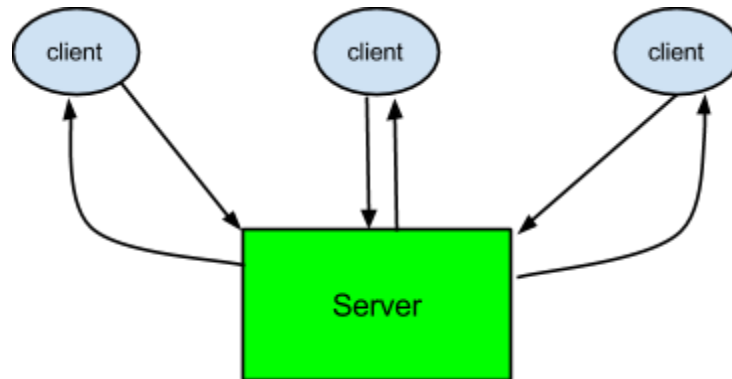


## Project 2

By: Rikin Marfatia and Stephen Pardue



This project uses a client-server model that needs the capability of handling multiple clients concurrently, while being a singular server model.

### Protocol

Client Requests:

LIST, DIFF, PULL, LEAVE

The client will have a very simple interface where they type in a letter corresponding to a request that they want to send, 'L' for list, 'D' for diff, 'P' for pull, 'E' for leave. The server receives these commands and responds appropriately, keeping track of the actions it does for each client (via console statements and logging). Logging takes into account the current action and the client's ip address that it is currently handling (ex: "Sending Diff to: 127.0.0.1").

### Implementation

There are two structs that were created to help properly implement this project: *FileInfo* and *DirectoryInfo*. *FileInfo* objects are basically the nodes of a linked list, controlled by *DirectoryInfo*, that hold important information about a file like the name, and the checksum (using MD5), which is used in file comparison like for the DIFF functionality. The *DirectoryInfo* object controls the linked list and keeps track of how many files there are in the directory.

In addition to the added structs, we made some helper c files to help with populating the structs, and also to help with transferring files. These helper files are *listdir.c* and *netfileio.c*. "listdir" helped with populating *DirectoryInfo* structs, and does a lot of the logic and calculations to find the diff between two directories. "netfileio" is responsible for handling file sending and receiving (PULL).

As far as choosing which way we wanted to implement a server that could handle multiple clients, we chose to use pthreads because they were very simple to integrate into our server code and they didn't seem to have any downsides as far as we saw

## **Runthrough**

**LIST** - Client send's 'L' command. Server receives command, populates a DirectoryInfo struct representing the server's directory and sends it to the client. Client receives the struct and traverses through it and prints the file names out with the help of listdir.c.

**DIFF** - Client sends 'D' command. Server receives command, populates a DirectoryInfo struct representing the server's directory, and sends it to the client. Client receives the struct, creates its own DirectoryInfo struct representing its local directory, and then uses a utility function in listdir.c to compare files on the server to the local directory using the checksum of the files.

**PULL** - Client sends 'P' command. Server receives command, and sends its DirectoryInfo to the client so that the diff can be determined. The diff DirectoryInfo is returned to determine which files need to be sent to the client, if any. Then the server sends the files using netfileio.c, and the client receives them in a similar fashion.

**LEAVE** - If the client type 'E', the client closes the socket, ending the connection.