

MIPS 排序实验

PB18151866 龚小航

本实验利用 MIPS 汇编，实现输入数据降序排列。

代码主要利用跳转语句实现状态之间的转换，包括执行函数，判断分支等。

程序结构为典型的代码声明+数据声明，以 main 作为程序入口。

将 main 部分代码展示如下，以说明程序结构。

.text 声明代码段开始，其中利用了 syscall，它根据\$**v0**的取值不同而会有不一样的效果：

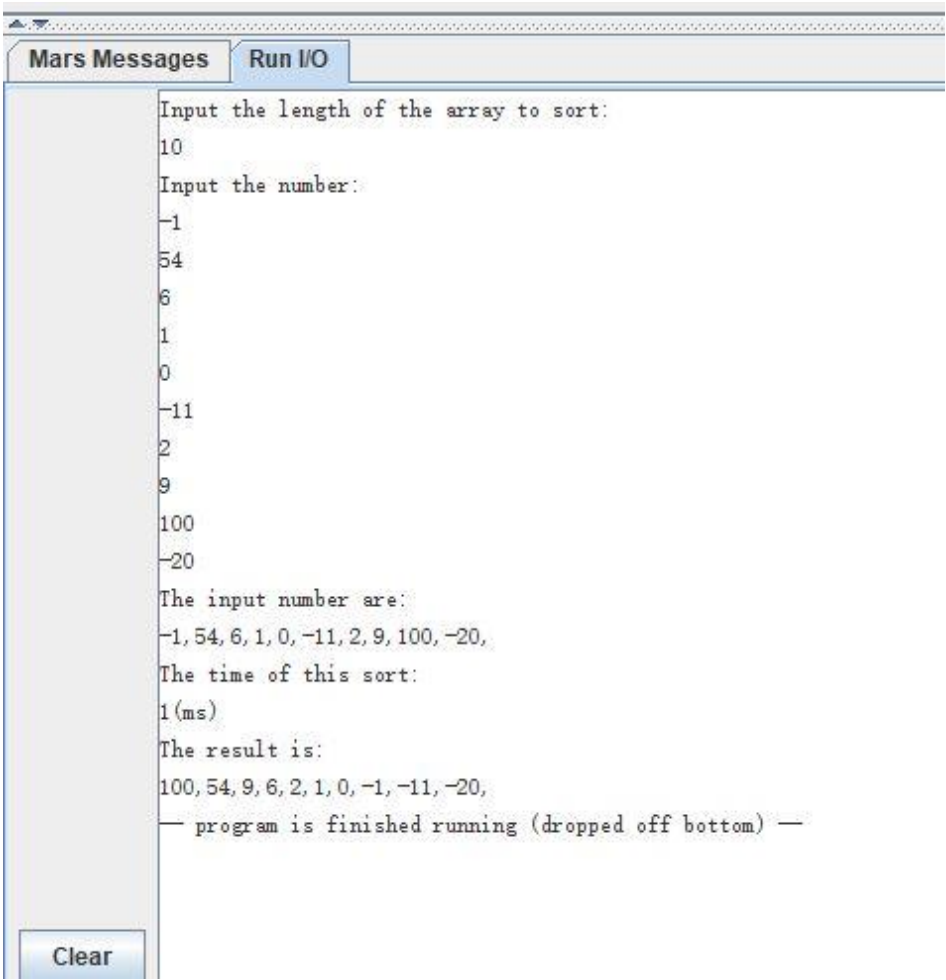
- 1: 输出\$a0 中的值；
- 4: 输出给定字符串；
- 5: 将控制台的数据读入\$v0；
- 11: 打印给定字符；
- 30: 计算代码执行时间

```
01  .text
02  .globl main
03  #$gp存数组基址
04  #$s0存数组大小
05  #函数调用时分别传给$a0和$a1
06
07 main:
08  la $a0, str_1      #输出提示用户输入数组大小
09  li $v0, 4
10  syscall            #输出给定字符串
11
12  li $v0,5           #系统调用把控制台中的数据读入$v0
13  syscall
14
15  move $s0, $v0      #把从控制台读入的数组大小保存到s0
16  la $a0, str_2      #输出字符串2，提示用户开始输入数据
17  li $v0, 4
18  syscall            #输出给定字符串
19
20
21  #调用read函数
22  move $a0, $gp      #把$gp作为参数传递给read函数拿到数组基址
23  move $a1, $s0
24  jal read           #跳转到read函数同时保存主函数地址到$ra
25
26
27  #打印刚才的输入结果
28  li $v0, 4
29  la $a0, str_4
30  syscall
31
32  move $a0, $gp
33  move $a1, $s0
34  jal print          #调用打印函数
35
36
37  #调用排序函数
38  move $a0, $gp
39  move $a1, $s0
40  jal sort
41
42
43  #调用输出函数
44  li $v0, 4
45  la $a0, str_3      #输出给定字符串
46  syscall
47
48  move $a0, $gp
49  move $a1, $s0
50  jal print
51
52  j EXIT
```

