

# EE 533- LAB 1

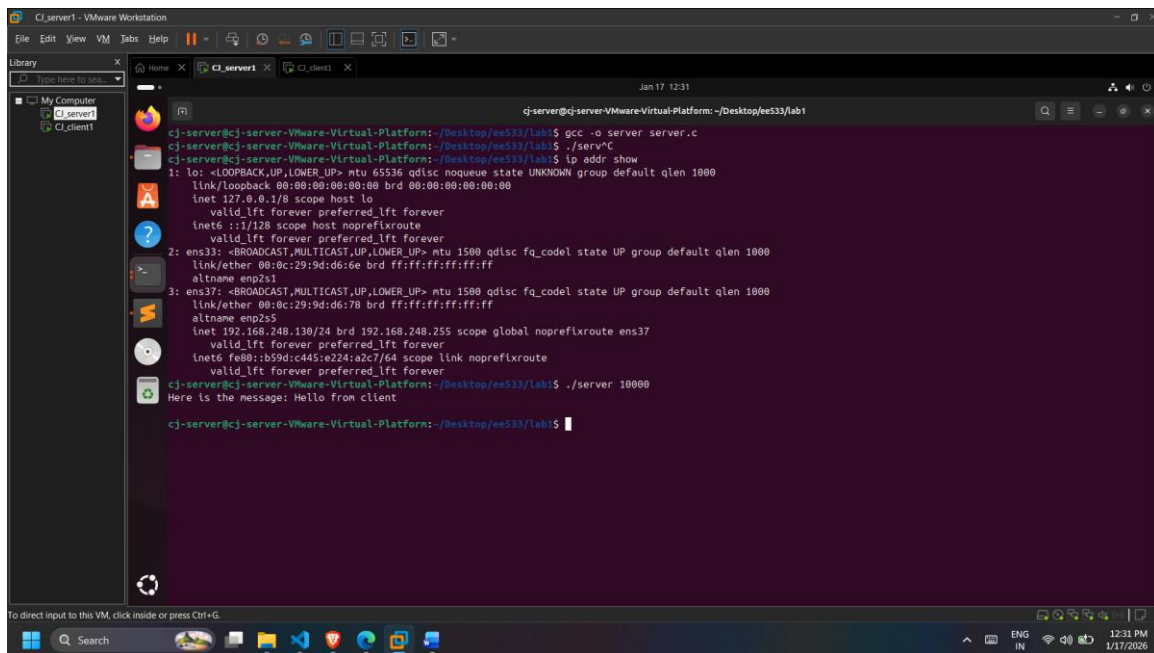
Name: Chaitanya Joshi

USC ID: 4991436817 ([joshic@usc.edu](mailto:joshic@usc.edu))

GitHub repo: [https://github.com/chaijosh/sp26\\_ee533\\_lab1](https://github.com/chaijosh/sp26_ee533_lab1)

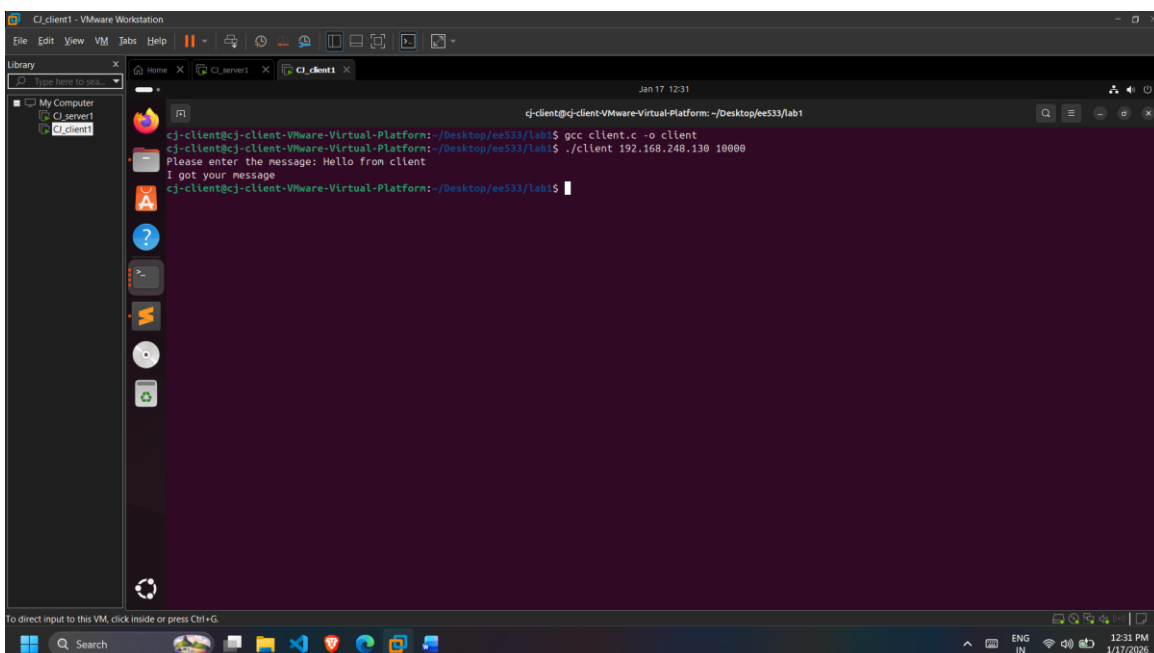
## Stream socket (TCP-IP) protocol with single server and client:

