

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+).
- Python can be run over both Window and Linux, but you must NOT use Window OS dedicated APIs.
 - If TA cannot run your program on Linux, you will get zero points.

* You can use simple basic python libraries (ex: socket, os, thread...), but must implement the main functionalities by yourself. You are NOT allowed to use some libraries that make it easy to implement the assignment. If you want to know whether a particular library is acceptable or not, ask me via the anonymous google sheet.

=> Allowed library example: "os", "time", "threading"

=> NOT allowed library example: "HTTP", "requests", "ServerSocket",
"flask", "urllib", "websocket"

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** (e.g. jpg, png, gif... etc.) 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 multiple/parallel HTTP request messages from browsers. (Usually commercial browsers open multiple TCP connections.)
- 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.
- Even if you receive the request on "favicon", you can send the actual "favicon" or "404 Not Found" error message.
- Submit a "me.html" file to introduce yourself. The "me.html" file must include some text and **(at least three)** image files on your introduction.
- 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 (e.g. Chrome).
- Write a URL in the browser address bar;
 - ◆ http://server_IP:port_number/html_file (e.g. http://115.145.x.x:10080/me.html)
 - ◆ http://server_IP:port_number/image_file (e.g. http://115.145.x.x:10080/1.jpg)

● For Evaluation,

- 1) TAs can run multiple web browsers on the same/different machines to communicate you're your web server program.
- 2) When TAs access "me.html", the browser must correctly show some text with **(at least) three** image files on your introduction.
- 3) TAs can use other html and image files to evaluate your web server program.
- 4) TAs can access your web server repeatedly to download html and image files.

4. Evaluation Scenario (total 100 points)

4.1 Basic HTTP requests & responses (70 points)

- User requests HTML or image (jpg, gif, png...) files.
If the HTML file includes embed images, the browser will request them automatically. Therefore, your web server program must handle those subsequent requests.
- Put all files 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 via the Web server program.
- 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 Persistent HTTP mode (20 points)

- Upgrade your web server program to be persistent HTTP mode.
(For the same browser, reuse the previous TCP connections for the next HTTP requests.)

4.3 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 4.11(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) For supporting a Persistent HTTP mode, need to know how to implement it first.
- 4) Use Wireshark or tcpdump to check incoming and outgoing HTTP & TCP traffic