Assignment 6: Implement a Simplified Email Service

Deadline: March 21, 2025 EOD

SUBMISSION INSTRUCTIONS

Zip all these files as instructed as the end of this assignment into a single zip file <NAME>_<ROLL NO>.zip and submit on MS Teams.

NOTES:

1. You should mention your name (as per ERP), roll number and the Assignment number within the comment line at the beginning of each program. A sample header will look as follows.
   ==========
   Assignment 6 Submission
   Name: <Your_Name>
   Roll number: <Your_Roll_Number>
   ==========
2. The code should have proper documentation and indentation. Unreadable codes will not be evaluated.
3. The code should get executed to the lab machines. If we get a compilation error or runtime error during executing your code on the lab machines, appropriate marks will be deducted.
4. Any form of plagiarism will incur severe penalties. You should not copy the code from any sources. You may consult online sources or your friend, but at the end, you should type the program yourself. We may ask you to explain the code, and if you fail to do so, you'll be awarded zero marks.

In this assignment, you will implement a custom email transfer protocol called My_SMTP using POSIX sockets in C. The goal is to develop both a server and a client that can communicate over a local area network (LAN) to send, receive, and retrieve emails. My_SMTP is inspired by SMTP but includes features for listing and retrieving stored emails.

Since email transfer requires a reliable transport layer protocol, you need to implement over TCP sockets. You have been introduced to several function calls to work with TCP sockets. You can use any function call you have explored in earlier assignments.

For a simplified and efficient email exchange process, My_SMTP introduces the following commands and response codes:

Command	Description
HELO <client_id>	Initiates communication with the server.
MAIL FROM: <email>	Specifies the sender's email address.
RCPT TO: <email>	Specifies the recipient's email address.

DATA	Signals the start of the email body. The message ends with a single dot (.) on a new line.
LIST <email>	Lists all stored emails for the given recipient.
GET_MAIL <email> <id>	Retrieves a specific email by its ID.
QUIT	Ends the session.

Response Code	**Meaning**
200 OK	Command executed successfully.
400 ERR	Invalid command syntax.
401 NOT FOUND	Requested email does not exist.
403 FORBIDDEN	Action not permitted.
500 SERVER ERROR	Internal server error.

Functionality to Implement:

Server:

- Listens on a specified port (e.g., 2525) for incoming client connections.
- Handles multiple clients concurrently.
- Parses and validates My_SMTP commands from clients.
- Stores emails in a local directory (mailbox/<recipient>.txt).
- Supports email retrieval through LIST and GET_MAIL.
- Sends appropriate response codes to clients.

Client:

- Connects to the My_SMTP server.
- Allows users to send emails using My_SMTP commands.
- Supports listing and retrieving received emails.
- Displays server responses and error messages.

Example Interaction:

Client Side

> ./mysmtp_client 192.168.1.100 2525
Connected to My_SMTP server.
> HELO example.com

```
200 OK
> MAIL FROM: alice@example.com
200 OK
> RCPT TO: bob@example.com
200 OK
> DATA
Enter your message (end with a single dot '.'):
Hello Bob,
This is a test email.
.
200 Message stored successfully
> LIST bob@example.com
200 OK
1: Email from alice@example.com (12-03-2025)
2: Email from charlie@example.com (13-03-2025)
> GET_MAIL bob@example.com 1
200 OK
From: alice@example.com
Date: 12-03-2025
Hello Bob,
This is a test email.
> QUIT
200 Goodbye
```

Server Side

```
> ./mysmtp_server 2525
Listening on port 2525...
Client connected: 192.168.1.101
HELO received from example.com
MAIL FROM: alice@example.com
RCPT TO: bob@example.com
DATA received, message stored.
LIST bob@example.com
Emails retrieved; list sent.
GET_MAIL bob@example.com 1
Email with id 1 sent.
Client disconnected.
```

What to submit:

You should submit the following files in a single tar.gz file:

- mysmtp_client.c and mysmtp_server.c
- a makefile to create the two executable files to run from mysmtp_client.c and mysmtp_server.c respectively

- The server should store all emails properly in the server machine.
- Client and server can run on different machines within a LAN and not necessarily in the same machine.