**Assignment 2: OCTOPUT TRANSFER PROTOCOL**

**How to Compile:**

1. Once client.c and server.c is downloaded to the designated folder, open 2 terminals.
2. Go to the directory in which client.c and server.c are located.
3. Type in the terminal:
   1. 1st terminal: gcc -o server server.c
   2. 2nd terminal: gcc -o client client.c
4. The server and client are now compiled.

**How to Configure:**

1. Type in the terminal:
   1. 1st terminal: ./server
   2. 2nd terminal: ./client
2. Both should say that the client and server are now running and connected.

**How to Use:**

1. The client can put 3 commands: quit, get, filename
   1. quit – terminates both client and server
   2. get – server sends to the client what files are available for transfer
   3. filename – type in a filename from the list shown in the server
2. When a file name is requested, the server will send the file and the client will save the file in a text document named newFile.txt located in the same folder where the client.c and server.c are saved.
3. If something went wrong, try restarting and compiling the file again it should work fine again.

**Indicate why Push-based or Pull-Based and justify?**

* This is a push-based server and client set-up. I believe that it should be the server sending the files and always listening to the client what the status of the communication between them is. The client sends the request and the Acks to the server and the server will do things based from what responses it had gotten from the client.

**Any special features?**

* This is able to transfer files even the 256KB test file from the server end to the client end.

**Please include a paragraph or two about your use of UDP as transport layer protocol and the implication of this choice.**

* UDP is used in this assignment so that we could achieve fast transfer of files and reduce the size and complexity of the source code as we could keep connections going on and sending files back and forth without using a lot of protocols done in a TCP version of this. Since we have been sending in an Octo-block, then the range of parameters sent over the network is about a maximum of 1111 bytes (not including the sequence number for each) in 8 octo-leg. By using UDP, we have used protocols only necessary for this assignment and reduced the size of parameters sent over the network.

**Where and How your testing was done:**

* Student Laptop using Virtual Box and Ubuntu
  + Testing done:
    - Transferred all test files including the 256kt ASCII test file
    - Tried if it could resend files that had lost segments/corrupted
      * Not so much success.

**What works:**

* Can transfer and save files on the client end with the title newFile.txt
  + Supports from 736-byte test file up to 256KB ASCII test file.

**What does not Work:**

* Use of multi-processing with timers and ack messages sent in parallel.
  + I have talked to Professor Carey, and we are not able to find out a way for the multi-processing to work for my source code and hence, he had told me to tell in the document my problems so that I could get partial marks for putting effort for the assignment as much as I could.