Lab Activity 6b
Intro to Computer Science (420-121-VA)
Perry R. James, 1 October 2020

MIPS – Part 2

Second Program

You can load data into memory using the .data instruction in MIPS. You can also assign labels to the data to make it easier to reference.

For this assignment, you will work with the following data in memory:

```
        .data
d1:     .word 25
d2:     .word 18
d3:     .word 0

        .text
main:       #your program goes here
```

This program loads three pieces of data into memory and provides them with labels for easy reference.

Write a MIPS program to perform the calculation `d3 = d1 + d2`. You will want to load the data from `d1` and `d2` into two registers and then perform the calculation. Store the result in the memory location `d3`. The following instructions might be useful:

`lw`	loads a word from memory into a register
`sw`	stores a register in memory
`add`	adds signed integers (operand size is word)

Answer the following questions:

1. How many registers did you use to accomplish this task?
2. Is it possible to use fewer registers?
3. How many MIPS instructions did it take to perform the calculation?
4. How many Java statements would it take?
5. How many *characters* are in the MIPS and Java versions?

Lab Activity 6c
Intro to Computer Science (420-121-VA)
Perry R. James, 1 October 2020

MIPS – Part 3

Third Program

Performing mathematical calculations was one of the first uses of electronic computers. To get a feel for how difficult it is to write even simple programs in assembly, write a program that calculates $y = ax^2 + bx + c$ for given a, b, c, and x.

First calculate x^2, then multiply it by a. Then multiply b by x and add it to the value you computed for ax^2. Next, add c to that value. Use the code below to get started. The `mul` opcode is similar to the `add` that you used earlier. *(Since we're working with small numbers, we'll ignore the complications caused by the product of two words not always fitting in a single word.)*

```
      .data
a:    .word 3
b:    .word 2
c:    .word 5
x:    .word 4
y:    .word 0

      .text
#calculate y = ax^2 + bx + c
```

Answer the following questions:

1. How many registers did you use to accomplish this task? _____
2. How many multiplications and additions did your solution perform to find the solution?

   _____ additions and _____ multiplications
3. Note that $ax^2 + bx + c = (ax + b)x + c$. Does this representation point to another way of finding the solution? If so, write a program that uses it.
4. How many instructions are needed to compute the two versions?

   original _____ modified _____