Practical 7

Aim : To do Socket Programming

Scenario : An organization named Albert Enterprise has established two departments for better performance of the company, as each department will be having some specific set of tasks to perform. So, this will reduce the time and increase the efficiency of the work. As both the departments are dependent on each other, they need to communicate more frequently. To solve the problem, the IT department has suggested the option to create a chat application which will work only in the office premises. So, help the IT professionals to create the chat application.

Make sure that the application has the below mentioned features :

1) Department 1 will be set as the server while department 2 will be set as a client device.

2) If any of the device irrespective of client or server has sent the message that the "work is done", then connection should be closed on both the ends.

3) Client device can send multiple messages at the same time, while the server device can send a single message at a time.

4) There is no restriction on the protocol selection, you can use UDP or TCP. Justify the reason for selection of the specific protocol.

Procedure :

1. Write a code for Department-1 acting as a server.

Code :

```python
import socket
import YSL_io

def server_program():
    host = socket.gethostname()
    port = 4200 # initiate port no above 1024
    server_socket = socket.socket() # get instance
    server_socket.bind((host, port)) # bind host address and port together
    server_socket.listen(2)
    conn, address = server_socket.accept() # accept new connection
    YSL_io.printCYN(f'\nConnection from client :  {str(address)}\n')
    str1 = ''
    str2 = ''
    while str2.lower().strip() != 'work is done':
        str1 = conn.recv(1024).decode()
        if not str1:
            # if data is not received break
            break
        YSL_io.printBLU('\tClient : ' + str(str1))
        str2 = YSL_io.inputGRN('\t-> ')
        conn.send(str2.encode())
    conn.close()
```

```python
server_program()
```

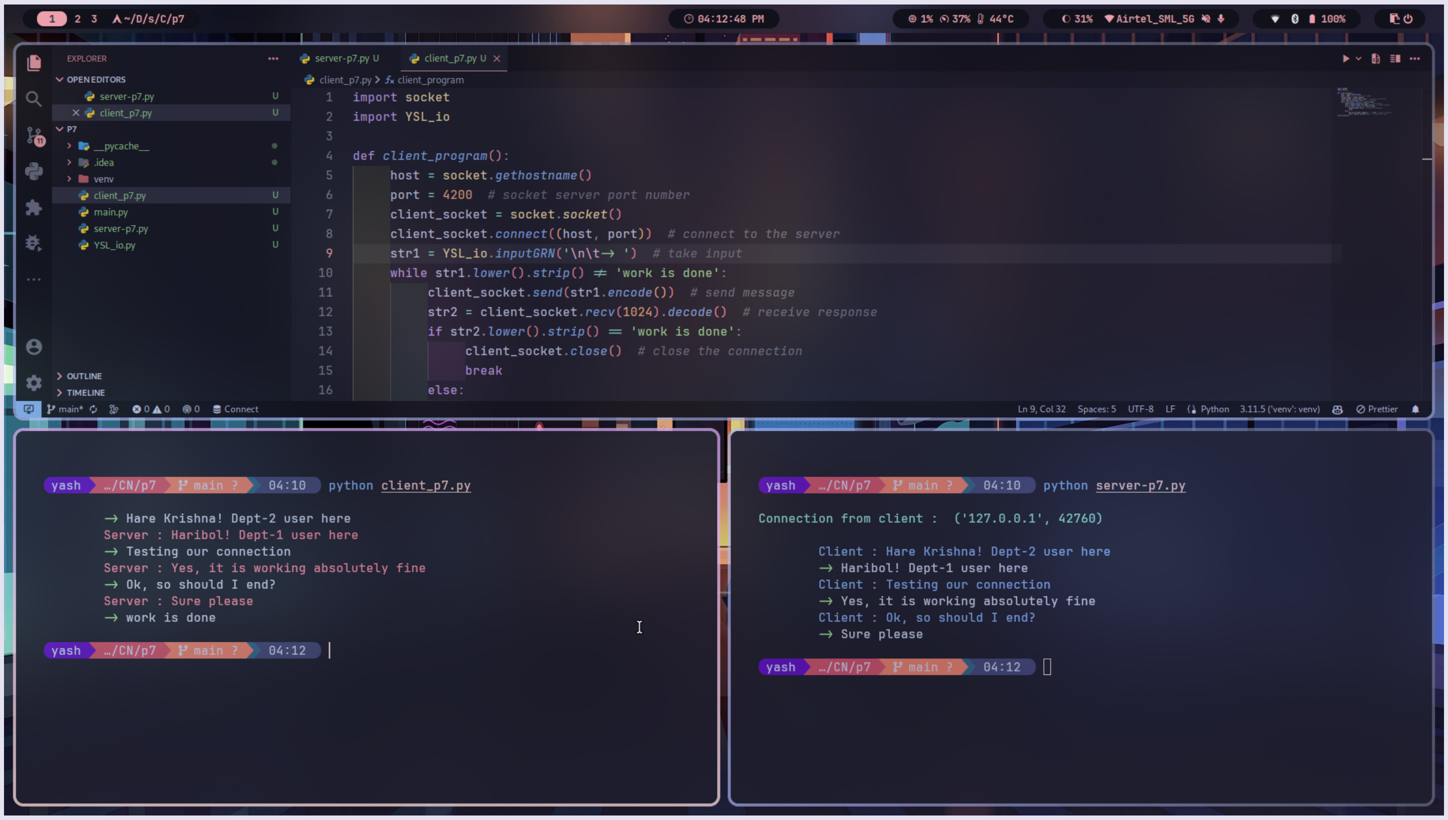
2. Similarly, for department-2 acting as a client, write its respective code.

