

# Key Value Store - Client Server using TCP

---

## Developed By:

---

- RADHIKA AGUMBE SRIDHAR 1550753
- AMRUTHAA RAJAN 1750155

## Introduction:

---

This project is an application with Client-Server communication through a socket. The application used here is Key-Value store.

## Implementation:

---

The following operations are supported:

1. PUT
2. GET
3. DELETE
4. QUIT

The client reads the input from the user and sends the same to the server. The processed data is sent back to the client from the server and is displayed in the output.

## Files:

---

We have two files namely `serverTCP.c` and `clientTCP.c`. `serverTCP.c` does various operations in server after reading the buffer and parsing them as characters from `char[]` array. The operations are done as separate cases in `switch()` by identifying the operation integer. `clientTCP.c` reads the input and writes it to the buffer in client which is a `char[]`. Operations, Keys and Values of the user-input are delimited by `:`, `;` and `\0`.

## Operations:

---

### 1. PUT:

- looks into buffer
- opens a file
- changes the file name as the given key
- puts the value in the file

- file close

## 2. GET:

- opens the file which is named after key
- gets the value in that file
- closes the file
- puts the value in the buffer

## 3. DELETE:

- checks for the availability of the file in the directory.
- if found, deletes the file.

## 4. QUIT:

- quits from the operation

## Usage:

---

### Server

- For compiling server:

```
gcc serverTCP.c -o server
```

- For starting server :

```
./server <portno>
```

### Client

- For compiling client:

```
gcc clientTCP.c -o client
```

- For running client :

```
./client hostname <portno>
```

Note: Server should be executed before the client.

## Example:

---

### Terminal 1:

```
$ gcc serverTCP.c -o server
$ ./server 9000
```

## Terminal 2:

```
$ gcc clientTCP.c -o client
$ ./client localhost 9000
```

## Walk Through:

### Server

```
$ ./server 9000
Service requested at: Sat Apr 22 21:25:45 2017

Service response sent to client at: Sat Apr 22 21:25:45 2017

Service requested at: Sat Apr 22 21:25:52 2017

Service response sent to client at: Sat Apr 22 21:25:52 2017

Service requested at: Sat Apr 22 21:26:03 2017

Service response sent to client at: Sat Apr 22 21:26:03 2017

KEY IS DELETED SUCCESSFULLY
Service requested at: Sat Apr 22 21:26:10 2017

File not found
Service requested at: Sat Apr 22 21:26:18 2017
```

### Client

```
$ ./client localhost 9000
Enter the operation
1: PUT, 2: GET, 3:DELETE, 4:QUIT
1
Option: 1
Enter the key
some-key
Key entered: some-key
Enter the value
some-value
Value is some-value
PUT service is requested at: Sat Apr 22 21:25:45 2017
Key Value Added
PUT service response is received at: Sat Apr 22 21:25:45 2017
Enter the operation
1: PUT, 2: GET, 3:DELETE, 4:QUIT
2
Option: 2
Enter the key
some-key
Key: some-key
GET service is requested at: Sat Apr 22 21:25:52 2017
Value: some-value
GET service response is received at: Sat Apr 22 21:25:52 2017
Enter the operation
1: PUT, 2: GET, 3:DELETE, 4:QUIT
3
```

```
Option: 3
Enter the key
some-key
DELETE service is requested at: Sat Apr 22 21:26:03 2017
Key: some-key
KEY IS DELETED SUCCESSFULLY
DELETE service response is received at: Sat Apr 22 21:26:03 2017
Enter the operation
1: PUT, 2: GET, 3:DELETE, 4:QUIT
2
Option: 2
Enter the key
some-key
Key: some-key
GET service is requested at: Sat Apr 22 21:26:10 2017
Value: KEY NOT FOUND
GET service response is received at: Sat Apr 22 21:26:10 2017
Enter the operation
1: PUT, 2: GET, 3:DELETE, 4:QUIT
4
Option: 4
QUIT operation is requested at: Sat Apr 22 21:26:18 2017
```