Advanced Computer Networks

Assignment 1 – Talk Application
Assignment given on: 22nd Aug 2023
Due Date: 3rd Sep 2023, on Moodle
(5% Penalty per 24-hour period)

Assignment 1 - Talk Application

The first objective of this project is to implement a simple Talk (2-users) application, between two users called U1 and U2. Assume that U2 is running the server to which U1 will connect. The two users can be in two different terminal windows on the same machine or on different machines.

Communication between the users is via UDP sockets.
At each user, the program will wait for user input or for network input. Let us consider U1 first.
If the user types a message (ended with carriage-return), then U1 sends this message to U2 via the socket. U2 will receive the message and simply output it to the screen as:

```
U1: What is the time?
```

On the other hand, if U1 gets a message from U2 *before* U1 types its message, then U1 will print the message as follows:

```
U2: Did you watch oppenheimer?
```

This continues back and forth until either one of them types "exit" or "quit" (not case-sensitive), at which point both programs terminate.

As an example, the output for a typical exchange (for U1) will be as follows:

```
U1: Hi, How are you doing? (Typed by U1)
U2: Hello, I am doing fine. What about you? (Typed by U2)
U1: I am doing good. (Typed by U1)
U2:  Did you watch oppenheimer? (Typed by U2)
U1: Yes, I have watched the movie. (Typed by U1)
U2: Ok. How is it? (Typed by U2)
U1: It is breathtaking!! A must watch (Typed by U1)
U2: I will watch it then. Thanks for the feedback. (Typed by
U2)
U2: EXIT (Typed by U2)
```

Summary of Command Line Options: The command line options provided to U1 are listed below:
- -s string – Receiver Name or IP address.
- -p integer – Receiver's Port Number

The command line options provided to U2 are listed below:
- -p integer – Receiver's Port Number

Note: getopt or similar function will be useful to learn.

1.1 Relevant Files
- The code for implementing a basic UDP client and server (in C) is available from Moodle or from Douglas Comer's (or Richard Stevens') book.
- You can also take a sample template of your choice and implement the core functionality of the application.
- You must generate your own Makefile and/or script files.
- You can implement it using either C/C++/Python or Java.

1.2 Sample Session
Assume that you have created the files U1.c and U2.c and the corresponding executables in your directory.

nif-c1% ./U2 -p 12345 &
nif-c3% ./U1 -s nif-c1 -p 12345

2. Grading (Evaluation Criteria)

Script file (equivalent to make file) which will contain instructions on how to run the code - 5%
Readme file explaining the directory structure, function of each file , etc. - 5%
Udp server - 40%
Udp client – 30%
Live demonstration of working program – 20% (please make a video recording of the application, demonstrating the client and the server application.)

Submission Details:

1. Please read the questions carefully and complete them.

2. Make a directory with name <Your_Roll_Number> and copy your all program (source code) and output file to that folder.

3. You can implement it using either C/C++/Python or Java.

4. Please copy your output and paste it to related program at the end of source code(please comment it)/ if necessary you can take screenshot name it with its question number and put it in the same folder.

5. Please make a video recording of the application, demonstrating the client and the server application. The video file may be uploaded on a shared drive folder, and the link to access the

same may be shared with the rest of the documents. **Kindly do not submit the video file on moodle.**

6. Test well before submission. Follow coding style uniformly. Provide proper comments in your code.

7. Submit only through moodle and well in advance. Any hiccups in the moodle/Internet at the last minute is never acceptable as an excuse for late submission. Submissions through email will be ignored.

8. **Please zip your folder and submit to moodle within 03-10-2023 (23:55 PM)**