

**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** 程序执行时间与代码长度优化

**实验时间： 2016-4-12，14：30-17：30 实验地点： 南一楼804室102号实验台**

**指导教师： 张勇**

**专业班级：计算机科学与技术201409班**

**学 号： U2014414808 姓 名： 王林**

**同组学生： 无 报告日期： 2016年 4 月 13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： 王林

日期： 2016.4.13

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**

[实验目的与要求 1](#_Toc448344765)

[实验内容 1](#_Toc448344766)

[实验过程 1](#_Toc448344767)

[4.1任务1 1](#_Toc448344768)

[4.1.2流程图 2](#_Toc448344769)

[4.1.3源程序 5](#_Toc448344770)

[4.1.4实验步骤 12](#_Toc448344771)

[4.1.5实验记录 12](#_Toc448344772)

[体会 14](#_Toc448344773)

[参考文献 15](#_Toc448344774)

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。

实验内容

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：

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=成绩排序，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。

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。

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。

排序的基本要求是按照平均成绩从高到低排序，也可以考虑按照指定课程的成绩排序。输出成绩单的基本要求是按照排序的结果依次显示每个学生的姓名和平均成绩，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）。

**提示：**可以采用一个一维数组记录每个学生成绩的排序名次，这样就可以避免在排序算法里交换成绩表中的具体信息了。

将9号和10号DOS系统功能调用定义成宏指令并调用。

实验过程

4.1任务1

修改与增强实验二的学生成绩查询程序的功能4.1.1设计思想及存储单元分配

显示菜单界面，输入学生姓名和成绩，计算平均成绩，并按照从高到低排序，最后输出排序后的成绩。

4.1.2流程图



**图 4-1 主程序流程图**



**图 4-2 功能1流程图**



**图4-3 功能2流程图**



**图 4-4 功能3流程图**



**图 4-5 功能4流程图**

4.1.3源程序

以下为两部分的源程序，分别由我和冷威同学编写，然后各自编译通过，生成两个OBJ文件，然后链接，生成一个可执行文件：AS\_LAB4.exe

**【test1.asm】**

;test1 source code

;@author strawberrylin

name first

extern Sort:far,Print:far

public store,num,TIPS

.386

;宏定义9号功能

write macro a

lea dx, a

mov ah, 9

int 21h

endm

read macro b

lea dx, b

mov ah, 10

int 21h

endm

data segment use16 para public 'data1'

bufc db '1:Input the name and score',0ah,0dh,

'2:Figure the sum and the average score',0ah,0dh,

'3:Sort the score',0ah,0dh,

'4:Output the score from high to low',0ah,0dh,

'5:Exit...',0ah,0dh,

'Enter your choose:','$'

buft2 db 0ah,0dh,'Input name:$'

buft3 db 0ah,0dh,'Input chinese score:$'

buft4 db 0ah,0dh,'Input math score:$'

buft5 db 0ah,0dh,'Input english score:$'

buf db 20

db ?

db 20 dup(0)

num dw 0

store db 200 dup(0)

TIPS db 0ah,0dh,'INFORMATIONS AS　FOLLOWS:',0ah,0dh,'$' ;信息输出提示语

PRJE DB 0ah,0dh,'PROJECT IS GOING TO EXIT!PRESS ANY KEY TO EXIT!$' ;结束提示语

data ends

stack segment use16 para stack 'stack'

db 200 dup(0)

stack ends

code segment use16 para public 'code'

assume ds:data,cs:code,ss:stack

start:

mov ax, data

mov ds, ax

lea di, store ; 信息储存位置

mov cx, 0 ; 学生数

show:

write bufc

mov ah, 1 ;input choose

int 21h

cmp al, '1'

jz function1

cmp al, '2'

jz function2

cmp al, '3'

jz function3

cmp al, '4'

jz function4

cmp al, '5'

jz function5

jmp show

function1:

call far ptr INFOIN

jmp show

function2:

call AVG

jmp show

function3:

call far ptr Sort

jmp show

function4:

call far ptr Print

jmp show

function5:

write PRJE

mov ah,1

int 21h

mov ah, 4ch

int 21h

code ends

;子程序名：AVG

;功能：计算平均成绩

;入口参数：储存学生信息的内存地址 store

;出口参数：无

procg segment use16

assume cs:procg

AVG proc far

push si

push cx

push dx

push ax

mov cx, num

mov ax, 0

lea si,store

add si, 10

fig:

mov bl, [si]

mov bh, 0

add ax, bx

mov bl, [si+1]

