King Fahd University of Petroleum and Minerals

Computer Engineering Department

COE523 Assignment 1

Chatting application

Prepared by

Malek Aljemaili - 201431340

Due Date

Mar 02, 2024

# Introduction

In the process of completing the Distributed Computing course (COE523), we went deep in the concept of threading and processing. And to ensure the student understanding the first assignment has been released. The assignment requires two processes, a server and client. There should be multiple clients (multiple processes running client code) and one server. The clients should communicate with each other through the server. In this report I will explain how to run and use the project and the implementation of some key features.

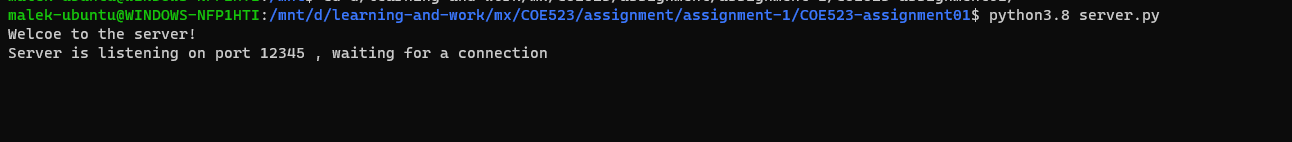
# Components

The assignment consists of four files. this report, client.py file, server.py and server\_client\_handler.py. The last two python files have the implementation of the server process while client.py contains the client process implementation.

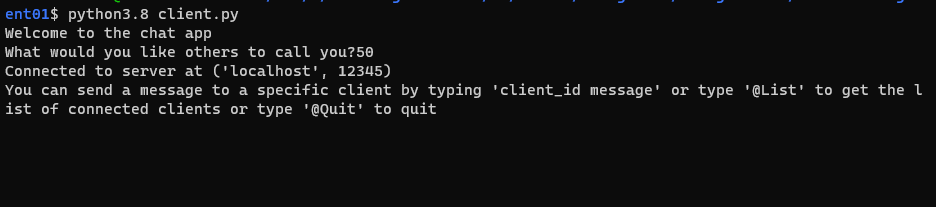
# Setup and Run

To run the assignment, make sure you have python3.8 and terminal window (preferable Linux terminal). After downloading the three python files, follow the next steps to run the assignment:

1. In a new terminal tab, run (python3.8 server.py). By default, the server will run on address “localhost” and port 12345. Once the server finishes initializing itself you will see output like below.



1. Now in a new terminal tab run the client program (python3.8 client.py). You will be asked to enter the client id (between 1 to 8 bytes). After choosing the client ID the program will try to connect to the server (“localhost” and port 12345). Make sure that the server is running before that. Now you will see output like below.



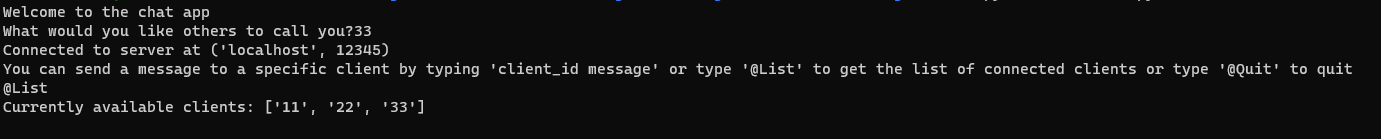
1. Repeat step 2 to create at least 2 clients.

After completing these steps, you are now ready to use the program. For the purpose of continue explaining the program I have set the following clients : 11,22 and 33

## Features

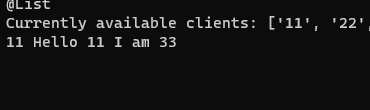
In this program you have multiple message types that you can send:

1. Write “@List” to ask the server about the currently connected clients.



1. Now you can send a message from client 33 to client 11 by typing “11 “then your message.

From client 33 terminal tab:

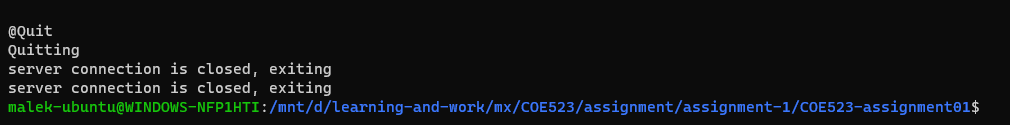


From client 11 terminal tab:

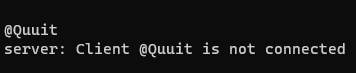
A black background with white text

Description automatically generated

1. After finish communicating you can close the client program by typing “@Quit”



1. If you type a message that doesn’t follow any of the previous message formats you will see error message from the server like:



## Hidden messages

The program has another few messages that are sent automatically by code. The first message is “Connect” followed by the client ID. This message is meant to tell the server that I am a new client, and this is my ID. Without this message the server will not consider the connection to be a client even if the TCP connection is available. The other message is “Alive” that is used by the server to update the time counter of this client. Each client should send this message to the server before the end of keep alive period that has been sent by the server to the client as a reply to “Connect” message. Any client that fails to send this message within the keep alive period will be considered offline. When that happens, the server will drop this client and send the new list of connected clients to all connected clients.

# Conclusion

The assignment consists of two programs, the server and the client. It can be used for simple chatting between different people across the same LAN. In the future we may have the time to support clients across the world by running the server in a public hosting service with public IP. The work on this assignment has been done by Malek Aljemaili.