Viswanathan Subramanian
Kevin Sandoval

Assignment 3: Remote Files Readme

In our implementation, We needed to define four functionality within the server portion of our code: netopen, netread, netclose, and netwrite. These functions perform equally to their counterparts (open, read, close, and write) but are made to work throughout the server/client procedure. Once we applied relevant definitions, we utilized these functions and demonstrated their usage/partiality on the client side of the assessment.
For our assignment, we chose not to provide multi-threading purposes and used one socket for multiple file connections. Every socket we had relates to a port on the server. Since a server can only really have a certain number of ports, we know there is a limit to how many files which can be opened at any given time. Now, this does raise the question: what if we have a large amount of clients open many files at once? What happens is the following: Whenever data is stored, we have to ensure that there is some technique being used. Otherwise all we are looking at can easily be misinterpreted. For example, in strings, when they are stored, we want to make sure that they have a \0 when they are read into memory. With a character array, however, this will not stored because of the buffer being a character array requiring a \0. Therefore, what is done is we treat the entire buffer as a string, meaning we take all the strings and use indexes to parse the data inputted. This process proved useful throughout most of the assignment, as each defined function sent a signal to the client, which represented its usage, would be utilized and sent back to the server.

Useful Info:
Buff is our buffer which used packets to transmit data server to and from client
Buff sends a packet of info to the client and the client confirms that packet
This program is based on the assumption the pathname/flags will never exceed 224

Our program should utilize all the given error conditions:
EACCES: Permission denied
EINTR: interrupted function call
EISDR: is a directory
ENFILE: too many open files in system
ENOENT: no such file or directory
EPERM: operation not permitted
EROFS: read-only filesystem
EWOULDBLOCK: Operation would block
EBADF: bad file descriptor
ECONNRESET: conection reset

ETIMEOUT: Connection timed out
HOST_NOT_FOUND: the specified host is unknown

Client:
-send header
-send data over buff
-receive header, checking for errors
-received data

Server:
-received header
-receive the data
-initiate system calls, checks for error, and set header
-send header and then data