计算机系统概论 A 实验报告

实验题目：实验四

学生姓名：牛庆源

学生学号：PB21111733

完成日期：2022.12.28

【实验题目】

排序 16 个学生的成绩，并统计 A 类学生和 B 类学生的人数，存放在指定的地址。

【实验目的】

1. 掌握汇编语言各种指令的写法。
2. 熟练使用 BR 指令。
3. 可以对照 c 语言代码写出汇编代码

【实验原理】

首先我们写一个 c 语言的冒泡排序

```
for(int i = 1; i < n; i++){
   for(int j = 0; j < n - i; j++){
      If(score[j] < score[j + 1]){
         swap(score[j], score[j + 1])
      }
   }
}
```

然后我们用汇编语言改写该内容即可，再加上对地址的处理。

1. R1 存初始的分数地址，R2 存学生人数，在学生人数使用结束后可以用来存其他的东西。R7 存放结果的首地址。
2. R3 存放 R1 指针指向的成绩。
3. R4 和 R5 分别存放 SCORE【j】和 SCORE【j+1】。R6 为-R5，并存放 R4-R5 的结果，用于判断两者大小。

4. 最后对于 A 类和 B 类人数，分别用 R5，R6 存放即可。
5. R1 用来计数（用于判断是否在 25%和 50%）。R2 和 R3 分别存放-85 和 -75 分别为 A 类和 B 类的界。
6. 将 R7 移动成绩结果的最后一个地址。（因为要从高分向低分存放 A 类和 B 类的学生，而存放结果是从低到高分，所以要移动到最后一个地址每次向前移动一个地址取成绩）
7. R4 取 R7 存放的内容，并-85 和判断是否是在 A 类，再通过 R1+1 判断 R1 是否在 25%来判断是否继续增加 A 类的人数（存放在 R5）。
8. 否则计数 B，计数 B 和计数 A（第 7 步）同理。（存放在 R6）
9. 将 R5 和 R6 的内容存放在指定内存。

【实验步骤】

1. 初始地址。
2. 初始化 R1, R2, R7。
3. 复制到指定的地址后再进行冒泡排序，排序结果直接就在指定位置。首先取指针内容，存在 R7 所指的位置，人数减一，两个指针移动，然后继续重复。COPY 结束的条件是人数减到非正（0）。
4. 参照 c 语言代码改写成汇编，其中 loopout 为外层循环，loopin 为内层循环。R2 为 i，从 1 开始。R7 指向 j 的位置，内容存放为 SCORE【j】。R4 和 R5 分别存放 SCORE【j】和 SCORE【j+1】。然后直接改写即可。对于大小判断的操作，用另一个寄存器存放负数的补码并 add 即可通过 BR 指令判断。
5. 最后按照实验原理所述的 5、6、7、8 步实现对 A 和 B 的统计即可。

【实验结果】

1. 遇到的最多的 bug 是控制指令取 nz 还是 n 或者是 z，进行调试后即消除 bug

2. 对于每一步的跳转都必须有清楚的意识，不然很容易弄混。

1..asm 文件截图如下：

初始化：

```
        .ORIG    x3000
        LD       R1, SCORE   ;R1为指针，指向初始的分数地址
        LD       R2, StuNUM  ;R2为学生人数（16）
        LD       R7, RESULT  ;R7为指针，指向要存放的分数初始地址
```

Copy 步骤：

```
COPY    LDR      R3, R1, #0  ;R3存成绩
        STR      R3, R7, #0  ;成绩存入结果
        ADD      R1, R1, #1  ;移向下一个成绩
        ADD      R7, R7, #1  ;结果成绩指针移向下一个
        ADD      R2, R2, #-1 ;R2减1
        BRp      COPY        ;全部复制
```

Loop 步骤：

```
        AND      R2, R2, #0
        ADD      R2, R2, #1  ;R2从1开始，代表i
LOOPout
         LD      R7, RESULT ;R7重新移动到指向要存放的分数初始地址
         AND     R3, R3, #0 ;R3清零
         ADD     R1, R2, #-16   ;R1=-(n-i)
    LOOPin
                 LDR     R4, R7, #0  ;R4存SCORE[j]
                 LDR     R5, R7, #1  ;R5存SCORE[j + 1]
                 NOT     R6, R5
                 ADD     R6, R6, #1  ;R6 = -R5
                 ADD     R6, R6, R4  ;R6 = R4 - R5
                 BRnz    FLAG    ;如果R4 <= R5
                 STR     R5, R7, #0
                 STR     R4, R7, #1  ;交换R4, R5
    FLAG
                 ADD     R7, R7, #1      ;指向下一成绩
                 ADD     R3, R3, #1
                 ADD     R6, R3, R1
                 BRn LOOPin
        ADD      R2, R2, #1
        ADD      R6, R2, #-16;
        BRn      LOOPout
```

