Design/Implementation

I started off reading the file and allocating memory given by the .size directive. Then I moved onto the other directives to populate this memory, or get the data and addresses. Execute runs all the machine instructions based on opcodes. Flags for instructions that need to be set are set in setFlags; cases include when the result is negative, zero, or overflow. printProgramStatus prints the status of the program and informs us of the success or failure in running the program. The other 3 methods are mainly used in execute. calcRegisters is used to get the values in memory at regA, source register, and regB, destination register, given the byte provided. readStoredInt is used to find a 32-bit int stored at some memory location. While storeInt is used to store a 32-bit int at some memory location. These 3 methods are the key in manipulation of values for our instructions.

Challenges

At first, writing this program seemed very overwhelming, but after reading up on how the instructions and program as a whole worked, it didn't seem so difficult. Unfortunately, there aren't a lot of resources about movsbl and cmpl, so I'm not completely sure they're implemented entirely. However, prog1 and prog2 ran successfully.

Extra Credit

Prog3.y86 is a simple program that multiplies two numbers, 10 and 8, and prints the result, 80.