INTERNET TECHNOLOGIES

ASSIGNMENT-1
SUBMISSION DATE:09-11-2020

TAMAGHNA SARKAR
BCSE-IV
GROUP-A2
ROLL: 001710501057

Problem Statement:

Implement a TCP-based key-value store. The server implements the key-value store and clients make use of it. The server must accept clients' connections and serve their requests for 'get' and 'put' key value pairs. All key-value pairs should be stored by the server only in memory. Keys and values are strings.
The client accepts a variable no of command line arguments where the first argument is the server hostname followed by port no. It should be followed by any sequence of "get <key>" and/or "put <key> <value>".
./client 192.168.124.5 5555 put city Kolkata put country India get country get city get Institute India
Kolkata
<blank>
The server should be running on a TCP port. The server should support multiple clients and maintain their key-value stores separately.
Implement authorization so that only few clients having the role "manager" can access other's key-value stores. A user is assigned the "guest" role by default. The server can upgrade a "guest" user to a "manager" user.

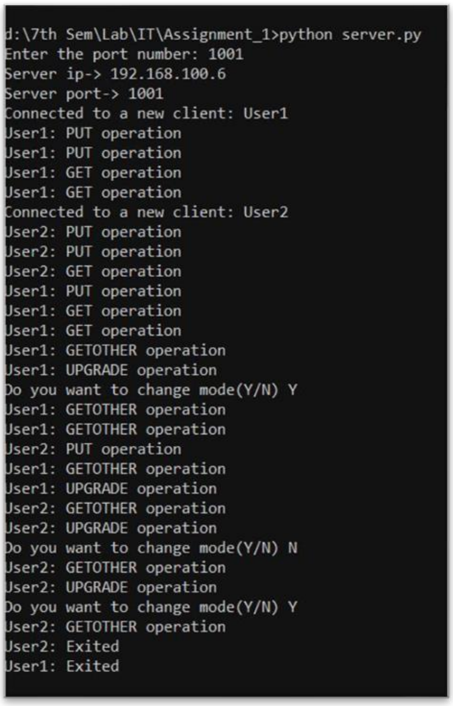
Design Details:

- Purpose of the program:
  - Multiple-clients can be connected to server using TCP sockets.
  - Server acts as the database where clients' information is stored.
  - Access to database in server by clients are determined by the status of the clients (Guest/Manager).
  - Multiple servers can be implemented by inputting a different port number when any server is created.

- Database:
  - Server has a dictionary where data for each client is stored. For each new client a new dictionary entry is created and it is automatically deleted when the client leaves the server.
  - As long the client is connected to the server, only the client can modify or add to its entry.
  - Client can retrieve its own data from the server but cannot access any other client's data while its status is "Guest". Only "Manager" can access all data. The server checks the status of the client only when it asks for any records other than its own.

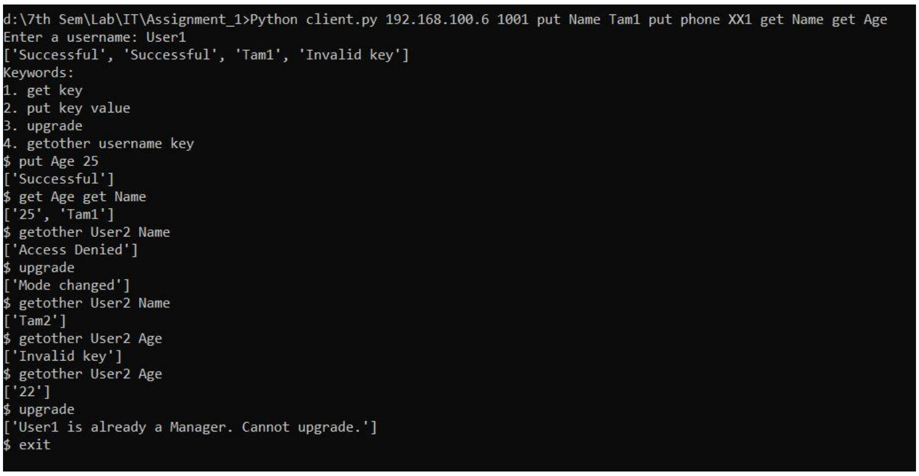
- Multiple-clients:
  - Each client is connected to the server using TCP sockets using the IP of the server and the port.
  - Server handles connection with each client in multiple threads running in the server. When a client disconnects, its thread is joined with the main thread of the server and subsequently its data in the database is deleted.
  - For a client to be a manager it has to sent a request to the server. The server can either accept its request and promote it to "Manager" or keep it as "Guest".
  - Once a "Manager", a client can access to all clients' information stored in the server.