- Code at server side- *server.c*  
Code at client side- *client.c*
- Output screenshots- server and client communicating on port 10000.



```
cj-server@cj-server-VMware-Virtual-Platform: ~/Desktop/ee533/lab1
cj-server@cj-server-VMware-Virtual-Platform: ~/Desktop/ee533/lab1$ gcc -o server server.c
cj-server@cj-server-VMware-Virtual-Platform: ~/Desktop/ee533/lab1$ ./server
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
    valid_lft forever preferred_lft forever
inet6 ::1/128 scope host noprefixroute
    valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
link/ether 00:0c:29:9d:d6:6e brd ff:ff:ff:ff:ff:ff
    altname enp2s1
3: ens37: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
link/ether 00:0c:29:9d:d6:7b brd ff:ff:ff:ff:ff:ff
    altname enp2s5
inet 192.168.248.130/24 brd 192.168.248.255 scope global noprefixroute ens37
    valid_lft forever preferred_lft forever
inet6 fe80::b59d:c445:e24:a2c7/64 scope link noprefixroute
    valid_lft forever preferred_lft forever
cj-server@cj-server-VMware-Virtual-Platform: ~/Desktop/ee533/lab1$ ./server 10000
Here is the message: Hello from client
cj-server@cj-server-VMware-Virtual-Platform: ~/Desktop/ee533/lab1$
```

Figure 1: Server output with single client



```
cj-client@cj-client-VMware-Virtual-Platform: ~/Desktop/ee533/lab1
cj-client@cj-client-VMware-Virtual-Platform: ~/Desktop/ee533/lab1$ gcc client.c -o client
cj-client@cj-client-VMware-Virtual-Platform: ~/Desktop/ee533/lab1$ ./client 192.168.248.130 10000
Please enter the message: Hello from client
I got your message
cj-client@cj-client-VMware-Virtual-Platform: ~/Desktop/ee533/lab1$
```

Figure 2: Client output

## Stream socket (TCP-IP) protocol with single server and multiple client:

- Code at server side by adding infinite loop and process forking- *server\_multiple\_clients.c*  
Code at client side- *client.c*
- Output screenshots- server and 4 clients communicating on port 10000.
  - We can see in the server side (Fig. 3 & 4) that there are multiple forks created that individually cater to simultaneous client connections.
  - At client side (Fig. 5, we run *client.c* 4 times in different terminals and run them concurrently.

