CSCI537

Introduction to Distributed Computing

Report on

FTP Using Socket
A2

Dept. of Computer & Information Science,

IUPUI

Name: Keyur Kirti Mehta

Contents

1. Introduction

The assignment is to implement file transfer protocol using the socket communication Java API. Designing of client server architecture which will transfer the file based on the client's request. The client (user) needs to first register and logged in before requesting the file transfer. The file should be encrypted before transmitted and checksum should be verified upon receive.

The assignment is implemented using multi-threaded server so as multiple client can be connected and request for file transmission.

2. Design Decisions and Implementation

1. Socket Communication

The assignment is implemented using Java Socket API. The server is running on CSCI machine rrpc01 (10.234.136.55) and listing the requests on port 9001. Server will open port and listen to client's request on socket port. Once client sends request on ip-port a new server thread will be created for that specific client. For each new client connection a separate server thread is started. So, multiple client can simultaneously request the file.

The communication between client and server will take place using object. A user class object will act like session object for client. All the information will be send across in that user object. User class object holds below mentioned information:

- Username
- Password
- File name
- Message ID (to identify the request/ response type)
- User authentication flag
- File checksum

Once server and client is started, it will initiate the Object input stream and Object output stream for both of them using ObjectOutputStream and ObjectInputStream class. These stream will write and read on other stream using socket's output stream and input stream. So, client and server will write/read objects using writeObject and readObject method.
File will be transferred using DataStream. On the same stream file data will be written using write method and will be read using read method.

2. User Authentication and Registration

Once the client starts, user needs to register using username and password. Server maintains one common list of all the registered user. A HashMap is created when server is started which stores all the username and their corresponding passwords. So, once the client session is finished, and new session is started on same or different machine, client can authenticate using earlier credentials. As server doesn't use any persistent storage, once it is stopped all the users' credentials will be vanished. Next time the user needs to register again in order to request for file transmission.

If the username is already occupied, then the client will be prompted Username exists.
If the user enter invalid credentials then the user will be prompted for incorrect login credentials.

Pros/Cons:

As credentials are verified at the end only authenticate user can access the system.

3. File Transfer

Once the registered user is logged in, the system will ask for the file name to enter. User needs to enter the file name along with the extension. Eg. Story.txt All the files are placed under the 'Docs' folder. Currently 5 sample files are placed in the folder for transmission

```
  53700-1.ppt, Area51.txt, Gandhi.txt, Story.txt, temp1.txt
```

Any new file which needs to be transmitted can be placed in 'Docs' folder.

Server may exhibit Byzantine Behavior while transmitting the file content. Server many choose to update the last byte of 1st iteration to 0 based on the probability value of 0.4

```
if(flag == 0)
      if(rand.nextInt(10) < 4 ){
            System.out.println("Server's Byzantine behavior");
            buffByte[readByte - 1] = 0;
      }
```

If the requested file is not present then user will receive the message and session will be terminated.

End to End check:

1. Encryption-Decryption

The file is encrypted before transmission and decrypted upon receive on the client side. The system uses excess 3 encryption technique in order to encrypt the content of the file. The content of file is read as Byte and the encrypted. Each time 1024 bytes of data is transferred.

For encryption	For decryption
buffByte[i] = (byte) ((buffByte[i] + 3))	buffByte[i] = (byte) ((buffByte[i] + 3))

