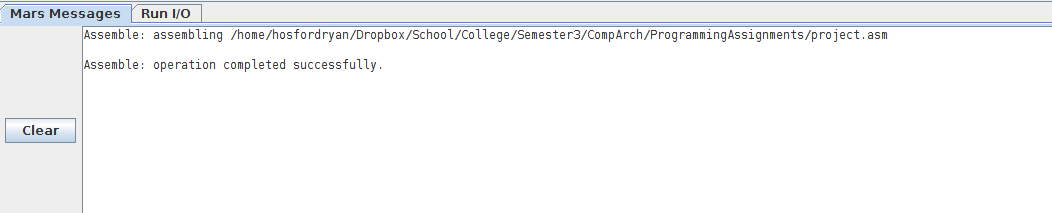
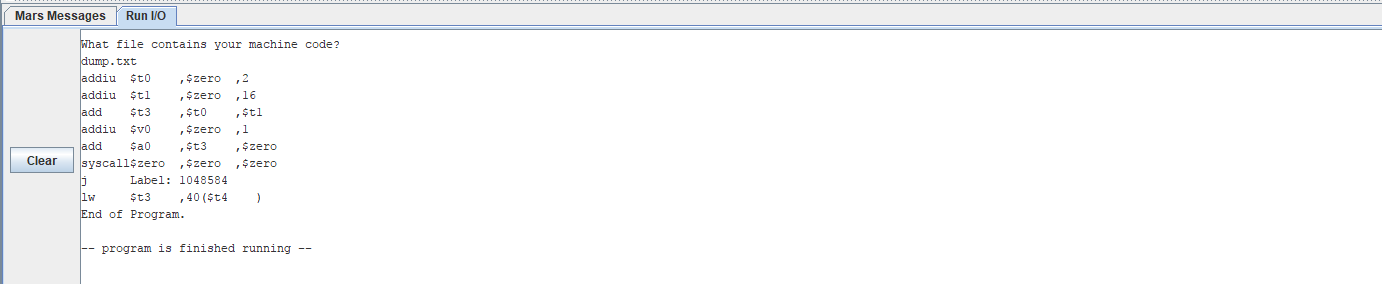
Final Project Report

For this project, I was tasked with making a disassembler for MIPS Machine Language code. The program needed to take binary machine code as input and then output the instructions that correspond to the binary machine code. This project is important because it demonstrates and forces a deeper understanding to how Assembly code gets translated down into machine code to be run.

The approach that I used to complete this project was to parse the first 6 bits of the machine code to determine the OpCode for that instruction. From there, I was able to determine which type of instruction it was, so I could look up that OpCode in the appropriate table of instructions. From there, I would parse the rest of the line to get the corresponding register and immediate values. After I had the register values, I looked up those values in a table to get the full name of those registers. After I had all the pieces for the instruction, I outputted everything, and then repeated this process until I had reached the end of the file. The language used was MIPS Assembly and was programmed in the Mars IDE.

There are a few special requirements to build and 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 cannot be greater than 100 characters long. For this project, I reused some code I wrote for a previous assignment for the parsing of the machine code to determine the type of instruction and the numerical values of the registers.

 Here is a screenshot showing that the program builds without error.

 Here is a screenshot showing the execution of the program. It prompts the user for the input file and then outputs the assembly code. I used an input file that shows the use of all 3 kind of instruction in the MIPS Assembly Language.