

ICS lab4

学号: PB21111715

姓名: 李宁

日期: 2022年12月1日

实验目的

- 1 学习汇编语言
- 2 熟练使用 lc3 指令
- 3 了解计算机系统

实验原理

h5 选择排序:

用 R0 作为指向原数组的指针(x4000), R1 作为指向排序后数组的指针(x5000), R2 存储当前最小值, R3 存储选择下界, R4 存储当前数字。R5 控制原数组的扫描结束, R6 记录已排序个数。

循环扫描原数组, 在大于 R3 的情况下, 判断 R4 是否小于 R2, 若小于, 更新 R2, 否则直接扫描下一个; 扫描一遍后存储 R2 的值到排序后的数组地址中, 回到原数组起点再次扫描, 直至排序后的数组已满。(这里是通过设置下界来判断某个元素是否被取到过, 但前提是数组中没有重复元素, 否则会只能取该元素一次)

h5 计算AB个数:

扫描排序后的数组, 依然用 R1 作为指针, R1 - 1 就是最后一个元素的位置, 也即最高成绩。R2 存储 A 的个数, R3 存储 A+B 的个数, R4 存储当前成绩, R5 存储常量 -85, R6 存储常量 -75。

向前扫描 4 位或者碰到成绩小于 85 的结束 A 的计算, 开始 B 的计算; 继续向前扫描至 8 位或者碰到成绩小于 75 的结束 B 的计算。

实验过程

h5 汇编代码:

```
1 .ORIG x3000
2 LD R6, COUNT ;剩余未排序个数
3 ADD R3, R3, #-1 ;下界设为-1
4 LD R1, ANS ;指向排序后数组的指针
5 ;初始化
6 AGAIN LD R0, ARRAY ;指向原数组的指针
7 LD R2, MAX ;最小值设为100
```

```

8 LD R5, COUNT ;剩余未扫描个数
9 ;扫描原数组
10 SUBAGAIN LDR R4, R0, #0 ;R4=M[R0]
11 ADD R0, R0, #1 ;R0++
12 ;判断是否取过
13 AND R7, R7, #0
14 NOT R7, R3
15 ADD R7, R7, #1
16 ADD R7, R4, R7 ;R7=R4-R3
17 BRnz SKIP
18 ;判断是否小于最小值
19 AND R7, R7, #0
20 NOT R7, R2
21 ADD R7, R7, #1
22 ADD R7, R4, R7 ;R7=R4-R2
23 BRzp SKIP
24 ;更新最小值
25 AND R2, R2, #0
26 ADD R2, R2, R4
27 SKIP ADD R5, R5, #-1
28 BRp SUBAGAIN
29 ;存储最小值, 更新下界
30 STR R2, R1, #0 ;M[R1]=R2
31 ADD R1, R1, #1 ;R1++
32 AND R3, R3, #0
33 ADD R3, R3, R2 ;R3=R2
34 ADD R6, R6, #-1 ;R6--
35 BRp AGAIN
36 ;计算A的个数
37 LD R5, ACONST ;R5= -85
38 LD R6, BCONST ;R6= -75
39 AND R2, R2, #0
40 AND R3, R3, #0
41 AGAIN1 ADD R1, R1, #-1 ;R1--
42 LDR R4, R1, #0 ;R4=M[R1]
43 ADD R7, R4, R5
44 BRn SKIP1
45 ADD R2, R2, #1 ;R2++
46 ADD R7, R2, #-4 ;判断优秀率是否超了
47 BRzp SKIP2
48 BRnzp AGAIN1
49 ;计算B的个数, 同理
50 SKIP1 ADD R1, R1, #1
51 SKIP2 ADD R3, R3, R2;
52 SKIP3 ADD R1, R1, #-1
53 LDR R4, R1, #0
54 ADD R7, R4, R6
55 BRn THEND

```

```

56 ADD R3, R3, #1
57 ADD R7, R3, #-8
58 BRzp THEND
59 BRnzp SKIP3
60 ;存储结果
61 THEND NOT R7, R2
62 ADD R7, R7, #1
63 ADD R3, R3, R7;
64 STI R2, NUMA
65 STI R3, NUMB
66 HALT
67 ;占位
68 ARRAY .FILL x4000
69 ANS .FILL x5000
70 NUMA .FILL x5100
71 NUMB .FILL x5101
72 MAX .FILL #100
73 COUNT .FILL #16
74 ACONST .FILL #-85
75 BCONST .FILL #-75
76 .END

```

实验结果

h5 测评结果：

汇编评测

3 / 3 个通过测试用例

- 平均指令数: 3352
- 通过 100:95:90:85:80:60:55:50:45:40:35:30:25:20:10:0, 指令数: 3424, 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: 3304, 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: 3328, 输出: 9,10,11,21,22,33,44,53,55,57,66,77,88,97,98,99,4,1

h5 自测结果：

汇编评测

1024 / 1024 个通过测试用例

- 平均指令数: 3262.8818359375
- 通过 58:5:18:50:55:73:84:31:61:41:40:4:11:87:75:22, 指令数: 3246, 输出: 4,5,11,18,22,31,40,41,50,55,58,61,73,75,84,87,1,2
- 通过 22:97:25:90:67:88:4:39:52:23:27:30:98:82:59:20, 指令数: 3276, 输出: 4,20,22,23,25,27,30,39,52,59,67,82,88,90,97,98,4,1
- 通过 87:83:21:36:57:10:37:28:51:96:22:84:52:86:2:47, 指令数: 3294, 输出: 2,10,21,22,28,36,37,47,51,52,57,83,84,86,87,96,3,2
- 通过 45:23:68:78:31:67:94:86:91:85:71:34:49:13:36:20, 指令数: 3248, 输出: 13,20,23,31,34,36,45,49,67,68,71,78,85,86,91,94,4,1

实验总结

- 还是先写伪代码，有了大框架再开写
- 善用 Label 和 BR 指令可以很方便的控制循环跳转，或跳过分支语句
- 注意指令的执行顺序，尤其是在跳转比较多的程序中，很容易因为顺序不对而达不到预期效果