EE533 NETWORK PROCESSOR DESIGN & PROGRAMMING

LAB#1: Familiarity with VM & Sockets

Instructor: Prof. Young Cho, PhD

CREATED AND COMPILED BY:
SARTHAK JAIN

(MS EE, UNIVERSITY OF SOUTHERN CALIFORNIA)

GITHUB LINK FOR MY REPOSITORY:

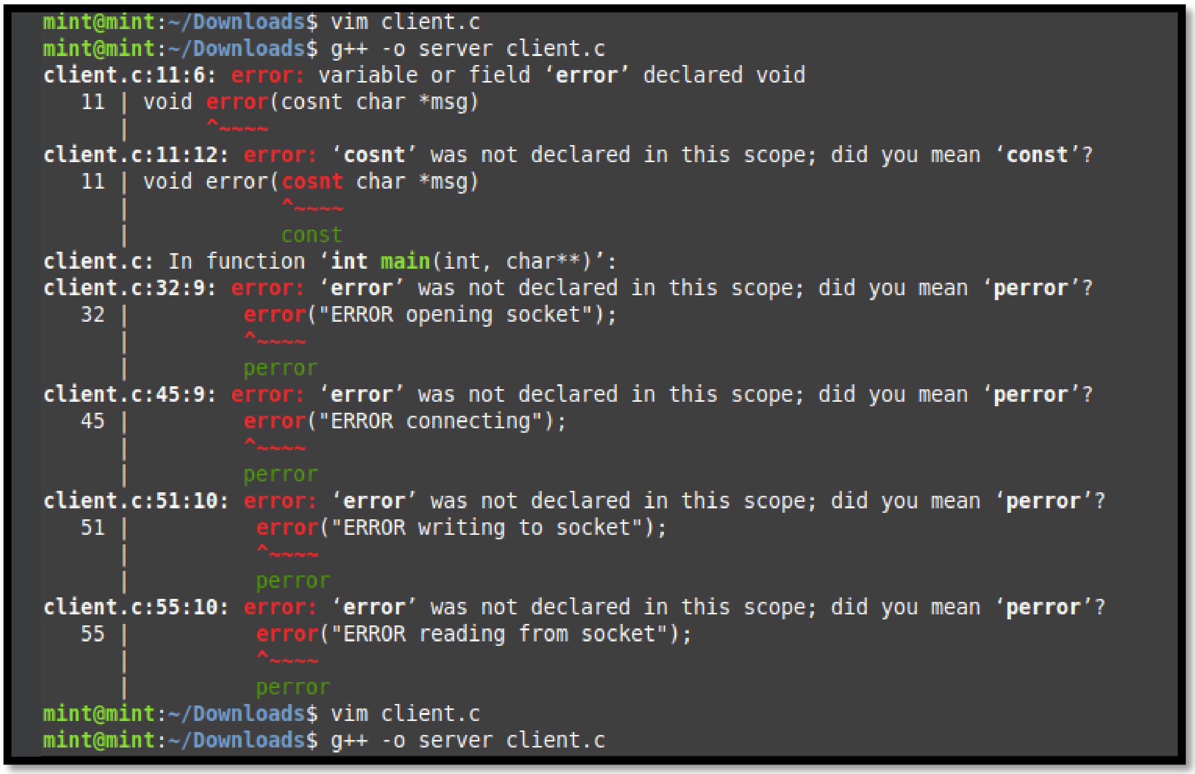
You'll find all the codes and the executables on this repository.

