Assignment-4 Guide
Design of a Mail Server and Client

Index

Assignment-4 Guide
Design of a Mail Server and Client

Introduction:
Follow the below steps to design Mail Server and Client.

Files and Folders:

Sl.No.	Name	Description
1	Client.c	Write client source code
2	SMTPServer.c	Write SMTP server source code
3	POP3Server	Write POP3 server source code
4	User.txt	Each line should contain username and password separated by space.
5	MyMailBox.txt	*Should be present in users directories. List of emails to be stored in a format

Note: Create a folder for each username manually. The mailboxes for each user will be stored in the respective sub directories.

EMAIL Format: Fomat to store the email in MyMailBox.txt, **note the email should end with '.'**

From: <username>@<domain_name>
To: <username>@<domain_name>
Subject: <subject string, max 100 characters>
Received: <time at which received, in date : hour : minute>
<Message body – one or more lines>
.

Client.c

Step 1:
The program will first ask for the username and password, and store it locally. It will then ask for options from the user and wait for user input from the keyboard. The program should support the following 3 options:
1. Manage Mail : Shows the stored mails of the logged in user only
2. Send Mail : Allows the user to send a mail
3. Quit : Quits the program.

Step 2: When user selects option 1 – Manage Mail (POP3-server)

a) Client program should do validation of username and password and do authentication using POP3Server by passing “USER” and “PASS” .

b) If authentication is not successful, display an error message and prompt user to re-enter the credentials.

c) If authentication is successful, then establish the connection between client and POP3server and client should recieve and display below message from server.

    +OK POP3 Server ready.

d) Program receives list of emails from the pop3 server for authenticated user and display it on the console below format.
Sl. No. <Sender’s email id> <When received, in date : hour : minute> <Subject>

And provide below option
POP3 commands:
STAT, LIST, RETR, DELE, QUIT
STAT – Count the number of emails.
LIST – should list all the email for the user
RETR – Retrieve email based on the serial number
DELE – Should delete the email based on the serial number
QUIT – Close the connection and terminate the program

Here, user should provide multiple options in the terminal to enter and accordingly manage mails. Now each of these options will correspond to the commands mentioned above. For example, if the user wants to retrieve a particular mail, then he will enter some character or option number from terminal, and after receiving the option number from the terminal, the client sends the appropriate command (which is RETR, in this case) to the POP3 server, and gets the response back.

Step 3: When user selects option 2 – Send a mail (SMTP-server)

(Authentication is Optional)
Client program should do authentication of username and password using User.txt file.
If authentication is not successful, display an error message and prompt user to re-enter the credentials.
If authentication is successful, then
Get following data from user
From
To
Subject
Email body (end with "." on a new line)

Once the data/mail is received from user, client side validations will happen and those validations are given below

1. The From, To, and Subject field must be there, in that order and in proper format. The message body can be empty (just the fullstop at the last line).
2. The format for the From and To fields must be X@Y. If the format is not correct, then "Incorrect format" is printed, and the three options are given again. The entire mail has to be entered again, there is no editing facility.

If the format is correct establish the connection between client and SMTP server and Program should handle following commands:

HELO, MAIL, RCPT, DATA, QUIT

Example:

```
C: HELO iit.edu // sending host identifies self
S: 250 OK Hello iit.edu // receiver acknowledges
C: MAIL FROM: <ag@iit.edu> // identify sending user
S: 250 <ag@iit.edu>... Sender ok // receiver acknowledges
C: RCPT TO: gb@iit.edu // identify target user
S: 250 root... Recipient ok // receiver acknowledges
C: DATA // Client program should send the message.
S: 250 OK Message accepted for delivery
C: QUIT // sender signs off
```

Step 3: Quit the program.

SMTPServer.c

SMTP Server is used to communicate with client to receive messages from the client and store the emails for specified user mail box(MyMailbox.txt – present in each user directory).

Once the connection is established the connection with the client, server should send below message to client.
S: 220 <iitkgp.edu> Service ready

SMTP Responses for the commands.

Commands from client	**Description**
HELO iit.edu	Server should send 250 OK Hello iit.edu
MAIL FROM: <ag@iit.edu>	Server should send 250 <ag@iitkgp.edu>... Sender ok
RCPT TO: gb@iit.edu	Server should send 250 root... Recipient ok
DATA	Server stores the email received from client in MyMailbox.txt for respective user, with below response 250 OK Message accepted for delivery
QUIT	Send below message and close the connection 221 iitkgp.edu closing connection

POP3Server.c

POP3 server is used to handle the emails operations such as list the emails, display an email, delete an email, count the number of messages.

Once the connection is established the connection with the client, server should send below message to client.
+OK POP3 server ready

Also the POP3Server should read the emails from MyMailBox.txt file for specified user(username should get from client) and send it to client.

POP3Server Responses for the commands. USER, PASS, STAT, LIST, RETR, DELE

Commands from client	**Description**
USER	Validate the username using user.txt file.
PASS	Authenticate the user using the password.
STAT	Server should count the number email present in MyMailBox.txt file and should return the count to client.
LIST	Server should read all the emails from MyMailBox.txt file and should return the list of email to client.
RETR	Server should read specified email number from MyMailBox.txt file and send it to the client.

DELE	Server should delete(or mark as deleted) the specified email from MyMailBox.txt.
	`Possible Responses:`
	`+OK message deleted`
	`-ERR no such message`

QUIT	When the client issues the QUIT command, the server checks if there are any messages marked to be deleted. If yes, the messages are deleted. If all marked messages are deleted, +OK is sent, otherwise –ERR is sent. The server then sends a “goodbye” message, releases locks and any other resources, and closes the connection