Saugat Tripathi
@02752994

Github: https://github.com/saugatt/DataComm_Project

TCP Programming Project

How to use the program:

There are two folders server/ and client/ in the main project folder.

For Server:
cd server,

1. **Compile the server code using command:**
   gcc –o server server.c

2. **Run the server using command:**
   Syntax: ./server <port>
   Example : ./server 1234

For Client:
cd ..
cd client/ folder,

1. **Compile the client code using command:**
   gcc –o client client.c

2. **Run the client using command:**
   Syntax: ./client <server IP> <server port> <file path> <to format> <to name>
   Example ./client 127.0.0.1 1234 ../practice_project_test_file_1 1 outputFile

Go to: server/ folder,
Compile the server code using command:
gcc –o server server.c

Run the server using command:
./server <port>
eg. ./server 1234

Go to client/ folder,

Compile the client code using command:

gcc –o client client.c

Run the client using command:

./client <server IP> <server port> <file path> <to format> <to name>

eg. ./client 127.0.0.1 1234 ../practice_project_test_file_1 target

Protocol

The protocol for communication between Client and server has the following rules:

1. The maximum file size is 1 Mb
2. A header of approximately 20 bytes is made.
3. The message type is a single integer buffer that is less than 4 Mb

When starting the the server a port number is provided, if not a default port number of 5002 is set.

In the client side, when starting the client, following parameters are provided:

- The IP address of the server where the application is running
- Port number of the application
- The file to be sent to the server
- The format that the file is converted into
- The name of the target file

The fields used are:

- Total count,
- type count, type
- file name size
- file name
- data count
- data

Total count: Total number of integers in array counting header and data from file
Type count: Single integer with counting the length of type sent by client.
Type: Type received from command line as an ASCII value in integer.
File name size: Single integer for size of filename.

In server side, runs continuously without interrupting and is listening for connection through its life.

After the connection is established the conversion is done in the server side, which is not part of the application and writes the file. Waits until any message is received from client.

If receives a message from client.

Reads the first 4 bytes which is total count integer. Reads the remaining (count) integer from the socket and puts into respective fields.
Sends back error if there is format error in type or amount or in numbers.

Parses the type, file name and data and converts the type accordingly and saves into the folder where the server resides. Reads four bytes from the socket to a integer named status.

If state == 0: prints “Format Error”
If state == 1: prints “Success”

Closes the connection and exits.

Closes the connection.

Test Cases

Description Content/Error
File size > 1MB
Making sure my server reads a fixed amount from the socket. As my server reads only 4MB, I am requiring my client to send file less than 1MB.
Client: File length exceed! Undefined error: 0
Server: No response, just waits.

Type error (Type other than 0, 1, 2 or 3)
Have to handle type error in client as server is made only to process type 0/1/2/3.
Client: Type variable is not correct. Please use type 0, 1, 2 or 3: Undefined error: 0

Challenges:

Learning quirks of C programming
Learning about socket programming and networking.

References

http://www.paulgriffiths.net/

https://www.tutorialspoint.com/cprogramming/c_quick_guide.htm
https://stackoverflow.com