https://github.com/SARTHAK-JAIN-ASIC/EE533/tree/main/LAB1

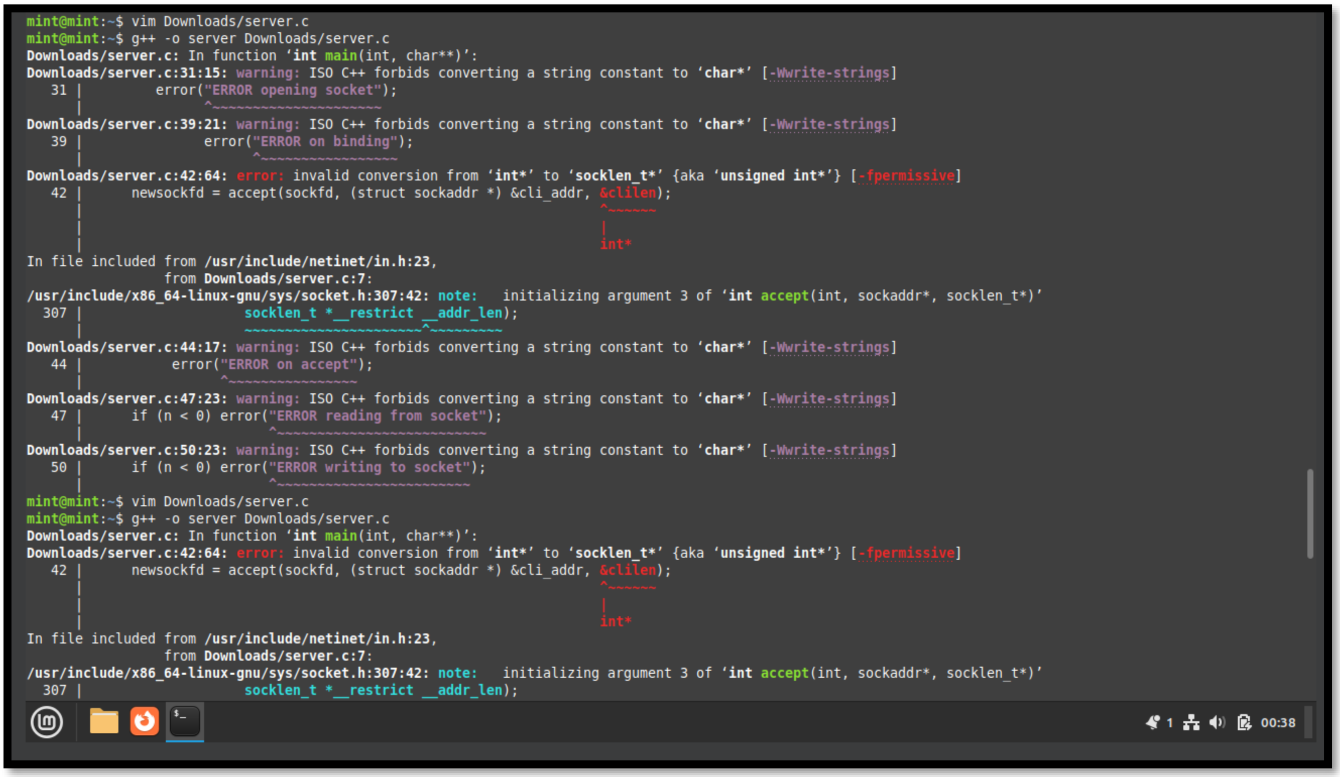
PART 1:

In the first part, we first had to make changes to the original code provided.

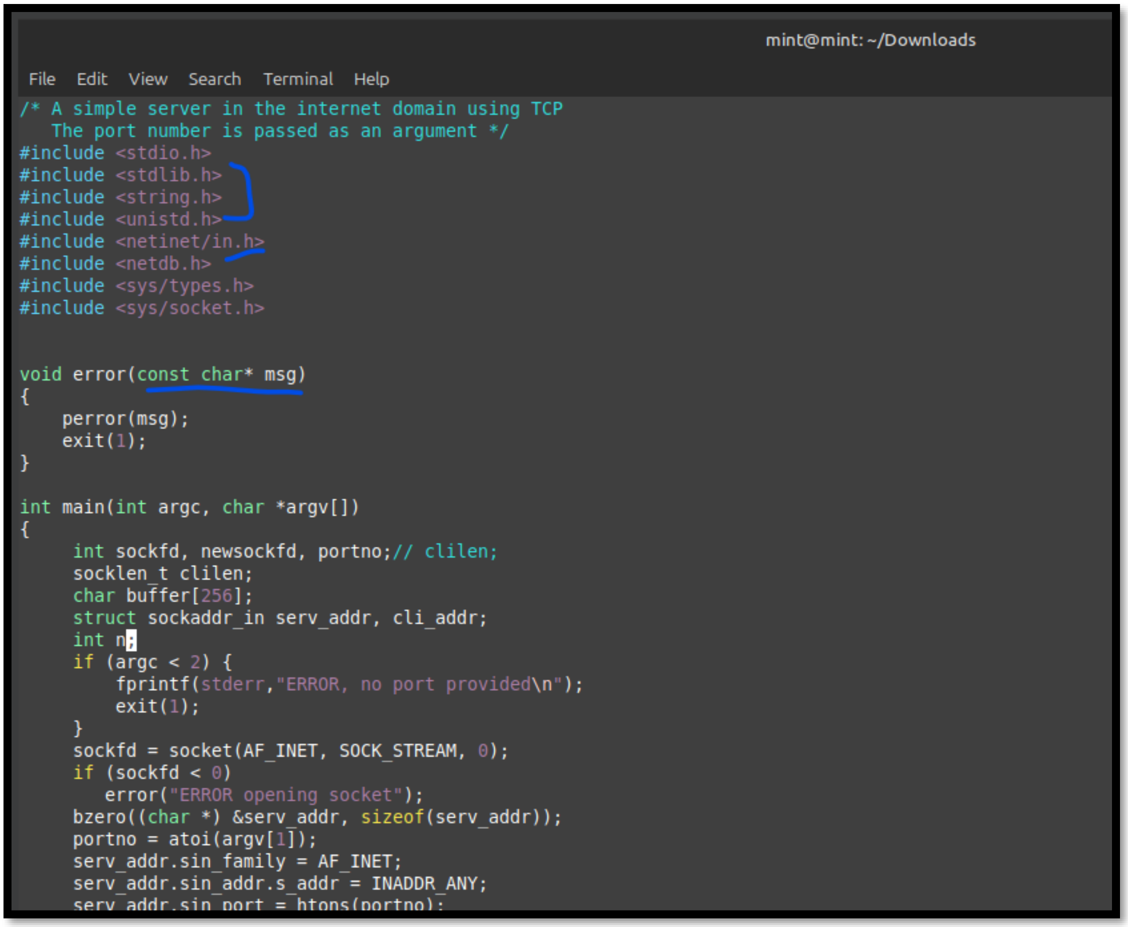
1. We added a few header files (the lack of which was causing compilation errors on my end).
2. We changed the error function's arguments to be const instead of variable since we cannot convert a constant string to char*
3. Changed data type of clilen from int to socklen_t.



