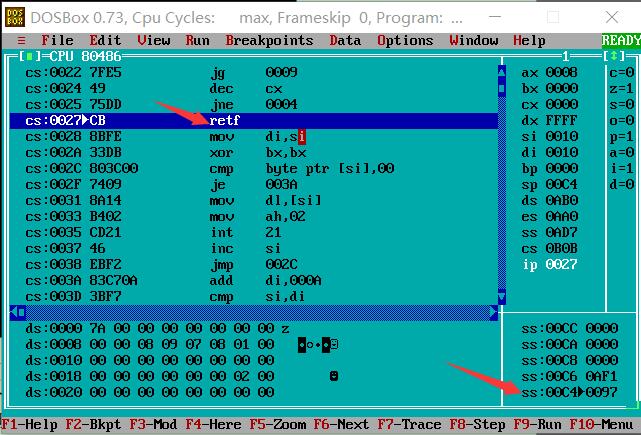


图3.1.17 退出FAE类型子程序时

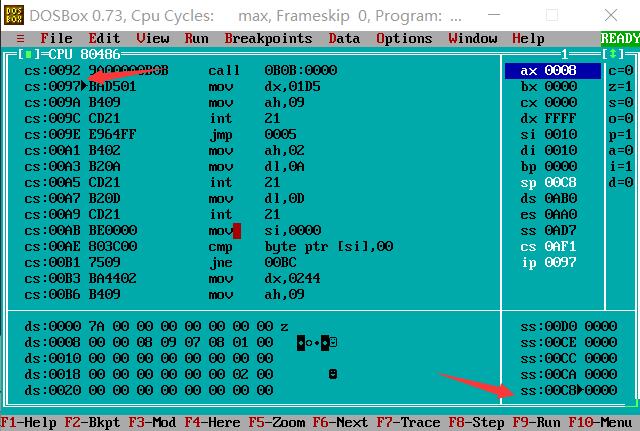


图3.1.18 退出FAR类型子程序，运行到调用行下一行

1. 修改公共变量JISHU，在模块二中修改为EXTRN JISH:WORD，结果模块二编译不通过，显示JISHU未定义，如图3.1.19所示

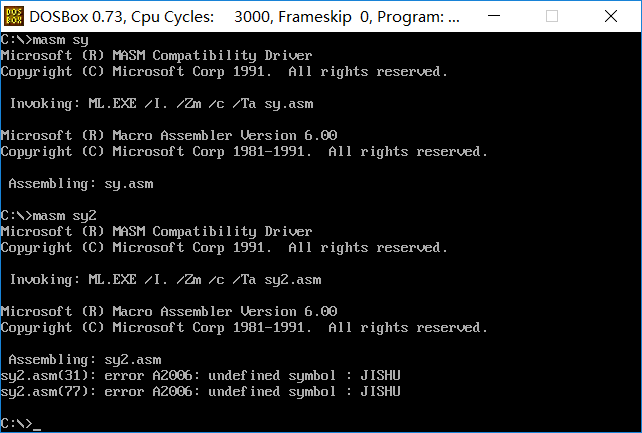


图3.1.19 编译不通过（一）

修改为EXTRN JISHU:BYTE，模块二编译不通过，如图3.1.20所示

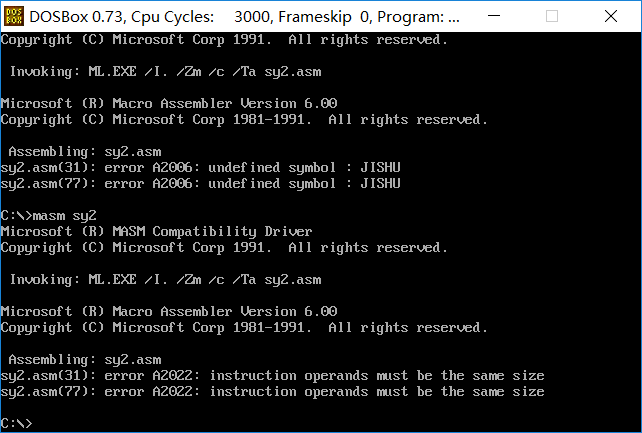


图3.1.20 编译不通过（二）

1. 从TD中跟踪到子程序内部可以通过按F7进入，也可以修改IP值进入，也可以在子程序内部设置断点，按F9运行到子程序。
2. FAR的RET指令为‘retf’，NEAR的RET指令为‘ret’。
3. 在程序中使用宏指令，仅仅是方便编写者，但这些在编译时仍然会显示出来，对实际代码长度没有影响；而使用子程序时，会修改IP值，进入子程序，等运行完之后再返回，如果只看主程序，代码长度变短，通过保护现场，在子程序中的寄存器不会影响主程序中的值。

## 任务二

### 设计思想及存储单元分配

1. 计算学生平均值

寄存器分配

SI：指向信息存储单元

CX：存储学生人数

AX：存储成绩加权和

DL：取每项成绩

BP：指向栈顶，取出参数

1. 按平均成绩从高到低排序

寄存器分配

SI：指向前一个学生的信息

DI：指向后一个学生的信息

AL：取出前一个学生的平均成绩

AH：取出后一个学生的平均成绩

CX：存储学生人数

DX：计数，控制比较次数

BP：指向栈顶，取出参数

交换子程序中寄存器分配

SI：参数，指向需要交换信息的前一个学生

BP：存储参数值

AL：取出前一个学生的信息

AH：取出后一个学生的信息

CX：计数，判断是否交换完毕

### 源程序

* 1. C语言主程序

#include <stdio.h>

#include <stdlib.h>

#define N 5

typedef struct sheet{

char name[10];

char chinese;

char math;

char english;

char average;

short rank;

}XINXI;

XINXI marklist[N];

//参数为结构数组首地址以及学生人数

extern void \_JISUAN(XINXI \*, short x);

extern void \_PAIXU(XINXI \*, short x);

void menu(void);//菜单显示函数

