National Institute of Technology Calicut

Department of Computer Science and Engineering

Winter Semester 2021 – 2022

CS3093D: Networks Laboratory

Experiment – 07

This assignment aims to learn about the SMTP protocol and POP3 protocol and build a simple mail server and client to send and receive mails. SMTP is used to send mails from a mail client to the mail server, while POP3 is used to access mail from a mail server by a mail client. More details of SMTP and POP3 are available in RFC 821 and RFC 1939, respectively.

The SMTP server and the POP3 server will run on the mail server. Your home machine will run the POP3 client to access the mailbox and an SMTP client to send mail through the SMTP server.

Server Setup:

Before initiating server programs, you need to set up a few things. In the directory from which you will run the program, create a file called ***userlogincred.txt***. Each line of the file contains a one-word user login name and a one-word password, separated by one space.

userlogincred.txt file:

Stallings network$ecurity

Fourouzan d@t@comm

Andrew @dministr@tor

Then in the same directory, create a subdirectory for each of the users, which will contain a corresponding mailbox file to store the received mails. Subdirectory should be named after the users.

Now write a C program named ***smtpmail.c*** to act as an SMTP mail server to receive mails from another mail client/server using SMTP and store them. The program will take a command-line integer argument ***my_port*** to indicate the port on which the mail server will run. It should handle receive of incoming mail. It simply waits on a socket bound at port ***my_port***. When a sender makes a connection, the process follows the SMTP protocol to receive the message.

The received message is appended to the end of a file named mymailbox that is stored in the user's subdirectory. The username for the mail will be in the mail address so that the program will redirect to the corresponding subdirectory. The mail is appended in the following format.

From: <username>@<domain_name>

To: <username>@<domain_name>

Subject: <string, max 100 characters>

Received: <mail_received time; date: hour: minute>

< Message body = (Number of lines entered in the message body > 0) >

Implement ***popserver.c*** program to manage the mailbox for POP3 server. The program will take a command-line integer argument ***pop3_port*** that will indicate the port on which the POP3 server will run.

Host machine setup:

You have to create a program called ***mailclient.c*** that implements both send and receive of mail. The program would first ask for the username and password. It will then ask for options from the user and wait for user input from the keyboard.

The program should support the following three 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.

Option 1: Manage Mail

This function should list all the mails in the user's mailbox on the screen in the following format:

Sl. No. <Sender's email id> <received time; date: hour: minute> <Subject>

Users should be able to view the mails based on the serial number. The program will need to communicate with the POP3 server running on the MailServer machine to read and delete mails. This will be done by opening a TCP connection to the POP3 server and using the POP3 protocol. Similar to SMTP, the client first sends the username and password. The POP3 server checks these.

The program uses getchar() function to read the input character from the std input/output terminal.

1) If the entered character is 'q', then POP3 server sends a "goodbye" message by closing the connection and returning to the options menu.
2) If the entered character is 'd', the mail is deleted.
3) Otherwise, it returns to show the list of emails again when the user hits any other character after reading the mail.

Option 2: Send mail:

The user should enter the mail to be sent in exactly the following format:

From: <username>@<domain name>

To: <username>@<domain name>

Subject: <string, 50>

<Message body = (Number of lines entered in the message body > 0); terminated by a final line with only a full stop character>

You can assume that no single line in the message body has more than 80 characters, and the maximum no. of lines in the message is 50. Also, you can assume that the To line will not contain more than one email address.

For example:

From: stallings@localhost

To: andrew@localhost

Subject: Network Lab Experiment 7

You have learned SMTP very well. Keep It Up!!!

.

Note the single full stop, immediately followed by an <Enter> at the last line.

On getting the complete message, the process first checks the following format of the message:

1. The From, To, and Subject field must be there, in that order, and in the proper format. The message body can be empty (just the full stop at the last line).
2. The format for the From and To fields must be X@Y.

If the format is incorrect, 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. The process sends the mail to the SMTP mail server if the format is correct. Then the client process will print the message "Mail sent successfully" on the screen.

Option 3: Quit

When the client issues the QUIT command, the server cleans up any resources used by the connection, sends the message "goodbye" and closes the connection.