统计 A，B 结果的初始化：

```
39      LD      R2, A     ;R2 = -85
40      LD      R3, B      ;R3 = -75
41      AND     R1, R1,#0        ;R1清零
42      AND     R5, R5,#0        ;R5清零存A人数
43      AND     R6, R6,#0        ;B人数
44      LD      R7, LAST         ;R7存放成绩结果末地址
```

对 A，B 的处理：

```
48  CountA  LDR     R4, R7, #0
49          ADD     R4, R4, R2  ;R4=score-85
50          BRn     CountB      ;如果score<85，开始计数B
51          ADD     R5, R5, #1  ;人数增加
52          ADD     R7, R7, #-1 ;移向下一分数
53          ADD     R1, R1, #1  ;计数加1
54          ADD     R0, R1, #-4 ;R0=计数-4
55          BRn     CountA      ;计数小于4，还在前25%继续判断A人数
56  ;
57  ; 获得B的人数
58  ;
59  CountB  LDR     R4, R7, #0
60          ADD     R4, R4, R3  ;R4=score-75
61          BRn     STORE       ;如果score<75，跳出B计数，进行保存
62          ADD     R6, R6, #1  ;人数增加
63          ADD     R7, R7, #-1 ;移向下一分数
64          ADD     R1, R1, #1  ;计数加1
65          ADD     R0, R1, #-8 ;R0=计数-8
66          BRn     CountB      ;计数小于8，还在前50%继续判断B人数
67
68  STORE   STI     R5, Anum    ;保存A，B人数
69          STI     R6, Bnum
70          HALT
```

所用到的 FILL 指令：

```
72  SCORE   .FILL   x4000       ;成绩存放起始地址
73  RESULT  .FILL   x5000       ;成绩结果起始地址
74  StuNUM  .FILL   #16         ;学生人数
75  A       .FILL   #-85        ;
76  B       .FILL   #-75        ;
77  LAST    .FILL   x500F       ;成绩结果末地址
78  Anum    .FILL   x5100       ;存放A地址
79  Bnum    .FILL   x5101       ;存放B地址
80          .END
```

2. 运行测试样例的结果如下：

lc3 评测

单样例最大指令数

1000000

选择评测实验

○ lab1 ○ lab2 ○ lab3 ◉ lab4 ○ 自定义

测试样例，样例之间以逗号分割

100:95:90:85:80:60:55:50:45:40:35:30:25:20:10:0,95:100:0:50:45:40:80:65:70:75:35:20:25:15:10:90,88:77

代码文本

```
        HALT

SCORE   .FILL   x4000           ;成绩存放起始地址
RESULT  .FILL   x5000           ;成绩结果起始地址
StuNUM  .FILL   #16                 ;学生人数
A           .FILL       #-85            ;
B           .FILL       #-75        ;
LAST    .FILL   x500F           ;成绩结果末地址
Anum    .FILL   x5100           ;存放A地址
Bnum    .FILL   x5101       ;存放B地址
            .END
```

调试模式

评测

汇编评测

3 / 3 个通过测试用例

- 平均指令数: 1627
- 通过 100:95:90:85:80:60:55:50:45:40:35:30:25:20:10:0, 指令数: 1682, 输出: 0,10,20,25,30,35,40,45,50,55,60,80,85,90,95,100,4,1
- 通过 95:100:0:50:45:40:80:65:70:75:35:20:25:15:10:90, 指令数: 1601, 输出: 0,10,15,20,25,35,40,45,50,65,70,75,80,90,95,100,3,2
- 通过 88:77:66:55:99:33:44:22:11:10:9:98:97:53:57:21, 指令数: 1598, 输出: 9,10,11,21,22,33,44,53,55,57,66,77,88,97,98,99,4,1