Contents

List of Figures

1 Introduction

File server with backend authentication server has been implemented using TCP sockets in C++. There are two servers: Server1 and Server2. Server1 (File server) receive TCP requests from client and requests a username and password from client for either creating new account or for requesting a file. Server1 will send this username and password to Server2 (Backend authentication server) to authenticate the user or add new user account. Once authenticated, the client can then request files from the server1 and if file is present, server1 will send the file to client.

Server1 is multithreaded to handle multiple client requests. Server2 will is multiprocess to handle multiple authentication requests from Server1. The choice to make server1 multi threaded and server2 multi-process is made to compare the performance of these in next phases of project.

2 Architecture

Multiple clients can connect to file server(server1). Once connected, client sends username and password to server1 for either creating new account or to authenticate himself for fetching file. The same will be sent to Authentication server (server2) for creation of account or authentication. Once authenticated, client will send filename with extension to server1 and will download file from server1 if file exists.

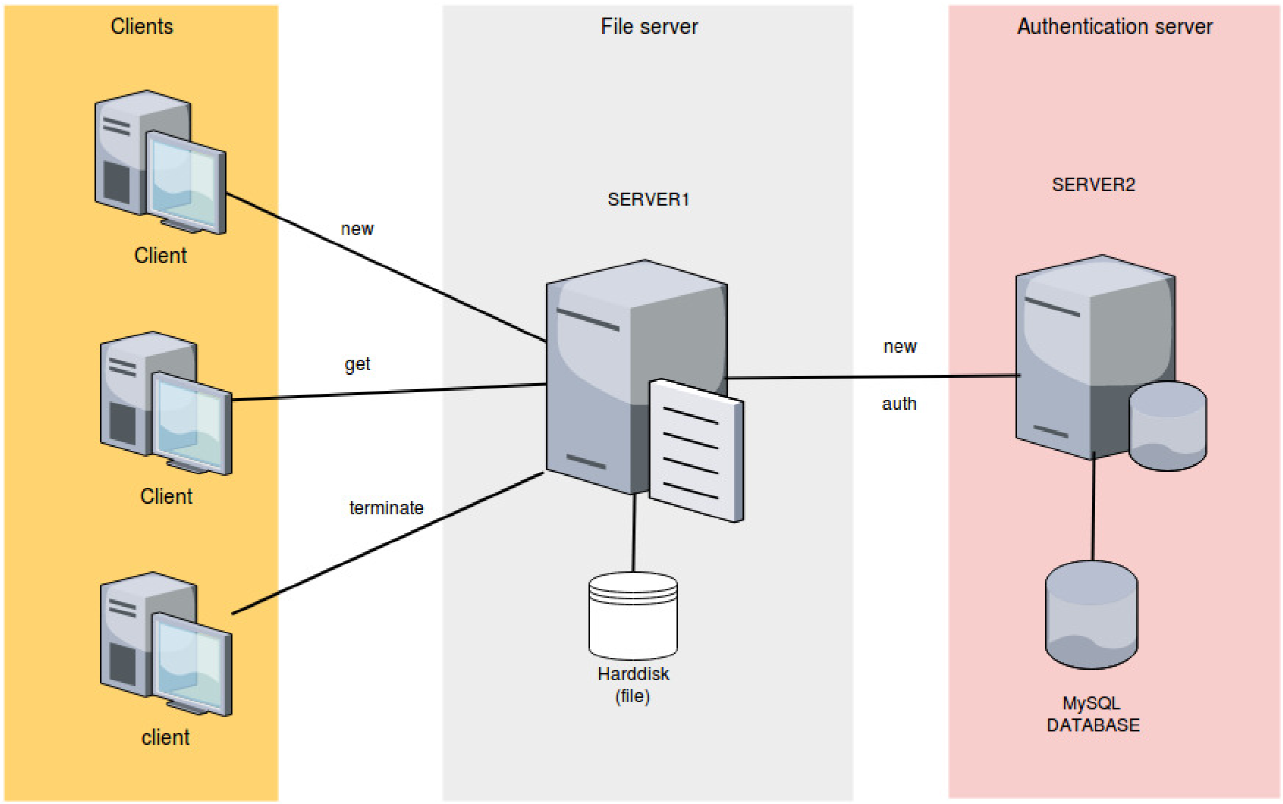


Figure 1: System Architecture

2.1 Client : Requests

Client has three options(or Type of requests):

- **To create new user account**. In this case, "new" keyword will be sent to Server1. Followed by username and password. Server1 will send "new" keyword to Server2 with username and password. Server2 will try to create user account.If some user account with that username exists, account will not be created else new user account will be created. Same will be informed to server1 and server1 will inform the same to client.