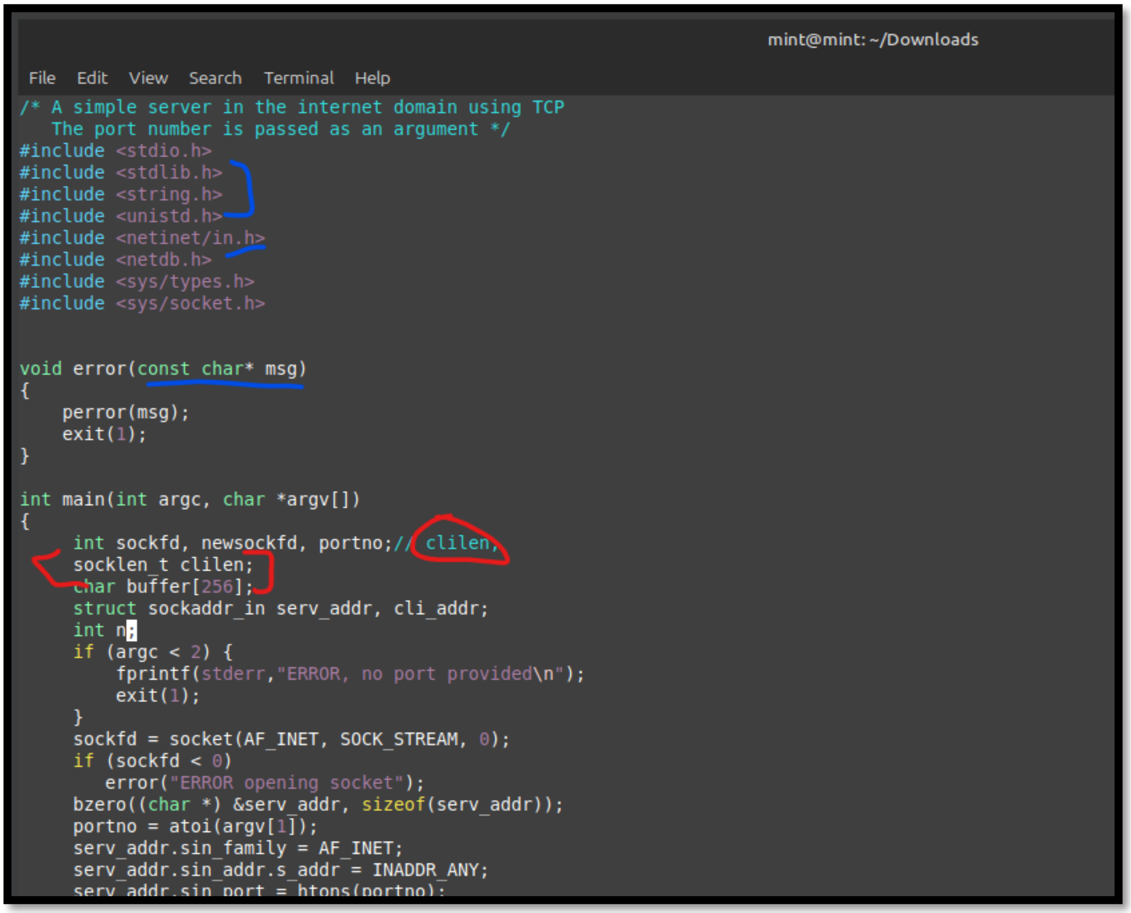
```
mint@mint:~/Downloads$ vim client.c
mint@mint:~/Downloads$ g++ -o server client.c
client.c:11:6: error: variable or field ‘error’ declared void
   11 | void error(cosnt char *msg)
      |      ^~~~~
client.c:11:12: error: ‘cosnt’ was not declared in this scope; did you mean ‘const’?
   11 | void error(cosnt char *msg)
      |            ^~~~~
      |            const
client.c: In function ‘int main(int, char**)’:
client.c:32:9: error: ‘error’ was not declared in this scope; did you mean ‘perror’?
   32 |         error("ERROR opening socket");
      |         ^~~~~
      |         perror
client.c:45:9: error: ‘error’ was not declared in this scope; did you mean ‘perror’?
   45 |         error("ERROR connecting");
      |         ^~~~~
      |         perror
client.c:51:10: error: ‘error’ was not declared in this scope; did you mean ‘perror’?
   51 |          error("ERROR writing to socket");
      |          ^~~~~
      |          perror
client.c:55:10: error: ‘error’ was not declared in this scope; did you mean ‘perror’?
   55 |          error("ERROR reading from socket");
      |          ^~~~~
      |          perror
mint@mint:~/Downloads$ vim client.c
mint@mint:~/Downloads$ g++ -o server client.c
```



