CS 421 – Computer Networks

Programming Assignment I

HTTP Authorization and Range Requests

Due: March 19, 2021 at 11:59PM

1) Introduction

In this programming assignment, you are asked to implement a program in either **Java or Python 3**. You should code a program that downloads and processes data that is obtained from an HTTP Server.

The goal of the assignment is to make you familiar with the HTTP Protocol and TCP socket programming. You must implement your program using either the **Java Socket API of the JDK** or the **Socket** package in your default Python distribution. If you have any doubt about what to use or not to use, please contact your teaching assistant Ayhan Okuyan at **ayhan.okuyan[at]bilkent.edu.tr.** For Python, the use of any class/function from **http or the requests package is prohibited.**

When preparing your project please keep in mind that your projects will be first evaluated by a computer program. Any problems in the formatting can create problems in the grading. Errors caused by incorrectly naming the project files and folder structure will cause you to lose points.

2) Specifications

The server program that is provided to you (in Python 3) uses a HTTP/1.1 similar application layer protocol that uses TCP underneath to communicate with the client. In server side, there are three entities that you will separately download and process. For easy understanding, this project is divided into three parts, all contributing to each other. It is suggested that you start from the first part and continue sequentially. In order to work with these problems, first you will be running the server program. In order to run the program use the following command in the homework folder via terminal. Note that one should download the appropriate Python 3 software. This server code is written in **Python 3.7.7.**

```
python3 serv.py
```

This program is written such that it opens a socket and creates a listening port on your local machine on port 8000 (http://localhost:8000/). If you receive a binding error while running this program, port 8000 may be used by another process, so consider changing it.

A. First HTTP Connection and TCP Socket Programming

This first part of the project asks you to create a TCP Socket as an HTTP client and connect it to the socket that the server program is listening. Then retrieve the webpage data by sending a simple HTTP GET request, issued as follows:

```
GET <ENTITY> HTTP/1.1\r\n
Host: <SERVER HOST> \r\n\r\n
```

Where <ENTITY> is the filename that you are required to enter with a "/" in the beginning to indicate that and <SERVER HOST> is the domain URL that you are trying to reach e.g. localhost:8000/. When a webpage entity is requested for the first time, its 'index.html' page is requested, or you can simply leave the <ENTITY> as '/' to request this page. **Once the content is retrieved, store it as an HTML file with the name 'index2.html'.** Then you need to extract the information of the entity that you will try to obtain next, hidden in the HTML code. Parsing and obtaining the entity name is part of the assignment and your grade will be deducted in the absence of it. **In the report, provide a brief explanation as to how the GET requests and corresponding responses operate.**

B. HTTP Authentication

In the second part, you are asked to reach and download the content whose name you have covered in the previous part. However, this part of the site is protected with a Basic Authorization scheme. For further information, examine the HTTP Authentication page.

First, try to access the page using the method you have followed in the previous part. Then use a GET request with a Basic Authorization header. You are provided the username and password, **'bilkentstu'** and **'cs421s2021'** respectively. Use these credentials in the form **<USERNAME>:<PASSWORD>** and obtain a base64 encoding as the authentication key. You may face the following response codes from the server program.

STATUS CODE	NAME	FUNCTION
200	OK	The request has succeeded.
401	Unauthorized	The client must authenticate itself to get the requested response.
403	Forbidden	The client does not have access rights to the content; that is, it is unauthorized, so the server is refusing to give the requested resource.
404	Not Found	The server cannot find the requested page.

In your report answer these questions.

- Why did the first attempt resulted with an error code? Why do we use authorization?

- Why do we use an encoding when sending the authorization request?

Save the HTML file obtained from this page and save it as **'protected2.html'.** Then, repeat the previous part and extract the name of the entity that will be processed in the following part. Parsing and obtaining the entity name is part of the assignment and your grade will be deducted in the absence of it.

C. HTTP Range Requests

When downloading or streaming particularly large entities from the web e.g videos, It may not be feasible to do it with a single GET request. For these types of occasions, Range Requests are used to retrieve parts of the data. In order to be able to do that, you will send a **HEAD request** to retrieve information on the length of the data with the following format.

```
HEAD <ENTITY> HTTP/1.1\r\n
Host: <SERVER HOST> \r\n\r\n
```

Use this request to get the information on the text entity whose name is obtained in the previous part and the 'index.html' file. **In your report, explain what the received headers do and compare the two.** After covering the length, you should write a code to download the text file in **ranges [10,100,1000,10000,15000] bytes** and save the total download time for each case**. In your report, provide the execution times and the plot of execution time vs. range. Comment on the reason why the graph has the shape that you obtained. Discuss on what would happen if you try to download the text file with a single GET request (not using range requests). Also give a brief description of the challenges you have faced during the implementation process.** Since HTTP is a stateless protocol, client should be aware and track the arrival of the last part of the file with the information derived from the incoming headers.

