Net-centric Programming

Lab: Simple Web Server using TCP Socket

In this lab, we will develop a so-called Very Simple Web Server (VSWS) which user can use a web browser to request data from our web server. The web server only supports simple document retrieval. To test your program, open a Web Browser and type the URL in the address bar, for example: *http://127.0.0.1:9999/index.html* (if your server listens on port 9999 and your want to retrieve the file index.html from the server)

Our web server needs to parse the request or http GET message send from the client browser. The format of GET message:

```
GET (space) URL (space) Version (\r\n)
Header field name: (space) value (\r\n)
(header fields)
Blank line (\r\n)
```

Example: for the URL above, you will see the request message similar to the following

```
GET /index.html HTTP/1.1
Host: 127.0.0.1:9999
(other header fields)
```

Then the web server will send back a response message. The format of the response message:

```
Version (space) status code (space) phrase (\r\n)
Header fieldname: (space) value (\r\n)
….
Blank line (\r\n)
(data data ….)
```

Mostly we will use the status code 200 OK to indicate the successful of data sending to client (more status code can be found in the HTTP RFC 2616)

Example:

```
HTTP/1.1 200 OK
Content-Length: 100   //Size of data (i.e. size of index.html)
Content-Type: text/html

(content of index.html file)
```

1. Reuse the TCP server code from previous lab (which server receive a string from client and display it on the screen). Modify the buffer size so it can handle a long string for example, 1024 would be enough for receive most request message from browser. Use a web browser to type in the URL and check whether you receive the correct request message or not

2. Construct the response message by hard-coding the data return to browser, for example, the data could be: *<html>Welcome to Very Simple Web Server</html>*

3. Create a simple html file (i.e. index.html) and this file back to browser and check whether the browser display the exact content or not

4. Assume the index.html file contains a link to an image file (.jpg) on your web server (likely you can put the .jpg file in the same folder with index.html. Improve your web server so it can handle the .jpg file as well. Repeat the process for audio files (mp3, wave, midi..)

5. Improve your server to deal with unavailable resources, error codes.