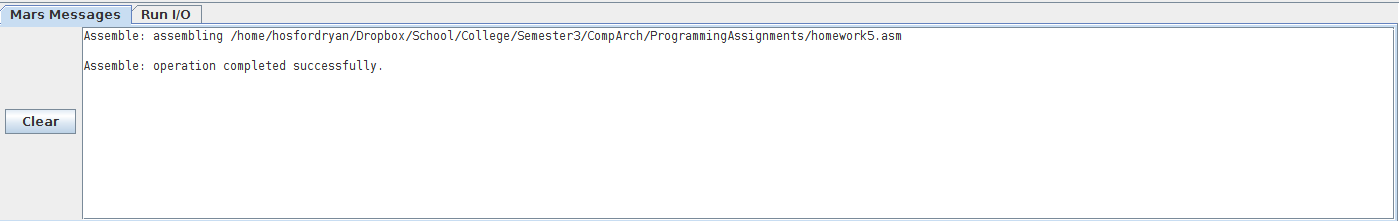
Programming Assignment 5 Report

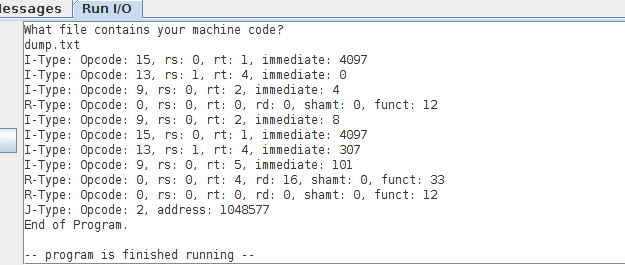
For this program, I was tasked with taking binary machine code as input and then for each instruction found in that input, outputting what kind of instruction it was (R-Type, I-Type, or J-Type) along with the corresponding register or immediate values. The program is important because parsing machine code input is one of the main portions of the project we are doing next which is a disassembler.

The approach I used to complete this assignment was to parse the first 6 bits of the machine code to determine the OpCode for that instruction. With the OpCode, you are able to figure out which kind of instruction it is, and from there can parse specific numbers of bits for the corresponding registers and immediate values. All binary values were converted into decimal values for readability in the output. The language used was MIPS Assembly and was programmed in Mars IDE.

There are some special requirements to build or execute this program such as the input file has to be saved by a windows machine using the CRLF line terminators, and the filename given by the user can’t be greater than 100 characters long and the file itself can’t be greater than 1000000 bytes long. Use the default settings in Mars IDE for building and executing.



Here is a screenshot showing that the program builds successfully



Here you can see the execution of the program. The user is prompted for the name of the file to read, then the program outputs the type of instruction found along with it’s corresponding register and immediate values.