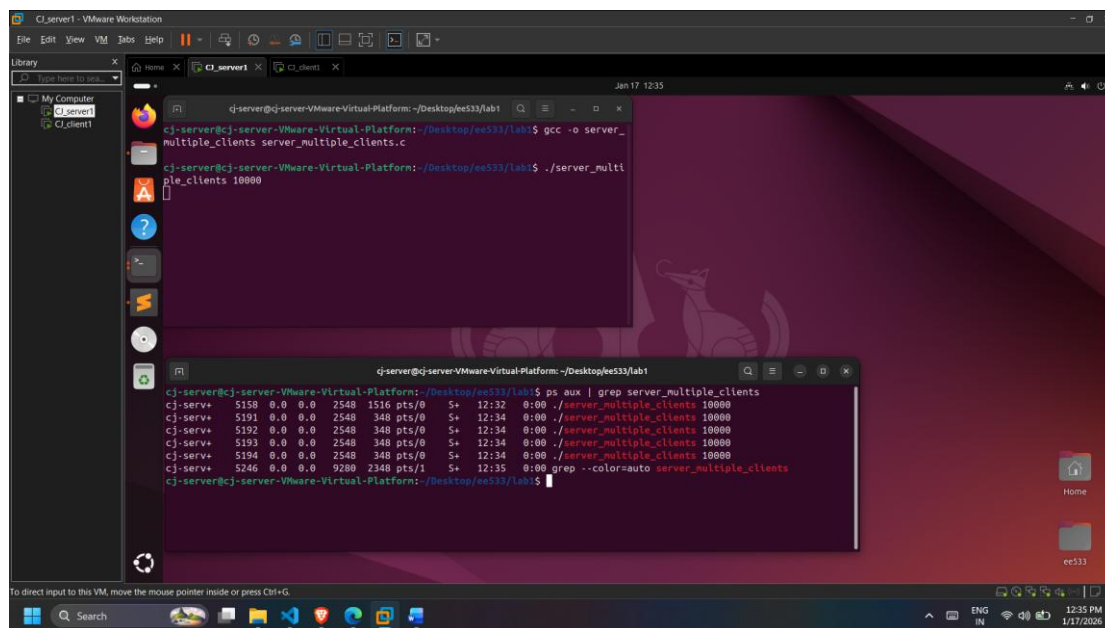


Figure 3: Server with multiple parallel connections (bottom terminal shows number of forked processes)

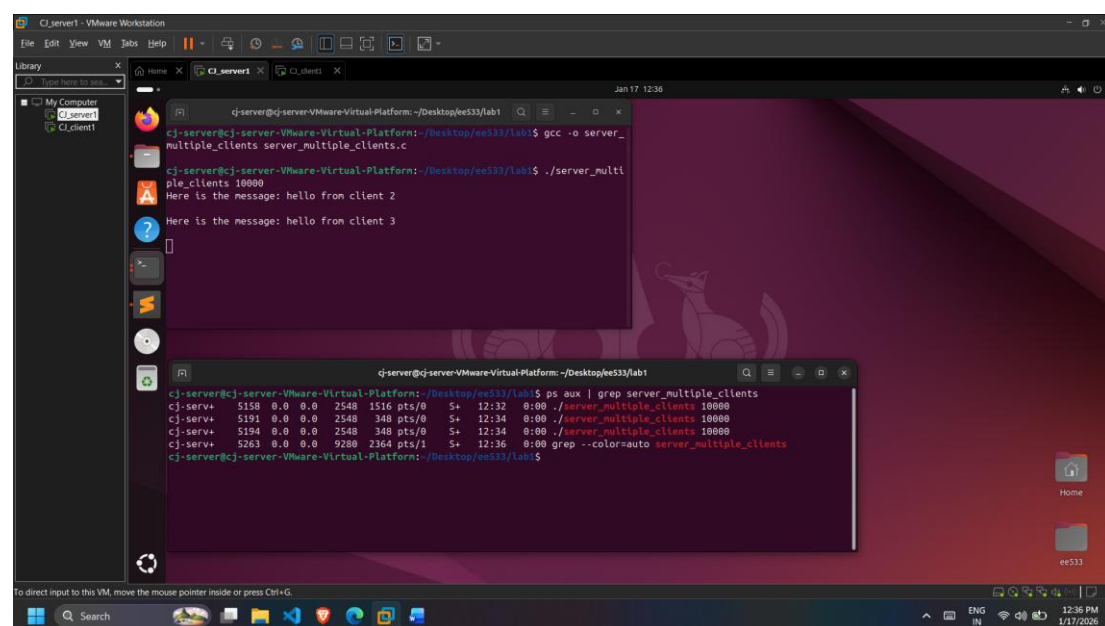


Figure 4: Server side receives few client messages and still has other clients connected