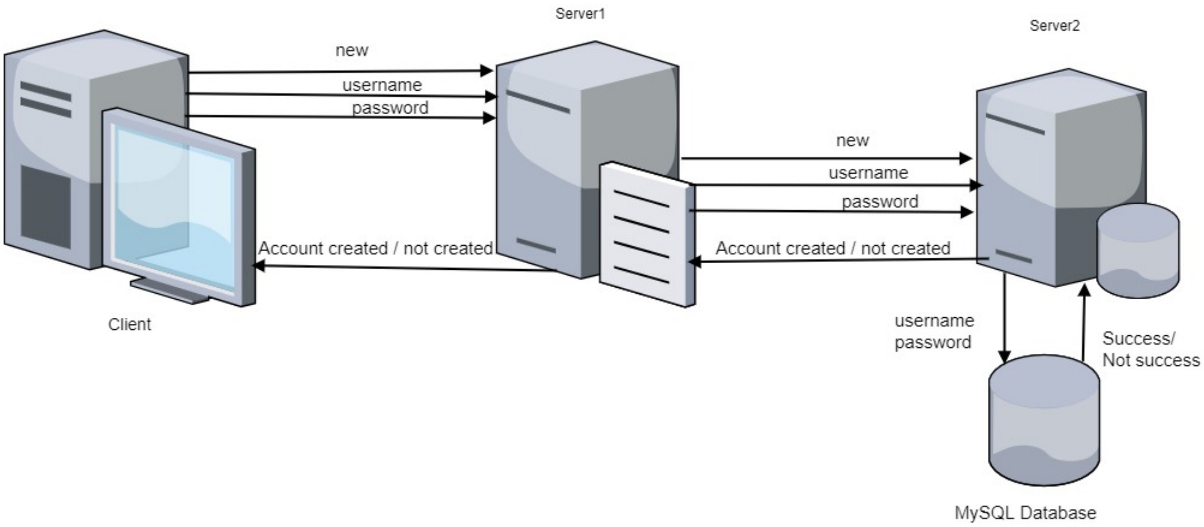


Figure 2: "New" request for creating new account

- **To request file from server1**. In this case, "get" keyword will be sent to server1. Followed by username and password. Server1 will send "auth" keyword to Server2 with username and password. Server2 will check the database for user account.If user details are correct, server1 will be informed and server1 will request filename from user. Client will then send filename with extension to server1(File server) and then file will be sent to client if that file exists at server1.

