CSCI 6300 Project: Cloud Prototype

(100 + 15 bonus* + 15 bonus) points**

1. Project description

Cloud computing is known as "on-demand computing", which is a kind of Internet-based computing, where shared resources, data and information are provided to computers and other devices on-demand. The storage solutions of Cloud computing provides users and organizations with various capabilities to store and process their big data in third-party data centers. Cloud computing relies on sharing of resources to achieve coherence and economies of scale.

In this project, you will need to design and develop a **prototype** of the Cloud environment, which should at least have the following components and functions:

(a) Two servers and two clients.
- Both servers can accept and handle multiple user requests at the same time.
- Two clients may contact the same server at the same time or different servers, respectively.

(b) Both servers have user authentication method implemented.
- Client needs to log in on the server, every time when it starts a new connection with one of the servers.
- Each server uses a text file (account1.txt for server 1 and account2.txt for server 2, maintain consistency) to save the (user name, password) pair, and checks if this is a new user or if the login information is correct or not.
  - If new user (user name not found), ask user to enter user name and password, save to its corresponding account file, and copy it another server's account file to make them consistent;
  - If incorrect login information (password not found), ask user to reenter the password. Disconnected this connection if more than 3 attempts.

(c) Both servers have "Read", "Write" and "Delete" functions implemented.
- After log in, client can **read** information from the server. Server sends all the information to the client.
- After log in, client can **write** information to the server.
- After log in, client can **delete** information from the server.
- Client sends "exit" to terminate the connection to the server once it is done. Then server needs to check if the client has made any changes (i.e., write or delete). If yes, server needs to synchronize the local information with the other server
  - Copy the new information to the other server;
  - Delete the same information from the other server.

(Hints: Each server uses a permanent file to store all the user data, and uses two separate temporary files for user added data and deleted data, respectively, which will be used during the synchronization between servers after user exits.)

(d) *Both servers have "Critical Section" check implemented.

  - Other client can read from the server when one client is also reading from that servers.
  - Other client cannot access (read/write/delete) the server when one client is making changes (write/delete) to that servers.
  - Client cannot access (read/write/delete) the server when one server is synchronizing with the other one.

(e) **Instead of just using one file for all user data in Step (c), both servers use a separate file to store information for each individual client.

  - Each client can read every other's file.
  - Each client can **only write/delete** the content of its own file.
  - Critical section is associated with each individual file.
  - One client cannot read a file when the owner is making changes (write/delete) to it.

2. Requirements

(a) You are given the flexibility to choose one of your favorite programming languages for implementation either in Windows or Linux environment, though c/c++ is preferred for the network socket programming.

(b) You must submit

  - all your source code of your daemon program
  - executable files
  - Makefile (if use c/c++)
  - ReadMe file that describes
    - the use of your program
    - how to run it

(c) You will need to demonstrate your project in class at the end of the semester (usually the last week before the final).

(d) If you form a team of no more than two students and work on the bonus steps ((d) or (e)), both of you will have the full credit for basic part but the bonus points will be evenly distributed between you two.

(e) The following due dates (see the syllabus) still apply except for the first two dates if assigned project is chosen.

  - Topic Selection: Feb. 4 (do a 5-minute presentation in class);
  - Project Proposal (3+ pages): Feb. 12 (by midnight);
  - Progress Report (4+ pages): Mar. 18 (by midnight);
  - Project Presentation and demo: Apr. 28 and May 3;
  - Final Report: May 5 (by midnight).