Introduction to Computer Networks

Assignment 2: Web Server Imitation

1. Goal

- Develop a simple web server using socket programming.

2. Development environments

- TA will evaluate your results on Linux. Therefore we recommend you use C or C++ on Linux. You can also use Python (version 3.6+) on Linux or Windows.
- You have to describe your development environment information in detail in the report.
  If not, TA cannot evaluate your program and you will get zero points.

3. Functionalities to implement

- Server
  - Develop a standard web server program.
  - The web server program runs with the port number of **10080**, and waits to receive HTTP requests from web browsers that **run on another (virtual) machine**.
  - When receiving HTTP request messages, the web server program sends HTTP response messages back to the browser.
  - Users can request any files (e.g. **html** or **image**) which are located in the same directory where web server program runs.
  - The web server must be able to handle concurrent HTTP request messages from multiple browsers.
  - You can extend your web server program to provide a "log in" functionality. (see below)
  - You can extend your web server program to be persistent HTTP mode. (see below)

- Client
  - Use an existing browser that follows the HTTP standard.
  - Write a URL in the browser address bar;
    - http://server_IP:port_number/html_file (e.g. http://115.145.179.59:10080/secret.html)
    - http://server_IP:port_number/object_file (e.g. http://115.145.179.59:10080/1.jpg)

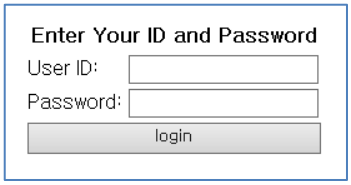
4. Evaluation Scenario (total 100 points)

4.1 Basic HTTP requests & responses (60 points)

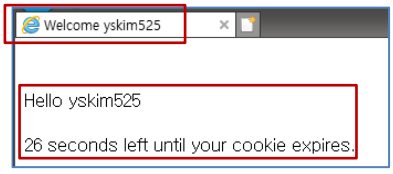
- User requests HTML files or object (e.g. image) files.
  If the HTML file includes objects, the browser requests them automatically. Therefore, your web server program handle those subsequent requests.
- Object (image) file size can be up to O(~10) Mbytes.
- User can repeatedly access / request content in Web server.
- If you requests a file that is not present in the web server machine, the web server should return a "404 Not Found" error message.
- The web server program operates in a non-persistent HTTP mode.

4.2 Log-in functionality (20 points)

- If you add 'Log-in functionality', users must access "http://serverIP:10080" for the first time, and the web server sends "index.html".
- The "index.html" file shows the following input forms;



- If a user enters ID & password (and push the login button), the server returns "secret.html".
  - You prepare your own "secret.html" file that includes any interesting story (text) about you and related image / video files (more than 3 objects).
- Without the login process, if users try to access any (even non-existing) URL directly, the web server program should return a "403 Forbidden" error message.
  - Once you logged in, you can access any content in the web server for 30 seconds. After 30 seconds, you will be logout status.
  - In the logout status, you have to log in again to access another content in the web server.
- When accessing "cookie.html", you can see "User_ID" in the browser title bar, and display how many seconds left before the cookie expires.



4.3 Persistent HTTP mode (10 points)

- Upgrade your web server program to be persistent HTTP mode.

4.4 Write your Report precisely (10 points)

5. Submission

- The deadline is 11.3(Sun) 23:59.
  - For delayed submissions, a penalty of -15 points applies every 24 hours. After 72 hours, you get zero points.
  - In the case of plagiarism, you will receive 0 points for the first time and ***F*** for the second.
- Submit a zip file in iCampus. The zip file must include a **report**, a **source code**, and **HTML and/or object** files for your demonstration
  - Name the zip file as flows StudentID_Name.zip (ex: 2019001_홍길동.zip)
  - In your report file;
    1) Describe your development environment information in detail. (Versions of operating systems, programming languages, compilers/interpreter versions, compile options…)
    2) Present how to design your assignment such as data structures and algorithms.
    3) **Explain how you test each item in "4. Evaluation Scenarios" (showing the screen capture). If you do NOT mention, TA will NOT evaluate.**

Notes:

1) You need to know the exact HTTP request & response message format.
2) Learn the HTML syntax, and create the necessary HTML files.
3) Study how to pass the input values in HTML to Web Server (GET or POST method)
4) Apply Cookie for the Log-in functionality
5) To implement a Persistent HTT mode, survey how to do first.
6) Use wireshark or tcpdump to check incoming and outgoing HTTP & TCP traffic