CS 214: Systems Programming: Asst3 Where's the File

Dov Kassai (dk778) and Vijay Challa (vkc17)

README

In our program we have set up a system that uses the client server connection based on what the user has put into the command line as an argument. On the client side we have catches for each of the twelve commands. The three that we don't need to access the server is configure, add, and remove. These functions are all local on the client's side.

When we connect to the server for the first time we would have a check that would see if the directory for the project exists or not. If the directory did not exist then we would create one so that the path to the project on the server side would be the same every time.

We also incorporated threading for the connection to the server so that it would be able to accept multiple connections from multiple clients. This is needed since there can be multiple clients that are working on the same server and would need to access the project at the same time. If there were no threads, then only one person would be able to access the server at the same time.

The design we have for the server is that it is constantly listening for a connection. The reason is because of the threading that we programmed in. It must always be listening for multiple threads that can come in.

The design for the client is that for every time it is run we have a check for what the argument given is. For example, if you call configure in the command line it would check to see if the first argument matches any of the functions that we have written. Since it does match, it goes into that function and does the work needed until it closes. For arguments, the arguments that connect to the client, each time that we call them then we reestablish the connection to the server. This is because we do not want to have a continuous connection between the server and the client. It should only be established when it is needed and closed when you are done with the process.

For when we were sending files between the client and server, we used a compression function so that we would only be sending one file between the two machines. We had a function that would tar the file that was going to be sent, then on the other machine we had a function that would decompress the files. This was so that it would be faster and less data would need to be sent over the network connection.