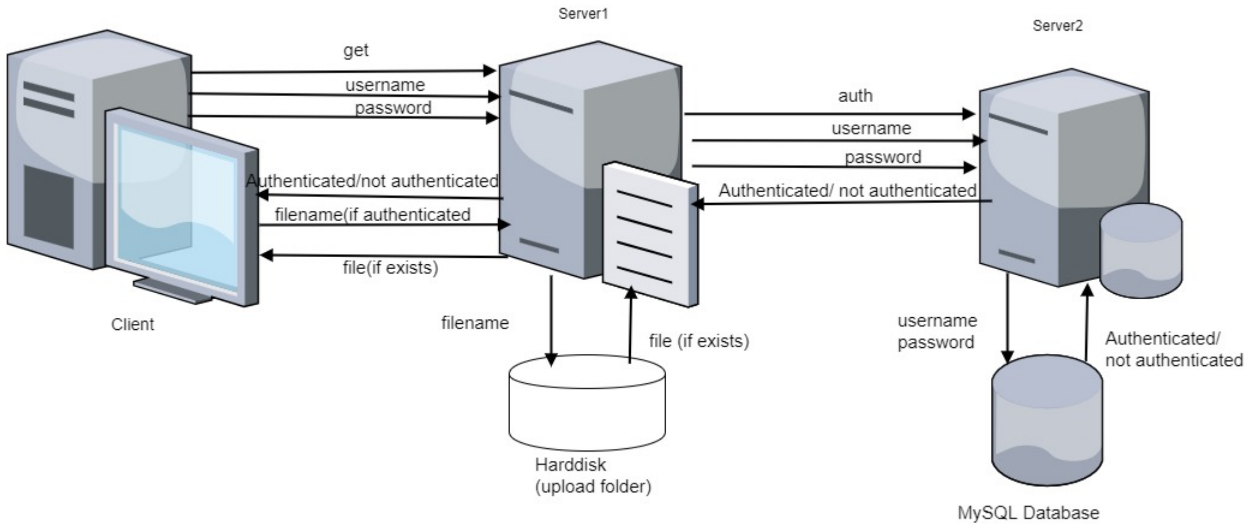


Figure 3: "Get" request for fetching file

- **To terminate the connection to server1**. In this case, "ter" keyword will be sent to server1 and connection will be terminated.

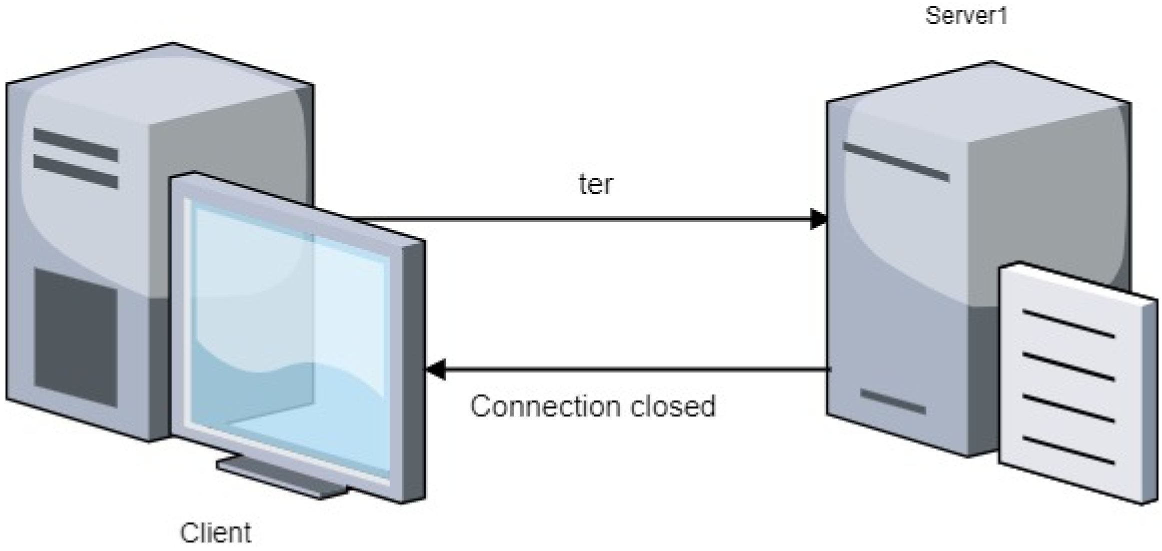


Figure 4: "Terminate" request for fetching file

2.2 Server2: Authentication server

Server2 is multiprocess server [1] where each new connection request from server1 is handled by different process.

