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 multiple Linux (virtual) machines.
- You have to use Python (version 3.6+).
- TA will evaluate your results on Linux. Python can be run over both Window and Linux, but you must NOT use window OS dedicated APIs.
  - If TA cannot run your program, and you will get zero points.

*** You can use simple basic python libraries (ex: socket, os, thread…), and must implement the main functionalities by yourself. You are NOT allowed to use a particular library to implement the assignment easy. If you want to know whether a particular library is allowed or not, ask TAs and me via the anonymous google sheet.**
For example, you can NOT use high-level APIs such as “HTTP”, “requests”, “ServerSocket”.

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 commercial 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 two type files; **html** and **image** which are located in the same directory where web server program runs. The html file can also include embed image files.
  - The web server must be able to handle concurrent HTTP request messages from browsers. (Usually commercial browsers open multiple TCP connections.)
  - You can extend your web server program to provide a "log in" functionality. (see below)
  - You can extend your web server program to be a 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.x.x:10080/secret.html)
    - http://server_IP:port_number/object_file (e.g. http://115.145.x.x:10080/1.jpg)
  - For evaluation, TA will run an only single web browser to communicate with your web server program.

4. Evaluation Scenario (total 100 points)

4.1 Basic HTTP requests & responses (60 points)

- User requests HTML or image files.
  If the HTML file includes embed images, the browser will request them automatically. Therefore, your web server program must handle those subsequent requests.

- All files located in the same directory where the web server program runs.

- Image file size can be up to ~ 10 Mbytes.

- User can repeatedly access / request any content in Web server.

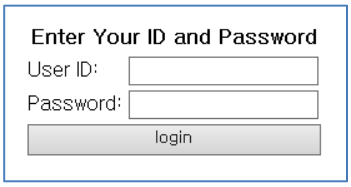
- If you request a file that is not present in the web server machine, the web server should return a "404 Not Found" error message.

- The web server program can work in a non-persistent or a 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;



Enter Your ID and Password

User ID:

Password:

- If a user enters ID & password (and push the login button), the server returns "secret.html".

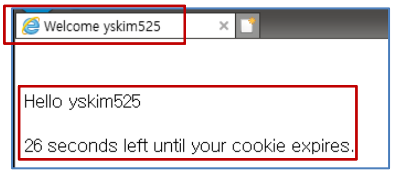
  - You prepare your own "secret.html" file that includes any interesting story (text) about you and related image files (at least 3 or more).

- Without the login process, if users try to access other URL (even non-existing file) directly, the web server program must 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, your must be expired.

  - After expiration, you have to log in again to access any content in the web server.

- When accessing "cookie.html", you have to show "User_ID" in the browser title bar, and display how many seconds left before the cookie expires.



Welcome yskim525

Hello yskim525

26 seconds left until your 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)

- Describe your development environment information in detail. (Versions of operating systems, what kinds of multiple machines that you used, how to run your programs…).
- Explain how to design your program in the view of data structures and algorithms.
- Must present which functionalities were successfully implemented (**with evidence / screenshots**).
  If you do NOT mention, TA will NOT evaluate.

5. Submission

- The deadline is 11.1(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 the ***F*** grade.
- Submit a zip file in iCampus. The zip file must include a **report**, a **source code**, **HTML and image** files for your demonstration
  - Name your source code file as "StudentID.py" (ex: 2018001.py)
  - Name your report file as "StudentID.pdf" (ex: 2018001.pdf)
  - Name the zip file as follows "StudentID.zip" (ex: 2018001.zip)

Notes:

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