All Copyright reserved for CMPT 361, MacEwan University (Instructor: Dr. Mohammed Elmorsy)
Please don't distribute or post to any publicly accessible website

CMPUT 361 – Assignment 1

Online File Storage System

Dr. Mohammed Elmorsy

Weight: 10% **Total points: 100 points** **Due: October 10th, 2020 and before 11 PM**

I. Objectives:

In this assignment you will develop an online file hosting system that follows the description below. During your work in this assignment, you will learn how to program both clients and servers in a UNIX-like environment.

II. Description:

In this assignment, you will develop an **online file storage system** that serves a **SINGLE** client (called ***"user1"***). Using this system, the client can upload his/her files and view the metadata of the uploaded files (the file names, sizes and their uploaded date and time). During your work in this assignment, you will develop **TWO** python 3 programs:

- **Server.py**: which runs in the server machine
- **Client.py**: which runs in the client machine

A. Definitions

- ***"user1"***: the username of the user who is authorized to upload files to the server and is using Client.py
- Unknown user: a user who is **NOT** authorized to upload files to the server
- Client application (Client.py): the client program that runs on the authorized client machine.
- Client user: the user who is running the client program.
- Server program (Server.py): the program running on the server machine
- Server user: the user who is running the server program.

B. Server-side description:

1. The server should listen to the client connection on port number **13000**
2. The server machine contains in a **SINGLE** directory (called *Server*) the following files:
   - **Server.py**
   - **Database.json:** that contains the metadata of all the uploaded files. For each file, the database saves the file name, size (in bytes) and upload date and time. An example of the format that the file can contain:

```
{"A.pdf": {"size": "39982", "time": "2020-06-06 21:32:31.081387"}, "B.jpg": {"size": "2000000",
                                    "time": "2020-06-06 21:33:14.133515"}}
```

     For more information about JSON file formats, please see:
     - https://www.json.org/json-en.html
     - https://docs.python.org/3/library/json.html
   - The files that the client uploaded.

All Copyright reserved for CMPT 361, MacEwan University (Instructor: Dr. Mohammed Elmorsy)
Please don't distribute or post to any publicly accessible website

C. Client-side description:

The client machine contains in a **SINGLE** directory (called *Client*) the following files:

- Client.py
- All the files that the client wants to upload. Note, the client can upload any file type (e.g. PDF, JPG) and any file size (e.g. 100, 1000, 1000000 and 10000000).

D. Server/Client Communication Protocol:

After the connection is established between the client and the server. The following protocol takes place:

1. **[Server Side]:** The server sends the client the following message *"Welcome to our system.\nEnter your username: "*
2. **[Client Side]:** After receiving the server message, the client program:
   a. prints the server message,
   b. receives an input from the client user and sends it to the server.
3. **[Server Side]:** After receiving client response, the server checks:
   a. if the response is "user1", the server proceeds to step 4
   b. otherwise, the server sends the following message *"Incorrect username. Connection Terminated."* to the client and terminate the connection.
4. **[Server Side]:** The server sends to the client the following menu: *"\n\nPlease select the operation:\n1) View uploaded files' information\n2) Upload a file \n3) Terminate the connection\nChoice: "*
5. **[Client Side]:** After the client receives the server response:
   a. If the response is "*Incorrect username. Connection Terminated.*", the client program prints this message to the user and terminate the client connection with the server.
   b. Otherwise, the client program:
      – Prints the received server menu to the client user,
      – Receives an input from the client user and sends it to the server side. The accepted client inputs should be one of the following {"1", "2", "3"}.
6. **[Server side]:** After receiving the client response:
   a. If the received response is "1", **the file metadata viewing subprotocol** is performed between the client and the server.
   b. If the received response is "2", the **file upload subprotocol** is performed between the client and the server.
   c. Otherwise, the **connection termination subprotocol** is performed between the client and the server.
7. **[Server side and Client side]:** If the connection is not terminated, steps 4 to 7 are repeated.

E. File Metadata Viewing Subprotocol:

1. **[Server side]** The server gets all the uploaded files information from **Database.json,** converts it to a formatted string and sends the formatted string to the client side.
2. **[Client side]** After the client receives the server message, the client program prints the received message to the client user.

Note: The formatted string should follow the same format presented in the sample runs section.

