Name: Sean Bartholomew
Date: 20160514
Current Module: Operating Systems
Project Name: Monitor

Project Goals:

The project is designed to model client server relationships. The dispatcher is to send user input to multiple instances of listener running on the same machine possibly in different locations, and from different users.

Considerations:

How to process signals
How to communicate between client and server.
How much memory will the program have access to.
How to ensure different users have necessary permissions.

Initial Design:

The project requires two programs listener.c and dispatcher.c Dispatcher will function as the server, accepting and maintaining socket connections with instances of listener. Listener will act as a client, attempting to connect to the server and simply echoing everything the server puts in the socket.

Data Flow:

The programs can be run separately, however listener will exit if dispatcher is not currently running. Dispatcher begins by creating a socket in my home directory on the machine. (The requirement is to run on the class VM) It then enters a forever loop and waits for notifications from open socket connections. If it receives a connection from a previously unknown socket it will save the connection to send data to. Listener calls out looking for the socket in my home directory created by dispatcher. If it is not found Listener will exit. If it cannot connect listener will exit. If successful at connecting Listener will enter a forever loop and simply print everything to the screen that is written to the socket.

Communication Protocol:

CMD line options:

None required.

Signals handled:

EOT will close the listener.

Potential Pitfalls:

Handling multiple connections, and ensuring that all connections receive the data.

Test Plan:

User Test / Test Cases:

Input data larger than buffer
Interrupts – ctrl-c ctrl-d etc
Non ASKI characters
Run listener and dispatcher in different directories
Run multiple listeners
Exit a listener while dispatcher is running
Exiting dispatcher while listeners are running
Input in a listener
Start listener without dispatcher running
Input in dispatcher without any listeners
Run dispatcher and listener by different users
Listener should Exit on ctrl-c
Run valgrind.

Expected Result:

I expect the behavior outlined in the initial design to be appropriately implemented. Additionally I expect that the program will have no memory leaks. The program will not seg-fault.

Conclusion:

The project functions as intended. There were a lesson that were learned in the course of the project. First is that when handling multiple connections to a socket, the connections must be stored somehow and each called individually. In order to facilitate multiple users, the socket must be accessible and have appropriate permissions, it also must be specified as an absolute path. (This is not the best security practice.) In the future I would probably implement the socket over IP instead.