Computer Communications & Networks

Febin Zachariah – 800961027

Aravind Prasannakumari Vijayakumar -800896112

Project-1

Implementation of HTTP Client and Server:

We have implemented the HTTP client and server in java using socket programming. GET and PUT methods are implemented in this project.

GET Method: In this method, client requests data from server. Steps involved GET method are described below:

1) Client establishes a TCP connection with the server.
2) Then, client will send the name of the file it wants to server in the form of request.
3) After getting the request from the client, server check whether the file is present or not.
4) If the requested file is not present, server will send the error code 400
5) If the file is present, it will send the fie with status 200.

PUT Method: In this method, client is sending files to the server and the server saves the file. Following steps are used in the PUT method:

1) Client establishes a TCP connection with the server.
2) If the file is present in the client side, client will send the file along with the request.
3) After receiving the request from the client, server saves the file and sends 200 status message.

Implementation Using Java

We have created two files in java

1**)MyClient.java**-> Performs the functions of a HTTP Client.

2) **MyServer.java**-> Performs the functions of a HTTP Server.

The server is multithreaded to handle multiple client connections at a time.

HTTP Client Implementation:

1) We have created a java file named “MyClient.java”
2) We are taking the parameters as command line arguments in the following format:

```
MyClient Hostname Port Method filename
```

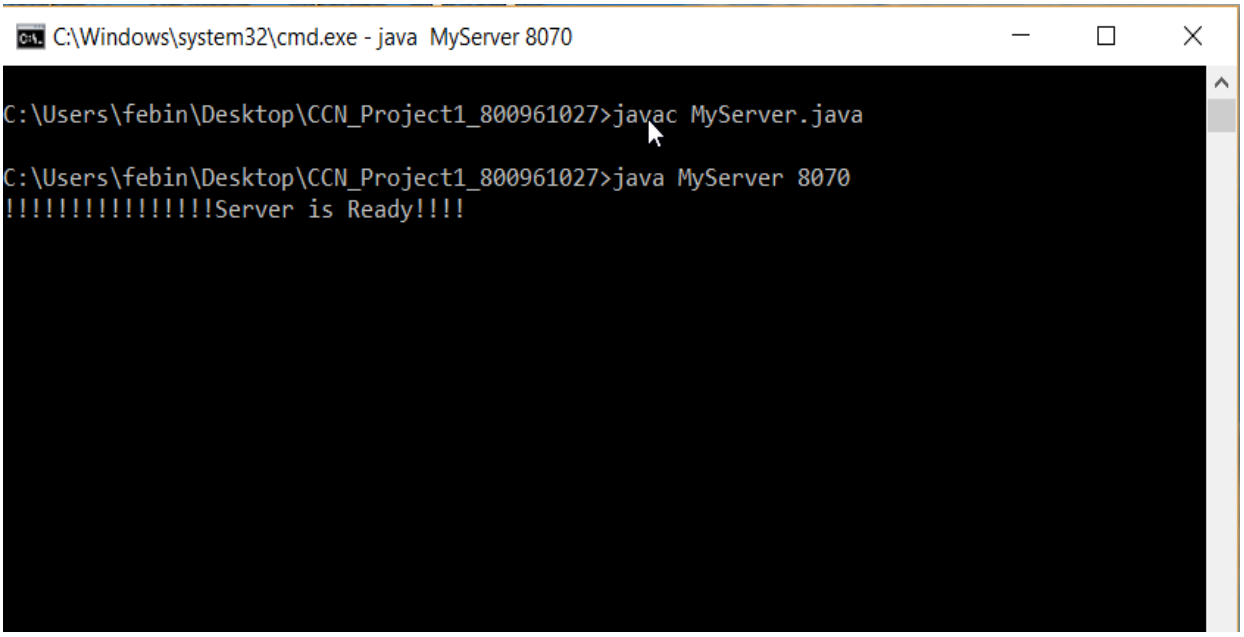
3) Program will check whether the arguments are correct and if the arguments are wrong it will throw error
4) Client will open a socket to connect to the server.
5) If the method is GET, we will use PrintStream to send the request to the server.
6) The response coming for the server is received by using DataInputStream.
7) In the case of PUT method, request is send by using PrintStream and FileInputStream.
8) The response coming from the server is handled by the DataInputStream.