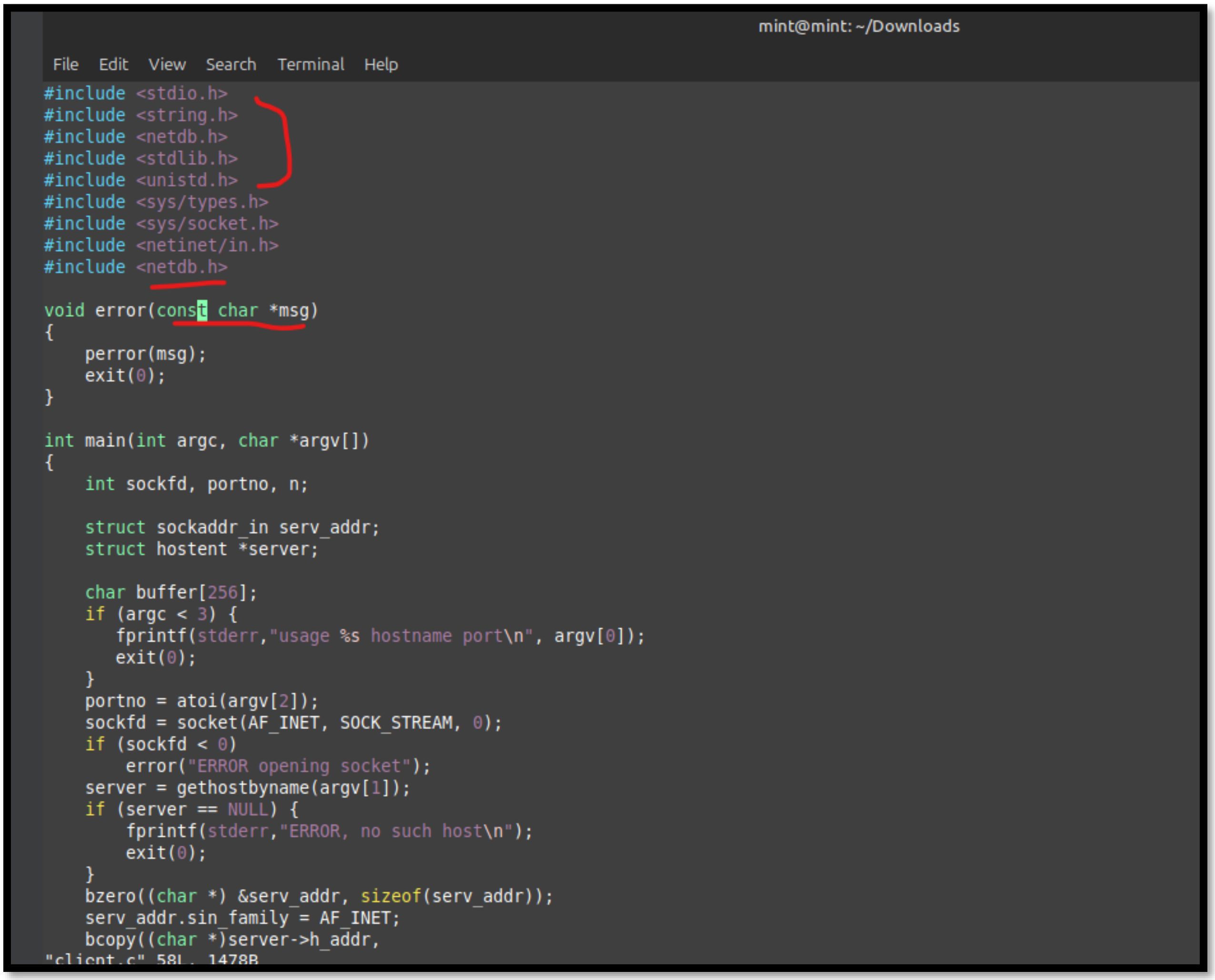
```
mint@mint:~$ vim Downloads/server.c
mint@mint:~$ g++ -o server Downloads/server.c
Downloads/server.c: In function ‘int main(int, char**)’:
Downloads/server.c:31:15: warning: ISO C++ forbids converting a string constant to ‘char*’ [-Wwrite-strings]
   31 |          error("ERROR opening socket");
      |                ^~~~~~~~~~~~~~~~~~~~~~
Downloads/server.c:39:21: warning: ISO C++ forbids converting a string constant to ‘char*’ [-Wwrite-strings]
   39 |                error("ERROR on binding");
      |                      ^~~~~~~~~~~~~~~~~~
Downloads/server.c:42:64: error: invalid conversion from ‘int*’ to ‘socklen_t*’ {aka ‘unsigned int*’} [-fpermissive]
   42 |      newsockfd = accept(sockfd, (struct sockaddr *) &cli_addr, &clilen);
      |                                                                ^~~~~~~
      |                                                                |
      |                                                                int*
In file included from /usr/include/netinet/in.h:23,
                 from Downloads/server.c:7:
/usr/include/x86_64-linux-gnu/sys/socket.h:307:42: note:   initializing argument 3 of ‘int accept(int, sockaddr*, socklen_t*)’
  307 |                    socklen_t *__restrict __addr_len);
      |                    ~~~~~~~~~~~~~~~~~~~~~~^~~~~~~~~~
Downloads/server.c:44:17: warning: ISO C++ forbids converting a string constant to ‘char*’ [-Wwrite-strings]
   44 |           error("ERROR on accept");
      |                 ^~~~~~~~~~~~~~~~~
Downloads/server.c:47:23: warning: ISO C++ forbids converting a string constant to ‘char*’ [-Wwrite-strings]
   47 |      if (n < 0) error("ERROR reading from socket");
      |                       ^~~~~~~~~~~~~~~~~~~~~~~~~~~
Downloads/server.c:50:23: warning: ISO C++ forbids converting a string constant to ‘char*’ [-Wwrite-strings]
   50 |      if (n < 0) error("ERROR writing to socket");
      |                       ^~~~~~~~~~~~~~~~~~~~~~~~~
mint@mint:~$ vim Downloads/server.c
mint@mint:~$ g++ -o server Downloads/server.c
Downloads/server.c: In function ‘int main(int, char**)’:
Downloads/server.c:42:64: error: invalid conversion from ‘int*’ to ‘socklen_t*’ {aka ‘unsigned int*’} [-fpermissive]
   42 |      newsockfd = accept(sockfd, (struct sockaddr *) &cli_addr, &clilen);
      |                                                                ^~~~~~~
      |                                                                |
      |                                                                int*
In file included from /usr/include/netinet/in.h:23,
                 from Downloads/server.c:7:
/usr/include/x86_64-linux-gnu/sys/socket.h:307:42: note:   initializing argument 3 of ‘int accept(int, sockaddr*, socklen_t*)’
  307 |                    socklen_t *__restrict __addr_len);
```



