Programming Assignment #2

CSC 688 Go Language and Programming
Department of Computer Engineering
Santa Clara University

Dr. Ming-Hwa Wang
Phone: (408) 805-4175
Course website:
Office Hours:

Summer Quarter 2017
Email address: mwang2@engr.scu.edu
http://www.cse.scu.edu/~mwang2/language/
Friday 9:30-10:00pm

Due date: midnight June 25, 2017

Centralized P2P File Sharing, Part I (200 points)

To implement a centralized P2P file sharing system, you need to run a centralized server first, (e.g., on the Linux server at 23.253.20.67) and the server program should display the available port number (from 1024 to 65535). The server can accept requests from clients and dispatch each request to a thread and immediately accept the next request. All threads are run concurrently to improve performance and it can handle more number of requests per time unit. The server will maintain the parallel hash table. You then can run multiple clients at different hosts (e.g., 104.130.67.11), and connect to the centralized server using its <ip> and <port>. Each client can issue put/get/delete commands concurrently. The client who issues put command needs to start a local server and use the <music name>, <ip> and <port> as parameters to issue the put command. The client who issue get command will get the <ip> and <port> from the centralized server and then call the local server to download and display the content of the music file locally. To simply your work, assume that each music file contains only text content. Your p2 should work on any number of Linux server other than the two Linux server listed above. Your p2 need to be tested on different computers and thus Autotest doesn't work. You need to test p2 manually and use Submit to submit your p2.

E.g., On machine with IP address 23.253.20.67, do:

```
$ server
server starts at IP address 23.253.20.67 and port number 9998
```

On machine with IP address 104.130.67.11, do:

```
$ client client_1 23.253.20.67 9998
client_1> put("listen to the music", <path to music file>)
```

On any Linux machine, do:

```
$ client client2 23.253.20.67 9998
client2> get("listen to the music")
# display the content of the music here
```

In order to make grading your p2 easier, your p2 should copy the shared music file to /tmp/<login>/napster/<client name>/<file name>, and make (by your code) /tmp/<login>, /tmp/<login>/napster and /tmp/<login>/napster/<client name> with mode 777 and /tmp/<login>/napster/<client name>/<file name> mode 666.

Student Name:

ID:

Score:

Correctness and boundary condition (60%):

Error Handling (5%):

Display output on both server and client windows whenever there is an event happens (10%):

Modular design, file/directory organizing, showing input, documentation, coding standards, sympathy/typing point with README (20%):

Automation (5%) for build executable automatically:

Subtotal:

Late penalty (20% per day):

Special service penalty (5%):

Total score: