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 <mark>$a0</mark>, str_1 #输出提示用户输入数组大小
09 li <mark>$v0</mark>, 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 <mark>$s0</mark>, <mark>$v0</mark> #把从控制台读入的数组大小保存到s0
16 la <mark>$a0</mark>, 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
   #调用read函数
21
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 <mark>$v0</mark>, 4
29 la <mark>$a0</mark>, 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, $s0
40 jal sort
41
42
43 #调用输出函数
44 li <mark>$v0</mark>, 4
45 la <mark>$a0</mark>, 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:

```
Run I/O
Mars Messages
            Input the length of the array to sort:
           10
           Input the number:
            -1
           54
            -11
            100
            -20
           The input number are:
            -1, 54, 6, 1, 0, -11, 2, 9, 100, -20,
           The time of this sort:
           1 (ms)
           The result is:
           100, 54, 9, 6, 2, 1, 0, -1, -11, -20,
             - program is finished running (dropped off bottom) -
  Clear
```

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:

```
Input the length of the array to sort:
          Input the number:
Clear
          10
          11
          12
          13
          14
          15
          16
          17
          18
          19
          20
          The input number are:
          1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
          The time of this sort:
          7(ms)
          20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1,
            - program is finished running (dropped off bottom) —
```

20 个数据在最坏情况下用了 7ms 完成了冒泡排序。