2. Verify Checksum

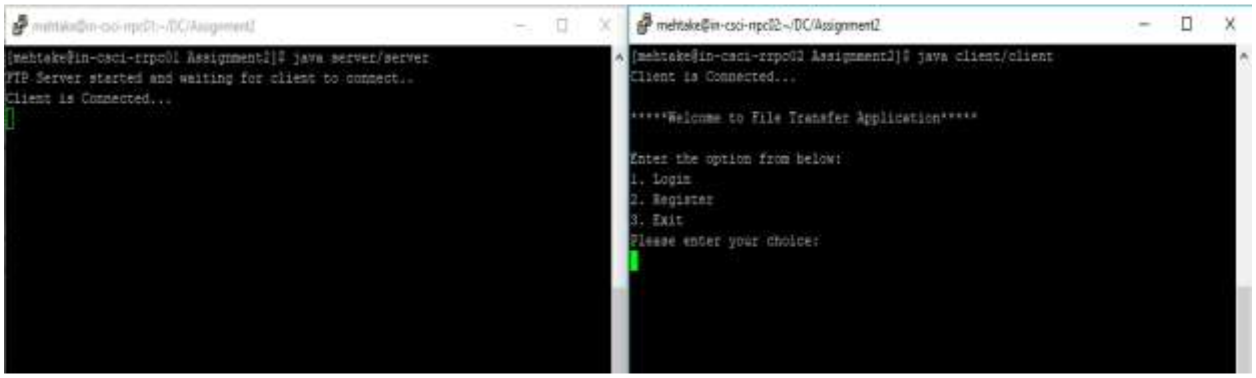
The server will send the checksum of the transmitted file and the client will calculate the checksum of the received file. If both checksum match then the file transmission is successful and client has received the correct file. And the session is terminated. If the checksum is not matched then the client will request for retransmission. If client receives the incorrect file 5 times then it will terminate the session and the corrupted file is deleted. The java library for checksum calculation is used. The checksum is calculated by using MD5 algorithm.

Pros/Cons:

As the file content is encrypted and decrypted at the end. Though the network is not trusted, the file transmission will be secured. Also at the end, file checksum is verified in order to ensure the correct file is received.

3. Screenshots

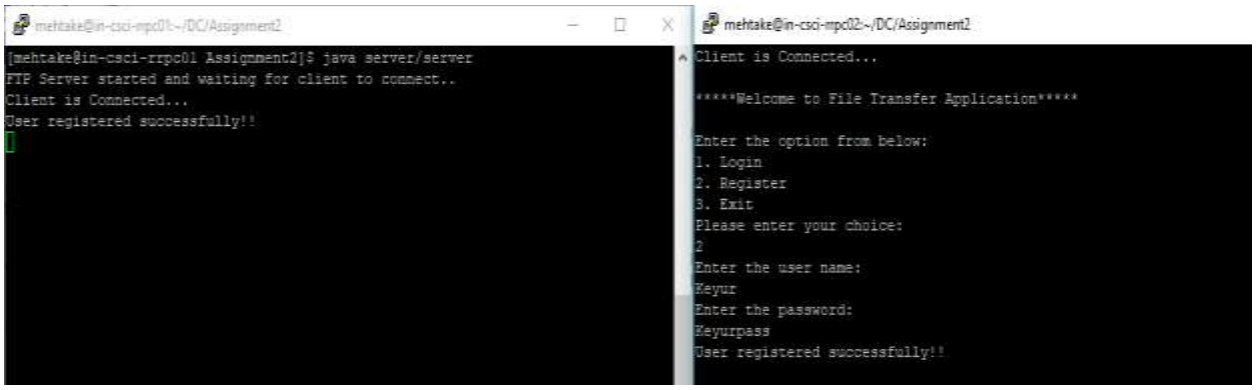
1. Client Connected


mehtake@in-csci-rrpc01:~/DC/Assignment2
mehtake@in-csci-rrpc02:~/DC/Assignment2

```
[mehtake@in-csci-rrpc01 Assignment2]$ java server/server
FTP Server started and waiting for client to connect..
Client is Connected...
```

```
[mehtake@in-csci-rrpc02 Assignment2]$ java client/client
Client is Connected...

*****Welcome to File Transfer Application*****

Enter the option from below:
1. Login
2. Register
3. Exit
Please enter your choice:
```

