COE 308 - Fall 2011
Computer Architecture

Writing, Testing, and Analyzing MIPS Assembly Code

Due Thursday, October 20 by Midnight

Objectives

- Using a MIPS simulator such as MARS
- Writing and debugging MIPS assembly language code
- Writing procedures and using system calls for input and output

Problem 1: Generating Prime Numbers

Write and test a MIPS assembly language program to compute and print the first *n* prime numbers. A number *x* is prime if no number except 1 and *x* divides it evenly. A sample run should look as follows. Limit the input *n* to a maximum of 1000 prime numbers.

```
How many Prime Numbers: 10
Generated Prime Numbers:
2, 3, 5, 7, 11, 13, 17, 19, 23, 29
```

Store the generated prime numbers in an array before displaying them. Your program should be well documented and divided into procedures.

Problem2: Reading a text file and sorting integers in descending order

Write and test a MIPS assembly language program to read a text file containing only decimal integers and sort them in descending order. The program should do the following:

- Open a text file and read its content into an array of characters. The array should be limited to 1000 characters. MARS provides the system calls for opening and reading from a text file.
- Traverse the array character by character. Convert each decimal string into binary. A decimal string consists of one or multiple decimal characters. It should terminate by white space or a newline character. Ignore and skip all other characters. Store all the decimal integers into an array of words. The size of the integer array should be limited to 100 words.
- Sort the integer array in descending order.
- Display the sorted array.

Your program should be well divided into procedures. A sample run is shown below:

```
Enter input text filename: input.txt
Sorted Array
1267
1209
130
5
```

Tool

Use the MARS tool to write, execute, and test your code. To get started, familiarize yourself with the MARS MIPS simulator. You should familiarize yourself with the assembly language syntax and system calls. The MARS Help provides a description of all the system calls that are needed to complete this project. It also provides a list of all the basic and pseudo instructions.

Submission Guidelines

All submissions will be done through WebCT.

Submit the source code of the program. Make sure that your program is well documented.

Grading Policy

The grade will be divided according to the following components:

- Correctness of code: program works properly and produces correct results
- Design and Coding: program is well designed and divided into procedures
- Documentation of code: program is well documented

Late policy

The project should be submitted on the due date by midnight. Late projects are accepted, but will be penalized 5% for each late day and for a maximum of 2 late days (or 10%). Projects submitted after the cutoff date will not be accepted.