Vladislav Tsoy
Brian Russell
CS211
Y86emulator READ ME

Design and implementation:

The y86emul.c program takes in and processes a .y86 text file. The program parses through the .y86 file token by token and allocates the memory array according to .size directives. According to the directive, the program takes the index and stores the directive information in memory. After the all the directives have been writes, the program then reads the .text instructions. According to the opcode is executes the 15 operations (nop, halt, rrmovl, irmol, rmmovl, mrmovl, OP1, jXX, call, push, return, pop, read, write, movsbl). Once all the instructions have been read and executed the program stops and outputs the results.

Challenges:

Some of the challenges I have ran into was reading the text file and writing the directives and their instructions into memory. Spent a lot of time trying to resolve segmentation faults. Another issue I had was figuring out the call, return, pop, pop, and push functions and how they the memory and registers were rewritten. Again, dealing with a few segmentation faults.