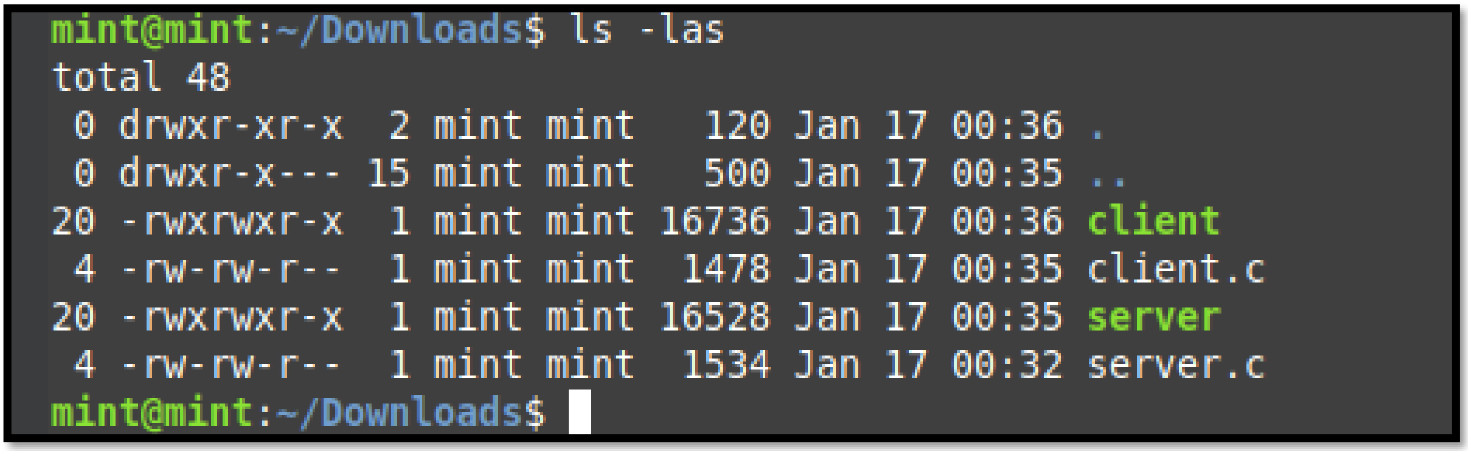
```
/* A simple server in the internet domain using TCP
   The port number is passed as an argument */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <netinet/in.h>
#include <netdb.h>
#include <sys/types.h>
#include <sys/socket.h>


void error(const char* msg)
{
    perror(msg);
    exit(1);
}

int main(int argc, char *argv[])
{
     int sockfd, newsockfd, portno;// clilen;
     socklen_t clilen;
     char buffer[256];
     struct sockaddr_in serv_addr, cli_addr;
     int n;
     if (argc < 2) {
         fprintf(stderr,"ERROR, no port provided\n");
         exit(1);
     }
     sockfd = socket(AF_INET, SOCK_STREAM, 0);
     if (sockfd < 0)
        error("ERROR opening socket");
     bzero((char *) &serv_addr, sizeof(serv_addr));
     portno = atoi(argv[1]);
     serv_addr.sin_family = AF_INET;
     serv_addr.sin_addr.s_addr = INADDR_ANY;
     serv_addr.sin_port = htons(portno);
```



```
/* A simple server in the internet domain using TCP
   The port number is passed as an argument */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <netinet/in.h>
#include <netdb.h>
#include <sys/types.h>
#include <sys/socket.h>


void error(const char* msg)
{
    perror(msg);
    exit(1);
}

int main(int argc, char *argv[])
{
     int sockfd, newsockfd, portno;// clilen;
     socklen_t clilen;
     char buffer[256];
     struct sockaddr_in serv_addr, cli_addr;
     int n;
     if (argc < 2) {
         fprintf(stderr,"ERROR, no port provided\n");
         exit(1);
     }
     sockfd = socket(AF_INET, SOCK_STREAM, 0);
     if (sockfd < 0)
        error("ERROR opening socket");
     bzero((char *) &serv_addr, sizeof(serv_addr));
     portno = atoi(argv[1]);
     serv_addr.sin_family = AF_INET;
     serv_addr.sin_addr.s_addr = INADDR_ANY;
     serv_addr.sin_port = htons(portno);
```



```
#include <stdio.h>
#include <string.h>
#include <netdb.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>

void error(const char *msg)
{
    perror(msg);
    exit(0);
}

int main(int argc, char *argv[])
{
    int sockfd, portno, n;

    struct sockaddr_in serv_addr;
    struct hostent *server;

    char buffer[256];
    if (argc < 3) {
       fprintf(stderr,"usage %s hostname port\n", argv[0]);
       exit(0);
    }
    portno = atoi(argv[2]);
    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd < 0)
        error("ERROR opening socket");
    server = gethostbyname(argv[1]);
    if (server == NULL) {
        fprintf(stderr,"ERROR, no such host\n");
        exit(0);
    }
    bzero((char *) &serv_addr, sizeof(serv_addr));
    serv_addr.sin_family = AF_INET;
    bcopy((char *)server->h_addr,
"client.c" 58L, 1478B
```