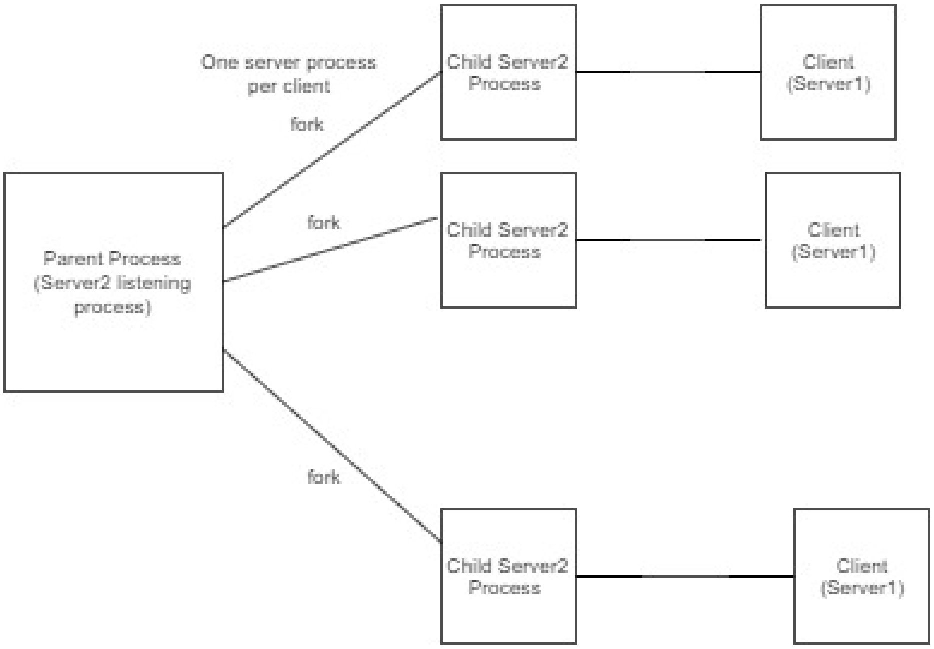


Figure 5: Multiprocess server :Server2

2.3 Server1 : File server

Server1 is multithreaded where each new connection request from client is handled by different thread.

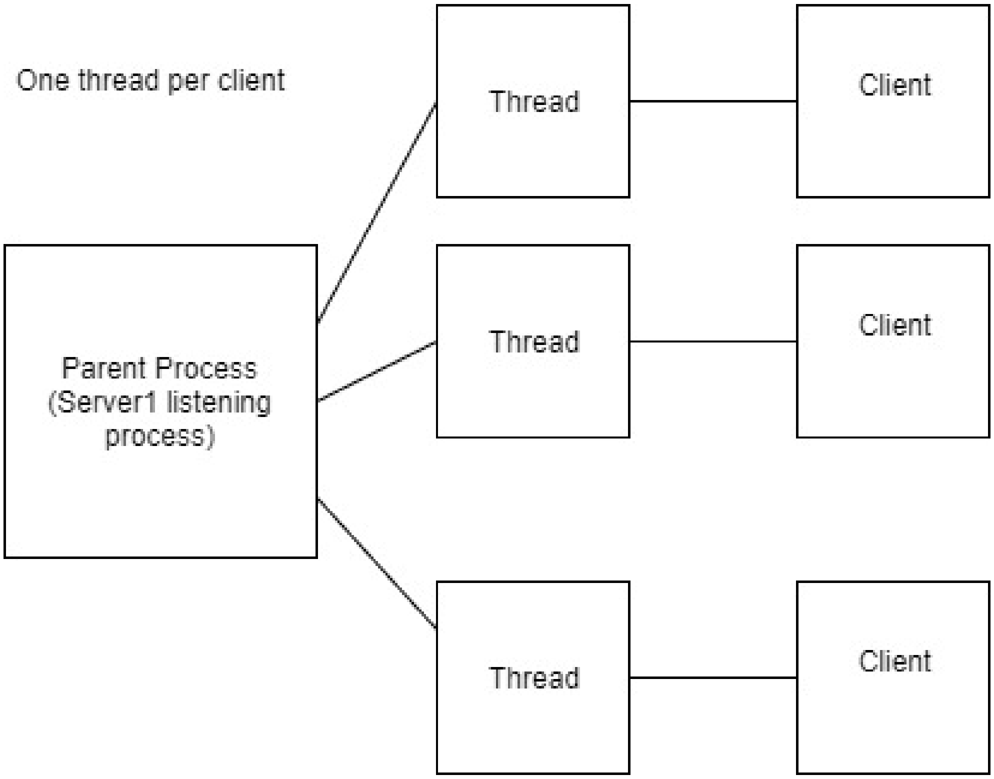


Figure 6: Multithreaded server : Server1

3 Implementation details

3.1 Server2 : Authentication server

Server2 (Backend authentication server) receives username and password from Server1 for either creating new account or to authenticate the user. It will have MySQL server installed. Step by step procedure for installation is as follows:

1. Command to install MySQL:
   sudo apt-get install mysql-server

2. To log in to MySQL as the root user:
   mysql -u root -p

    //USER ”root”
    //PASSWORD ”root”

3. To create a New MySQL Database ”ems_data”

    ```
    create database ems_data;
    use ems_data;
    ```

4. To create table ”user” with attributes ”user_name” and ”passwd”

    ```
    Field — Type — Null — Key — Default
    user_name — varchar(20) — NO — PRI — NULL —
    passwd — varchar(20) — NO — — NULL —
    ```

