Computer Organization Lab 1

教授:蔡文錦
TAs:林浩君、薛乃仁、許承壹

Table of Contents

- Objectives
- Example
- Tasks
- Report
- MARS installation
- Grading Policy
- Submission
- References

Objectives

In this lab, we are going to learn how to write assembly code of MIPS architecture, and understand the difference between it and high-level languages such as C/C++. We will give an example to you.

1. Learn how to write **assembly code** and understand how it works
2. Learn how to test your code using a **MIPS simulator** —Mars

Example: Computing Factorial

This is an example about computing factorial n!, where n is a given positive integer.

The attached files **factorial.c** and **factorial.s** are the given example code: the .s file is written in assembly and the .c file is the corresponded C code.

Please check the above files before moving on to the tasks, for you to get familiar with assembly code.

Tasks

For each task, the corresponding C code is given. And each task account for 20% of your score. Please don't worry about the test cases, we will not design an extreme one.

1. Prime Number
2. Simple Calculator Simulation
3. Drawing Triangles
4. Fibonacci Sequence

Task 1. Prime Number(20%)

We want you to write a program that can determine whether a given input number is a prime number or not.

Input: a positive integer n.

Examples:

```
Please input a number: 7
It's a prime
```

```
Please input a number: 9
It's not a prime
```

Task 2. Simple Calculator Simulation(20%)

We want you to write a pretty simple calculator that can do addition、 subtraction and multiplication on two given numbers and show the result.

Input: Please check the provided C code and the below example, which is clear enough.

Example:

```
Please enter option (1: add, 2: sub, 3: mul): 1
Please enter the first number: 5
Please enter the second number: 7
The calculation result is: 12
```

Task 3. Drawing Triangles(20%)

In this task, you are going to draw triangles facing up and upside down.

Input: a number of 1 or 2, which is the type of the triangle, and a number n for the layers of the triangle.

Examples:

```
Please enter option (1: triangle, 2: inverted triangle): 1
Please input a triangle size: 3
  *
 ***
*****
```

```
Please enter option (1: triangle, 2: inverted triangle): 2
Please input a triangle size: 3
*****
 ***
  *
```

Task 4. Fibonacci Sequence(20%)

Read a number from standard input. Output the n-th item of Fibonacci sequence. Students are **required to use recursive function** to solve this task.

Input: a positive integer n.

Example:

```
Please input a number: 10
The result of fibonacci(n) is 55
```

Report(20%)

There's no restriction about the report format, but please answer the following questions:

1. What is the usage of $zero? What happens if you execute

addi $zero, $zero, 5 ? (5%)

2. How to use the stack to ensure that the value of each register is correctly saved when executing a recursive function? (5%)
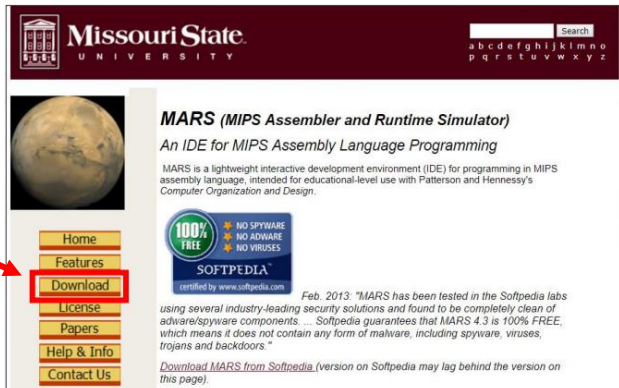
3. What was the most challenging part for you in this homework? (10%)

MARS Installation

We are going to use MARS to execute your assembly code, which is a MIPS simulator.

Step 1. Download MARS from this website

http://courses.missouristate.edu/KenVollmar/mars/

MARS Installation

Step 2. Execute your assembly code on MARS:

     1.File ->New

     2.Write MIPS code

     3.Run -> Assemble (F3)

     4.Run -> Go (F5)

Note: If the text on MARS is unreadably small, you may need to update your java version.

Grading Policy

- Each task have 5 hidden testcases, and you will get **4 points** for each correct testcase, totally 20 points for each task.
- We will use **diff** for task 3 (Drawing Triangles) and **diff -w** (ignore blank and newline) for other tasks to test your program. Any incorrect formatting may result in deduction of points.
- For all the tasks, test cases and results **will not overflow 32-bit registers**.
- Late submission will have **20%** penalty per day. The submission will no longer be accepted **three days** after deadline.
- **Any assignment work by fraud will get a zero point.**

Submission

- The files you should hand in include:
  - **triangle.s**
  - **calculator.s**
  - **prime.s**
  - **Fibonacci.s**
  - **HW1_{studentID}.pdf(e.g. HW1_0811510.pdf)**
- Compress the above file into one zip file, and name your zip file as **HW1_{studentID}.zip (e.g. HW1_0811510.zip)**
- **Wrong format will have 10% penalty.**

```
HW1_studentID.zip
├── HW1_studentID.pdf
├── prime.s
├── calculator.s
├── triangle.s
└── fibonacci.s
```

References

MIPS Instruction Set Quick Reference