

# Emulator

- My implementation focuses on a processor struct that contains all the registers and the program counter to go through the bytes.
- Once the file is read (fetched), the input is passed to the function (decode) to check the input line by line and recognize the directives.
- Once the input was decoded, the bytes saved will be passed to the function (execute) in order to do the operations recognized by the OPCODE of the instructions from the .text directive and doing the regular fetch-decode-execute process inside.

# Disassembler

- The program uses most of the code from y86emul.c with slight modification to print out the instructions in assembly language.

# Challenges

- I have faced many segmentation faults during my implementation and I couldn't find where my problem is exactly.
- I have overcome this problem with prog1. Still debugging prog2.
- However, you can see how the algorithm works