    ```
    CREATE TABLE user (user_name VARCHAR(20) NOT NULL PRIMARY, passwd varchar(20) NOT NULL);
    ```

5. To insert values in table

    ```
    INSERT INTO user VALUES (’username’,’password’);
    ```

6. To export databases:

    ```
    mysqldump -u <Username>-p <DatabaseName>><wantedsqlfile.sql>
    ```

7. To import database:

    ```
    mysql -u <username>-p <databasename><<filename.sql>
    ```

In order to make MySQL commands from C++ program, there is requirement to install required libraries for including header files. The command to install same:

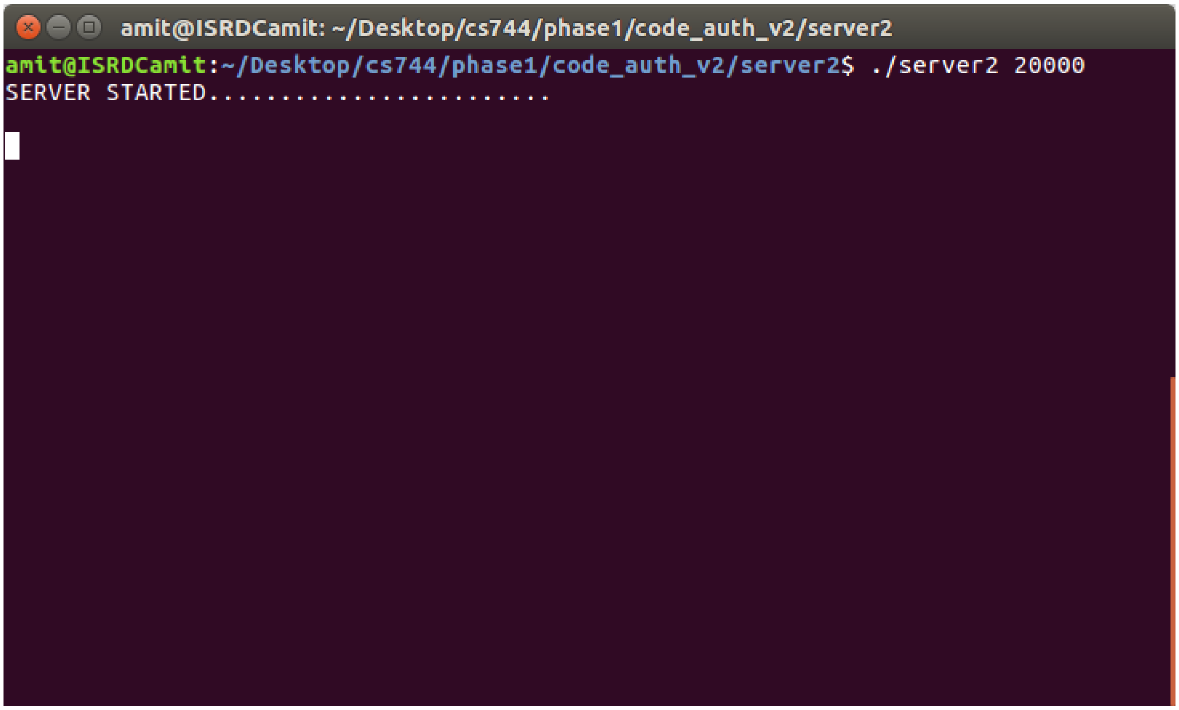
```
sudo apt-get install libmysqlclient-dev
```

Above steps have been consolidated in two shell scripts provided in code folder:

```
installmysql.sh
importdatabase.sh
```

Makefile has also been provided to compile C++ code. Steps to run Server2:

```
./server2 Listeningport
```



```
amit@ISRDCamit:~/Desktop/cs744/phase1/code_auth_v2/server2$ ./server2 20000
SERVER STARTED......................
```

