CS-35101

Thomas Moore

3/25/22

Summary:

This code operates by first reading and storing user given inputs on lines 7 through 21. It first compares input 1 with input 2. If input 1 is larger it branches off to store 1 in a new temporary register and then compares to see if input 1 is larger than input 3. If input 1 is not larger than input 2 it moves input 2 into another new temporary register and jumps to another comparison branch. Each of those comparison branches are labeled “Skip” and will compare the numbers again to find the 2nd largest input. These will branch off several times until it reaches an “Add” branch. The “Add” branches will combine the largest and 2nd largest number and then store that sum into the final temporary register $t5. All the “Add” branches jump to “Terminate” which prints the final sum and exits the program.

Conclusion:

This part of the project started off strong, but I ran into a major issue several times. If the 2nd input was the largest it would end up getting added to itself. This led me to create 5 different addition scenarios that would cover and check which input was the largest and then double check to see which input is the 2nd largest. I learned that comparison in MIPS requires precise knowledge of what data is in which register. Keeping track of data that is being compared several times can be tricky with limited registers.

Lab 3-2 Code:

```
1.  #Thomas Moore
2.  .data
3.          Intro: .asciiz "\nEnter 3 integers: "
4.          Sum: .asciiz "\nThe Sum of the two largest integers are: "
5.          test: .asciiz "\nThis is broken: "
6.  .text
7.          la $a0, Intro
8.          li $v0, 4 # print Intro string
9.          syscall
10.
11.         li $v0, 5 # read int
12.         syscall
13.         move $t0, $v0 # Number 1 = $t0
14.
15.         li $v0, 5 # read int
16.         syscall
```

```
17.         move $t1, $v0 # Number 2 = $t1
18.
19.         li $v0, 5 # read int
20.         syscall
21.         move $t2, $v0 # Number 3 = $t2
22.
23.         bgt $t0, $t1, Skip # if (Number 1 > Number 2) Jump to Skip
24.         move $t4, $t1 # if Number 2 is greater move it to $t4
25.         j Skip2
26. Skip:
27.         move $t3, $t0 # move Number 1 to $t3
28.         bgt $t3,$t2, Add # Number 1 > 3 Jump to Add
29.         bgt $t1, $t2 Add3 # 2 > 3 Jump Add3
30.
31. Skip2:
32.         bgt $t1, $t2, Add2 # Number 2 > 3 Jump to Add2
33.         add $t5, $t4, $t2 # if 3 is greater add
34.
35.         j Terminate
36.
37. Add:
38.         bgt $t1, $t2, Add4 #2>3
39.         add $t5, $t3, $t2 # Add 1 and 3
40.
41.         j Terminate
42.
43. Add2:
44.         bgt $t1, $t0, Add5 # 3 > 1 Jump Add5
45.         add $t5, $t4, $t1 # Add2
46.
47.         j Terminate
48. Add3:
49.         add $t5, $t2, $t3 # add 3 and 1
50.
51.         j Terminate
52.
53. Add4:
54.         add $t5, $t3, $t1 #add 1 and 2
55.
56.         j Terminate
57.
58. Add5:
59.         add $t5, $t4, $t2
60.
```

```
61.         j Terminate
62.
63. Terminate:
64.         la $a0, Sum
65.         li $v0 4
66.         syscall
67.
68.         move $a0, $t5
69.         li $v0, 1
70.         syscall
71.
72.         li $v0, 10
73.         syscall
```

Lab 3-2 Results:

```
Enter 3 integers: 1
2
3

The Sum of the two largest integers are: 5
-- program is finished running --


Reset: reset completed.


Enter 3 integers: 3
2
1

The Sum of the two largest integers are: 5
-- program is finished running --


Reset: reset completed.


Enter 3 integers: 1
3
2

The Sum of the two largest integers are: 5
-- program is finished running --


Reset: reset completed.


Enter 3 integers: 7
1
3

The Sum of the two largest integers are: 10
-- program is finished running --
```