

Kartik Saddi

Y86 Emulator and Y86 Disassembler

Y86 instructions are different from Intel x86 instructions. This program designs an emulator that can run Y86 instructions on an Intel machine. There are 2 input files included: prog1.y86 and prog2.y86 each which contain different instructions.

I compiled my program with GCC compiler, running `gcc -o y86emul y86emul.c`.

After being compiled, this program can be run with the command: `y86emul nameofinputfile` (Name of input file will be either prog1.y86 or prog2.y86.)

Each input file contains lines with what the line contains as the first word; ie the input file will have a line that starts with .text which will have the machine instructions. There will also be .size that specifies total memory of entire program, and .byte that specifies a location in memory and the value at that location., as well as others.

My program takes the input file, and parses through using `fscanf` to find out which directive we are looking at. When the machine instructions are found, I run the method `execute` that converts machine code to their respective assembly instruction and changes registers and values in memory as needed for each instruction. Memory is represented as a large char array in my program.

I have also included a disassembler called `y86dis.c`. After compiling, run with: `y86dis nameofinputfile`. (Do not include the period). This program takes all of the machine code, and prints out each assembly instruction (such as `rrmovl`) and the address in memory of that instruction.