Internet Technology Lab Assignment 1 Report

by **Priti Shaw, 001710501076**
Batch: **A3**
Submission Date: **12th October,2020**

1. Problem Statement

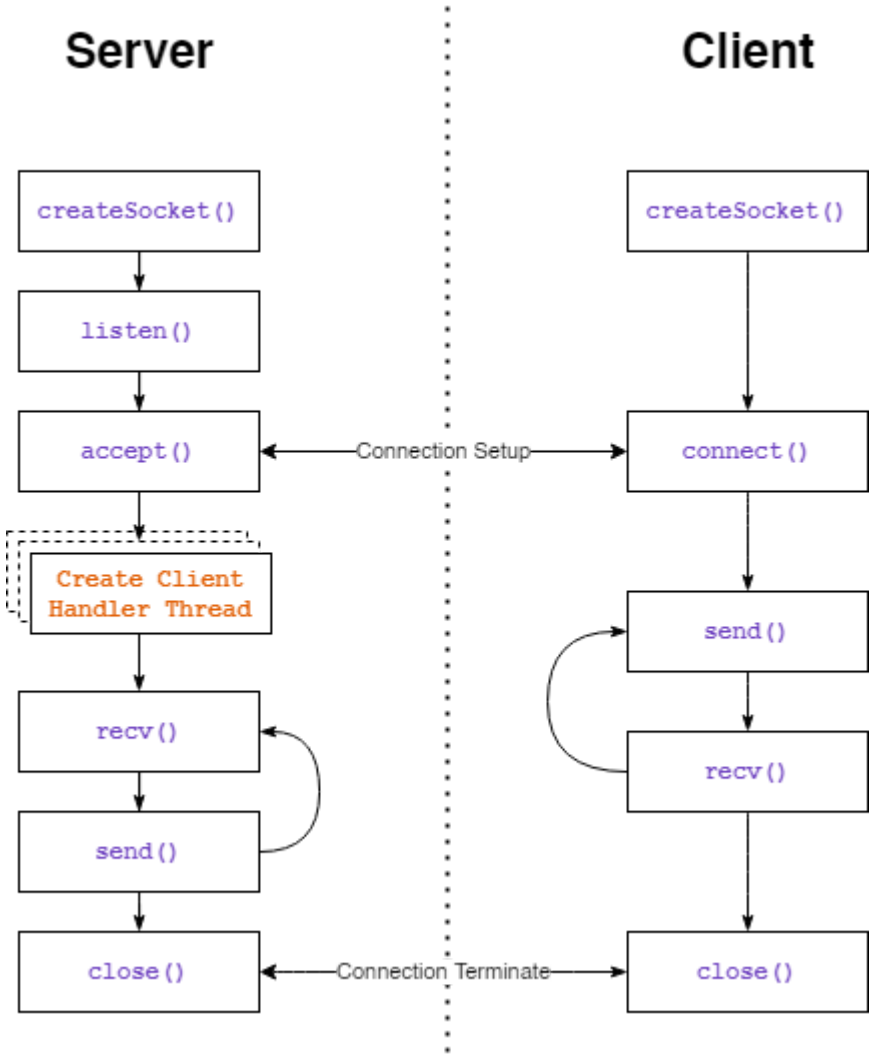
Implement a TCP-based key-value store. The server implements the key-value store and clients make use of it. The server must accept client's connections and serve their requests for *get* and *put* key value pairs.
All key-value pairs should be stored by the server only in memory. Keys and values are 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.

2. Solution Approach

TCP is a protocol used in transport layer of the OSI model, responsible for process to process communication. We need both IP address and port for creating a connection. Combination of IP address and Port number is known as socket.



The solution is divided into three parts. The first part consists of connection establishment i.e, TCP socket is created and the connection between client and server is established. The next part comprises of request processing i.e, PUT, GET, UPGRADE, GETALL instructions are handled. The final part deals with the response from the server.

3. Salient Features

- Every client is uniquely identified by the combination of *client's IP address and the incoming request port*.
- Can handle same *key* for different client
- Supports **multi-client** using the unique identification. Each client is given a thread with `ClientdatabseKeyValue` object

```
class ClientdatabseKeyValue:
    # variables
    valstore # Key-Value map for Client
    role     # Signifies present role of Client
    username # Unique Client Identifier

    # methods
    def getValue(key):          # Handle GET request
    def putValue(key, value):   # Handle PUT request
    def changeRole():           # To make present Client Manager
    def actionOnRequest(inst): # Handle incoming instruction from the Client
```

- Threading concepts has been used to handle concurrent connections.
  Each **thread**maintains seperate local key value mapping. The thread also maintains present role of the Client
- Multiple Servers can be run at different ports and each maintains seperate database.
- For handling the client role **manager**, the program maintains two key-value maps, one is global which holds key value for every client which is accessible only to the client whose role is manager and another map is locally available to each client's thread process for storing their key value pair.
- The **Manager** requests are served from the global key-value map.
- All instructions are through command line arguments.

4. Sample Input/Output

- Server started at default port 9999

```
pritishaw@DESKTOP-EMPON25:/mnt/e/JU/7/Internet_Technology/
Lab/Assignment1$ python3 server.py
Server started at 127.0.0.1:9999
```

- Client connection established

```
pritishaw@DESKTOP-EMPON25:/mnt/e/JU/7/Internet_Technology/Lab/Assignment1$ pyth
on3 client.py 127.0.0.1 9999 put city kolkata put country India get country get
 city get Institute
India
kolkata
<blank>
```

- Client with *Guest* role

```
pritishaw@DESKTOP-EMPON25:/mnt/e/JU/7/Internet_Technology/Lab/Assignment1$
python3 client.py 127.0.0.1 9999 getall country put city Pune get city
<blank>
Pune
```

- Client with *Manager* role

```
pritishaw@DESKTOP-EMPON25:/mnt/e/JU/7/Internet_Technology/Lab/Assignment1$ python3 client.py 1
27.0.0.1 9999 upgrade getall city
INFO: Role changed successfully
ClientIP        Port    Value
127.0.0.1       51934   kolkata
127.0.0.1       51941   Pune
```

- Multiple client

- Multiple Server