华 中 科 技 大 学

课 程 实 验 报 告

课程名称：汇编语言程序设计实验

实验名称：实验二分支程序、循环程序的设计

实验时间：**2019-9-30，19：00-22：20**

实验地点：南一楼

指导教师：鲁宏伟

专业班级：信息安全1802班

学 号：U201814867

姓 名：王证儒

报告日期：2019年9月30日

**成绩评定**

|  |  |  |
| --- | --- | --- |
| 实验完成质量（70分） | 报告撰写质量（30分） | 总成绩 |
| 实验步骤清晰、详细、深入，实验记录真实完整等 | 报告规范、完整、通顺、详实 |
|  |  |  |

**目 录**

[1实验目的 1](#_Toc21377406)

[2实验内容 1](#_Toc21377407)

[2.1 学生成绩查询程序 1](#_Toc21377408)

[2.1.1 实验说明 1](#_Toc21377409)

[2.1.2 功能一：提示并输入学生姓名 1](#_Toc21377410)

[2.1.3功能二：以学生姓名查询有无该学生 2](#_Toc21377411)

[2.1.4功能三：计算所有学生的平均成绩 2](#_Toc21377412)

[2.1.5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 2](#_Toc21377413)

[3实验过程 2](#_Toc21377414)

[3.1算法设计 2](#_Toc21377415)

[3.2源程序 3](#_Toc21377416)

[3.2.1 功能一 3](#_Toc21377417)

[3.2.2 功能二 4](#_Toc21377418)

[3.2.3 功能三 7](#_Toc21377419)

[3.2.4 功能四 8](#_Toc21377420)

[3.5实验步骤 11](#_Toc21377421)

[3.6实验结果及分析 12](#_Toc21377422)

[4心得与体会 13](#_Toc21377423)

# 1实验目的

1. 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；
2. 加深对转移指令及一些常用的汇编指令的理解；
3. 加深对常用DOS功能调用指令的理解，进一步熟悉TD的使用。

# 2实验内容

## 2.1 学生成绩查询程序

设计实现一个学生成绩查询的程序

### 2.1.1 实验说明

在以BUF为首地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：10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 1000

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, ?); 除了已经定义了的学生信息及成绩表外，其他学生的暂时成绩假定是一样的

db ‘WangWu’, 4 dup(0) ;最后一个必须修改为自己名字的拼音

db 85, 85, 100, ？

### 2.1.2 功能一：提示并输入学生姓名

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。输入的学生姓名字符串存放在以in\_name为首地址的存储区中。
3. 若只是输入了回车，则回到“（1）”处重新提示并输入；若仅仅输入字符q，则程序退出，否则，准备进入下一步处理。

### 2.1.3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，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。

### 2.1.4功能三：计算所有学生的平均成绩

1. 使用算术运算相关指令计算并保存每一个学生的平均成绩。
2. 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，再与数学成绩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运用。把小数3.5转换成分数后再运算以避免使用浮点指令。

### 2.1.5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。

1. 使用转移指令回到“功能一（1）”处（提示并输入姓名）。

# 3 实验过程

## 3.1 算法设计

总体上，所有的字符串输出使用 int 21H 的 09H 号中断，需要输出的内容全部保存在数 据段；所有需要输出的字符使用 int 21H 的 02H 号中断，需要输出的字符使用立即数表 示；需要输入数据则使用 int 21H 的 0AH 号中断，需要输入的数据存在数据区的数据缓 冲区中。

计算部分，发现计算平均成绩公式的特性，计算过程可以优化为一系列\*2（左移）和相加操作，最后利用div指令计算平均成绩，存入内存。

输入部分，对用户输入的数据进行判断，若为回车字符 (DH)，则判定为非法输入，提示 重新输入；若为’q\n’̊(71H, 0DH)，则判断为退出，直接终止程序；否则判断为姓名，进入查找部分继续执行。

查找部分，利用二重循环，遍历每条学生记录。在嵌套循环中：对姓名的每一个字进行逐字匹配，一旦发现失配则匹配不成功，进入下一个循环；若同时匹配到结尾，表明匹配成功，跳转进入判断部分；若所有的姓名均遍历完成仍没有匹配成功者，则表明姓名不存在，重新进入输入部分。

## 3.2 源程序

### 3.2.1 功能一

1: .386

2: carriage\_return macro

3: mov ah,02h

4: mov dl,0dh

5: int 21h

6: mov ah,02h

7: mov dl,0ah

8: int 21h

9: endm

10:

11: N equ 1000

12:

13: stack segment

14: db 200 dup(0)

15: stack ends

16:

17: data segment

18: db 'ZhangSan', 0, 0 ;学生姓名，不足10个字节的部分用0填充

19: db 100, 85, 80, ? ;平均成绩还未计算

20: db 'LiSi', 6 dup(0)

21: db 80, 100, 70, ?

