The WTF client is set up simply enough, the arguments are read in, and depending on which commands are input, the appropriate function is called. If the command requires a connection be made to the server first, it is first checked that a connection is successfully made, and then instructions are sent to the server. The only commands that do not require a connection to the server are add, remove, and configure. Every command that requires a connection to the server first sends the command using send, and then the project name. The server reads the command, checks if the project exists or doesn't exist, depending on the command, and creates a new thread that runs the appropriate function. In order to keep all the threads in sync, and prevent two threads from changing the same project at the same time, we used a linked list of a struct that contained both the mutex and the project name associate with the mutex, like a keychain. Whenever a project needed to be accessed, the appropriate key was found on the keychain and the mutex was locked. When the thread finished, the same mutex was unlocked. In order to prevent two threads from changing the keychain at once, through the create and destroy commands, there was a separate mutex that served as the master lock. Every time a mutex needed to be removed or added, that mutex was used. The way files were sent across would follow a specific set up steps. First the size of the incoming file is sent across, in addition to any other data the receiving end needs to know. Then the file is sent across repeatedly in a loop until all bytes have been sent, using the number of bytes received and sent to keep track.