Name: Stephen Paradis
Date: May 6, 2015
Current Module: Operating Systems
Project Name: Signaler

Project Goals:

The goal of this project was to generate prime numbers and use signals to manipulate the stream of printed numbers. There were specific signals we were asked to capture and each signal changed how the numbers were printing. There were also command line arguments needed to modify how the counting of numbers starts.

Considerations:

- How to capture the signals using 'signal' or 'sigaction'
- What algorithm I would use for the prime number generator
- What signals need to be captured for the program
- Take consideration of what the command line arguments need to do

Initial Design:

This program is structured in a functional way, there are no structures used in this program. There is a prime number algorithm that checks each number to check if the given number is prime or not. I used the signals.h header to capture my signals using sigaction. Along with other various header files needed for printing or conversions needed for optargs.

Data Flow:

The flow of the program starts with the make sure the signals are prepared to be caught. Once that is done I generate and check if my numbers are prime. If they are prime then they get printed to the screen. Depending on what signal is caught the prime number will get affected.

Potential Pitfalls:

- There is a possibility to get an algorithm for generating prime numbers that would give incorrect output.
- Not handling the signals I have gotten correctly.
- Depending on how you generate the prime numbers, there could be too much memory allocated.

Test Plan:

User Test:

- 1.Tested program with no flags
- 2.Tested program with the -s flag
- 3.Tested program with the -r flag

- 4.Tested program with the -e flag
- 5.Tested program when I used kill -HUP
- 6.Tested program when I used kill -USR1
- 7.Tested program when i used kill -USR2

Automated Test:

- Not available.

Test Cases:

- 1.Ran the signaler program just by itself
- 2.Ran the signaler program with the -s flag to specify the next prime number greater than n
- 3.Ran the signaler program with the -r flag to start the number decreasing from n
- 4.Ran the signaler program with the -e flag to exit the program if a prime number greater than n would be printed
- 5.While the program was running ran kill -HUP in a separate terminal tab.
- 6.While the program was running ran kill -USR1 in a separate terminal tab.
- 7.While the program was running ran kill -USR2 in a separate terminal tab.

Expected Result:

For this program each test case was run to see if they would affect the prime numbers printing. The expected result of the 3 command line arguments would be to change what number ‘n’ the program starts at. For the kill -HUP command the expected result is that the prime numbers restart. Also for the kill -USR1 the expected result would be that the prime numbers jump one. For the final signal kill -USR2 the expected result is that prime numbers start counting in reverse order.

Conclusion:

The project overall seems to be working well, with both the signals and command line input working as needed. The optargs and sigaction worked well with what was needed for this project. For future improvements I would update the prime number generator to a better algorithm.