CSE5306, DISTRIBUTED SYSTEMS
FALL 2017, PROJECT 1
REPORT

I have neither given nor received unauthorized assistance on this work
Team Members

Signed: Divya Kathirvel Date: 09/28/2017

Signed: Ankita Gandhi Date: 09/30/2017

PROGRAM IMPLEMENTATION:

We selected message oriented client-server communication for our project.

For the server side, the following steps were followed in the program to establish connection with client

1) The socket module is imported

2) A port is reserved for the file transfer

3) A socket object is created

4) Use the local machine host and port to bind

5) Listen to client for connection request

6) Once client requests accept the request and service the client by sending and receiving data

For UPDATE, DOWNLOAD, DELETE and RENAME functions, on receiving the request from the client, the server performs the corresponding action and responds back to the client the Success or Error message.

For the client side,

1) The socket module is imported

2) A port is reserved for the file transfer

3) A socket object is created

4) Use the host and port of server to connect with the server machine

5) Once server accepts the request, can ask for service from the server sending and receiving data

For performing functions, the client sends request to server and waits for positive or negative response from the server.

Note: If connection established on different systems, turn off the firewall, and both client as well as server needs to be connected on the same network (Do not use UTA college network)

Issues Encountered:

1) Code compatibility issues, such different Interpreter versions for Python (2.7 vs 3.5)
2) Version control of code
3) Understanding the functionality of sockets
4) Usage of single and multi-threaded server

Learning outcomes:

1. Understanding of client server model.
2. Understanding of message oriented communication between client and server
3. Socket programming
4. Understanding of version control tools
5. Understanding multi-threading
6. Importance of locking

Understanding importance of locking:

1. We connected 2 clients to our multi-threaded server from 2 different machines.
2. Each gave command to rename the same file at the same time.
3. One request was successful and one request ran into exception saying file not found.
4. By this we understand how locking protocols can be necessary.

5. Locking protocols will provide additional stability and guarantee consistency.
6. Most basic approach will be provide write lock to the file for functions like renaming. Read and write locks for delete requests, etc.