22: db N-3 dup('TempValue ', 0, 80, 90, 95, ?); 除了已经定义了的学生信息及成绩表外，其他学生的暂时成绩假定是一样的

23: db 'WangZR', 6 dup(0) ;最后一个必须修改为自己名字的拼音

24: db 85, 85, 100, ?

25: msg1 db 'Please input the name of student:', 13, 10, '$'

26: in\_name\_header db 10, ?;预留给DOS的区域, DOS回调时的长度信息存放处

27: in\_name db 10 dup(?);输入存放区域

28: poin dw 0;存放偏移地址

29: data ends

30:

31: code segment use16

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

33: start:

34: mov ax, data

35: mov ds, ax

36: mov ax, stack

37: mov ss, ax

38: mov sp, 20

39: input:

40: lea dx, offset msg1;输出提示信息

41: mov ah, 09h

42: int 21h

43: mov in\_name\_header, 10

44: lea dx, offset in\_name\_header

45: mov ah, 0Ah;交给DOS输入

46: int 21h

47: carriage\_return

48: cmp in\_name, 0dh;如果只输入了回车

49: jz input

50: cmp in\_name, 'q'

51: jnz process

52: cmp in\_name + 1, 0dh

53: jz over

54: process:

55:

56: over:

57: mov ax, 4c00h

58: int 21h

59: code ends

60: end start

### 3.2.2 功能二

1: .386

2: carriage\_return macro

3: mov ah,02h

4: mov dl,0dh

5: int 21h

6: mov ah,02h

7: mov dl,0ah

8: int 21h

9: endm

10:

11: N equ 100

12:

13: stack segment use16

14: db 200 dup(0)

15: stack ends

16:

17: data segment use16

18: BUF db 'ZhangSan', 0, 0 ;学生姓名，不足10个字节的部分用0填充

19: db 100, 85, 80, ? ;平均成绩还未计算

20: db 'LiSi', 6 dup(0)

21: db 80, 100, 70, ?

22: db N-3 dup('TempValue ', 0, 80, 90, 95, ?); 除了已经定义了的学生信息及成绩表外，其他学生的暂时成绩假定是一样的

23: db 'WangZR', 6 dup(0) ;最后一个必须修改为自己名字的拼音

24: db 85, 85, 100, ?

25: msg1 db 'Please input the name of student:', 13, 10, '$'

26: msg2 db 'NOT FOUND!', 13, 10, '$'

27: in\_name\_header db 10, ?;预留给DOS的区域, DOS回调时的长度信息存放处

28: in\_name db 10 dup(?);输入存放区域

29: poin dw 0;存放偏移地址

30: data ends

31:

32: code segment use16

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

34: start:

35: mov ax, data

36: mov ds, ax

37: mov ax, stack

38: mov ss, ax

39: mov sp, 20

40: input:

41: lea dx, offset msg1;输出提示信息

42: mov ah, 09h

43: int 21h

44: mov in\_name\_header, 10

45: lea dx, offset in\_name\_header

46: mov ah, 0Ah;交给DOS输入

47: int 21h

48: carriage\_return

49: cmp in\_name, 0dh;如果只输入了回车

50: jz input

51: cmp in\_name, 'q'

52: jnz process

53: cmp in\_name + 1, 0dh

54: jz over

55: process:

56: mov bl, in\_name\_header[1]

57: xor bh, bh

58: mov byte ptr in\_name[bx], 0h

59: mov cx, N

60: mov bx, offset BUF

61: find\_iterator:

62: lea si, [bx]

63: mov di, offset in\_name

64: push bx

65: push cx

66: mov cl, in\_name\_header[1]

67: xor ch, ch

68: strcmp\_iterator:

69: mov dl, [si]

70: cmp dl, [di]