All Copyright reserved for CMPT 361, MacEwan University (Instructor: Dr. Mohammed Elmorsy)
Please don't distribute or post to any publicly accessible website

F. File Upload Subprotocol

1. **[Server side]:** The server sends to the client the following message "Please provide the file name:"
2. **[Client side]:** After receiving the server message, the client:
    - **a.** prints the received message to the client user
    - **b.** receives an input from the client user that contains the file name
    - **c.** gets the file size
    - **d.** composes the following string "[filename]\n[filesize]" and sends this message to the server side where:
        - **i.** [filename] should be replaced by the file name that the client wants to upload
        - **ii.** [filesize] is the size of the file (in terms of Bytes) that the client wants to upload
3. **[Server side]:** After receiving the client response, the server sends the following message to the client "OK [filesize]" where [filesize] is the size of the file (in terms of Bytes) that the client wishes to upload.
4. **[Client side]:** After receiving the server response, the client prints this response to the client user and uploads the selected file to the server side. Then the client program prints the following message to the client user "*Upload process completed*".
5. **[Server side]:** After receiving the entire file, the server:
    a. Saves this file
    b. Update Database.json with the uploaded file metadata (name, size, upload date and time)

Note:
The server **MUST** make sure that it receives the entire file contents.

G. Connection Termination Subprotocol

1. **[Server side]:** The server terminates the connection
2. **[Client side]:** The client prints the following message *"Connection terminated."* And terminates the connection.

III. Sample runs:

The following sample runs show the terminal contents of the client side that are shown to the client user. The server side **MUST NOT** produce any output to the server user.

1. **Unauthorized client:**

```
$ python3 Client.py
Enter the server host name or IP: localhost
Welcome to our system.
Enter your username: user2
Incorrect username. Connection Terminated.
$
```

All Copyright reserved for CMPT 361, MacEwan University (Instructor: Dr. Mohammed Elmorsy)
Please don't distribute or post to any publicly accessible website

2. Authorized client uploads 3 files using the same connection, view their information and terminate the connection:

```
$ python3 Client.py
Enter the server host name or IP: localhost
Welcome to our system.
Enter your username: user1

Please select the operation:
1) View uploaded files' information
2) Upload a file
3) Terminate the connection
Choice:2
Please provide the file name: a.pdf
OK 39982
Upload process completed

Please select the operation:
1) View uploaded files' information
2) Upload a file
3) Terminate the connection
Choice:2
Please provide the file name: b.txt
OK 10
Upload process completed

Please select the operation:
1) View uploaded files' information
2) Upload a file
3) Terminate the connection
Choice:2
Please provide the file name: c.jpg
OK 1302245
Upload process completed

Please select the operation:
1) View uploaded files' information
2) Upload a file
3) Terminate the connection
Choice:1

Name            Size (Bytes)        Upload Date and time
a.pdf           39982               2020-06-09 20:29:15.481345
b.txt           10                  2020-06-09 20:29:26.957531
c.jpg           1302245             2020-06-09 20:29:45.247712

Please select the operation:
1) View uploaded files' information
2) Upload a file
3) Terminate the connection
Choice:3
Connection Terminated.
```

All Copyright reserved for CMPT 361, MacEwan University (Instructor: Dr. Mohammed Elmorsy)
Please don't distribute or post to any publicly accessible website

3. Authorized client opens a new connection and uploads a fourth file

```
$ python3 Client.py
Enter the server host name or IP: localhost
Welcome to our system.
Enter your username: user1

Please select the operation:
1) View uploaded files' information
2) Upload a file
3) Terminate the connection
Choice:2
Please provide the file name: d.txt
OK 50
Upload process completed

Please select the operation:
1) View uploaded files' information
2) Upload a file
3) Terminate the connection
Choice:1

Name                Size (Bytes)        Upload Date and time
a.pdf               39982               2020-06-09 20:29:15.481345
b.txt               10                  2020-06-09 20:29:26.957531
c.jpg               1302245             2020-06-09 20:29:45.247712
d.txt               50                  2020-06-09 21:29:13.207891

Please select the operation:
1) View uploaded files' information
2) Upload a file
3) Terminate the connection
Choice:3
Connection Terminated.
$
```

All Copyright reserved for CMPT 361, MacEwan University (Instructor: Dr. Mohammed Elmorsy)
Please don't distribute or post to any publicly accessible website

IV. Instructions:

1- This is an individual assignment.
2- The developed program:
   a. Must run with Python 3
   b. Must be well documented. The grader should be able to know the detailed logic of your code by reading the comments provided.
   c. Must **ONLY** import socket, sys, os, datetime and json modules. Your program **MUST NOT** import any other module.
3- Your submission should contain the following:
   a. The developed programs named "Server.py" and "Client.py"
   b. A word (or PDF) file that contains:
      - The status of your developed program and any deficiencies in this developed program
      - The tests conducted to verify the program operations (You must conduct some tests using two lab machines where one is used as a client and the other is used as the server). Also, you **MUST** include some sample runs' outputs.
4- The submission should be done to **the BlackBoard** before **11 PM, Saturday October 10th, 2020.**

V. Grading

Item	**Amount of points**
Implementation of the described features	80
Documentation Quality	10
Accuracy of the submitted text file and Quality of conducted tests	10
Total	100

Notes:

- Submissions, that fail to run, will not receive more than 50% of the available points.