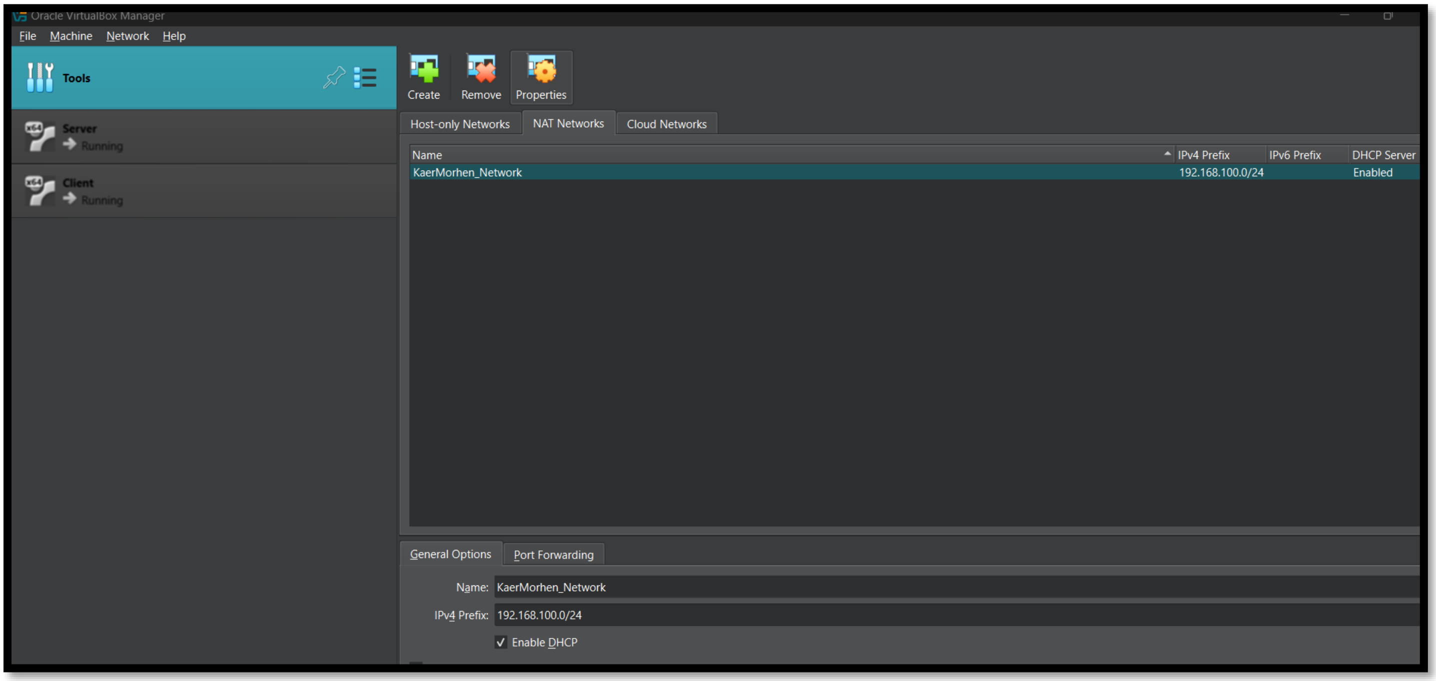
```
mint@mint:~/Downloads$ ls -las
total 48
 0 drwxr-xr-x  2 mint mint   120 Jan 17 00:36 .
 0 drwxr-x--- 15 mint mint   500 Jan 17 00:35 ..
20 -rwxrwxr-x  1 mint mint 16736 Jan 17 00:36 client
 4 -rw-rw-r--  1 mint mint  1478 Jan 17 00:35 client.c
20 -rwxrwxr-x  1 mint mint 16528 Jan 17 00:35 server
 4 -rw-rw-r--  1 mint mint  1534 Jan 17 00:32 server.c
mint@mint:~/Downloads$
```