void shuru(void);//成绩表录入函数

void shuchu(void);//成绩表输出函数

int main(void)

{

int i;

while(1)

{

menu();

scanf("%d", &i);

switch(i)

{

case 1: shuru();

break;

case 2: \_JISUAN(marklist, N);//调用汇编子程序，计算平均成绩

break;

case 3: \_PAIXU(marklist, N);//调用汇编子程序，对成绩进行排序

break;

case 4: shuchu();

break;

case 5: return 0;

}

}

}

//菜单显示函数

void menu(void)

{

printf("Please choose what you want to do:(1-5)\n");

printf("1.Enter name and grades(# = stop) \n");

printf("2.Calculate average\n");

printf("3.Rank from high to low\n");

printf("4.Print report card\n");

printf("5.EXIT\n");

}

//成绩表录入函数

void shuru(void)

{

int i;

for(i = 0; i < N; i++)

scanf("%s%d%d%d",marklist[i].name, &marklist[i].chinese, &marklist[i].math, &marklist[i].english);

}

//成绩表输出函数

void shuchu(void)

{

int i;

printf("NAME CHINESE MATH ENGLISH AVERAGE RANK\n");

for(i = 0; i < N; i++)

printf("%-10s%10d%10d%10d%10d%10d\n",marklist[i].name, marklist[i].chinese, marklist[i].math, marklist[i].english, marklist[i].average, marklist[i].rank);

}

* 1. 计算平均值汇编子程序

.model small,c

.386

.data

.code

PUBLIC \_JISUAN

\_JISUAN PROC

PUSH AX

PUSH DX

PUSH SI

PUSH CX

PUSH BP

MOV BP,SP

MOV SI,12[BP]

MOV CX,14[BP]

\_NEXT: XOR AX, AX

XOR DX, DX

MOV DL, 10[SI]

