FINAL REPORT

Design:

Firstly, I want to start with explaining my reasoning and design. My program has 2 separate part so to say. First is connection part and second is keeping synchronize part. In the first part, client requests to connect, server accepts the request when server is idle (when synchronization is not started) after that server compares its files with the client and keeps the one which is up to date. So if client has the same file as the server and it is more up to date than server's file, client's file is kept. Besides that if server does not have the file that exists in client the file is added to server and same scenario valid for client. That was the first critical part.

Second part is to keep synchronized. This part has 4 sub parts. Firstly server compares its all files with the client by sending file name, file type and file last modification time and if the client has the file then more up to date file is chosen, otherwise file will be added to client's directory later.

Secondly, client sends its remaining files which does not exist in server to server, and server keeps the track of files in case of another client has more up to date version of the same file.

After comparing all files with all clients, server starts updating itself. Server keeps the files in a linked list, and if a client has more up to date file server changes that file properties in linked list. Therefore, when updating itself, server knows which file to receive from which client. And updates itself.

Lastly, server has the most up to date files but clients do not. So, server updates all the clients separately by sending its files.

Results of Tests:

I tested with many clients and there was no problem about number of clients, program accepts as much client as thread pool size so if more client than pool size try to connect, they will be rejected.

My program synchronizes when a client delete a file, edit a file or create a file also when server edit a file or create a file or delete a file synchronization happens. But only one case is not achieved that when a client creates a file inside one folder in its directory, server gets the file but cannot send out to other clients. Other than this all the operations are successfully done.

Shortcomings:

For clients, log file is not created. Also signals are not handled properly so that when SIGINT is received, socket is closed and all the clients and the server shut down. Other than that when clients connect at the same time, this causes client to be stuck. They should be accepted one by one.