Output Log:



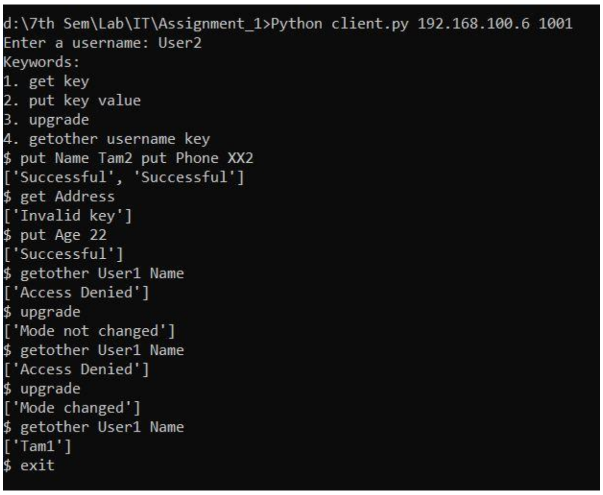
```
d:\7th Sem\Lab\IT\Assignment_1>python server.py
Enter the port number: 1001
Server ip-> 192.168.100.6
Server port-> 1001
Connected to a new client: User1
User1: PUT operation
User1: PUT operation
User1: GET operation
User1: GET operation
Connected to a new client: User2
User2: PUT operation
User2: PUT operation
User2: GET operation
User1: PUT operation
User1: GET operation
User1: GET operation
User1: GETOTHER operation
User1: UPGRADE operation
Do you want to change mode(Y/N) Y
User1: GETOTHER operation
User1: GETOTHER operation
User2: PUT operation
User1: GETOTHER operation
User1: UPGRADE operation
User2: GETOTHER operation
User2: UPGRADE operation
Do you want to change mode(Y/N) N
User2: GETOTHER operation
User2: UPGRADE operation
Do you want to change mode(Y/N) Y
User2: GETOTHER operation
User2: Exited
User1: Exited
```

SERVER
Port:1001



```
d:\7th Sem\Lab\IT\Assignment_1>Python client.py 192.168.100.6 1001 put Name Tam1 put phone XX1 get Name get Age
Enter a username: User1
['Successful', 'Successful', 'Tam1', 'Invalid key']
Keywords:
1. get key
2. put key value
3. upgrade
4. getother username key
$ put Age 25
['Successful']
$ get Age get Name
['25', 'Tam1']
$ getother User2 Name
['Access Denied']
$ upgrade
['Mode changed']
$ getother User2 Name
['Tam2']
$ getother User2 Age
['Invalid key']
$ getother User2 Age
['22']
$ upgrade
['User1 is already a Manager. Cannot upgrade.']
$ exit
```

CLIENT-1
Username: User1



```
d:\7th Sem\Lab\IT\Assignment_1>Python client.py 192.168.100.6 1001
Enter a username: User2
Keywords:
1. get key
2. put key value
3. upgrade
4. getother username key
$ put Name Tam2 put Phone XX2
['Successful', 'Successful']
$ get Address
['Invalid key']
$ put Age 22
['Successful']
$ getother User1 Name
['Access Denied']
$ upgrade
['Mode not changed']
$ getother User1 Name
['Access Denied']
$ upgrade
['Mode changed']
$ getother User1 Name
['Tam1']
$ exit
```

CLIENT-2
Username: User2

Comments

1. Multiple servers can handle multiple clients simultaneously.
2. Client's data is protected as only server can decide who are the manager and the guest clients.