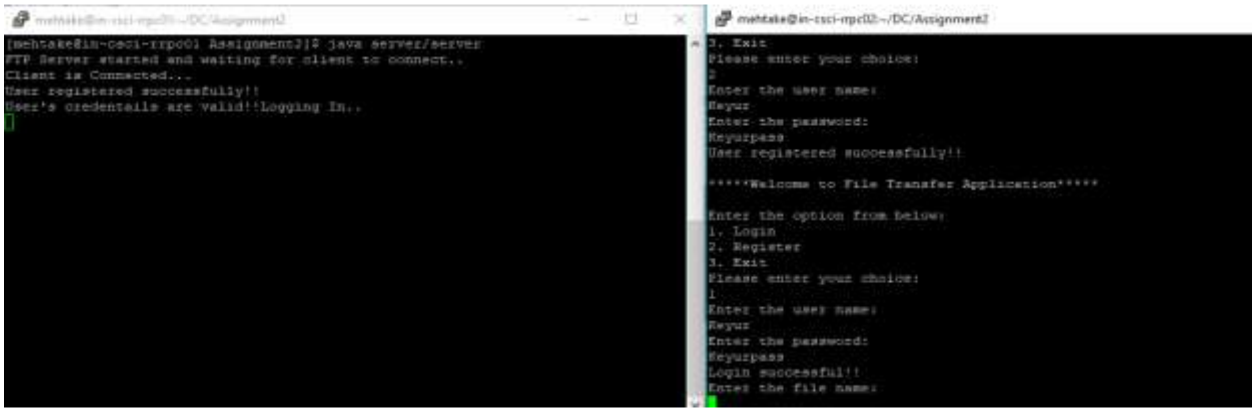
2. User Registration


mehtake@in-csci-rrpc01:~/DC/Assignment2
mehtake@in-csci-rrpc02:~/DC/Assignment2

```
[mehtake@in-csci-rrpc01 Assignment2]$ java server/server
FTP Server started and waiting for client to connect..
Client is Connected...
User registered successfully!!
```

```
Client is Connected...

*****Welcome to File Transfer Application*****

Enter the option from below:
1. Login
2. Register
3. Exit
Please enter your choice:
2
Enter the user name:
Keyur
Enter the password:
Keyurpass
User registered successfully!!
```