71: jnz next

72: inc si

73: inc di

74: loop strcmp\_iterator

75: jmp found

76: next:

77: pop cx

78: pop bx

79: add bx, 14

80: loop find\_iterator

81: not\_found:

82: mov ah, 09h

83: mov dx, offset msg2

84: int 21h

85: jmp input

86: found:

87: mov poin, bx

88: mov al, [bx + 13]

89: over:

90: mov ax, 4c00h

91: int 21h

92: code ends

93: end start

### 3.2.3 功能三

File: e:\Study\Assembly\asm4vscode\SRC\2-2-3.asm

1: .386

2:

3: N equ 100

4:

5: stack segment use16

6: db 200 dup(0)

7: stack ends

8:

9: data segment use16

10: BUF db 'ZhangSan', 0, 0 ;学生姓名，不足10个字节的部分用0填充

11: db 100, 85, 80, ? ;平均成绩还未计算

12: db 'LiSi', 6 dup(0)

13: db 80, 100, 70, ?

14: db N-3 dup('TempValue ', 0, 80, 90, 95, ?); 除了已经定义了的学生信息及成绩表外，其他学生的暂时成绩假定是一样的

15: db 'WangZR', 6 dup(0) ;最后一个必须修改为自己名字的拼音

16: db 85, 85, 100, ?

17: msg1 db 'Please input the name of student:', 13, 10, '$'

18: msg2 db 'NOT FOUND!', 13, 10, '$'

19: in\_name\_header db 10, ?;预留给DOS的区域, DOS回调时的长度信息存放处

20: in\_name db 10 dup(?);输入存放区域

21: poin dw 0;存放偏移地址

22: data ends

23:

24: code segment use16

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

26: start:

27: mov ax, data

28: mov ds, ax

29: mov ax, stack

30: mov ss, ax

31: mov sp, 20

32: calc\_avg:

33: mov bx, offset BUF

34: mov cx, N

35: calc\_avg\_iterator:

36: xor ah, ah

37: mov al, [bx+10];A成绩

38: shl ax, 1

39: add al, [bx+11]

40: adc ah, 0;处理1字节无法存储的情况

41: shl ax, 1

42: add al, [bx+12];这里相当于算((((A<<1)+B)<<1)+C)/7

43: adc ah, 0

44: mov dl, 7

45: div dl

46: mov [bx+13], al

47: add bx, 14

48: loop calc\_avg\_iterator

49: over:

50: mov ax, 4c00h

51: int 21h

52: code ends

53: end start

### 3.2.4 功能四

1: .386

2: carriage\_return macro

3: mov ah,02h

4: mov dl,0dh

5: int 21h

6: mov ah,02h

7: mov dl,0ah

8: int 21h

9: endm

10:

11: N equ 100

12:

13: stack segment use16

14: db 200 dup(0)

15: stack ends

16:

17: data segment use16

18: BUF db 'ZhangSan', 0, 0 ;学生姓名，不足10个字节的部分用0填充

19: db 100, 85, 80, ? ;平均成绩还未计算

20: db 'LiSi', 6 dup(0)

21: db 80, 100, 70, ?

22: db N-3 dup('TempValue', 0, 80, 90, 95, ?); 除了已经定义了的学生信息及成绩表外，其他学生的暂时成绩假定是一样的

23: db 'WangZR', 4 dup(0) ;最后一个必须修改为自己名字的拼音

24: db 85, 85, 100, ?

25: msg1 db 'Please input the name of student:', 13, 10, '$'

26: msg2 db 'NOT FOUND!', 13, 10, '$'

27: in\_name\_header db 10, ?;预留给DOS的区域, DOS回调时的长度信息存放处

28: in\_name db 10 dup(?);输入存放区域

29: poin dw 0;存放偏移地址

30: data ends

31:

32: code segment use16

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

34: start:

35: mov ax, data

36: mov ds, ax

37: mov ax, stack

38: mov ss, ax

39: mov sp, 20

40: input:

41: lea dx, offset msg1;输出提示信息

42: mov ah, 09h

43: int 21h

44: mov in\_name\_header, 10

45: lea dx, offset in\_name\_header

46: mov ah, 0Ah;交给DOS输入

47: int 21h

48: carriage\_return

49: cmp in\_name, 0dh;如果只输入了回车

50: jz input

51: cmp in\_name, 'q'