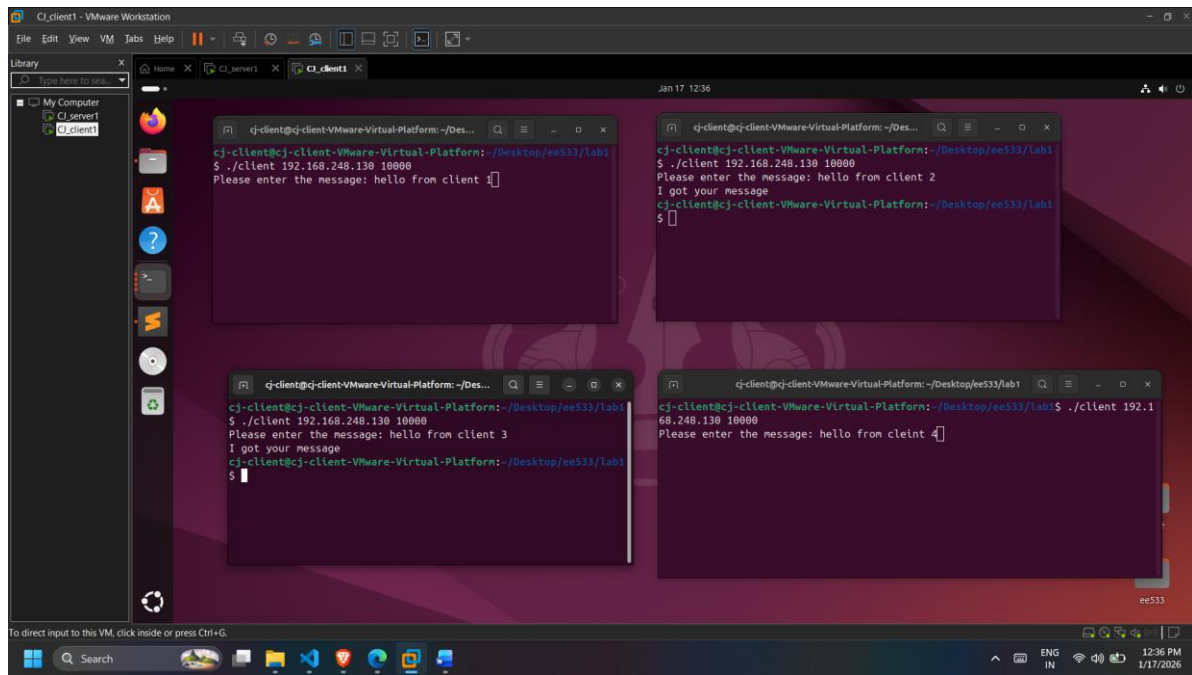


Figure 5: Multiple clients messaging the server

### Datagram socket (UDP-IP) protocol with single server and multiple clients:

- Code at server side- *server\_datagram.c*  
Code at client side- *client\_datagram.c*
- Output screenshots- server and clients communicating on port 10000 using UDP protocol.