mov bh, 0

add ax, bx

mov bl, [si+2]

add ax, bx

mov dh, 0

mov dl, 3

div dl

mov [si+3], al

add si, 16

dec cx

jnz fig

pop ax

pop dx

pop cx

pop si

ret

AVG endp

procg ends

;子程序名：INFOIN

;功能：录入学生信息

;入口参数：储存学生信息的内存地址的di

;出口参数：无

proce segment use16

assume cs:proce

INFOIN proc far

inc cx

mov num,cx

push dx

mov dx, num

add dx, 30h

mov ah, 2

int 21h

push ax

push cx

push si

write buft2

read buf

mov cl, buf+1

mov ch, 0

lea si, buf+2

begin:

mov al, [si]

mov [di], al

inc si

inc di

dec cx

jnz begin

mov byte ptr [di], '$'

mov cl, buf+1

mov ch, 0

add di, 10

sub di, cx

write buft3

call GSCORE

write buft4

call GSCORE

write buft5

call GSCORE

inc di

mov byte ptr [di], 0ah ;在平均成绩后加入'0ah' '0dh' '$'

inc di

mov byte ptr [di], 0dh ;

inc di

mov byte ptr [di],'$' ;输出

pop si

pop cx

pop ax

pop dx

ret

INFOIN endp

;子程序名：GSCORE

;功能：得到输入的成绩

;入口参数：储存学生信息的偏移地址di

;出口参数：ax，存储对应的成绩

GSCORE proc

push dx

push bx

push si

push cx

read buf

mov cl, buf+1

lea si, buf+2

mov ax, 0

loapi:

mov bl, [si]

sub bl, 30h

mov bh, 0

imul ax, 10

add ax, bx

inc si

dec cx

jnz loapi

mov [di], al

inc di

pop cx

pop si

pop bx

pop dx

ret

GSCORE endp

proce ends

end start

**【test2.asm】**

;function3&function4 code

;@author: Lancyi

name second

extern num:word,store:byte,TIPS:byte

public Sort,Print

.386

data segment use16 para public 'data1'

data ends

stack segment use16 para stack 'stack'

db 200 dup(0)

stack ends

code segment use16 para public 'code'

assume ds:data,cs:code,ss:stack

code ends

;子程序名：Sort

;功能：按平均分成绩由高到低进行排序

;入口参数：储存学生信息的内存地址，即store,学生数cx

;出口参数： 无

procd segment use16

assume cs:procd

Sort PROC FAR

PUSH AX

PUSH DX

PUSH DI

PUSH CX

MOV CX, num

LOAP0:

LEA DI,store

MOV DX, CX

LOAP1:

ADD DI,13 ;获取当前学生平均成绩

MOV BL,16[DI] ;下一学生平均成绩（每个学生占14个字节），加上0AH0DH共16个

CMP BL,[DI] ;若当前学生平均成绩小于下一学生平均成绩则交换

JB IAE

PUSH CX

MOV CL,14

LOAP2:

MOV CH,[DI]

XCHG CH,16[DI]

MOV [DI], CH

DEC DI

DEC CL

JNE LOAP2

ADD DI,16

POP CX

IAE: ;获取下一学生的信息

ADD DI,3

DEC DX

CMP DX,1

JNE LOAP1 ;内层循环

DEC CX

CMP CX,1 ;外层循环

JNE LOAP0

POP CX

POP DI

POP DX

POP AX

RET

Sort ENDP

procd ends

;子程序名：Print

;功能：按平均分成绩由高到低将学生信息打印

;入口参数：储存学生信息的内存地址，即store

;出口参数： 无

procf segment use16

assume cs:procf

Print PROC FAR

push si

push cx

push dx

push ax

push bx

mov bx, num

lea dx, TIPS

mov ah, 9

int 21H

lea si, store

loapx:

mov cx, 4

mov dx, si

MOV AH, 9

INT 21H

add si, 10

loapy:

mov al, [si]

mov ah, 0

mov dx, 0

push bx

mov bl, 10

div bl

mov bx, ax

add bl, 30h

add bh, 30h

mov dl, bl

mov ah, 2

int 21h

mov dl, bh

mov ah, 2

int 21h

pop bx

inc si

dec cx

jnz loapy

inc si

inc si

dec bx

jnz loapx

pop bx

pop ax

pop dx

pop cx

pop si

RET

Print ENDP

procf ends

end

4.1.4实验步骤

1、按照实验和同学组队，然后分工，并且统一程序变量命名风格和规范。

2、然后各自完成自己部分的代码，实现自己部分的功能。

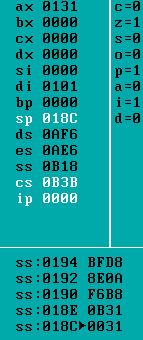
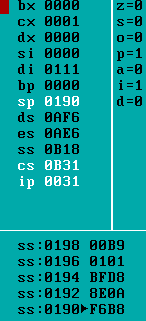
3、然后两部分编译，链接，生成可执行文件AS\_LAB4.exe