SAL DX, 2

ADD AX,DX

XOR DX,DX

MOV DL,11[SI]

SAL DX,1

ADD AX,DX

XOR DX,DX

MOV DL,12[SI]

ADD AX,DX

MOV DL,7

IDIV DL

MOV 13[SI],AL

ADD SI,16

DEC CX

JNZ \_NEXT

POP BP

POP CX

POP SI

POP DX

POP AX

RET

\_JISUAN ENDP

\_TEXT ENDS

END

* 1. 按平均值从大到小汇编子程序

.model small,c

.386

.data

.code

PUBLIC \_PAIXU

\_PAIXU PROC ;排序子程序

PUSH AX

PUSH CX

PUSH DX

PUSH SI

PUSH DI

PUSH BP

MOV BP,SP

MOV CX,16[BP]

BIG: MOV DX,CX

DEC DX

MOV SI,14[BP]

\_SMALL: MOV DI,SI

ADD DI,16

MOV AL,13[SI]

MOV AH,13[DI]

CMP AL,AH

JNB CONTU

CALL JIAOHUAN

CONTU: ADD SI,16

DEC DX

CMP DX,0

JG \_SMALL ;如果ax！=0，则比较下一对数

DEC CX

JNZ BIG ;如果cx!=0，则进行下一轮冒泡

MOV SI,14[BP] ;排名次

MOV CX,0

MINGCI: INC CX

CMP CX, 16[BP]

JG FLAG

MOV WORD PTR 14[SI], CX

ADD SI, 16

JMP MINGCI

FLAG: POP BP

POP DI

POP SI

POP DX

POP CX

POP AX

RET

\_PAIXU ENDP

JIAOHUAN PROC

PUSH AX ;把两个学生的内容交换一下

PUSH CX

PUSH BP

MOV BP,SI

MOV CX,0

LOOPA: MOV AL,DS:[BP]

MOV AH,DS:16[BP]

MOV DS:[BP],AH

MOV DS:16[BP],AL

INC BP

INC CX

CMP CX,14

JB LOOPA

POP BP

POP CX

POP AX

RET

JIAOHUAN ENDP

\_TEXT ENDS

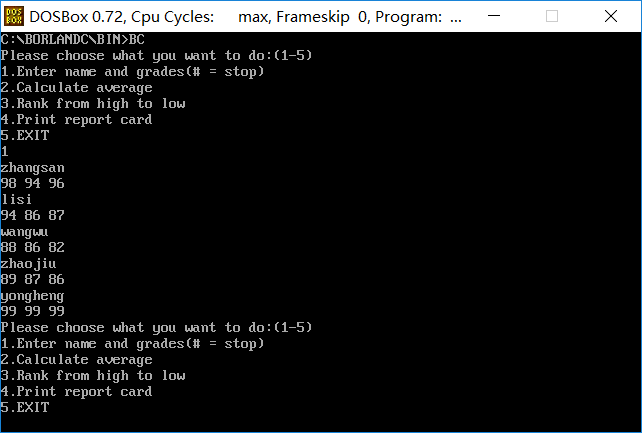
END

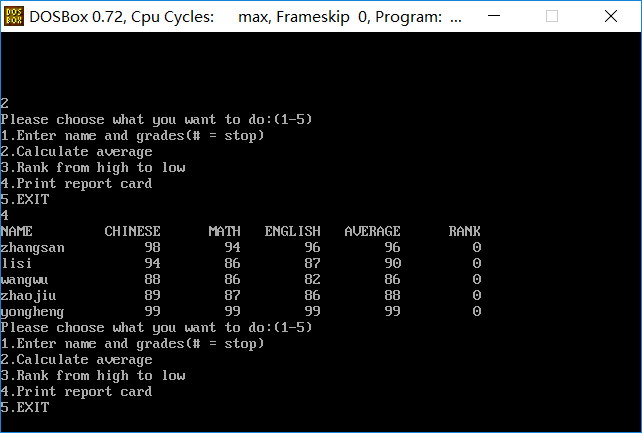
### 实验步骤

1. 将两个asm文件编译，生成的obj文件放到bin文件目录下，再将c文件放到该目录下。
2. 生成一个新工程，将这三个文件添加进去，编译，连接，运行，检验功能是否正常。
3. Td调试该程序，观察主、子程序之间参数传递的机制。