Code :

```python
import socket
import YSL_io


def client_program():
    host = socket.gethostname()
    port = 4200  # socket server port number
    client_socket = socket.socket()
    client_socket.connect((host, port))  # connect to the server
    str1 = YSL_io.inputGRN('\n\t-> ')  # take input
    while str1.lower().strip() != 'work is done':
        client_socket.send(str1.encode())  # send message
        str2 = client_socket.recv(1024).decode()  # receive response
        if str2.lower().strip() == 'work is done':
            client_socket.close()  # close the connection
            break
        else:
            YSL_io.printRED('\tServer : ' + str2)  # show in terminal
            str1 = YSL_io.inputGRN('\t-> ')  # again take input
client_program()
```

3. Now, execute server code first in one environment and client one in another environment. And test their communication in network 127.0.0.0.



```python
import socket
import YSL_io

def client_program():
    host = socket.gethostname()
    port = 4200  # socket server port number
    client_socket = socket.socket()
    client_socket.connect((host, port))  # connect to the server
    str1 = YSL_io.inputGRN('\n\t→ ')  # take input
    while str1.lower().strip() != 'work is done':
        client_socket.send(str1.encode())  # send message
        str2 = client_socket.recv(1024).decode()  # receive response
        if str2.lower().strip() == 'work is done':
            client_socket.close()  # close the connection
            break
        else:
```

```
yash …/CN/p7  main ? 04:10 python client_p7.py

        → Hare Krishna! Dept-2 user here
        Server : Haribol! Dept-1 user here
        → Testing our connection
        Server : Yes, it is working absolutely fine
        → Ok, so should I end?
        Server : Sure please
        → work is done

yash …/CN/p7  main ? 04:12
```

```
yash …/CN/p7  main ? 04:10 python server-p7.py

Connection from client :  ('127.0.0.1', 42760)

        Client : Hare Krishna! Dept-2 user here
        → Haribol! Dept-1 user here
        Client : Testing our connection
        → Yes, it is working absolutely fine
        Client : Ok, so should I end?
        → Sure please

yash …/CN/p7  main ? 04:12
```

<u>Conclusion</u> : From this experiment, the communication among the server and the client is observed and implemented using socket programming (in python here), in which two hosts can communicate between each other using the network 127.0.0.0 locally here.