

NAME: RAKESH SEHGAL

STUDENT ID: 905618787

**TOOLS: XCODE v.4.5., Java version "1.6.0_35",
Java(TM) SE Runtime Environment**

OPERATING SYSTEM: MAC OSX

**COURSE: NETWORK ARCHITECTURE AND
PROTOCOL (CS/ECE-5565)**

PROJECT 1

DATE OF SUBMISSION: 10/02/2012

BRIEF DISCUSSION OF THE IMPLEMENTATION

The project involves design, implementation, and testing of a simplified version of an HTTP server and an HTTP client program. The design of the HTTP server is such that any client on the network can send an HTTP request for a file on the server. If that file is found by the server, the server returns an HTTP 200 OK message along with the file in the content body and its length. Although, if the file is not found, the server returns an HTTP 404 file not found error. If any client sends a request that is not recognized by the server, it returns an HTTP 400 Bad request as a response to the request. The implementation of both the client and server was done in Java.

SERVER IMPLEMENTATION

The following pre-define libraries were used for the server:

1. `import java.io.*;`
2. `import java.net.*;`
3. `import java.net.ServerSocket;`

When the server is started, the user can specify has to specify the port they want to run the server on. The client uses this port to connect to the server and request files. If the user does not specify the port, the application results in an exception (`IllegalArgumentException`) and displays a message “Port number not entered” to let the user know that they have to specify a port number in order to run the server. If there is some application already running on the same port, it displays “Could not start the server on the given port”.

The application then opens a `ServerSocket` on the given port number and waits infinitely (until terminated) to accept connections (`ServerSocket` method `accept()`) from a client. After the server has accepted the connection, it reads the message sent in the request. If the message contains request in the correct format (`GET /filename HTTP/1.0`), the server reads the file from the local disk, creates a response in the following format:

HTTP/1.0 200 OK

Content-Length: *length*

file-content

and sends it back to the client, replacing *length* with the length of file and *content-body* with the content of the file.

In case the body of the request differs (for my case instead of `GET`, the server finds a `SET`), it sends the client the following response:

HTTP/1.0 400 Bad Request

The server then closes the client connection and awaits connection from another client.

CLIENT IMPLEMENTATION

The following pre-defined libraries were used for the implementation of the client:

1. `import java.io.*;`
2. `import java.net.*;`
3. `import java.net.Socket;`

When a user starts the client, they can enter the server host. If the user does not specify this, the client tries to connect to the “localhost” by default. The server port must be specified at all times. The client then asks the user if they would like to simulate a bad request, if the user presses ‘b’, a request with following content is sent:

SET /filename HTTP/1.0

which results in the server returning a HTTP 400 Bad request. Whereas, if the user presses ‘return’ or any other key, the client asks the user to enter a filename to request to the server (also done for a bad request). In case the user specifies the correct filename, the file is returned from the server and the client then asks the user to enter a filename to store the file on the user’s local disk, it displays

HTTP/1.0 200 OK

Content-length: *length*

HTML OR TEXT FILE CONTENT

and stores the file on user’s local disk. If the user does not specify any filename, the filename in original request is used.

TESTING RESULTS

1) My_Server started with no port number:

Rakeshs-MacBook-Pro:My_Server rakeshseghal1\$ java Server
Please input <port number>

2) My_Server running successfully:

Rakeshs-MacBook-Pro:My_Server rakeshseghal1\$ java Server 5554
Server running successfully on port#5554
Waiting for clients to connect.....

3) My_Client started without port number:

Rakeshs-MacBook-Pro:My_Client rakeshseghal1\$ java Client
Please input <host name > <port number>

(If the user does not specify <host name> "localhost is used by default")

4) My_Client sends Bad Request (SET /filename HTTP/1.0):

Rakeshs-MacBook-Pro:My_Client rakeshseghal1\$ java Client 5554
If you would like to send a bad request enter (b):
b
Enter the name of the file to retrieve from the server:
test.html
Connected to the server on port#5554
Request sent to server: SET /test.html HTTP/1.0

Response from server: HTTP/1.0 400 Bad Request
Done reading response from server. Terminating client.

