

CS3357A COMPUTER NETWORKING

Assignment #1: Build a multi-client HTTP web server.

Assignment Purpose

The purpose of this assignment is to use network/socket programming to build a web server that consist of two separate parts: a server and a client. The server can receive HTTP GET requests and respond according to the HTTP protocol with the requested file or with the appropriate error message and code in case of an error. And the client can connect to the server, send HTTP requests, receives the response and parse it according to the HTTP protocol, and finally saves the received file.

Assignment Description

Using socket programming in python, develop a web server that accepts connections from clients (e.g. browsers), receives and handles ONLY HTTP GET requests according to the HTTP protocol. The web server that you will implement should be able to handle multiple clients simultaneously, therefore, once the server starts it must be ALWAYS accepting new connections.

The GET request is used to request resources (e.g. files) from an HTTP web server. Other HTTP requests such as POST, DELETE exist as well. However, you are only required to implement the GET request in this assignment.

The server script should accept two arguments. The first is the port on which the server is listening for new client connections and the second argument is the maximum number of clients that the server can accept. The IP address should be set to local host if the server and the client are running on the same machine (on different terminals) or the LAN IP of the server machine if the server and the client are running on different machine on the same LAN network, choose any either of these setups!

The client script should accept two arguments. The first argument is the port that the client should connect to (which is the server port the server should be listening on) and the second argument is the file name which the client will request from the server.

Specifically, the server should be able to:

- Accept new connections from clients trying to connect on the server's port.
- Accept and parse HTTP GET requests.
- Search and retrieve the requested HTML file in the server's file system.
- If the requested HTML file is found, an HTTP response message including the header and the requested file should be created and sent to the client.
- If the requested HTML file is NOT found, the server should send a 404 HTTP message to the client.

The client should be able to:

- Connect to the server

- Send HTTP GET requests
- Receive and parse GET responses.
- If a file is returned, the client should save that file.

Testing the server

You should test the server in two different ways:

- 1. Run the server.py file to run the server and use any browser as a client. In this scenario, the browser acts as the client that requests resources from the browser using the URL.**

Assume the server has an HTML file called hello.html. In the server code, assume you set the IP address to local host (<http://127.0.0.1/>) and the port to 9200 accepting 3 maximum clients. To test if the server is running as expected, run the server as follows:

`Python server.py 9200 3`

and type the following URL in a browser <http://127.0.0.1:9200/hello.html>

If the server works appropriately, the hello.html file should be displayed on the browser.

- 2. Run the server.py file to run the server and run the client.py**

In this scenario run the server as follows:

`Python server.py 9200 3`

and run the client as follows:

`Python client.py 9200 hello.html`

If your server works, the hello.html file should be received and saved by the client.

Deliverables

You should submit the client-side script (client.py) and the server-side script (server.py) of the web server.

Late submission

- Late assignments will be accepted for up to two days after the due date, with weekends counting as a single day; the late penalty is 20% of the available marks per day. Lateness is based on the time the assignment is submitted.
- Extensions will be granted only by your course instructor. If you have serious medical or compassionate grounds for an extension, you **must** take supporting documentation to the Academic Counselling unit of your faculty, who will contact the instructor.

Resources required.

To work on this assignment, you will need to install python programming language and a python package manager (e.g., Anaconda, pip)

Rubric

This assignment is marked out of 10. Grades are distributed as follows.

1. The server successfully accepts multiple client connections simultaneously. 2 marks
2. The server successfully receives HTTP GET requests, parses, and handles them correctly. 2 marks
3. The server sends the appropriate HTTP response. 1 mark
4. The server handles client disconnections. 1 mark
5. The client can connect to the server. 1 mark
6. The client can send HTTP GET requests to the server. 1 mark
7. The client receives the HTTP response from the server and parses it correctly to save the received file. 2 marks

Happy coding!