Note that the server program is adopted for single client use and inherent persistent connection. This means after connecting the client, you won't be able to connect with another client and also you can use the open connection to send and retrieve data more than once, without the need of another client. Also note that, HTTP is a synchronous protocol which waits for the response from the server before sending another request (excluding HTTP2.0). Save all of the obtained files with the format **'big<RANGE>.txt'** where <RANGE> is the used range value for that execution**.**

The responses you may observe in this part are as follows.

STATUS CODE	NAME	FUNCTION
206	Partial Content	This response code is used when the `Range` header is sent from the client to request only part of a resource.
416	Requested Range is not Satisfiable	The requested byte range is not available and is out of bounds.
404	Not Found	The server can not find the requested page.

The persistent connection is established with the use of 'Connection' header in HTTP1.1. The connection stays active when connection is 'keep-alive' until the status is changed to 'close'. However, the connection between the server is implemented as a persistent one inherently in this assignment and in order to close the connection with the server, it is necessary to send an EXIT request which is defined as follows.

```
EXIT HTTP/1.1\r\n\
Host: <SERVER HOST> \r\n\r\n
```

3) Running your Program

Your program must be **a console application** (no graphical user interface, GUI, is allowed) and should be named as **httpclient.py** or **httpclient.java** based on your preference of language. Your program should not take any argument from the command line. You are free and encouraged to place print statements in your code to describe the functionality. **Please note that you must run your program after you start the server program**.

4) Final Remarks

- **Please contact your teaching assistant Ayhan Okuyan (ayhan.okuyan[at]bilkent.edu.tr) if you have any questions about the assignment.**
- **Do not forget to check the response message after sending each command to see if your code is working properly** and debug it if it is not. Note that the server cannot detect all the errors that you make; therefore, you might have to experiment to make sure that you correct all your errors.
- You can modify the source code of the server for experimental purposes. However, do not forget that your projects will be evaluated based on the version we provide.
- You might receive some socket exceptions if your program fails to close sockets from its previous instance. In that case, you can manually shut down those ports by waiting for them to timeout, restarting the machine, etc.
- Remember that all the commands must be constructed as strings and encoded with US-ASCII encoding.

- **Please put the downloaded files the under the same directory with the server and client codes.**

8) Submission rules

You need to apply all the following rules in your submission. You will lose points if you do not obey the submission rules below or your program does not run as described in the assignment above.

- The assignment must be submitted to Moodle**.** Any other methods (Email/Disk/CD/DVD/Cloud Drive) of submission will not be accepted.
- Zip all of the downloaded files, your report in PDF format, and your source code for submission. The submission should only include a single **ZIP** file. Any other compression is not accepted.
- The name of the zip file must be **AliVelioglu20141222** if your name and ID are **Ali Velioglu and 20141222**, respectively. If you are submitting an assignment done by two students, the file name should include the names and IDs of both group members like **AliVelioglu20141222AyseFatmaoglu20255666** if group members are Ali Velioglu and Ayse Fatmaoglu with IDs 20141222 and 20255666, respectively.
- For group submissions, **ONLY ONE MEMBER** must make the submission. The other member must **NOT** make a submission.
- All the files must be in the root of the zip file; directory structures are not allowed. Please note that this also disallows organizing your code into Java packages. The archive should not contain any file other than the source code(s) with .java or .py extension. The archive should not contain these:
  - Any class files or other executables including the pycache files,
  - Any third-party library archives (i.e., jar files),
  - Project files used by IDEs (e.g., JCreator, JBuilder, SunOne, Eclipse, Idea, PyCharm or NetBeans, etc.). You may, and are encouraged to, use these programs while developing, but the end result must be a clean, IDE-independent program.

- The standard rules for plagiarism and academic honesty apply; if in doubt refer to Academic Integrity Guidelines for Students and Academic Integrity, Plagiarism & Cheating.