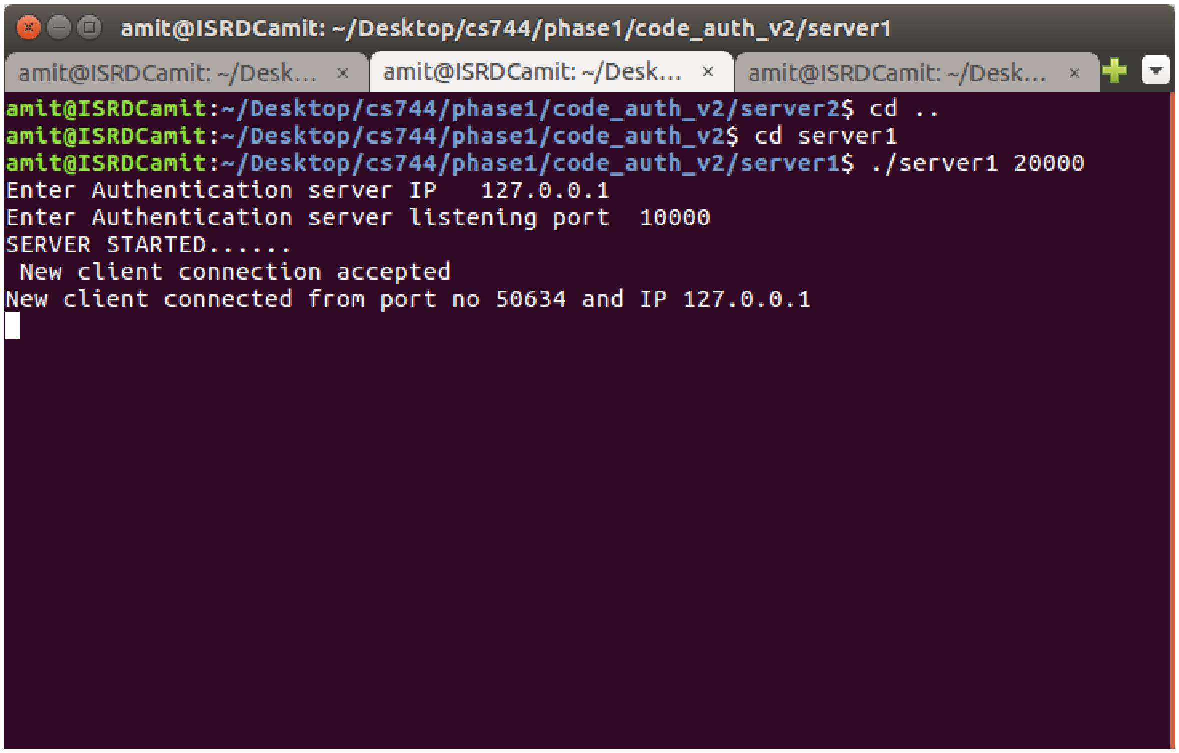
Figure 7: Screenshot of Server2 terminal.

3.2 Server1 : File server

Server1 receives username and password from client for either creating new account or to authenticate the user for fetching file and will send this username and password to Server2(Authentication server). Once authenticated by Server2, Server1 will get filename from client and will provide file to client if file exists. Makefile has been provided to compile C++ code. Steps to run Server1:

```
./server1 Listeningport
```

The command prompt message to enter Authentication server IP address and listening port.


amit@ISRDCamit: ~/Desktop/cs744/phase1/code_auth_v2/server1
amit@ISRDCamit: ~/Desk... ×
amit@ISRDCamit: ~/Desk... ×
amit@ISRDCamit: ~/Desk... ×

```
amit@ISRDCamit:~/Desktop/cs744/phase1/code_auth_v2/server2$ cd ..
amit@ISRDCamit:~/Desktop/cs744/phase1/code_auth_v2$ cd server1
amit@ISRDCamit:~/Desktop/cs744/phase1/code_auth_v2/server1$ ./server1 20000
Enter Authentication server IP   127.0.0.1
Enter Authentication server listening port  10000
SERVER STARTED......
 New client connection accepted
New client connected from port no 50634 and IP 127.0.0.1
```

