Y86 Emulator

Avijeet Sachdev (as2221) - Computer Architecture

- This program is essentially an emulator to the x86 assembly language and follows the main principle and flow of fetch, decode, and execute.
- Right in the beginning of the program of y86emul.c are the movl operations that are rrmovl, irmovl, rmmovl, and mrmovl. The four operations are register-to-register move operations, immediate-to-register operations, register-to-memory operations, and memory-to-register operations, respectively. They are fairly straightforward when moving around the `reg` array, *numPtr which holds values for what to move, and *memArrAddress which is the int pointer to the memory array address.
- Afterwards, comes the op1() function, which holds the logic for case 0-4 which are addl (addition), subl (subtracting), andl (bitwise AND), xor(exclusive-OR), and mull (multiplication), respectively. All that is written describes the conditionals and register moves to mimic the assembly operations' functionality.
- jXX(), the next function, handles all of the jumps involved with assembly programming. For this emulator, we deal with the cases 0-6 with jmp, jle, jl, je, jne, jge, and jg conditions. They follow the specific conditions of the given jump type, with the given SF (sign flag), OF (overflow flag), and ZF (zero flag) flags to handle them.
- call() and ret() are also fairly straightforward in that they move the ESP (reg[4]) either forward and back 4 positions.
- pushl() and popl() handle pushing and popping off of the stack, where they deal with the lower register and move ESP 4 forward or backward after, as well.

- Lastly, the readL() and writeL() tie everything together with working with the given byte. The method will check if it is valid then set charCurrByte to be the character prepared to be printed out and keep moving the EIP forward to keep printing.
- The provided prog1.y86 will print out:
    - Congratulations. You have made it this far.
      This first program was all of 10 Y86 machine instructions, which I apparently did not screw up.
      Susan sends her best wishes to you all.
- The provided prog2.y86 will serve as a factorial calculator that can print factorials up to 12! because 2^{31} < 13!.