HTTP Server Implementation

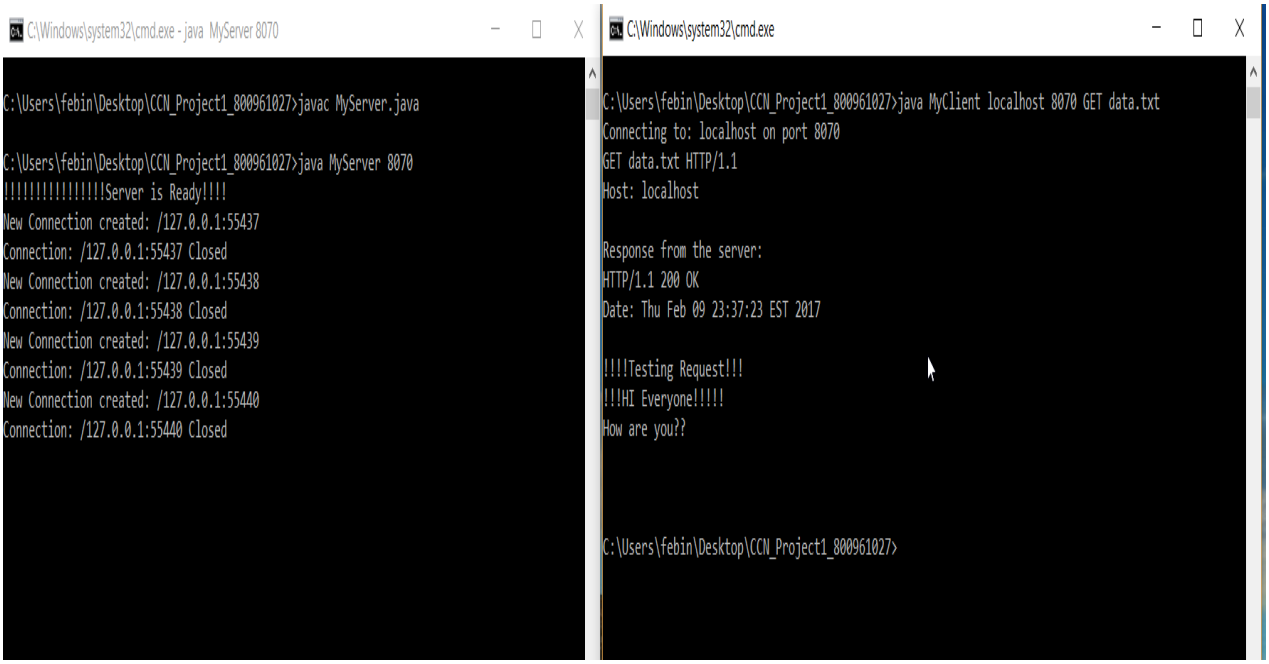
1) Server code is written in the file named “MyServer.java”
2) Start Server by using the following command
   MyServer Port.
3) Create ServerSocket and wait for the clients to connect.
4) For each client, create a separate thread to handle the connection.
5) Request coming from each client is handled by using DataInputStream Class
6) After getting the request, server perform the tasks based on the method type.
7) If it is a GET request, server searches for the requested file and sends 404 status if the file is not found. If the file exists, it sends status 200 and data by using PrintStream class.
8) If it is a PUT request, server writes the contents to a file by using FIleOutputStream class.
9) If the file is successfully created, server sends status 200 by using PrintStream class.
10) addShutDown () method is used to close the server.