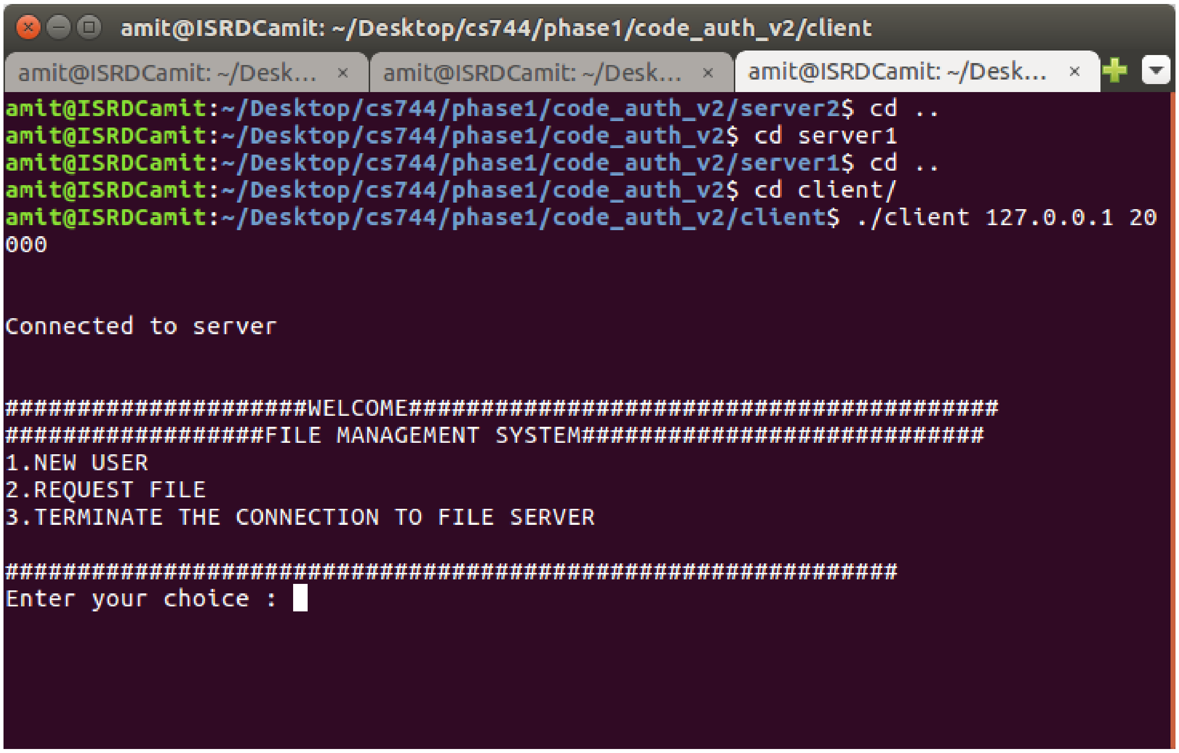
Figure 8: Screenshot of Server1 command prompt message.

3.3 Client

Client sends username and password to server for either creating new account or to authenticate himself for fetching file. Once authenticated, client will send filename to server1 and will downlaod file from server1 if file exists. Makefile has been provided to compile C++ code. Steps to run client:

./client Server1_IP Server1_Listeningport

Follow command prompt message for further details/requests.


amit@ISRDCamit: ~/Desktop/cs744/phase1/code_auth_v2/client
amit@ISRDCamit: ~/Desk...
amit@ISRDCamit: ~/Desk...
amit@ISRDCamit: ~/Desk...

