Computer Architecture Summer 2018
Masood Karimi
Professor Brian Russell

Synopsis:

y86emul is a program that takes a y86 input file and executes it in C.

y86dis is a program that takes a y86 input file and prints out the corresponding assembly code.

y86emul and y86dis Implementation and Design:

y86emul has a step-by-step design. The first thing this program does is parse the directives. After parsing the directives, it executes the instructions (which was parsed in the .text directive). While the code was tedious to write, it is very simple in its application. There are 15 main instructions and these are identified by the low opcode byte. Some instructions have a subset of actions, such as *op1, jmp, readX,* and *writeX*. The way that the program identifies which sub-instruction to carry out is through a secondary opcode, which is stored in the high field. After the instruction is identified, the program carries out the instruction and continues the cycle until it completes.

y86dis follows the same design as y86emul. The difference here is that instead of carrying out the instructions we print out the assembly code and then move onto the next provided instruction (we are just reading essentially and not doing anything with the instruction).

Overall Analysis:

The overall run time, or Big-O, of this program would be based upon the amount of directives and inputted instructions. Because of this I believe the Big-O runtime of this program is O(n) + O(m) where n is amount of directives we are interpreting and m is amount of instructions we carry out after parsing the directives.