cs281: Introduction to Computer Systems

Assembly Exercise with Stack Pointers

The purpose of this exercise is to continue to familiarize ourselves with DDD, the x86 ISA, and specifically the basic stack operations.

1. Obtain the `stack.s` file from the class webpage. This is a ”blank template” that contains a barebones main program. The only instructions in main are to set up the stack frame for the main function call. You will leave these alone and write some code in the middle.

2. Open `stack.s` in an editor.

3. Compile it with `gcc -m32 -g -O0 stack.s`

4. You will add three instructions in your file to push three integers on the stack. You should push 1, 2, and 3 in that order. Make each push operation a full double word (32 bit push). Use the immediate addressing mode with your push operations.

5. You will add three instructions to then pop these values off the stack. Put them in register edx as you pop them.

6. In between you want to add some code that will access and change these values. Specifically you are to fetch from the stack the top two values, add them together and write the result in the 3rd slot in the stack (adding 3+2 and storing the result overtop the 1). Do not use push/pop – you do not want to change the stack. Instead use the indirect addressing mode (with base/displacement) to access these stack locations.

7. Start ddd with `ddd a.out &`

8. Set a breakpoint at main.

9. Run the program by stepping through your instructions. Draw a memory model of the stack and record the values that are stored there while your program is running. Also keep track of the various register values. Use the register display and memory display in your debugging.