Introduction to Network Programming

1st Exam Makeup

Date: 2020/12/21 Time: 15:40-18:20

Set up:

Download **np_1st_exam_makeup.zip** from new E3. After extracting this file, you will see the following directory structure:

np_1st_exam_makeup/

└── **setup.sh** --- This script is used to set up directories of your submission.

Submission:

Change directory to **np_1st_exam_makeup**. Type **chmod u+x setup.sh** then type **./setup.sh <your_student_id>** to set up the directories. Then, you will see the following directories.

<your_student_id>/

├── P1/ --- This directory is for storing your answer to the problem 1.

├── server/

Put your **server code** and a readme file that describes how to compile your program here.

└── client/

Put your **client code** and a readme file that describes how to compile your program here.

└── P2/ --- This directory is for storing your answer to the problem 2.

Put your **server code** here and a readme file that describes how to compile your program here. We will test your code with telnet or your own client code if you put it in.

Type **zip –r <your_student_id> <your_student_id>** and upload the **<your_student_id>.zip** to new E3. **Upload the file with wrong format will get some penalty.**

Notes: late submission will get 20% score penalty, please take your time.

1) (50%) UDP file uploader

Implement a UDP Server and UDP Client. Your client should be able to transfer files to the server side. Your server can only bind to an assigned port and the server will wait for the client's request on this port and receive the files from the client.

Command format:

- Server
  - **./server {port}**

  Example:

  ./server 1234

  Server can only receive the client's request using port 1234 (1234 is just for example. It can be any valid port)

- Client: connect to the server.
  - **./client {server-ip} {server-port}**

  Example:

  ./client 127.0.0.1 1234

After connecting to the server, client should be able to do the following functions:

- **send-file {file-name1} {file-name2} {file-name3}…**

Client can send variable number of files to the server. When the server receives the file, the received file name must be the same as the original file name.

- **exit**

Client will close the connection to the server, but server will still running and wait for another connection.

Notes:

- The specified files will send to the same directory as the server program.
- Use "% " as the command line prompt. Notice that there is only one space after the prompt.
- We will ONLY test **one** client at the same time. The maximum size of the transferred file will not be larger than **256 bytes**.
- You will get some penalty if your server or client breaks after transferring file(s).

2) (50%) Implement a TCP server. The server does the following functions:

- When client connects to the server, client will receive this message from the server:

  Hello, please assign your username:

- After input the username **{userID}**, client will receive this message from the server if the username is not be used:

  Welcome, {userID}.

  Otherwise the client will receive the message:

  The username is already used!

- When client connect/disconnect to the server, **server** should output message:
  New connection from {ip}:{port} / {userID} {ip}:{port} disconnected

The client does the following functions:

- The service accepts the following commands and at least 10 clients:

  Remember to handle incomplete client input. (you can send an error message whatever you want or just skip it.)

Command	Description	Output
list-users	List all **online** users' username in **any** order with their IP and client port number.	{userID} {ip}:{port} {userID} {ip}:{port}…
sort-users	List all **online** users' username in **alphabetical** order with their IP and client port number.	{userID} {ip}:{port} {userID} {ip}:{port}…
exit	Disconnect from the server.	Bye, {userID}.

Example:

Scenario: Bob connect to the server and assume that there is no other user connects to the server after him:

Hello, please assign your username:

% Bob

Welcome, Bob.

% list-users

Chris 140.113.12.34:12345

David 140.113.56.78:56765

Alice 140.113.211.33:34253

Bob 140.113.32.87:23232

% sort-users

Alice 140.113.211.33:34253

Bob 140.113.32.87:23232

Chris 140.113.12.34:12345

David 140.113.56.78:56765

% exit

Bye, Bob.

Notes:

- Use "% " as the command line prompt. Notice that there is only one space after the prompt.
- It's OK to print the client IP as "127.0.0.1" if the client runs on localhost.
- The result of "list-users" could be changed after using the "sort-users" command.