### 实验数据记录与分析

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. 运行该程序，如图3.2.1所示





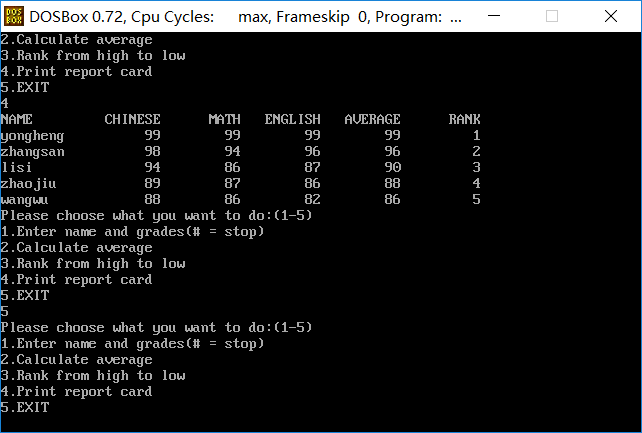


图3.2.1 C语言和汇编语言混合程序

1. 退出，调试该程序，观察主、子程序之间参数传递，如图3.2.2、图3.2.3、图3.2.4所示

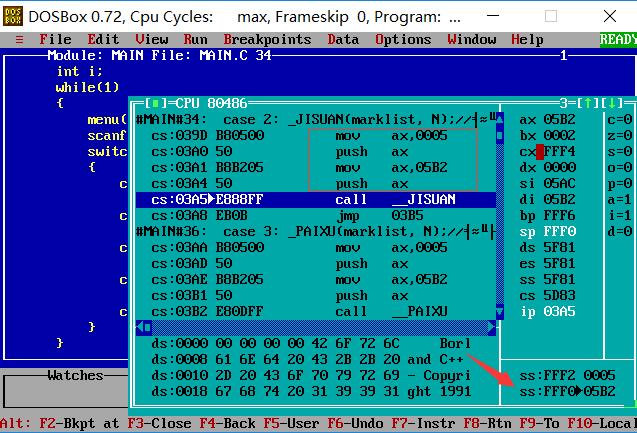


图3.2.2 进入子程序前，参数依次压栈

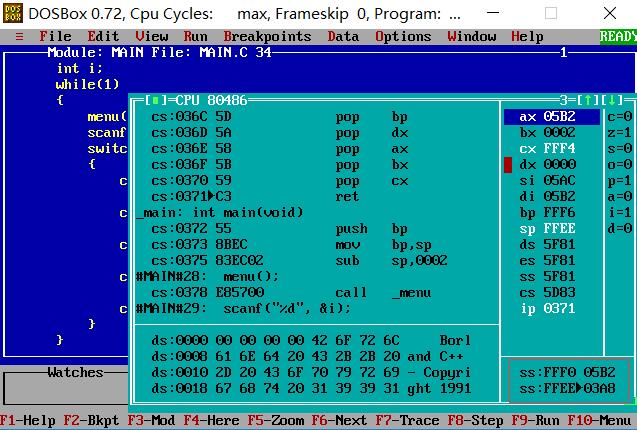


图3.2.3 退出子程序前，堆栈区为IP值和参数

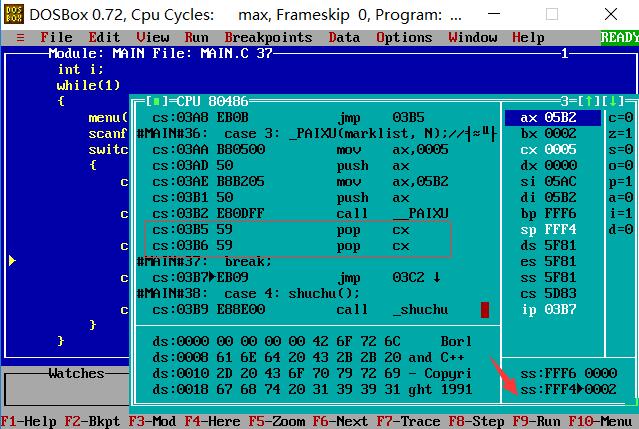


图3.2.2 退出子程序后，参数依次出栈

## 选做题

### 实验数据记录及分析

1. 通过控制循环，依次输出空格，以此控制输出格式，如图3.3.1所示

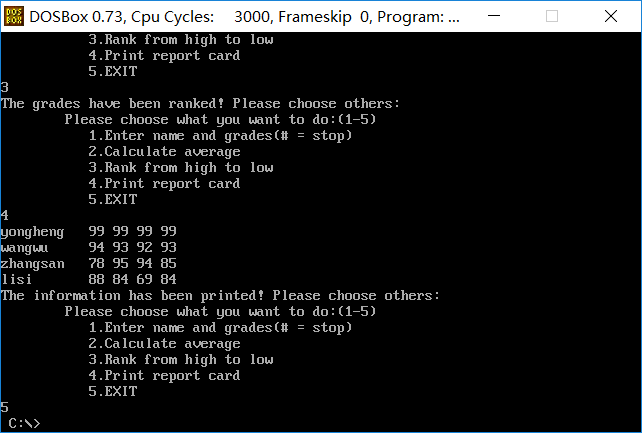


图3.3.1 输出菜单

1. EXTRN说明语句放在.386之前，由于在EXTRN之前没有处理器选择伪指令，汇编程序将选择缺省值‘.8086’。而EXTRN语句放在.386后面，EXTRN后面的子程序会被当作32位段翻译，此时将与自己编写的子程序的意义不一致。
2. 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）不能带入到本模块中。