5) My_Client request for a non existing file on My_Server (HTTP/1.0 404 Not Found):

Rakeshs-MacBook-Pro:My_Client rakeshseghal1\$ java Client 5554
If you would like to send a bad request enter (b):

Enter the name of the file to retrieve from the server:
badfile.html
Connected to the server on port#5554
Request sent to server: GET /badfile.html HTTP/1.0

Response from server: HTTP/1.0 404 Not Found
The requested resource was not found on the server!!!
Done reading response from server. Terminating client.

6) Sample-server with My_Client:

Rakeshs-MacBook-Pro:My_Client rakeshseghal1\$ java Client 5554
If you would like to send a bad request enter (b):
(user pressed returned here, meaning do not send a bad request)
Enter the name of the file to retrieve from the server:
test.html

Connected to the server on port#5554
Request sent to server: GET /test.html HTTP/1.0

Please enter a name for the received file
my_local_file.html
Response from server: HTTP/1.0 200 OK
Content-Length: 18167

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
*****CONTENT HERE*****
</html>
```

Done reading response from server. Terminating client.

7) My_Server with Sample-client (Good and Bad request) :

Rakeshs-MacBook-Pro:My_Server rakeshseghal1\$ java Server 5554
Server running successfully on port#5554
Waiting for clients to connect.....

Accepted connection from client#1
Client Request: SET /test.html HTTP/1.0
Bad request from user. Sending response: HTTP/1.0 400 Bad Request
Accepted connection from client#2
Client Request: GET /test.html HTTP/1.0
User request is: GET /test.html HTTP/1.0
Trying to read file
File read successfully. Sending response to client: HTTP/1.0 200 OK

8) Successfully running My_Server with My_Client:

Bad Request:

Rakeshs-MacBook-Pro:My_Client rakeshseghal1\$ java Client 5554
If you would like to send a bad request enter (b):
b
Enter the name of the file to retrieve from the server:
test.html
Connected to the server on port#5554
Request sent to server: SET /test.html HTTP/1.0

Response from server: HTTP/1.0 400 Bad Request
Done reading response from server. Terminating client.

Good Request:

Server:

Rakeshs-MacBook-Pro:My_Server rakeshseghal1\$ java Server 5554
Server running successfully on port#5554
Waiting for clients to connect.....

Accepted connection from client#1

Client Request: GET /test.html HTTP/1.0
User request is: GET /test.html HTTP/1.0
Trying to read file
File read successfully. Sending response to client: HTTP/1.0 200 OK

Client:

Rakeshs-MacBook-Pro:My_Client rakeshseghal1\$ java Client 5554
If you would like to send a bad request enter (b):

Enter the name of the file to retrieve from the server:

test.html

Connected to the server on port#5554

Request sent to server: GET /test.html HTTP/1.0

Please enter a name for the received file

my_local_file.html

Response from server: HTTP/1.0 200 OK

Content-Length: 18167

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
*****CONTENT HERE*****
</html>
```

Done reading response from server. Terminating client.

9) Successfully running My_Server with My_Client on local network:

Rakeshs-MacBook-Pro:My_Client rakeshseghal1\$ java Client 192.168.1.104
5554

If you would like to send a bad request enter (b):

Enter the name of the file to retrieve from the server:

test.html

Connected to the server on port#5554

Request sent to server: GET /test.html HTTP/1.0

Please enter a name for the received file

my_local_file.html

Response from server: HTTP/1.0 200 OK

Content-Length: 18167

```
*****CONTENT HERE*****
```

Done reading response from server. Terminating client.

Summary of test results:

All the above test procedure were successfully run and expected output were obtained. All the test results obtained have been pasted above without any manipulations. All of the features described and discussed in the project design have been implemented and were observed to be working properly on the local machine and on local network.