4.1.5实验记录

1、实验环境条件：P3 1GHz，256M内存；WINDOWS XP命令行窗口；Sublime Text 2；MASM.EXE 6.0； LINK.EXE 5.2; TD.EXE 5.0。

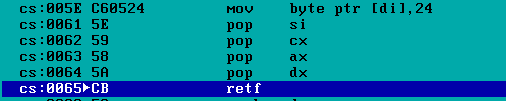
2、在TD中跟踪子程序的方法有：1、单步执行到对应的跳转指令，然后按F7键可以进入子程序内部；2、直接找到子程序的入口地址，然后通过在CS段直接找到子程序部分，然后观察子程序。

3、在子程序部分进入子程序前，会有push操做，即将子程序返回地址入栈，然后进入子程序。子程序结束后，最后有pop操作，返回到之前的返回地址。

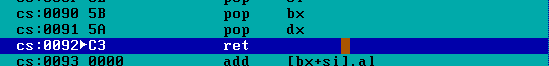
 

如图，进入输入学生信息的子程序后，有一个压栈操作，返回子程序时，有一个出栈操作。

4、在far调用子程序和near调用子程序时，返回指令不同，段间调用是retf,段内调用是ret。



段间调用返回指令如图。



段内调用返回指令如图。

5、把一个模块拆分成多模块后，要说明公开和引用的变量，还要统一的输入和输出说明，以便统一程序的编写。同时写好各部分的注释，比如子程序的输入参数和输出参数等等。

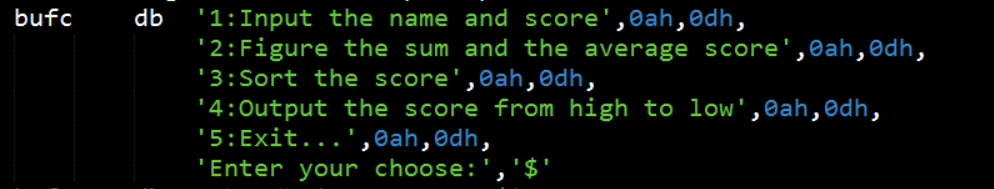
6、模块间通信时，若两个模块间公开和引用的变量名称不一样，则会出现无法解析的会未定义的变量错误，如果类型不同，则出现size不匹配的错误。

7、使用宏调用时，是直接替换对应符号，没有跳转操作，二调用子程序时，有跳转操作。



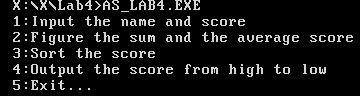
如图为将9号输出功能写成宏定义后，运行程序反汇编的情况，可以看到只是发生了符号的替换。

8、未使菜单更加漂亮，可以在数据段直接写出一长串字符，最后加上‘$’,然后直接调用9号功能，输出一长串字符，即是功能菜单界面。

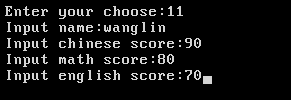


如图，就是这样。

9、实验运行记录如下：



如图为文字菜单界面。



选择功能1，输入学生信息，同时屏幕有显示当前输入学生个数。

然后执行功能2计算平均成绩，执行功能3排序，执行功能4输出排序后的成绩，结果如图



本次输入：qwe 1 2 3;asdf 4 5 6;经过功能2、3、4后输出如图。

体会

通过本次上机，更加熟悉了对TD工具的使用，通过对其他EXE文件的反汇编，观察反汇编代码，进一步熟悉子程序的运行情况，宏调用的操作，以及压栈出栈操作。

本次上机的具体收获有：（1）练习编写子程序，学习了段间调用和段内调用两种不同的方式。（2）练习了宏调用，将9号功能和10号功能写成宏调用的形式，是代码更加简洁。（3）学习了多模块编程，通过和同学组队，来完成各自的功能部分，然后组合在一起，更加了解了源程序生成可执行文件的机理，同时增加了程序开发了经验。

同时，对多模块编程还不够熟悉，需要进一步学习。

# 参考文献

[序号] 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