# 总结与体会

终于写完了这次实验，虽然本次实验有两周的完成时间，但是由于自己比较拖沓，再加上这周的党课，导致自己没有时间编写修改程序，报告也迟交，这是一个教训，自己以后一定要提前写好程序。

在任务一中，我负责主控程序和任务一、二，主控程序比较容易实现，任务二由于之前写过，也比较简单，比较复杂的是任务一，首先我在确定存储结构时很纠结，想用结构，但是觉得不熟悉，怕出错，但是又想尝试一下，也想直接开辟存储单元，进行存储，这样很方便后续处理，但这又没有什么新意，最后我还是选择了直接开辟存储单元。在输入名字时，我是以字符形式存入的，通过一些判断以实现名字的正确输入，输入成绩时，由于输入的是字符串，需要调用串转换数子程序，因此，特意写了一个比较简单的子程序。在同伴写好后，我们尝试连接合并，初次连接并没有成功，原因是公共变量和子程序申明出错，调整后成功连接。

最后我们编写的程序正确运行，通过这种分开编写各个模块，最后合并连接，我感受到合作的力量，以及子程序的强大功能，即无论主程序中怎么样，编写子程序时都不用考虑，只需考虑子程序中情况即可。

任务二是C和汇编的混合编写。对于我来说，这很新奇，从没想过两个语言可以共同编写一个程序，在写完后，运行调试过程中，我更加了解了C语言函数的传参方式，以及清除参数方法。

这次的实验中，我没有投入时间，问了一些同学，导致完成很慢，但我会找时间再仔细学习的。

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》“第四章 程序设计的基本方法”，“第五章 程序设计的其他方法和技术”.华中科技大学出版社

[2] 许向阳，《80X86汇编语言程序设计上机指南》“第十一章 汇编语言程序与C程序的连接”。