Lab 3: MIPS!

Due Sunday 12 May 2019, 11:59 PM

Minimum Submission Requirements

- Ensure that your Lab3 folder contains the following files (note the capitalization convention):
  - Lab3.asm
  - README.txt
- Commit and push your repository

Lab Objective

This lab will introduce you to the MIPS ISA using MARS. You will write a program that calculates the factorial of an integer between 0 and 10.

Lab Preparation

Read chapters 1 (section 1.3), 2, 3, and 7 from Introduction To MIPS Assembly Language Programming. In addition, watch videos 2-3 and 19 from this video playlist:

https://www.youtube.com/watch?v=BlOLrVo4Nkk&index=2&list=PL5b07qlmA3P6zUdDf-o97ddfpvPFuNa5A

Specification

You will write a program in the MIPS32 language using the MARS integrated development environment to calculate the factorial of a number.

Example Outputs

```
Enter an integer between 0 and 10: 11

Invalid entry!

Enter an integer between 0 and 10: 0

0! = 1

-- program is finished running --
```

```
Enter an integer between 0 and 10: 10

10! = 3628800

-- program is finished running --
```

For full credit, **the output should match this format exactly**. Take note of the:

1. Exact wording of the prompts
2. Space after each colon
3. Space before and after the equal sign
4. New line character after the prompt
5. New line character after the error message
6. New line character after the final result

Functionality

This program will prompt the user for an integer between 0 and 10 and calculate the factorial. If the user enters a number outside of the acceptable range, the program must notify the user, and prompt the them again for an integer. You may assume that the grading script will only provide decimal integer inputs that can be stored in 32 bits.

The ASCII values for the characters you will be printing are here:

CHARACTER	ASCII CODE (HEX)
!	0x21
=	0x3D
(space)	0x20

Table: ASCII Codes

Syscalls

You **must** use the following syscalls for their specified purposes as shown below:

SYSCALL #	FUNCTION	PURPOSE
1	Print integer	Print integers
4	Print string	Print prompt and error message
5	Read integer	Read user input
10	Exit	Exit your program cleanly
11	Print character	Print exclamation point, spaces, and equal sign in answer printout

Table: Required Syscalls

Automation

Note that part of our grading script is automated, so **it is imperative that your program's output matches the specification exactly**. Output that deviates from the spec will cause point deduction.

Your code should end cleanly without error. **Make sure to use the exit syscall** (syscall 10).

Files

Lab3.asm

This file contains your code.

Follow the code documentation guidelines here. Make sure to include a header comment and pseudocode.

README.txt

This file must be a plain text (.txt) file. For full credit, it should contain your first and last name (as it appears on Canvas) and your CruzID. Your answers to the questions should total at least 150 words. Your README should adhere to the following template:

```
------------------------
Lab 3: MIPS!
CMPE 012 Winter 2019

Last Name, First Name
CruzID
-------------------------

The text of your prompts are stored in the processor's memory. After
assembling your program, what is the range of addresses in which these strings
are stored?
Write the answer here.

What were the learning objectives of this lab?
Write the answer here.

Did you encounter any issues? Were there parts of this lab you found
enjoyable?
Write the answer here.

How would you redesign this lab to make it better?
Write the answer here.

What external resources did you use to complete this lab?
(Not including course materials)
Write the answer here.

Did you work with anyone on the labs? Describe the level of collaboration.
Write the answer here.
```

Grading Rubric

point values to be determined

assembles without errors
output matches the specification
- format of prompt
- format of error message
- format of final result
- final result
- print character syscall (11)
- exit syscall (10)

5 pt documentation
- 1 pt complete header comments in code and README
- 1 pt useful & sufficient comments
- 1 pt comment on register usage
- 1 pt clean visual structure / use of white space
  - Note: line up instructions, operands, and comments using spaces
    indent code from labels
- 2 pt readme file complete (should total at least 150 words)