README Assignment4

Azizur Rahman

April 26, 2016

y86emul:

The program is able to read a text file and allocate space in a memory array corresponding to the given directives. The memory array is of type char and each character in the array is allocated to one cell in the array. I then use a while loop and switch statement to go through every element in the array starting at the proper index given to me by the .text directive. Through this, I am able to execute the desired opcodes and increment a program counter accordingly that keeps track of the location that I am at currently. To overcome many of the obstacles I encountered, I wrote down the machine code and divided them up into sections to understand exactly what is happening in both prog1 and prog2.

Challenges:

- Storing each character in the memory array character by character which lead to complications dealing with jumps and addresses
- Displacements and addresses have to be reversed and converted to decimal to take into account little endian
- Was not sure when to cast the memory array as an unsigned char
- Readx and writex being unclear as to the input they take in and where to store them

y86dis:

This program prints the address of the y86 instruction, followed by the equivalent x86 assembly instructions, with the corresponding y86 machine code given. It follows the algorithm y86emul has to traverse the memory array but instead of executing the commands, this program prints.

happy.y86.txt

Run this with y86emul.c for a picture ☺.