Niroojen Thambimuthu(10153928)

CPSC 441 Assignment 2 - Not completed

How to compile: - server: gcc udp_server.c
- client: gcc udp_client.c

Both server and client are on seperate files. Test files are in server directory folder

How to run: - server: ./a.out
- client: ./a.out “textfile”.txt Eg: ./a.out 736.txt

How to use: Run server, and run client with Input text file as command line argument. It will put received data into a file within the client folder directory. I deleted existing file within the client folder during testing.

For this assignment, I used the tutorial udp transfer files and the read_write program as a base for the implementation of this assignment. My application is of a client-server implementation and I also used a push-based solution to deal with this assignment. The server will be running indefinitely until a client response has been capture and once it has a response, it will work its ways to send back all octolegs to the client who requested it. I used some form of acknowledgment for testing a text file, however I did not implementation other acknowledgment factors. I couldn’t implement any reliability features into my udp implement because my main point of focus is to make my octoblock work. For all udp implementations within the code, it was mainly harvested from the example udp client and server posted in the tutorial slides. I managed to isolate which coding sends data and which receives data, thus I was able to implement the required action whenever I wanted it. The udp implementations are adequate, however I'm not sure of its reliability.During university time and at home, I used my mac laptop using the Xcode application, and the terminal for testing. I would use 2 terminal windows to simulate both server and client. Honestly, I did start slightly earlier, but I did put many hours trying to understand and debug my code. I had to restart my octoblock algorithm a few times because I was becoming less and less productive. What doesn’t work is the micro octoblock implementation and test text file with at most 2 full octoblock sets (more than 8888*2 bytes). However the other test cases work. So to conclude, the octoblock somewhat works, but isn’t perfected enough.