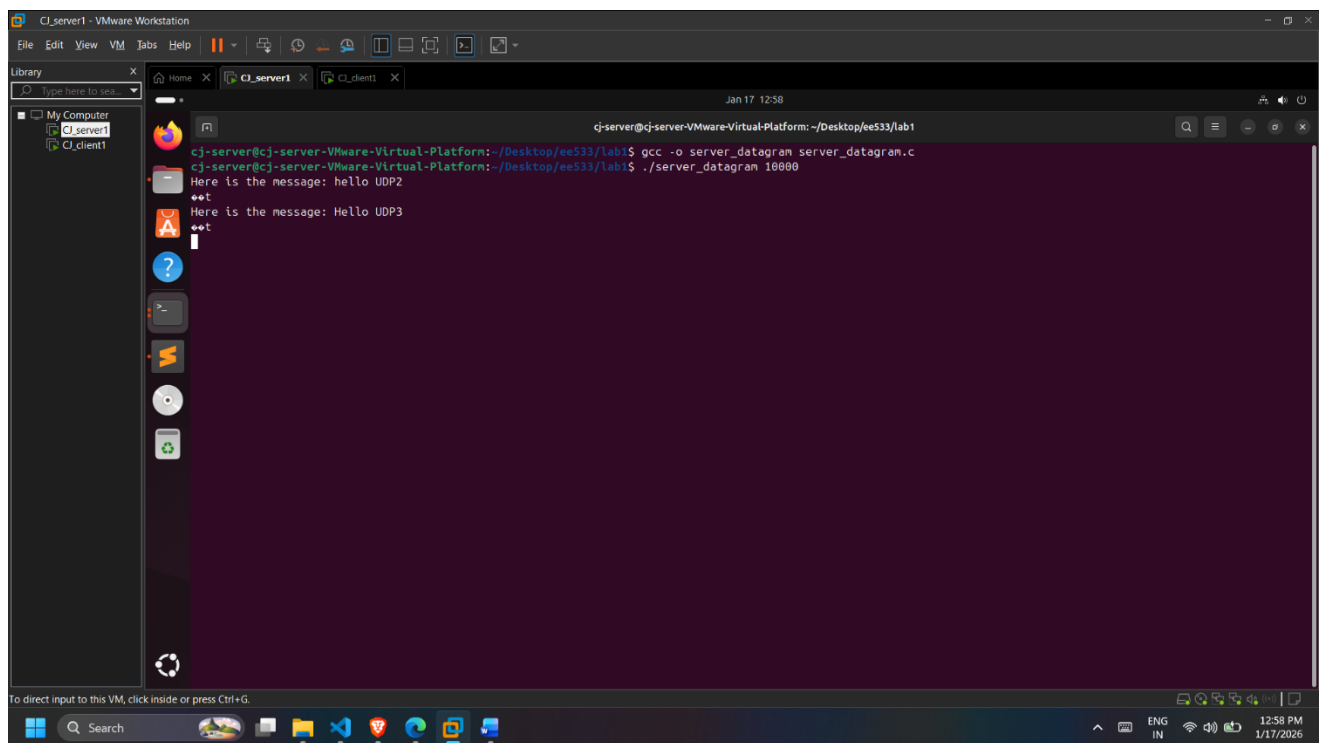


Figure 6: Server with datagram socket running UDP

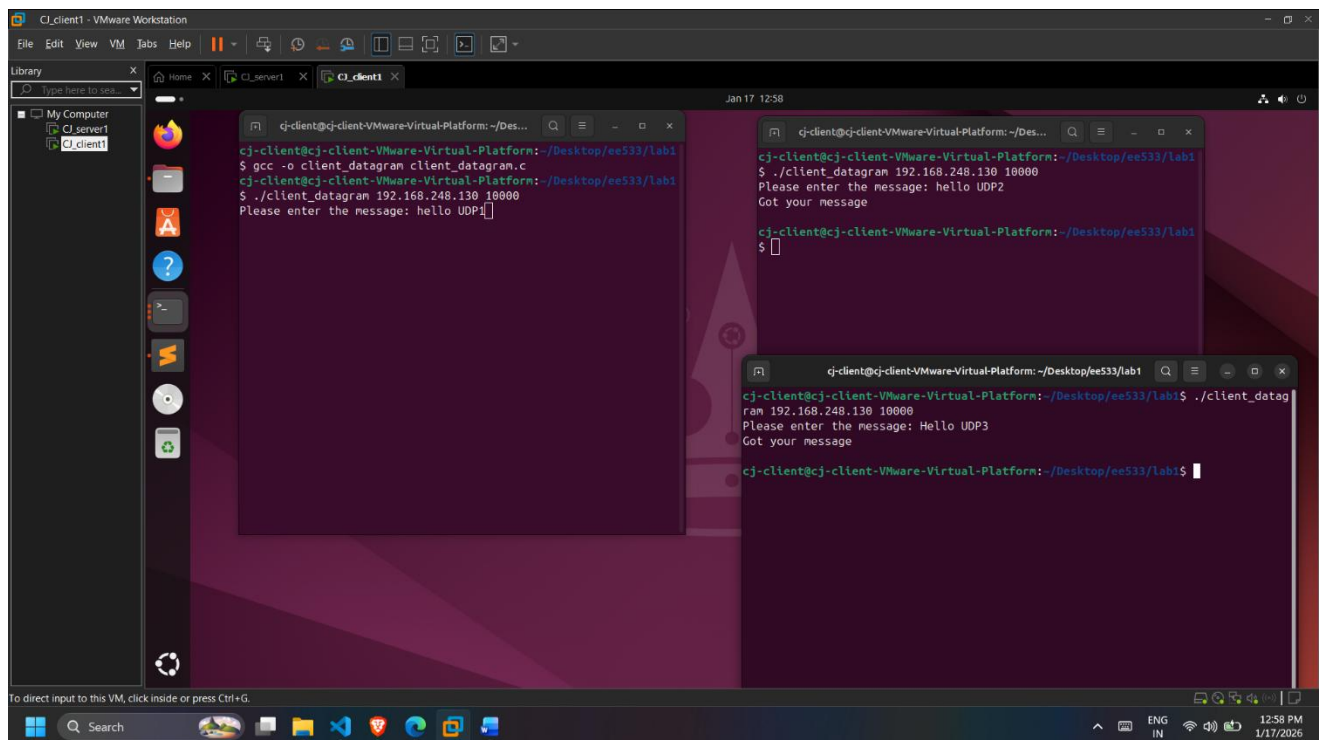


Figure 7: Multiple clients communicating with server using UDP

## Unix domain sockets:

- Code at server side- `U_server.c` (taken from [linuxhowtos.org/data/6/U\\_server.c](https://linuxhowtos.org/data/6/U_server.c))  