最后的 EXIT 在程序中指向数据声明部分，即跳过下方的函数声明，main 与数据声明相接，构成完整的 MIPS 汇编程序。

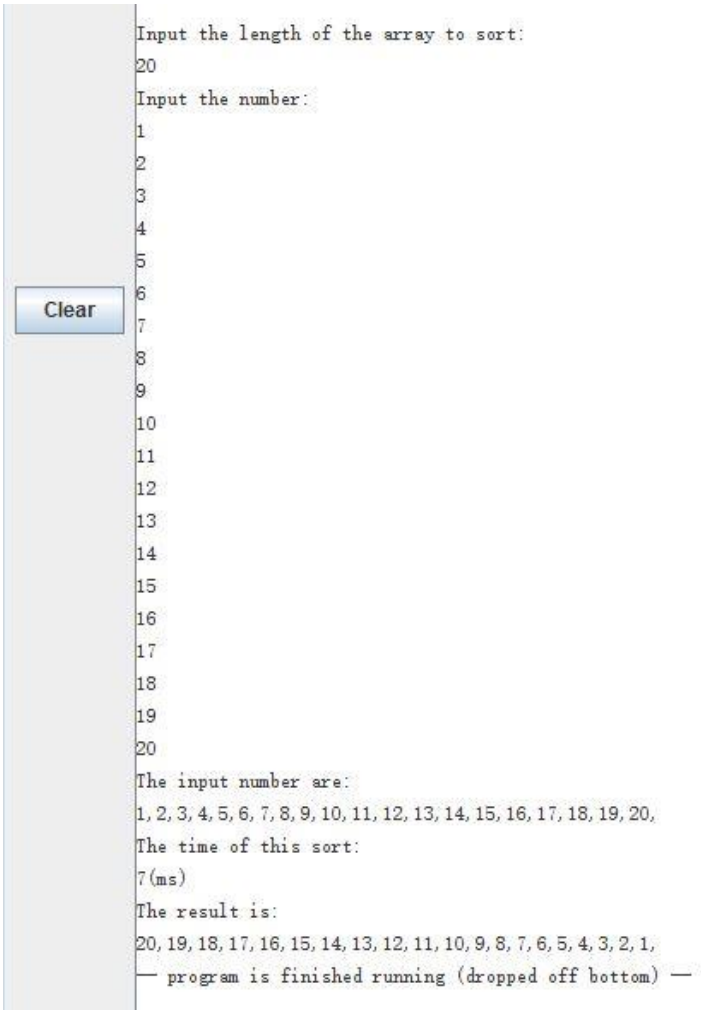
各个函数段的具体实现与注释附于 asm 文件中。

以下为测试用例：



可见十个数的排序所耗时间较短，仅为 1ms。运行时间与各个方面的多种因素均有关联，这种 syscall 的计时方法精度有限。

增加排序数据个数，并根据冒泡排序的特点，设计最坏情况：



20 个数据在最坏情况下用了 7ms 完成了冒泡排序。