Michael Zaragoza
016803317

Lab Assignment #1 Report

In summary, this lab allows two computers to act as a client and server to synchronize files with each other. Basically, the client will connect to the server and they will both share files with each other. Any changes to them will have the client and server synchronize those changes so that they reflect the same content. I did run into a couple problems while doing this lab. It was not easy whatsoever.

The first problem was trying to implement some sort of distribution hash table for the program. In Python, I decided to create a dictionary to act as a hash table to store the IP address and all of their respective files. This made sure that all the correct content stays in respect to their nodes. This also makes comparing files easier.

The second problem I ran into was synchronization. I was trying to figure out how exactly I should go about syncing all the files from every node. I started off by making a while loop that will continuously execute until the user ended either program. In this loop, a for loop executes, checking every node in the network and verifying that they all have the same files. If a new file was added to any of the nodes, the server will update every other node and add the files to them. If a file was deleted, then the server will notify every node and remove the file from every node.

The third problem was working with two computers for the assignment. The program works for one computer, but I can't run it the exact same way on two. If I run a client on another computer, it connects to the server, but it cannot retrieve the files from the client, since the client's directory is not on the server. This was the drawback for testing everything on one machine. I made sure to test all my synchronization implementations beforehand, but failed to test these with two computers.

There were definitely a few things I couldn't implement in time. First one was checking if files were updated. I was thinking of obtaining the timestamps for each file and comparing them to the server's local table's file. If they were different, then I would simply "add" the updated file to the nodes, replacing the original. Another implementation I couldn't complete is testing the assignment with two computers earlier on. As stated above, I was able to connect to the server from the client on a separate computer. However, not only did I not write the code to retrieve the client's files instead of it's directory, but I didn't have permission to see the newly created folder on the client. This makes using a peer-to-peer network not happen, which is a little disappointing, since this was a requirement for the lab.