

CA Lab: Homework 2

student: Dimitri Tabatadze

March 17, 2023

Task Description

We should have written 4 MIPS programs, each evaluating different mathematical expressions. The user should have been able to input up to three variables (depending on the program) and the program should have output the result of the mathematical expression. The results should have been as accurate as possible.

Solution

All four programs reuse the same part of the code, that is the I/O, so the instruction for inputting the numbers and outputting the result are basically the same for all files.

For inputting a single integer and storing it in a register `$t0` I do

```
% la      $a0, x
% li      $v0, 4
% syscall
% li      $v0, 5          # input integer
% syscall
% or      $t0, $v0, $zero # store in $t0 (x)
```

For printing the string corresponding to the expression, I do

```
% la      $a0, expression
% li      $v0, 4
% syscall
```

For outputting the number stored in `$a0`, I do

```
% li      $v0, 1          # print the value stored in $a0
% syscall
```

The code for each calculation is the following.

a)

variable map: $\$t0 = x$, $\$t1 = y$, $\$t2 = z$, $\$a0 = \langle \text{final value} \rangle$

```
% add    $t3, $t0, $zero
% mul    $t3, $t3, 5
% add    $t4, $t1, $zero
% mul    $t4, $t4, 3
% add    $t3, $t3, $t4
% add    $t3, $t3, $t2
% add    $a0, $t3, $zero
```

The data looks like this:

```
% .data
%      x:      .asciiz "x > "
%      y:      .asciiz "y > "
%      z:      .asciiz "z > "
%      expression: .asciiz "5x + 3y + z = "
```

b)

variable map: $\$t0 = x$, $\$t1 = y$, $\$t2 = z$

```
% add    $t3, $t0, $zero
% mul    $t3, $t3, 5
% add    $t4, $t1, $zero
% mul    $t4, $t4, 3
% add    $t3, $t3, $t4
% add    $t3, $t3, $t2
% div    $t3, $t3, 2
% mul    $a0, $t3, 3
```

The data looks like this:

```
% .data
%      x: .asciiz "x > "
%      y: .asciiz "y > "
%      z: .asciiz "z > "
%      expression: .asciiz "((5x + 3y + z) / 2) * 3 = "
```

c)

variable map: $\$t0 = x$

```
% add    $t0, $v0, 2
% mul    $t0, $t0, $v0
% add    $t0, $t0, 3
% mul    $t0, $t0, $v0
% add    $a0, $t0, 4
```

The data looks like this:

```
% .data
%      x: .asciiz "x > "
%      expression: .asciiz "x^3 + 2x^2 + 3x + 4 = "
```

d)

variable map: $\$t0 = x$, $\$t1 = y$

```
% mul    $t2, $t0, 4
% div    $t2, $t2, 3
% mul    $a0, $t2, $t1
```

The data looks like this:

```
% .data
%      x: .asciiz "x > "
%      y: .asciiz "y > "
%      expression: .asciiz "(4x / 3) * y = "
```

Conclusion

With the help of this task, I found out how to do simple useful operations, such as prompting input from the user. I also understood how I would have to sequence the instructions to implement the given mathematical.

Reference

I used the help window provided by MARS for MIPS documentation