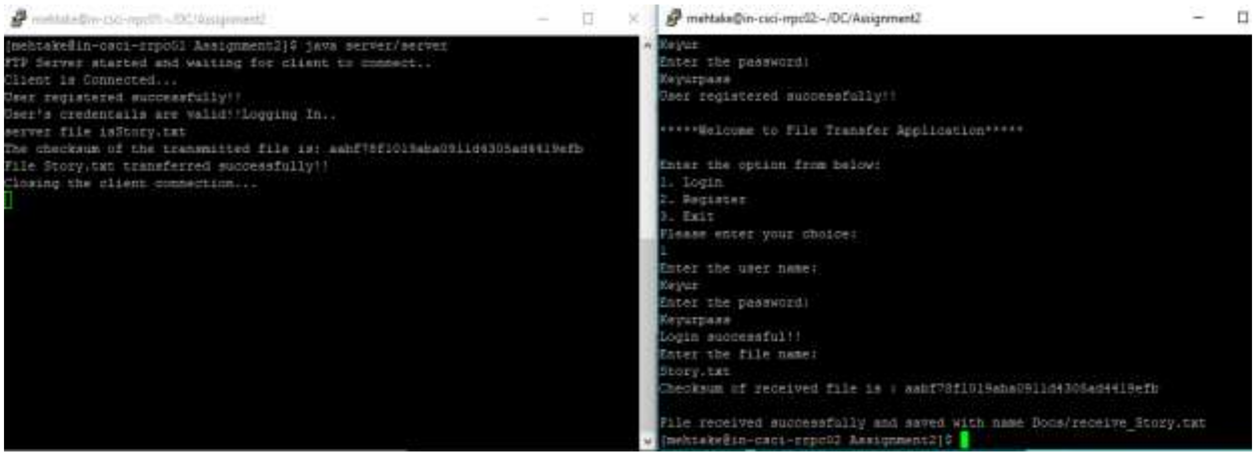
3. User Login


mehtake@in-csci-rrpc01:~/DC/Assignment2
mehtake@in-csci-rrpc02:~/DC/Assignment2

```
[mehtake@in-csci-rrpc01 Assignment2]$ java server/server
FTP Server started and waiting for client to connect..
Client is Connected...
User registered successfully!!
User's credentails are valid!!Logging In..
```

```
3. Exit
Please enter your choice:
2
Enter the user name:
Keyur
Enter the password:
Keyurpass
User registered successfully!!

*****Welcome to File Transfer Application*****

Enter the option from below:
1. Login
2. Register
3. Exit
Please enter your choice:
1
Enter the user name:
Keyur
Enter the password:
Keyurpass
Login successful!!
Enter the file name:
```

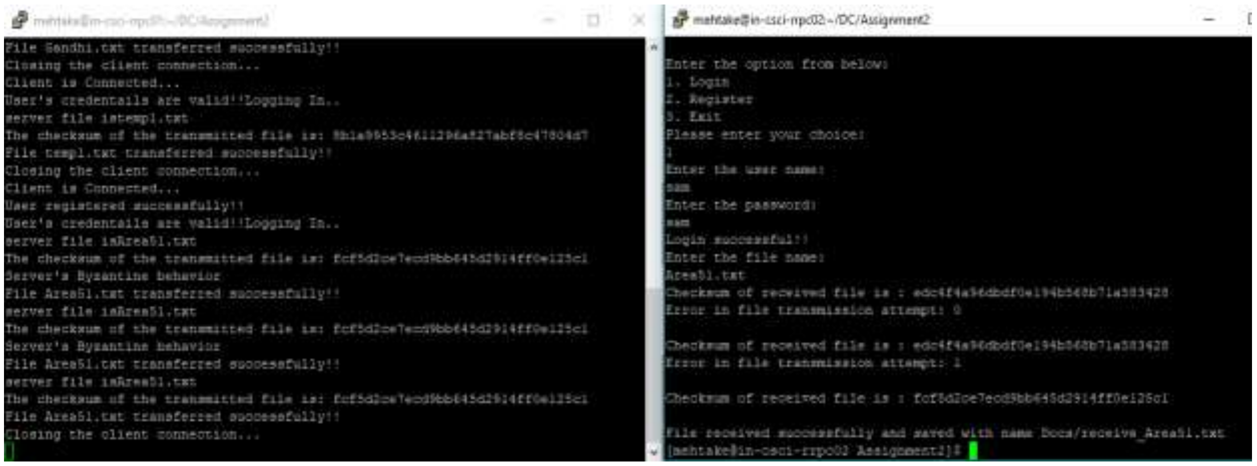
4. File transfer


mehtake@in-csci-rrpc01:~/DC/Assignment2
mehtake@in-csci-rrpc02:~/DC/Assignment2

```
[mehtake@in-csci-rrpc01 Assignment2]$ java server/server
FTP Server started and waiting for client to connect..
Client is Connected...
User registered successfully!!
User's credentails are valid!!Logging In..
server file isStory.txt
The checksum of the transmitted file is: aabf78f1019aba0911d4305ad4419efb
File Story.txt transferred successfully!!
Closing the client connection...
```

```
Keyur
Enter the password:
Keyurpass
User registered successfully!!

*****Welcome to File Transfer Application*****

Enter the option from below:
1. Login
2. Register
3. Exit
Please enter your choice:
1
Enter the user name:
Keyur
Enter the password:
Keyurpass
Login successful!!
Enter the file name:
Story.txt
Checksum of received file is : aabf78f1019aba0911d4305ad4419efb

File received successfully and saved with name Docs/receive_Story.txt
[mehtake@in-csci-rrpc02 Assignment2]$
```

