Name: SGT Schoenwald-Oberbeck, Jesse
Date: 19MAY2017
Current Module: Assembly
Project Name: Fibonacci

Project Goals:
Build a program in assembly which returns the nth Fibonacci number.

Considerations:
o A fibonacci generator must be built.
o User input must be taken.
o Print functions must be called/used.
o The stack must be managed.
o Assembly...

Initial Design:
A number representing an index in the sequence of Fibonacci numbers is taken from the user. This number is used as a counter, and the Fibonacci sequence is run in order until the counter reaches zero, and the current number, which matches up with the user's choice, is printed.

Data Flow:
Arguments can be taken via the command line at program launch, or just afterward if no command is supplied. Invalid commands shall be treated as zero. Input goes through checks to catch inputs of 1, 0, or less than 0. These are passed to proper exit channels. Otherwise the generator begins as previously described, and the result is printed via printf.

Communication Protocol:
None.

Potential Pitfalls:
o Stack alignment.
o Assembly.

Test Plan:

User Test:

Multiple runs of the program, using every variation/combination of options and inputs the user can think of.

Test Cases:

All test cases completed with correct output.

Conclusion:
Assembly is simultaneously not as bad as I would have thought prior to this week's lessons, and a bit of a nightmare. A limited number of 64 bit registers that have to be juggled is a great

deal more difficult than programmer defined variables of any number or type, and making space for things that won't work with/in a register is a chore. However, if and when I have to build a tiny payload, or reverse engineer some compiled code this exposure will be immensely valuable.