**CS 4414 Operating Systems – Fall 2014**

**FTP Server**

Homework #5

Duevyn Cooke

I completed the assignment entirely. My FTP server allows remote clients to list, store, and retrieve files from the local file system.

**Problem Description**

The goal of this assignment is to implement a minimal FTP server capable of interacting with off-the-shelf ftp clients. The protocol is enacted through a pair of TCP connections: a constant control connection for the server to respond to client commands; and separate data connections accepted by the client in which data is exchanged. My server will implement the bare-bones requirements/commands set forth in Section 5.1 of the FTP specification, IETF RFC 959, with negligible exceptions.

**Approach**

The first part in developing a solution to the problem is enabling the server to accept connections from client programs in order to communicate. I took the conventional route to accomplish this task: I utilized Berkley Sockets (BSD sockets). Berkeley sockets is a computing library with an application programming interface (API) designed to allow interprocess communication. The initialization and utilization of sockets is procedural and must be done in a specific order. The server side progresses as follows:

* the socket() function returns a file descriptor for the socket and allocates system resources to it
* the bind() function associates the socket with a local port and IP address
* the listen() function causes the previously bound socket to accept incoming connections from client programs
* the accept() function, as the name implies, accepts incoming attempts to create a TCP connection to a remote client; it does so by creating a new socket

Only after this total ordering of function calls can the server program communicate with remote clients; the first objective in this assignment.

After this server initialization process, an active FTP session begins. It is important to note that sockets are equally as important in this second objective of the program as they were to the previously mentioned task. In the FTP protocol, the initial connection between the server and client is known as the control connection. My server constantly reads this connection stream for incoming FTP commands from the client. Once the server receives a command, it sends a response back to the client that informs the client of the status of its desired command. If necessary, the server then opens a new socket that connects it to the client. This new socket, known as the data connection in FTP protocol, allows the server to send file information to, or receive file information from the client.

**Results & Conclusion**

As previously stated in the opening, my server handles the rudimentary FTP commands outlined in the IETF RFC 959. Thus, my server handles the following commands: USER, QUIT, PORT, TYPE, MODE, RETR, STOR, and NOOP. My server accepts any name and password as an acceptable login combination. Also, only a single client can connect to the server at a time.

The goal of this assignment was to familiarize me with remote procedure, remote file systems, and the well-known file transfer protocol. This was more than accomplished as I went from being unfamiliar with FTP to building a primitive FTP server.

**Pledge:** On my honor as a student, I have neither given nor received unauthorized aid on this assignment—Duevyn Cooke.