Server – Client Socket Programming

Objective

The objective of the project is to observe the server-client behavior over TCP connection, read and write TXT, HTML, IMAGE files between both and simulate the behavior of HTTP requests.

Project Flow:

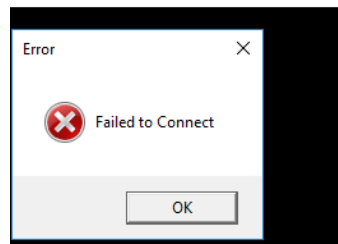1. Server starts and awaits listening for any connections
   SERVER

2. Client starts connecting to server, if no connection is found, it throws an Error, and server grants permission for connections.
   CLIENT:

   SERVER

3. Client sends request parameters to server, and server awaits and receives it. Server send request confirmation.
   CLIENT

   SERVER

4. If server cannot find file, it sends 404 error message.
   CLIENT:

5. If file was successfully received, client receives success message.

CLIENT:

```
Reading in binary
Succesfully read
```

SERVER:

```
Content-Length: 11164
Writing in binary
File successfully saved.
File has been uploaded to server.
```

6. Server can handle multiple connections using threads:

```
Awaiting Connections ...
Client Connected with Socket fd is 300 ip is 127.0.0.1
Permission message sent to the client
Client 0 has been added to socket array
Receiving Request: POST magdy.png hostname
errorSending Response: HTTP/1.0 200 OK
Content-Length: 11164
Writing in binary
File successfully saved.
File has been uploaded to server.

Client Connected with Socket fd is 316 ip is 127.0.0.1
Permission message sent to the client
Client 0 has been added to socket array
Receiving Request: POST magdy.png hostname
errorSending Response: HTTP/1.0 200 OK
Content-Length: 11164
Writing in binary
File successfully saved.
File has been uploaded to server.
```

Organization of Project:

The project contains 2 main projects, one for server and the other for client, each has 2 helper extension files in server and client which have functions to

1. Parse Client Request
   It takes the request as a string, takes the first 3 arguments of the request (the ones we're interested in this project), and returns an array of 3 to be used later.

2. Read from file and insert into buffer
   Function takes three parameters filename, buffer and pointer to int to return the length of content being read.
   The function reads any file in binary, and then it terminates the buffer with '\0' to indicate end of buffer and prevent garbage.

3. Write from buffer and insert into server/client
   Function takes filename, buffer and length as parameters and then uses the write function in fstream to write the buffer into the new file.