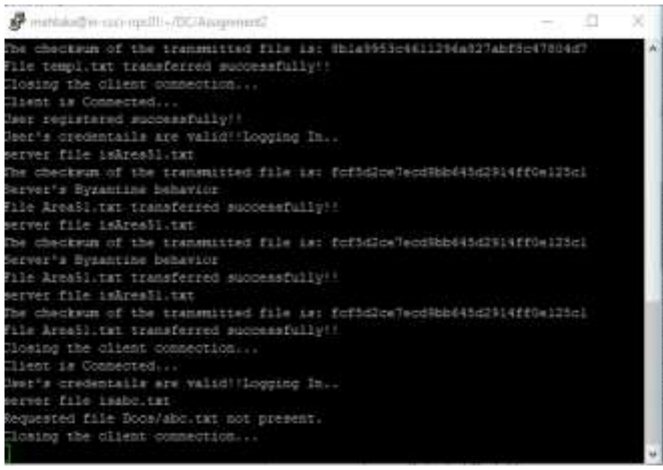
5. Byzantine Behavior


mehtake@in-csci-rrpc01:~/DC/Assignment2
mehtake@in-csci-rrpc02:~/DC/Assignment2

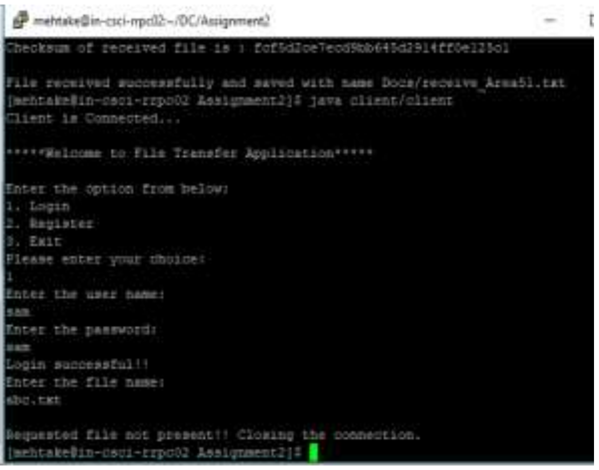
```
File Sandhi.txt transferred successfully!!
Closing the client connection...
Client is Connected...
User's credentails are valid!!Logging In..
server file istempl.txt
The checksum of the transmitted file is: 8b1a9953c4611296a827abf8c47804d7
File templ.txt transferred successfully!!
Closing the client connection...
Client is Connected...
User registered successfully!!
User's credentails are valid!!Logging In..
server file isArea51.txt
The checksum of the transmitted file is: fcf5d2ce7ecd9bb645d2914ff0e125c1
Server's Byzantine behavior
File Area51.txt transferred successfully!!
server file isArea51.txt
The checksum of the transmitted file is: fcf5d2ce7ecd9bb645d2914ff0e125c1
Server's Byzantine behavior
File Area51.txt transferred successfully!!
server file isArea51.txt
The checksum of the transmitted file is: fcf5d2ce7ecd9bb645d2914ff0e125c1
File Area51.txt transferred successfully!!
Closing the client connection...
```

```

Enter the option from below:
1. Login
2. Register
3. Exit
Please enter your choice:
1
Enter the user name:
sam
Enter the password:
sam
Login successful!!
Enter the file name:
Area51.txt
Checksum of received file is : edc4f4a96dbdf0e194b548b71a583428
Error in file transmission attempt: 0

Checksum of received file is : edc4f4a96dbdf0e194b548b71a583428
Error in file transmission attempt: 1

Checksum of received file is : fcf5d2ce7ecd9bb645d2914ff0e125c1

File received successfully and saved with name Docs/receive_Area51.txt
[mehtake@in-csci-rrpc02 Assignment2]$
```

