

Computer Networks

Instructor: Shashi Prabh

Lab 3: A simple TCP client-server application

Finish by: Feb 10/13

In this lab, you will gain familiarity with socket programming by modifying the a simple client-server code provided in the textbook. *This lab is to be done individually.*

1 Running the sample code

Save the client-server codes of Peterson and Davie, Section 1.4.2 as `client.c` and `server.c` (saving in separate folders will be helpful down the line. You can get the code from the course website as well.) Compile and test the client-server code. For compiling, you can invoke:

```
gcc srcfilename -o execfilename
```

Use the client-server program to chat with your neighbor! We will announce the address of a test server that we'll be running in the lab. Send from the client your name to the test server.

2 Make the client connect to a specified address

Modify the server code so that it binds to a specified address instead of `INADDR_ANY`. Test the code using your neighbor's IP address. You can find IP address by using `ifconfig -a`, `hostname -I` or `ip addr`.

3 Requesting and receiving files

Modify the code to support the following *sequence* of instructions:

1. Client establishes connection
2. The server sends "Hello"
3. The client receives "Hello" and then sends "GET"
4. The server sends "OK"
5. The client may send any number of file requests. The client makes the requests by sending filenames, one at a time.
6. If the server has the file, it sends the file. Otherwise, it sends "File not found" message.
7. The client sends "Bye", waits for any message from the server for 2 seconds and then terminates the connection.

The client and server should keep displaying on the terminal all the messages (sent and received) and requests. After sending "Bye" to the test server, print the last message from the test server to see your demo result.

Submission Explain and demo your code to one of the TAs.

Evaluation

- Client can connect to a server running at specified IP and port. TA: _____

Can send and receive strings. TA: _____

Client follows the specifications and can receive files. TA: _____

Server follows the specifications and can send files. TA: _____