Programming Project

Dr. Hana Khamfroush

Due Date: <placeholder>

1 Description

It is now time to get your hands dirty with programming applications in a networked environment. To do this, you will use socket programming in order to create a client-server application that allows for the client to upload files to the server. For this project, you will implement a server program and a client program. The server program will wait for a connection from a client and upon the establishment of a connection, the server will wait for files from the client. Upon a successful file transfer, the client program will then end, however the server program will continue to run waiting for a new client to connect and send a file.

The client program will take input in as a path to a file to upload, and a network address to connect to (see below). The client should verify that the file path they have received is valid before connecting to the server, loading the file and transferring it to the server. The server will then wait to receive the file and the client will disconnect once it has finished sending.

There is a catch, though, your application **must** allow socket connections to take place over a network other than 'localhost' (127.0.0.1). Essentially, if you have two laptops (both connected to the same network), one laptop should be able to run the server program and the other one should be able to run the client program to transfer files between them. Design your code in such a way that the IP address of the server is parameterized and I can set it to my other machine's IP address for evaluating your program — this can either be through a direct change of a constant variable, command-line user input, or a GUI (specify in your README). Regardless of how you approach this detail, make sure it is well-documented.

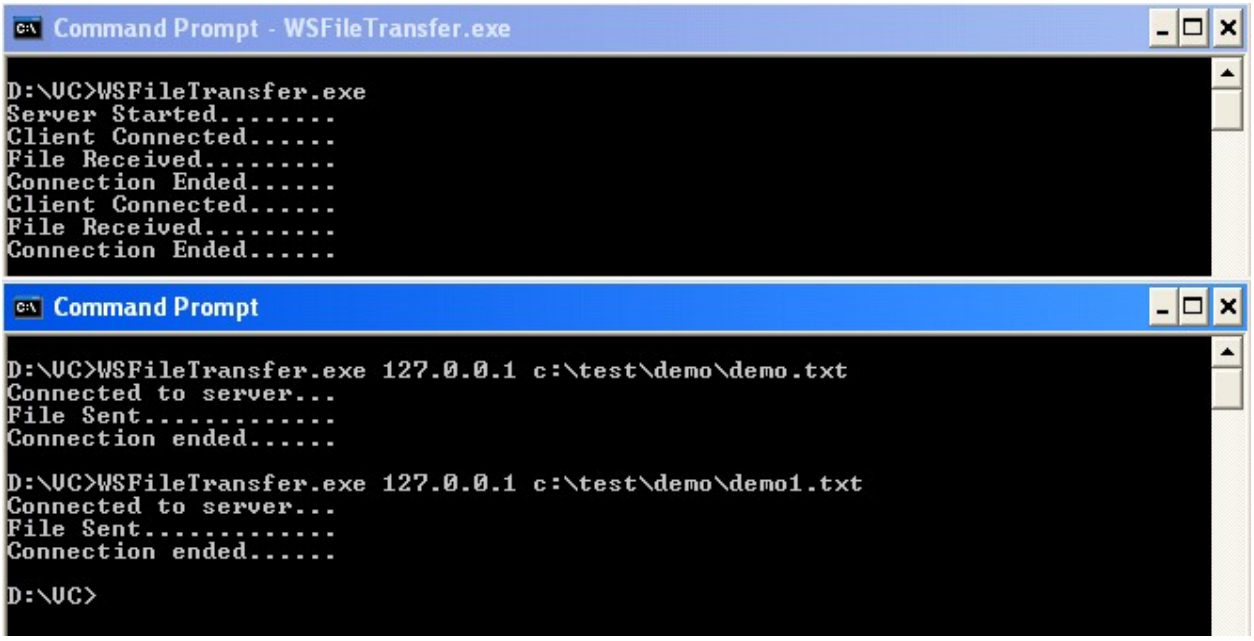
Since this is a 300-level course, well-designed code is expected. As such, you will be graded with respected to style, code structure, etc. For more information regarding what is expected in terms of code style, please refer to Section 2. Since C/C++ are reknown languages for security vulnerabilities, it should be noted that this does not need to be considered for this application. In professional contexts, you will need to consider security of socket applications, but secure software development is beyond the scope of this course. (Though, memory leaks or segmentation faults can also be considered security vulnerabilities, and these **must** be handled by your implementation.)

2 Code Style Guide

Writing code that maintains clear style is incredibly important as software engineers and computer scientists. Your code must be readable such that another developer can understand your code and meaningfully extend it without you. As such, maintain the following guidelines listed on Dr. Finkel's checklist.

3 Example

Below is a sample execution of the file transfer application, with the standard output from `client.c` and `server.c` (in addition for input for the former).



4 Requirements

For this assignment, you must have/maintain the following:

- This project is an **individual** effort.
- Application must be developed in C/C++.
- You must use the following socket framework for establishing and handling connections `<sys/socket.h>`[1].
- Your application must include at least two source code files named "client.c" and "server.c". You may develop as many additional source code files if you wish — within reason (e.g., do not have a source code file for each constant you may use).
- Your source code must be legible and maintain a cohesive/coherent code style. For more information on code style, refer to Section 2.

[1] Documentation: `http://pubs.opengroup.org/onlinepubs/7908799/xns/syssocket.h.html`

- Your application must be runnable in such a way that one physical computing device serves as the server and another physical computing device serves as a laptop. For this, you can develop your software in such a way that this is rather trivial.
- Each submission **must** be submitted as a compressed .zip or .tar file.
- You **must** include a README file that details your application, and all details needed for testing and running your application.

5 Tips

- This project is worth 25% of your final grade. Do not hold off working on this project.
- The socket framework required for this project should come pre-installed on your system. If you do not have it on your system already and have difficulty installing it, please reach out to Christian Tooley at christian.tooley@uky.edu.
- I would advise against incorporating too much of an object-oriented design for this project if you choose C++ as your language of choice.
- When you're ready to incorporate sockets into your application, there are a lot of resources to better understand to build a simple socket program. This resource is a great place to start familiarize yourself with socket programming.