6. File not present

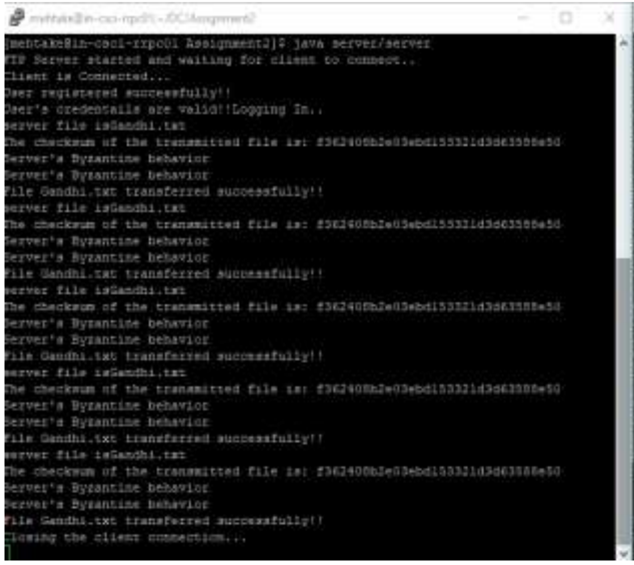


```
The checksum of the transmitted file is: 8b1a9953c4611296a827abf8c47804d7
File templ.txt transferred successfully!!
Closing the client connection...
Client is Connected...
User registered successfully!!
User's credentails are valid!!Logging In..
server file isArea51.txt
The checksum of the transmitted file is: fcf5d2ce7ecd9bb645d2914ff0e125c1
Server's Byzantine behavior
File Area51.txt transferred successfully!!
server file isArea51.txt
The checksum of the transmitted file is: fcf5d2ce7ecd9bb645d2914ff0e125c1
Server's Byzantine behavior
File Area51.txt transferred successfully!!
server file isArea51.txt
The checksum of the transmitted file is: fcf5d2ce7ecd9bb645d2914ff0e125c1
File Area51.txt transferred successfully!!
Closing the client connection...
Client is Connected...
User's credentails are valid!!Logging In..
server file isabc.txt
Requested file Docs/abc.txt not present.
Closing the client connection...
```

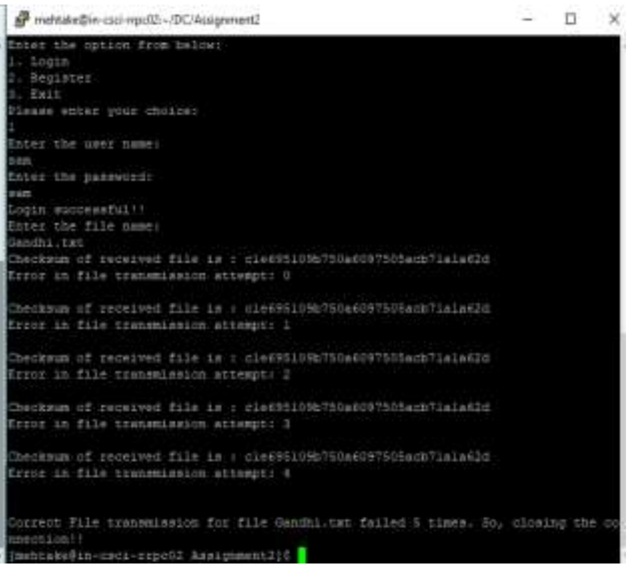


```
Checksum of received file is : fcf5d2ce7ecd9bb645d2914ff0e125c1

File received successfully and saved with name Docs/receive_Area51.txt
[mehtake@in-csci-rrpc02 Assignment2]$ java client/client
Client is Connected...

*****Welcome to File Transfer Application*****

Enter the option from below:
1. Login
2. Register
3. Exit
Please enter your choice:
1
Enter the user name:
sam
Enter the password:
sam
Login successful!!
Enter the file name:
abc.txt

Requested file not present!! Closing the connection.
[mehtake@in-csci-rrpc02 Assignment2]$
```