Execution Results

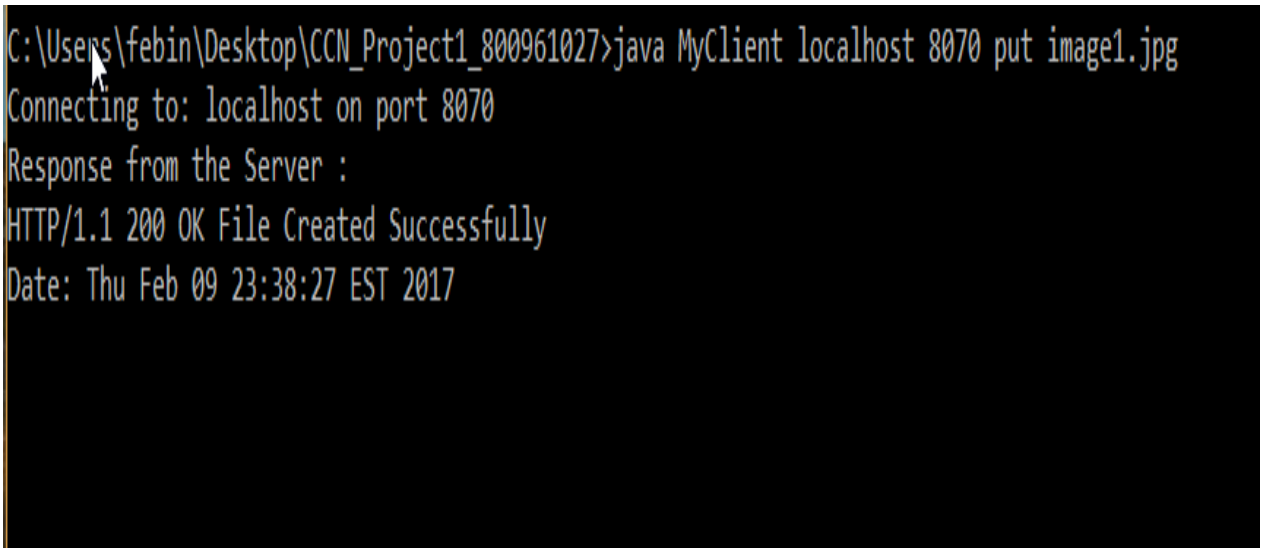
1) Compile both server and client by using the following commands
   javac MyServer.java
   javac MyCleint.java
2) Start the server by using the command java MyServer 8070



3) Create a GET request to the server to read the contents of the text file data.txt
java MyClient localhost 8070 GET data.txt

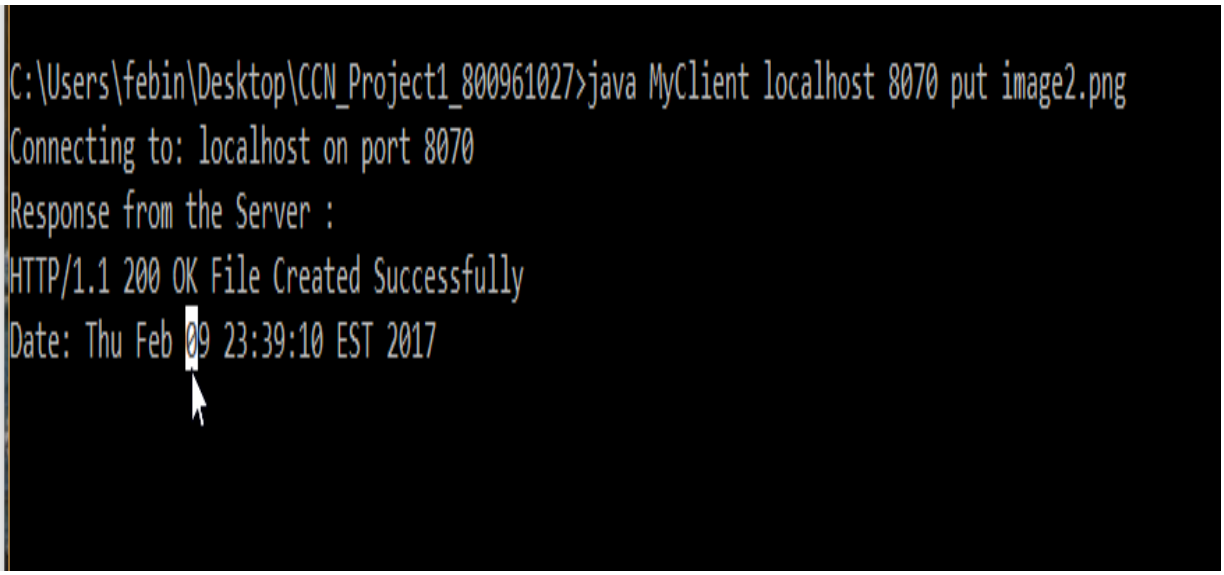


4) Sending a put request to the server to send an image to the server.
Java MyClient localhost 8070 put image1.jpg



Sending another image to the server.

Java MyClient localhost 8070 put image2.png



5) Server is shut down by using (ctrl-c) command from the prompt