

CSE 322

Offline Assignment 1

Problem # 1

In this problem, you will be designing an SMTP client for a standard mail server using Socket Programming. Your task is to develop a simple mail client that sends email to any recipient. Your client will need to connect to a mail server, dialogue with the mail server using the SMTP protocol, and send an email message to the mail server. The details of SMTP has been described in the class. You can get help from the given documents on SMTP or from Wikipedia.

In order to limit spam, some mail servers do not accept TCP connection from arbitrary sources. For this experiment, you may want to consider “webmail.buet.ac.bd” as the mail server. You need to be in the BUET network to access this mail server.

You are given a skeleton code for this assignment. You need to complete the skeleton code. In some cases, the receiving mail server might classify your e-mail as junk. Make sure you check the junk/spam folder when you look for the e-mail sent from your client.

- You must be able to send mail (with subject) to multiple users.
- You need to implement the protocol as specified in the slide; therefore, **you cannot use any JAVA high level library.**
- You must maintain the states of the SMTP clients according to the protocol
- You must output all the replies from the server
- If server replies with an error, you must output the appropriate error message
- You must maintain timeout for each command. Timeout should be 20s.

BONUS

Try to send attachment with your email using SMTP.

Problem # 2

In this problem, you need to develop a web server that can handle multiple incoming **HTTP GET** and **POST** requests and send HTTP response message back to the client, with appropriate Content-type in the response (according to the standard http protocol). The requirements of the web server are-

- Listen on a specified port, other than the standard port 80, and accept HTTP requests.
- Handle each HTTP request in a separate thread.
- Handle HTTP version 1.0 GET and POST requests. You will be given an http form as a template to demonstrate the POST request.
- Extract filename from HTTP request and return the file or an appropriate error message (e.g. if we type “localhost:8080/index.html” in the address bar of a browser, then we will get the index.html page if found. **If not** the browser will show a **404 Not Found Message**).
- If the user **inputs “localhost:8080” only** in the browser, the server must search for an **“index.html”** file. If found then it will display the file. Otherwise a 404 Error message must be displayed in the browser as well as in the output.
- Return a HTTP response message and close the client socket connection.
- Return appropriate **status code**, e.g. **200 (OK)** or **404 (NOT FOUND)**, in the response.
- Determine and send the **correct MIME type** specifier in the response message.
- You need to generate an **appropriate log file** for the corresponding http request.

You need to implement the basics of http protocol accordingly; **therefore, you cannot use any JAVA high level library. You cannot use Javascript. All the operations should be strictly limited to Socket Programming and html manipulation.**

Reference: <https://tools.ietf.org/html/rfc7231>

Submission Guideline:

- Form a two person group
- Two separate links will be given for two offlines in moodle. Create a folder having only the source files (*.java). The folder name should be your student id+Problem<id>. Zip the folder and submit it in moodle.
- You must follow the specific instruction of submission given in the class.
- Deadline: November 11 (01:00 AM)