7. Incorrect File for 5 times

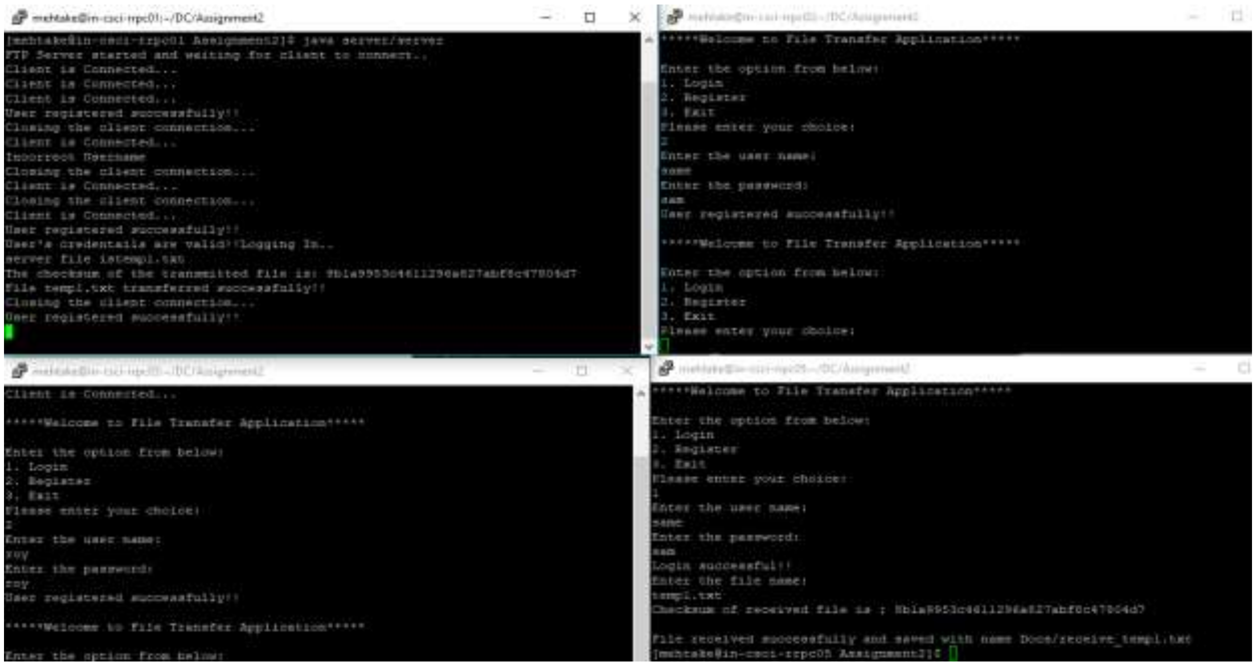


```
[mehtake@in-csci-rrpc01 Assignment2]$ java server/server
FTP Server started and waiting for client to connect..
Client is Connected...
User registered successfully!!
User's credentails are valid!!Logging In..
server file isGandhi.txt
The checksum of the transmitted file is: f362400b2e03ebd153321d3d63588e50
Server's Byzantine behavior
Server's Byzantine behavior
File Gandhi.txt transferred successfully!!
server file isGandhi.txt
The checksum of the transmitted file is: f362400b2e03ebd153321d3d63588e50
Server's Byzantine behavior
Server's Byzantine behavior
File Gandhi.txt transferred successfully!!
server file isGandhi.txt
The checksum of the transmitted file is: f362400b2e03ebd153321d3d63588e50
Server's Byzantine behavior
Server's Byzantine behavior
File Gandhi.txt transferred successfully!!
server file isGandhi.txt
The checksum of the transmitted file is: f362400b2e03ebd153321d3d63588e50
Server's Byzantine behavior
Server's Byzantine behavior
File Gandhi.txt transferred successfully!!
server file isGandhi.txt
The checksum of the transmitted file is: f362400b2e03ebd153321d3d63588e50
Server's Byzantine behavior
Server's Byzantine behavior
File Gandhi.txt transferred successfully!!
Closing the client connection...
```



```
Enter the option from below:
1. Login
2. Register
3. Exit
Please enter your choice:
1
Enter the user name:
sam
Enter the password:
sam
Login successful!!
Enter the file name:
Gandhi.txt
Checksum of received file is : c1e695109b750a6097505acb71a1a62d
Error in file transmission attempt: 0

Checksum of received file is : c1e695109b750a6097505acb71a1a62d
Error in file transmission attempt: 1

Checksum of received file is : c1e695109b750a6097505acb71a1a62d
Error in file transmission attempt: 2

Checksum of received file is : c1e695109b750a6097505acb71a1a62d
Error in file transmission attempt: 3

Checksum of received file is : c1e695109b750a6097505acb71a1a62d
Error in file transmission attempt: 4


Correct File transmission for file Gandhi.txt failed 5 times. So, closing the co
nnection!!
[mehtake@in-csci-rrpc02 Assignment2]$
```

