Microsoft Calculator Simulation

1. Problem Statement

The assigned task was to design a MIPS assembly language program that reads value from a file. The file is requested and read from file (.txt) and then opens the file, reads the contents which are ascii values, and then converts to internal, binary, storage. The value is then printed out. The operation supported are (+, -, \*, %, /, =, s, r, z)

1. Approach to Solution

By using MIPS assembly language in a simulator called, MIPS Assembler and Runtime Simulator (MARS). First step was to create the design of how the project is going to be created.

The design was outlined to read file name from the user with the .txt extension attached to the file, since they were all going to read the .txt file. This helped set the *.data* section of the code to start, and nothing else needed to be outlined for that. The *.text* design was simply to have the input statement stored in a register, then to load the string value into an argument, and convert the values read from file into register than covert those register by moving or multiplication. The *syscall* command is then used to execute the code. Using a loop to convert the string and open the file.

This proved to be difficult as values from file were being converted but not being displayed on the console. After many days of testing, the issue remained the same, and the program wouldn’t properly work.

In order to test this program, a text file was created in the same directory with a randomly generated hexadecimal values were listed inside, then followed to the system messages to see if there were any errors generated in the code. However, the program proved difficult even after design, and after multiple tries, and iteration, the program failed to compile properly.

1. Solution

The requested file name, and read, and converted but couldn’t display converted calculations. The task was designed and implemented however testing the program resulted in no result. By using the directions provided as guideline. It was a bit tedious, and outline was somewhat ambiguous. The file was to open, read the contents, and then write out the contents of the file to the console.

In this solution:

1. A Text file name has to be entered in by the User.
2. The file name must have a .txt extension which is then compared with the documents in the same directory as the .asm file.
3. That string is then stored into the register. (Code will be in .text)
4. The *la* command loads the value that was stored in the register, in this case file name, into an argument that will get read and printed to console.
5. The file is then loaded into register and then converted from ascii value to decimal value.
6. The system call command executes the display of the text file onto the console.
7. Test run failed.
8. Program then terminates