جامعــة خليفــة
Khalifa University

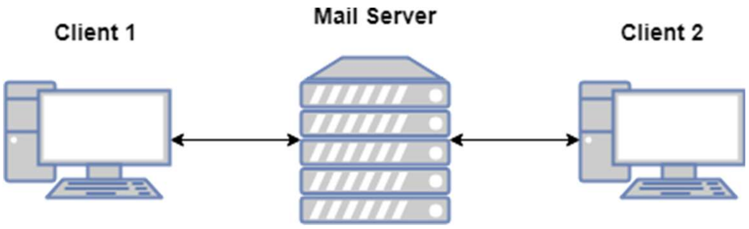
Khalifa University of Science and Technology
Department of Electrical and Computer Engineering
ECCE 356: Computer Networks
Spring 2022

Course Project: Mail Transfer

Deadline: Monday, May. 9th , 2022
@12:00 noon

Specifications and Requirements:

The goal of the project is to extend on the knowledge you gained in your assignments with regards to the basics of network programming within the **TCP/IP** Internet family environment, and its socket application programming interface (API) and also a little bit of **multi-threading**. You will use the TCP protocol as a "delivery service", to carry your packets. The particular problem that you have to resolve consists of **transferring Mail** between **multiple clients and one server** logged in at **different** computers.



Scenario and Functionalities:

The project scenario is divided to 2 parts: basic scenario and advanced functionalities. The basic scenario extends on the scenario developed in assignment 2 and introduces additional components mentioned below. You are required to **implement** the **complete basic scenario** and **at least one** of the **advanced features**.

Basic scenario:

- **Server:** creates a directory for each registered client with "inbox" and "sent" directories in each.
  - **Directory title:** clientemail
  - Assume server already has a list of 5 existing emails.
- **Client 1:** constructs email to send to client 2 with an attachment.
  - **Attachment specifications:** any type (pdf, word, png) and size more than 2 Mb. (i.e. email should be sent on **multiple packets** not only one)
- **Client 1:** connects to mail server
  - Mail server name entered by client.
- **Client 1:** sends email to the server.
- **Server:** saves the email to client 2's "inbox" directory.
  - **File name format:** Subject_Time

- **Server:** saves email to the client 1's "sent" directory.
    - **File name format:** Subject_Time
- **Client 2:** connects to the mail server.
    - Mail server name entered by client.
- **Client 2:** request updating local "inbox" directory by the server.
- **Server:** sends client 2 a copy of his "inbox" directory.
- **Client 2:** saves a copy of emails into the local inbox directory.
- **Client 1 and 2:** list emails in their inbox/ sent directories.
    - **Format:** From To Subject Timestamp.

Advanced features:

- **Multiple receivers:** allow a sender to send an email to multiple receivers (multiple "to" clients)
- **Forward emails:** a client can resend an existing email (from "inbox"/ "sent") where the new email will be automatically constructed from the existing email.
- **Mark emails as read:** once an email is read, a symbol should be added once listing the emails to indicate it was read

Assumptions and Specifications:

- The server that clients connect to can change its name.
- If an attachment is sent along with an email it should be saved in the same directory as the email with the same name. However, the extension (.png, .pdf) will be different.
- *TO* and *FROM* fields are email format and **NOT** host name
- Sender Client can send to different email IDs. (Do not restrict a single *To* and *From* Emails)
- There should be input check aside from the server validation to ensure that the input fields are valid.
- All the registered Clients' email addresses are stored in the mapping file on the server. i.e. Server has to check the mapping file to acknowledge that the client is registered.
- Any client can send and receive emails.
- All running codes should display its status (ex. Client print to screen, "email has been sent to the server", whenever it sends the email).
- Server can receive emails to unconnected client and will forward them according to the basic scenario once they connect.
- Your program should include reply codes (error and confirmation) used in assignment 2.

Supplementary Codes:

To develop the applications required in this assignment, please use the previously given programs in assignment 1 and the codes you developed in assignment 2.

Groups:

For this assignment, you are allowed to strictly work in groups of 2 students unless you have requested otherwise and received an email confirmation regarding your group formation. Every member must explain his/her contribution to the assignment. Every group will be given 10 mins to demo and 10 mins to answer questions related to the assignment.

Deliverables:

You will be required to submit the following by the deadline. Failure to submit by the deadline will result in deduction of marks.

- Word/PDF document which contains your C/C++ program for turnitin check. The document should include all the file you created and modified (cpp and header files).
- A ZIP folder which includes the used projects (server, sender and receiver). You will be required to download the folder from moodle in your demo slot and run it directly. Hence, make sure you submit a correctly running version.

Demo:

On demo day, each team should run the server and client codes on 2 or 3 PCs. One of the Clients running can be on the same machine as the server. In addition, you will be asked questions about the functioning of the program; any student of the group must be able to answer any question. Part of the marks will be assigned for demonstrating compliance with requirements at this demonstration. Marks assigned to each member of the group may be different, depending on ability to answer the questions.