8. Multi-Threading



```
mehtake@in-csci-rrpc01:~/DC/Assignment2
[mehtake@in-csci-rrpc01 Assignment2]$ java server/server
FTP Server started and waiting for client to connect..
Client is Connected...
Client is Connected...
Client is Connected...
User registered successfully!!
Closing the client connection...
Client is Connected...
Incorrect Username
Closing the client connection...
Client is Connected...
Closing the client connection...
Client is Connected...
User registered successfully!!
User's credentails are valid!!Logging In..
server file istempl.txt
The checksum of the transmitted file is: 9b1a9953c4611296a827abf8c47804d7
File templ.txt transferred successfully!!
Closing the client connection...
User registered successfully!!
```

```
mehtake@in-csci-rrpc05:~/DC/Assignment2
*****Welcome to File Transfer Application*****

Enter the option from below:
1. Login
2. Register
3. Exit
Please enter your choice:
2
Enter the user name:
same
Enter the password:
sam
User registered successfully!!

*****Welcome to File Transfer Application*****

Enter the option from below:
1. Login
2. Register
3. Exit
Please enter your choice:
```

```
mehtake@in-csci-rrpc05:~/DC/Assignment2
Client is Connected...

*****Welcome to File Transfer Application*****

Enter the option from below:
1. Login
2. Register
3. Exit
Please enter your choice:
2
Enter the user name:
roy
Enter the password:
roy
User registered successfully!!

*****Welcome to File Transfer Application*****

Enter the option from below:
```

```
mehtake@in-csci-rrpc05:~/DC/Assignment2
*****Welcome to File Transfer Application*****

Enter the option from below:
1. Login
2. Register
3. Exit
Please enter your choice:
1
Enter the user name:
same
Enter the password:
sam
Login successful!!
Enter the file name:
templ.txt
Checksum of received file is : 9b1a9953c4611296a827abf8c47804d7

File received successfully and saved with name Docs/receive_templ.txt
[mehtake@in-csci-rrpc05 Assignment2]$
```

4. Advantage / Disadvantage

Advantage:

1. Provide end to end encryption of the file. So the content of the file is secured.
2. Received file checksum is verified with checksum of transmitted file. It will ensure the correct file receive.
3. Only authenticate user can access the system.

Disadvantage:

1. The authenticated user list is not maintained as persistent storage. So, once the server is stopped all user credentials are vanished.

References

1. https://www.baeldung.com/a-guide-to-java-sockets
2. https://coderanch.com/t/205325/java/send-java-Object-socket
3. https://www.mkyong.com/java/how-to-generate-a-file-checksum-value-in-java/
4. https://stackoverflow.com/questions/10131377/socket-programming-multiple-client-to-one-server