CS-421

COMPUTER NETWORKS

PROJECT 1

Muhammet Musa Gezer

21703973

Section 1

Electrical and Electronics Engineering

24/03/2022

Introduction

The aim of this project assignment was to get familiar with the internals of the HTTP protocol. This program is supposed to download an index file to obtain a list of text file URLs and download some of these files depending on their sizes. I used python to write the program named Cloud Downloader. This report explains the important parts of the code.

URL Getter

URL getter is the first function used in the code. This function takes the URL of the index file and its authorization information (name and password, separated by a colon) to as arguments. Host name and path name are taken from the URL. The authorization information is encoded using base64 module as shown below:

```
base64.b64encode(authentication.encode('ascii'))
```

A get request header is created in the form given below:

GET <path>

Host: <host>

Authorization: Basic <encoded authentication information>

A socket is created using the socket module of python and it is connected to the hosts port 80 which is allocated for HTTP requests.

```
client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
client.connect((host, 80))
```

The request is encoded and sent to the client and a response with a header and the data is taken. They are returned separately after closing the socket.

Text Extractor

This function takes index file created in URL Getter function as input. By using regular expressions, it finds URLs, their authorization information and the bytes it contains. It creates a tuple of these three parameters and returns a list of them.

List Extractor

List extractor processes the list of authorization information for the servers that file is to be downloaded. The authentication information is encoded using base64 and get requests are created in order for all servers. A socket is created as in URL getter and requests are sent. In order to be able to obtain large chunks of data, a receive all function is created. The gathered data are merged after separated from the headers according to the index calculations which are done according to the given indexes and lengths of the acquired data. The status codes are checked for every response. In case of an unfavorable status codes, the program is exited after the status code is given. It returns final data in bytes.

Receive All

This function is used to get large responses. The response is taken 1024-byte chunks and merged in the function. The gathered data is returned as bytes and it used in List Extractor.

```
while True:

    pack = sock.recv(buffer_size)
    if not pack:
        break
    data += pack
```

The main code

The arguments given during running the program are taken using sys module. Th arguments are sent to URL getter to obtain the index file. The status code is checked. If there is no problem in status code (200), the data acquired is sent to the Text extractor. The output list of the Text extractor is sent to List Extractor. The final data acquired from List Extractor in bytes is taken and written into a file with the name given in the index file.

The format of the message is as shown below:

_____

Command-–line:

URL of the index file: dijkstra.cs.bilkent.edu.tr/~cs421/descriptor2.txt

File Name: Test2.txt

File size is 182700 bytes

Index file is downloaded

There are 5 servers in the index

Status Code: 200 OK

Connected to dijkstra.cs.bilkent.edu.tr/~cs421/partialt21.txt

Downloaded bytes 1 to 250 (size = 250)

Connected to dijkstra.cs.bilkent.edu.tr/~cs421/partialt22.txt

Downloaded bytes 251 to 101000 (size = 100750)

Connected to dijkstra.cs.bilkent.edu.tr/~cs421/partialt23.txt

Downloaded bytes 101001 to 127000 (size = 26000)

Connected to dijkstra.cs.bilkent.edu.tr/~cs421/partialt24.txt

Downloaded bytes 127001 to 162000 (size = 35000)

Connected to dijkstra.cs.bilkent.edu.tr/~cs421/partialt25.txt

Downloaded bytes 162001 to 182700 (size = 20700)

Download of the file is complete (size = 182700)

_____

References

[1] “Socket - low-level networking interface¶,” *socket - Low-level networking interface - Python 3.10.3 documentation*. [Online]. Available: https://docs.python.org/3/library/socket.html. [Accessed: 24-Mar-2022].