ASSIGNMENT 3 REPORT

Ajiitesh Gupta
1562557

Objectives:

This programming assignment is intended to give experience in developing client-server programs that utilize TCP signals for communication over the internet, FIFOs and Sockets for communication over peers, and I/O multiplexing for nonblocking I/O.

How to use:

Makefile commands (assignment spec):

- Executing 'make' produces the a3w22 executable file.
- Executing 'make clean' removes unneeded files produced in compilation. - Executing 'make tar' produces the 'submit.tar.gz' archive.

Running the program:

The program can be invoked as a master switch using:

"./a3w22 master nSwitch portNumber"
where master is a master switch and nSwitch specifies the number of switches in the network (at most MAX NSW= 7), and portNumber specifies a port number starting with 9xxx, where xxx is the last 3 digits of your school ID, so 9557 for me.

The program can also be invoked as a packet switch using:

"./a3w22 pswi dataFile (null|pswj) (null|pswk) IPlow-IPhigh serverAddress portNumber"
where the program simulates switch pswi by processing data read from file dataFile. Port1 and port2 of pswi are connected to packet switches pswj and pswk, respectively. Either, or both, of these two switches may be null.
Switch pswi handles traffic from hosts in the IP range [IPlow-IPhigh].
Each IP address is ≤ MAXIP (= 1000) and >= 0.
Sequence matters. The master should be set up first and should not be closed before the packet switches.

Design Overview:

In general, C++ code is used by me, and I made small functions that do one thing each to increase understandability of the functions. Code is also separated into separate files to decrease recompile time and create logical code separation.

1. *main.h*: Header file which contains the server, the client, and the rules class.
2. *main.cpp*: Main file which compiles all the header, and cpp files to finally execute my code.
3. *message.h*: Header file which contains the messages such as composeAdd, composeHello, etc.
4. *Makefile*: The file which makes the code into an executable, having the option of also making a compressed file in the tar.gz form, and also has the ability to delete any unneeded files.

Ajiitesh Gupta
1562557

Project Status:

All functionality delivered to match example test files and specification. I worked on this assignment from scratch because my assignment 2 submission had gone corrupt, not only on eclass, but my local system as well, and since this is a continuation of assignment 2, I had to start this assignment from scratch.

Testing and Results:

The programs were compared with the example inputs and the outputs were equivalent.

Acknowledgments:

New: How to check if a socket was closed: http://www.stefan.buettcher.org/cs/conn_closed.html

What libraries do I need for the commands I'm thinking of/ poll and mkfifo implementation help: https://linux.die.net/

How to read a file: http://www.cplusplus.com/reference/fstream/ifstream/open/

Sample code taken from the lab codes and the lab sessions

How to use poll guide:
http://www.unixguide.net/unix/programming/2.1.2.shtml

String token iteration:
https://stackoverflow.com/questions/236129/how-do-i-iterate-over-the-words-of-a-string