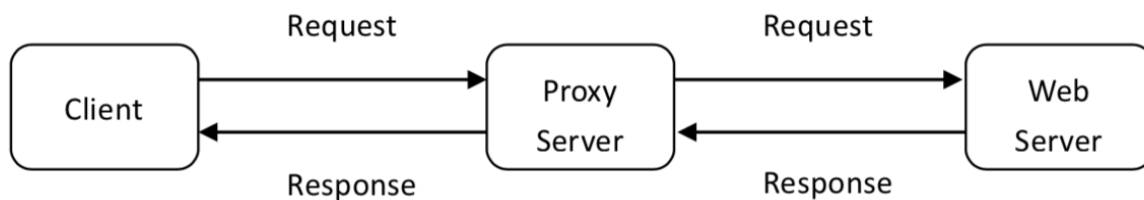


Assignment # 1

In this assignment, you will write a Python-based **socket program to implement a simple Web Proxy**. The following diagram can help you understand how a Web Proxy works. (1) The client sends an HTTP request to retrieve an object (e.g. HTML); (2) This request gets received by the proxy and it creates a fresh HTTP request for the same object to the origin server; (3) Send the new HTTP request to the Web Server; (4) Webserver sends the response back to the Proxy Server; (5) The proxy server creates a new HTTP response along with the object and sends back to the client.

Note: The web proxy must be **multi-threaded** to handle multiple requests at the same time.



Task Details:

1. Extract requested webpage

When the client will be browsing for a particular website (e.g. yahoo.com) through its web browser, it must be making the request directly to the proxy server. If the proxy server is running in localhost at port 8888, the URL could look like “127.0.0.1:8888/yahoo.com”.

- a) In this subtask, your goal is to capture this request at the proxy server (ProxyServer.py) and print the resource being asked (in our example, “yahoo.com”). This requires you to accept the TCP connection made by the web browser in the ProxyServer.py program.

2. Sending GET request to the Web Server and returning the response back to client

Once, the resource name is figured out, your proxy server program must do the following steps:

- a) Create a TCP socket connection (sock) to the Web Server (w/ port# 80) using socket and connect system calls. **Note:** Ensure that the request does have “www.” prepended.
- b) Create a temporary file on the above created socket and use it to pass the GET request. You can use the following code snippet.

```
tempFile = sock.makefile('r', 0)
tempFile.write("GET "+"http://" + resource_name + " HTTP/1.0\n\n")
```

- c) Use the client socket to send the HTTP response back to the Client’s browser by reading the tempFile and writing to the client’s socket.

Important Notes:

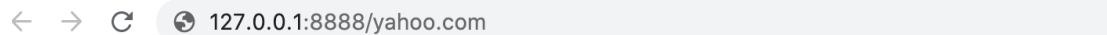
1. You should handle all the possible errors and exceptions.
2. A **Starter Code** is provided for you, which you must modify to complete for implementing the proxy server.

Running your program:

1. Run the proxy server program using your command prompt (example is given below)

```
[csadmins-MacBook-Pro:ProxyServer Dktosh$ python ProxyServer.py 127.0.0.1  
Ready to serve...  
█
```

2. Then request a web page from your browser (see the example below).



3. Your proxy server program should automatically accept this request and start fetching content of www.yahoo.com.
4. Your program should output something similar as following:

```
[csadmins-MacBook-Pro:ProxyServer Dktosh$ python ProxyServer.py 127.0.0.1  
Ready to serve...  
( 'Received a connection from:', ('127.0.0.1', 56804))  
GET /yahoo.com HTTP/1.1  
Host: 127.0.0.1:8888  
Connection: keep-alive  
DNT: 1  
Upgrade-Insecure-Requests: 1  
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.96 Safari/537.36  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9  
Sec-Fetch-Site: none  
Sec-Fetch-Mode: navigate  
Sec-Fetch-User: ?1  
Sec-Fetch-Dest: document  
Accept-Encoding: gzip, deflate, br  
Accept-Language: en-US,en;q=0.9  
sec-gpc: 1  
  
/yahoo.com  
yahoo.com
```

What to be submitted:

1. A PDF formatted report with all evidences, codes, execution samples, instructions for running the submitted programs, and references used. (10% grade)
2. PDF report and well-documented source codes must be submitted in **one single Zip file with name format lastname_assign1.zip**.

3. Readability of the program is highly valued. Program comments has a weight of 10% of total points.

Few resources you can use:

- [1] Socket programming in python: <https://realpython.com/python-sockets/>
- [2] Understand proxy server: https://en.wikipedia.org/wiki/Proxy_server