Jeremiah Ramilo

CS 3340.002

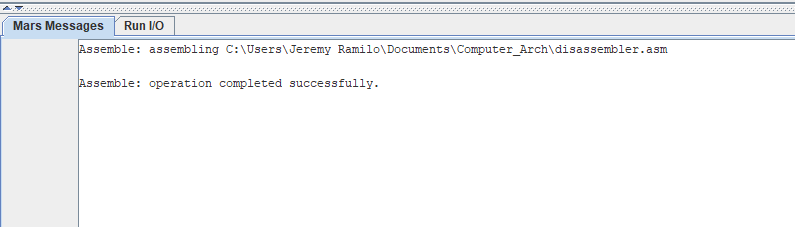
12/11/17

Disassembler Report

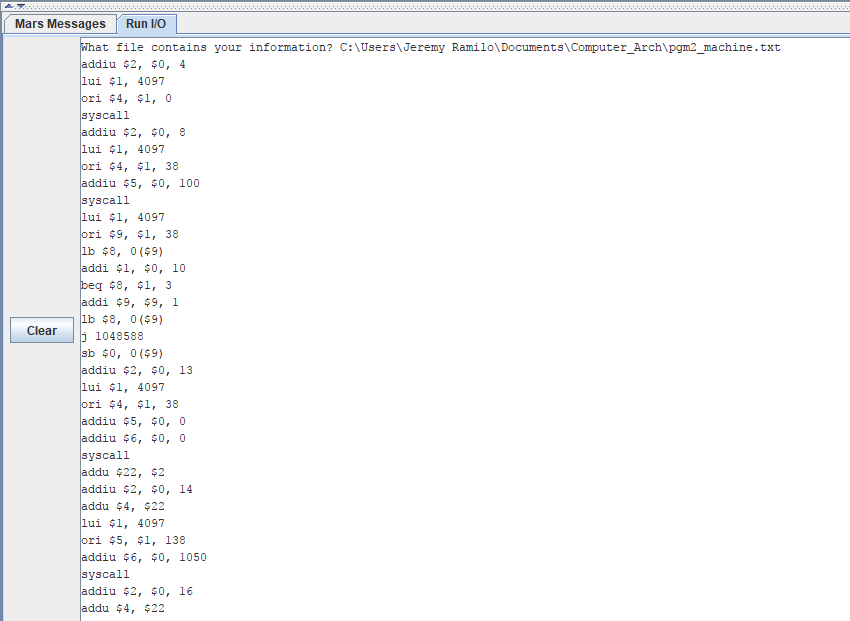
The disassembler required my program to read in any binary text file and convert it back into MIPS assembly language. The importance of this program was to take everything we have learned over the semester and bring it all together. This program required me to read in a text file of arbitrary length, convert binary to decimal, check opcodes, and print out the resulting disassembled assembly language.

My design is simple in thought, but a little complex when you look at the code. I start by reading in the text file and taking the first six bits (opcode). One I have the opcode I check which type it is and proceed to read in all the corresponding instructions based on its instruction format. Once all of that info is gathered, I will then print out the disassembled code based on the opcode, function, and other parts of the instruction format. Once it is printed out, I will check if there is another line of text in the binary text file. If there is another line, I will loop back to the top. If not, I will terminate the program.

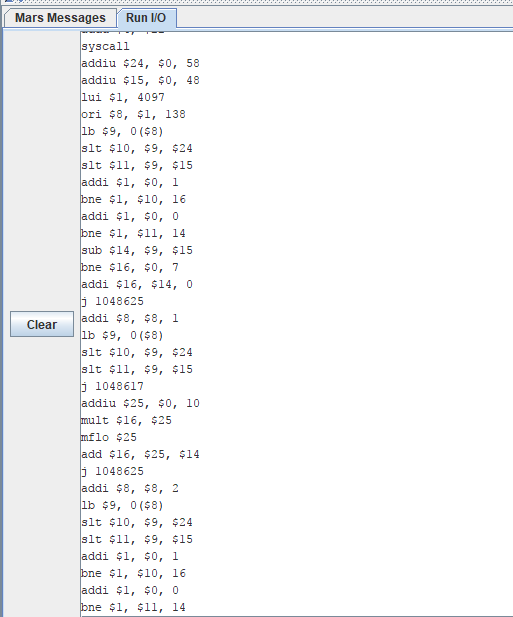
In order to execute my program a full file path has to be given. In order to move from one line of text to another I incremented the bytes by 2 (line 992). I reused code from the previous assignments when I had to read in the text file, convert the opcode, read all the instructions from the instruction format, and check if the text file is finished. I wrote new code in order to check the translations of the opcode and function types as well as when I had to print out all the disassembled code.



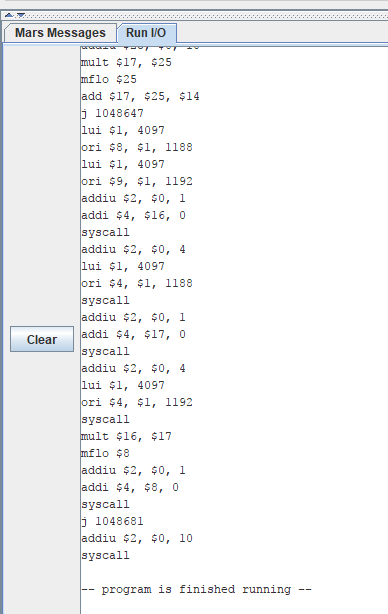
Picture of successful assembled code



Picture of code execution with a full file path given



Middle part of execution



End of execution