

Jesse-James Black

Russell

12/8/16

### Y86 Emulator

#### Extra Credit attempted: Personal program, prog3.y86

#### Implementation

My implementation for the y86 emulator was essentially a large switch statement with different instructions needed to perform operations on the memory. I first started by loading the program into memory, I then sent the memory to the instruction set function which then began to operate on it.

I use fopen to open the file and fgetc to get each individual char and load it into memory tokenizing it for \n\t\r while also making sure the relevant directives are valid. After this the execute function is called. This is where all the instructions are stored. In making these I saw that mrmovl and movsbl are very similar and that cmpl and subl can be very similar as well. While these instructions are executing checks for program status and flags are checked and printed if an error is found.

#### Challenges

Initially this program was very hard conceptually. After thinking about it and breaking it down it became much easier. I decide to emulate it in the fetch decode execute cycle. Where decode and execute were essentially the same function. This made it much more easy to conceptualize and program.

Overall I think this program was much easier then I initially thought it would be. It really helped to solidify my skills as a C programmer.

#### Extra Credit

Prog3.y86 prints a simple message to the terminal in color.