52: jnz find

53: cmp in\_name + 1, 0dh

54: jz over

55: find:

56: mov bl, in\_name\_header[1]

57: xor bh, bh

58: mov byte ptr in\_name[bx], 0h

59: mov cx, N

60: mov bx, offset BUF

61: find\_iterator:

62: lea si, [bx]

63: mov di, offset in\_name

64: push bx

65: push cx

66: mov cl, in\_name\_header[1]

67: xor ch, ch

68: strcmp\_iterator:

69: mov dl, [si]

70: cmp dl, [di]

71: jnz next

72: inc si

73: inc di

74: loop strcmp\_iterator

75: jmp found

76: next:

77: pop cx

78: pop bx

79: add bx, 14

80: loop find\_iterator

81: not\_found:

82: mov ah, 09h

83: mov dx, offset msg2

84: int 21h

85: jmp input

86: found:

87: mov poin, bx

88: mov al, [bx + 13]

89: calc:

90: mov bx, poin

91: xor ah, ah

92: mov al, [bx+10];A成绩

93: shl ax, 1

94: add al, [bx+11]

95: adc ah, 0;处理1字节无法存的情况

96: shl ax, 1

97: add al, [bx+12];这里相当算((((A<<1)+B)<<1)+C)/7

98: adc ah, 0

99: mov dl, 7

100: div dl

101: judge:

102: mov dl, 'A'

103: cmp al, 90

104: jge print

105: inc dl

106: cmp ax, 80

107: jge print

108: inc dl

109: cmp ax, 70

110: jge print

111: inc dl

112: cmp ax, 60

113: jge print

114: add dl, 2

115: print:

116: mov ah, 02h

117: int 21h

118: carriage\_return

119: jmp input

120: over:

121: mov ax, 4c00h

122: int 21h

123: code ends

124: end start

## 3.3 实验步骤

1.阅读代码，查看资料，确定程序功能。

2.准备上机实验环境，编写代码。

3.经MASM汇编，LINK连接后并且确认源程序正确无误后，直接运行，观察程序对输入的处理和输出有无明显错误。

4.测试各类输入，覆盖所有情况。

5.修改数据段内数据，重新编译链接进行测试。

## 3.4 实验结果及分析

1. 编译完成，无报错。直接执行二进制程序，出现提示字符串’Please input the name of the student:’

2. 输入一个列表中已经存在的姓名ZhangSan程序显示该学生平均分所对应的等级，并继续提示用户输入学生姓名。

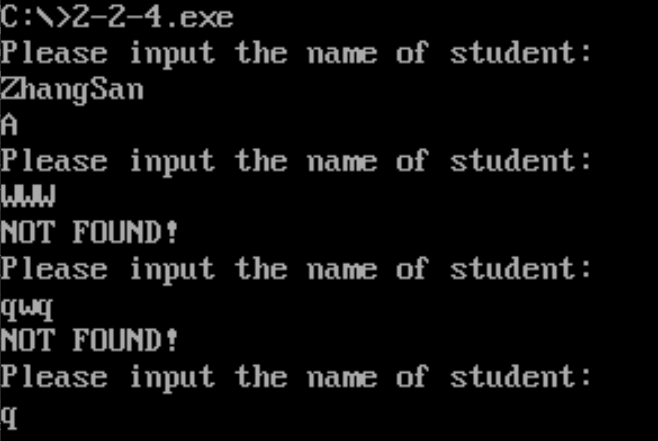
3. 输入一个列表中不存在的姓名WWW，程序提示该学生不存在，并要求重新输入。

4. 输入一个列表中不存在，且以字母q为前缀的姓名qwq，程序提示该学生不存在，并要求重新输入。

5. 直接输入回车，程序再次要求输入学生姓名。

6. 输入 q，程序直接退出。

1~6 步输出结果如图4所示，初步表明程序能够正常运行。



**图4初步测试记录**

# 4心得与体会

本次实验要完成一个相对大的程序。这是我写过的前所未有之长的汇编代码，同时也很好地锻炼了我的能力。

这次实验让我复习到了很多学过的汇编语法，这让我对汇编语言的指令、寻址方式更加熟悉。让我更加习惯汇编程序设计的流程和思路：用有限的寄存器和底层的简单操作完成更加复杂的功能，实现循环、分支等高级语言很容易做到但汇编不那么容易做到的feature。