Code at client side- `U_client.c` (taken from [linuxhowtos.org/data/6/U\\_client.c](https://linuxhowtos.org/data/6/U_client.c))
- Output screenshots- Communication between 2 processes using Unix domain on a single host

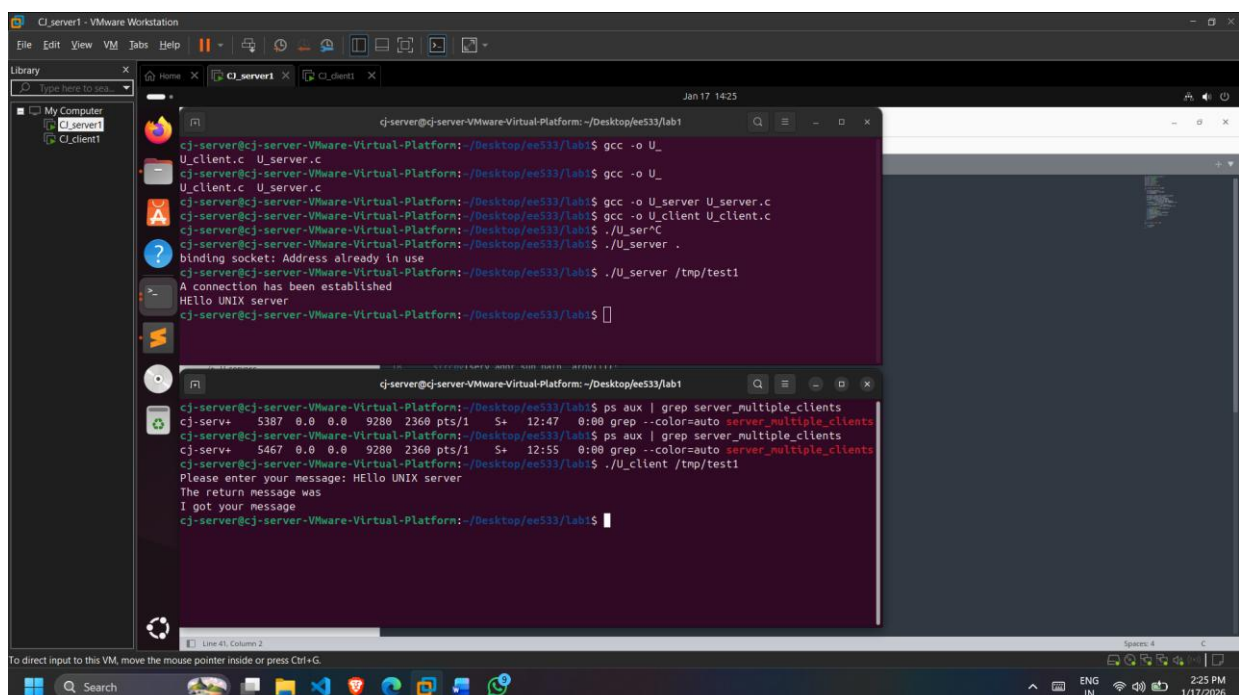


Figure 8: Unix domain communication