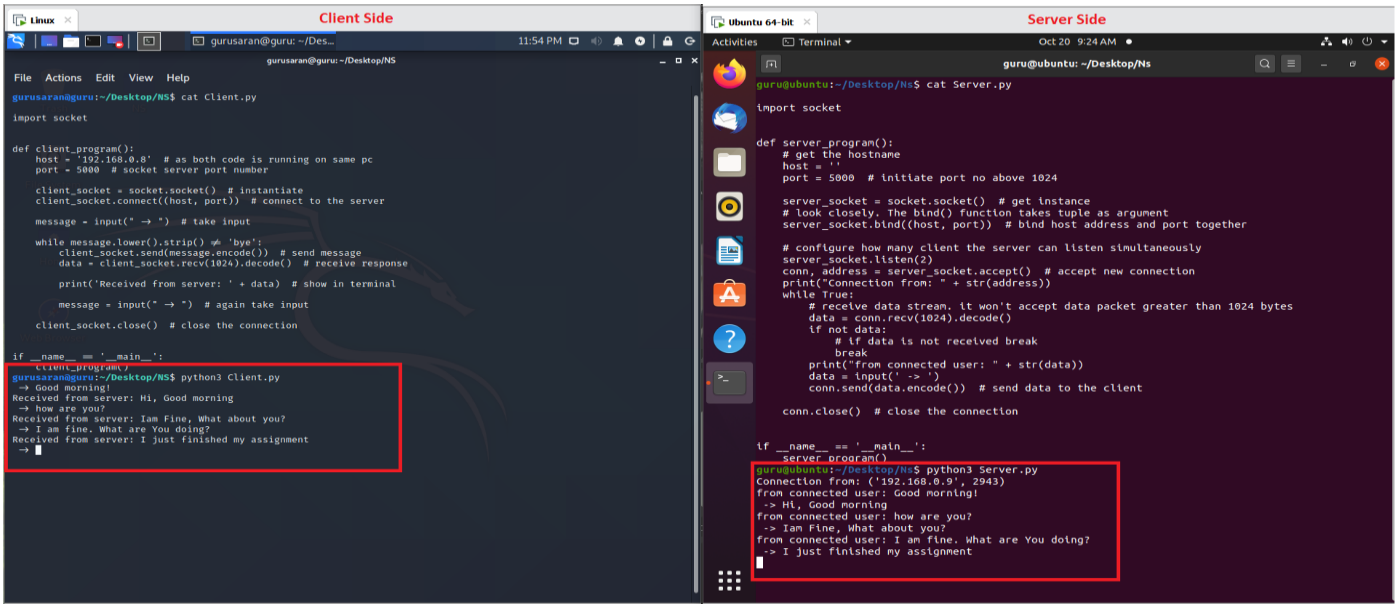
LAB-5

Implement the socket program as given in https://www.journaldev.com/15906/python-socket-programming-server-client. Keep one VM as a server another one as client and show the output on client and server sides

Output for Client-Server Socket Program:



```
gurusaran@guru:~/Desktop/NS$ cat Client.py

import socket


def client_program():
    host = '192.168.0.8'  # as both code is running on same pc
    port = 5000  # socket server port number

    client_socket = socket.socket()  # instantiate
    client_socket.connect((host, port))  # connect to the server

    message = input(" → ")  # take input

    while message.lower().strip() ≠ 'bye':
        client_socket.send(message.encode())  # send message
        data = client_socket.recv(1024).decode()  # receive response

        print('Received from server: ' + data)  # show in terminal

        message = input(" → ")  # again take input

    client_socket.close()  # close the connection


if __name__ == '__main__':
    client_program()
gurusaran@guru:~/Desktop/NS$ python3 Client.py
 → Good morning!
Received from server: Hi, Good morning
 → how are you?
Received from server: Iam Fine, What about you?
 → I am fine. What are You doing?
Received from server: I just finished my assignment
 → 
```

```
guru@ubuntu:~/Desktop/Ns$ cat Server.py

import socket


def server_program():
    # get the hostname
    host = ''
    port = 5000  # initiate port no above 1024

    server_socket = socket.socket()  # get instance
    # look closely. The bind() function takes tuple as argument
    server_socket.bind((host, port))  # bind host address and port together

    # configure how many client the server can listen simultaneously
    server_socket.listen(2)
    conn, address = server_socket.accept()  # accept new connection
    print("Connection from: " + str(address))
    while True:
        # receive data stream. it won't accept data packet greater than 1024 bytes
        data = conn.recv(1024).decode()
        if not data:
            # if data is not received break
            break
        print("from connected user: " + str(data))
        data = input(' -> ')
        conn.send(data.encode())  # send data to the client

    conn.close()  # close the connection


if __name__ == '__main__':
    server_program()
guru@ubuntu:~/Desktop/Ns$ python3 Server.py
Connection from: ('192.168.0.9', 2943)
from connected user: Good morning!
 -> Hi, Good morning
from connected user: how are you?
 -> Iam Fine, What about you?
from connected user: I am fine. What are You doing?
 -> I just finished my assignment
```