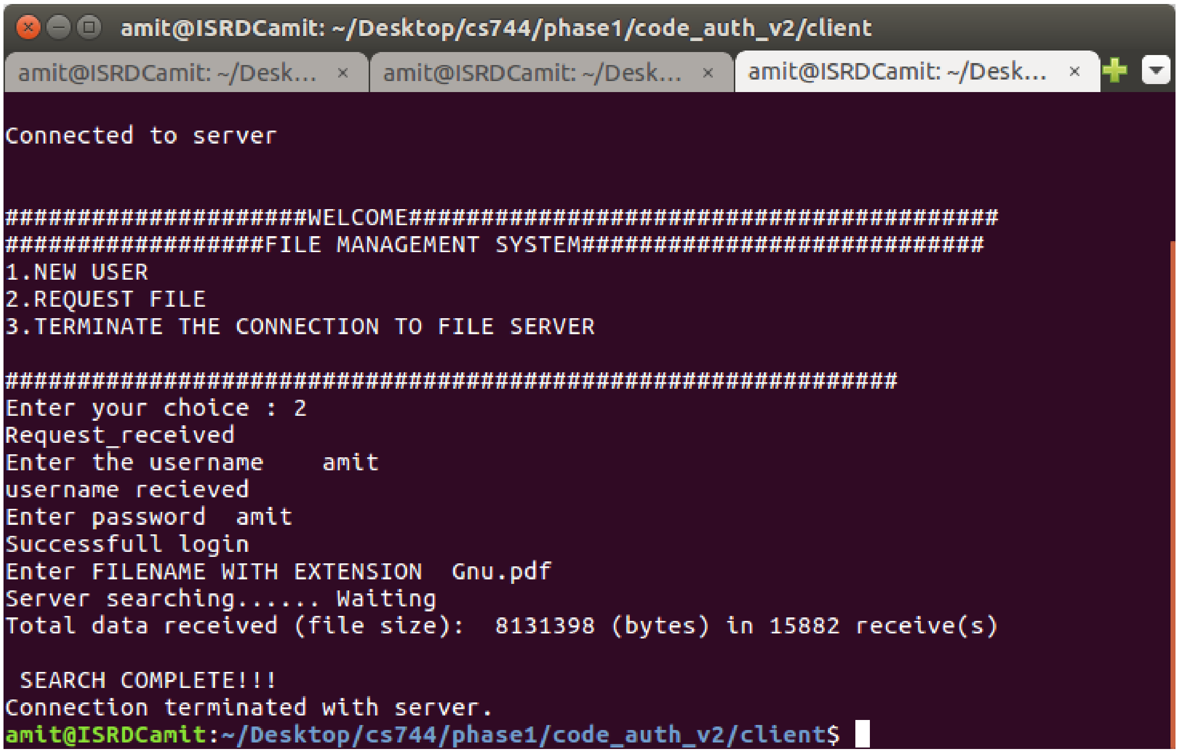
```
amit@ISRDCamit:~/Desktop/cs744/phase1/code_auth_v2/server2$ cd ..
amit@ISRDCamit:~/Desktop/cs744/phase1/code_auth_v2$ cd server1
amit@ISRDCamit:~/Desktop/cs744/phase1/code_auth_v2/server1$ cd ..
amit@ISRDCamit:~/Desktop/cs744/phase1/code_auth_v2$ cd client/
amit@ISRDCamit:~/Desktop/cs744/phase1/code_auth_v2/client$ ./client 127.0.0.1 20
000


Connected to server


#####################WELCOME#########################################
##################FILE MANAGEMENT SYSTEM############################
1.NEW USER
2.REQUEST FILE
3.TERMINATE THE CONNECTION TO FILE SERVER

##############################################################
Enter your choice : 
```

Figure 9: Screenshot of Client command prompt message.


amit@ISRDCamit: ~/Desktop/cs744/phase1/code_auth_v2/client
amit@ISRDCamit: ~/Desk... ×
amit@ISRDCamit: ~/Desk... ×
amit@ISRDCamit: ~/Desk... ×

```
Connected to server


#####################WELCOME#########################################
##################FILE MANAGEMENT SYSTEM############################
1.NEW USER
2.REQUEST FILE
3.TERMINATE THE CONNECTION TO FILE SERVER

##############################################################
Enter your choice : 2
Request_received
Enter the username    amit
username recieved
Enter password  amit
Successfull login
Enter FILENAME WITH EXTENSION  Gnu.pdf
Server searching...... Waiting
Total data received (file size):  8131398 (bytes) in 15882 receive(s)

 SEARCH COMPLETE!!!
Connection terminated with server.
amit@ISRDCamit:~/Desktop/cs744/phase1/code_auth_v2/client$ 
```

Figure 10: Client authentication and file download procedure

4 Submission folder

There are three folders inside : client, server1 and server2. The folder/directory structure and files are listed below:

4.1 Client

- Makefile
- client.cpp : Client code.
- ”downloads” folder : to store received files.

4.2 Server1

- Makefile

- server1.cpp : File server code.
- ”uploads” folder : to store files available for sharing/download.

4.3 Server2

- Makefile
- server2.cpp : Authentication server code.
- installmysql.sh : script to install MySQL server and required libraries for header files.
- importdatabase.sh : script to import database and user table from cs744.sql file.
- cs744.sql : contains MySQL database and table for import.

References

[1] Dr. Mostafa Hassan Dahshan. faculty.ksu.edu.sa/mdahshan/cen463fa09/07-file_transfer_ex.pdf.