CS-382 Network Centric Computing: Spring 2016

Programming Assignment 2
Due Date: March 9, 2016 11:55 PM
(10% penalty per 24 hours late upto 4 days - no credit afterwards)

About Cooperation with Other Students:

This assignment is meant to be done alone. Absolutely no cooperation is allowed. If there are questions, ask the course staff; they are there to help you. For this assignment, you must do all thinking, research, and coding by yourself. Do not even discuss with anyone how far you are with this assignment.

Individuals found guilty of violating above policy will promptly be referred to the disciplinary committee.

More Important Note:

We are also giving you the responsibility to REPORT all incidences of cooperation. You MUST report to the instructor (or the TAs) such incidences. If you do not, we will consider you as guilty as those who you witnessed cooperating. All coding, debugging, web search for functions, etc. must be done by individual students!

Even More Important Note:

Server will be called like this:
server <port>

Client will be called like this:
client <server-ip> <server-port> <username>

Server must work in the directory it's started in (i.e current working directory)
Client must monitor the directory it's started in (i.e current working directory)

Client and Server both can run on a single machine. Or, they can run on different machines as well.

Problem:

In the previous assignment we started building our own Dropbox clone, affectionately called Dropbin. In this part you will be extending it to a full fledge deployment-ready app.

The Assignment:

*You can continue to assume that there are no subdirectories in your **client's** dropbin folder and all your dropbin files are in the same directory so you don't have to worry about directory trees.*

Again, you must use TCP as your transport layer protocol for this assignment. Ask yourself: why? (then ask us if you don't get the answer)

Sharing:

- Similar to the original dropbox, one file can now be shared between multiple clients. Each of the collaborators of a given file should be able to update the file and the changes should be reflected to rest of the collaborators. Similarly if one of the collaborators of the file deletes a shared file, it should be deleted from the Dropbin directory of all the clients
- For sharing a file create a file "Sharefile.dropbin" in your dropbin directory. Each line of this file should contain the name of the file to be shared followed by a space separated list of usernames (the collaborators on that file). For example if user_1 shares file_A with user_2 and file_B with user_2 and user_3, user_1's Sharefile.dropbin will look like this:

```
file_A user_2
file_B user_2 user_3
```

- In case of conflicting updates e.g user_1 changes file_A and it's uploaded to the server but not yet propagated to all clients/collaborators, meanwhile user_2 (or even user_1 from a different machine) makes changes to its copy of file_A that gets sent to the server. Now every copy of file_A should have user_2's version or the later version of user_1 of file_A.

Dropbin Server:

The Dropbin server now is not only responsible for receiving updates from clients but also needs to propagate these updates (e.g. in case of file sharing). The server needs to keep track of what files are shared with which group of users and makes sure that shared files are in sync across users. In addition, now your server also needs to be fault tolerant. This means It restores its last known state on restart. To do this you would need to intermittently store the state at the server side. How you do that is an implementation decision you need to make. Moreover, now the server will also needs to:

- Keep track of usernames and their corresponding files for different users.
- Keep track of shared files.
- Push changes to the client when anything on the server changes e.g due to shared files being edited or users connecting from different machines.
- Handles conflicts that arise due to sharing (last update wins as discussed in the sharing section).

Dropbin Client:

- Dropbin client now takes username as its third command line argument and sends it to the server when first connecting.
- The Dropbin client can now connect from multiple locations. Client's directory in each location should be in synch and be a perfect mirror unless the client uses selective sync. In selective sync, a client can select a subset of files to be synced to one of its machines.
- The client can also disconnect and reconnect (from one or all of its locations). On reconnection it will pull all updates from the server and pushes any local changes it has made since disconnecting.
- You will have to implement some method to figure out what if any changes happened while the client was offline.

Submission instructions (MUST be STRICTLY followed)

1. Your code must work in a linux environment
2. Name your code files dropbin-client.java and dropbin-server.java (in case of java) or dropbin-client.py and dropbin-server.py (in case of python)
3. Include any other support files (other source files, Makefile, etc.) needed to successfully run your program
4. Use your 8-digit roll number (as per your email address) dot zip. e.g., 15100061.zip
5. Upload to LMS (in the Assignment Tab)

Goodluck!