<u>All these changes enabled us to compile the code correctly as seen above.</u>

PART 2:

Next task was to get the client and the server to ping each other.

For this, I created a NAT network, that would “connect” server and client VMs (as below)



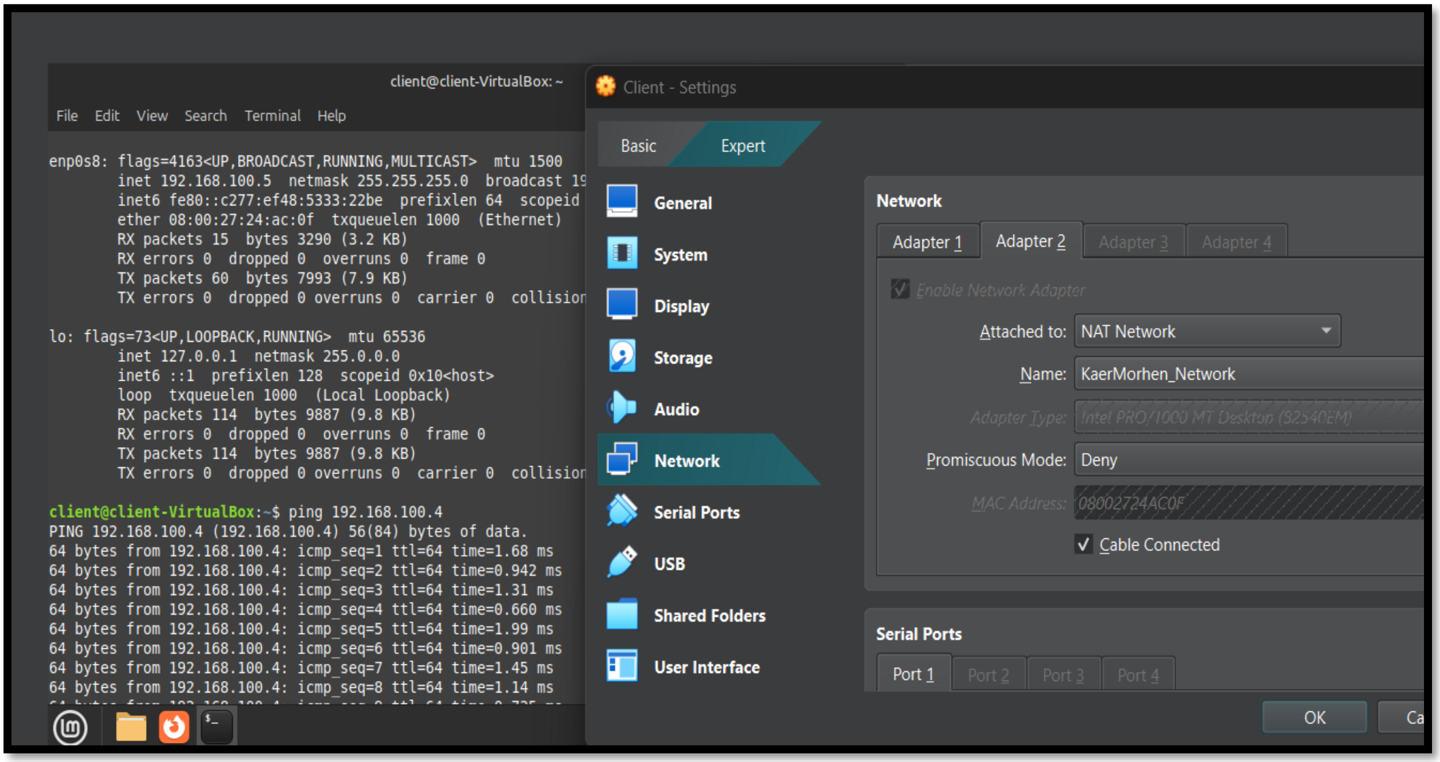
Name	IPv4 Prefix	IPv6 Prefix	DHCP Server
KaerMorhen_Network	192.168.100.0/24		Enabled

Name: KaerMorhen_Network

IPv4 Prefix: 192.168.100.0/24

☑ Enable DHCP

Then, I created one adapter for server and client (and attached it to the NAT network as above).



```
enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.100.5  netmask 255.255.255.0  broadcast 19
        inet6 fe80::c277:ef48:5333:22be  prefixlen 64  scopeid
        ether 08:00:27:24:ac:0f  txqueuelen 1000  (Ethernet)
        RX packets 15  bytes 3290 (3.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 60  bytes 7993 (7.9 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collision

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 114  bytes 9887 (9.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 114  bytes 9887 (9.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collision

client@client-VirtualBox:~$ ping 192.168.100.4
PING 192.168.100.4 (192.168.100.4) 56(84) bytes of data.
64 bytes from 192.168.100.4: icmp_seq=1 ttl=64 time=1.68 ms
64 bytes from 192.168.100.4: icmp_seq=2 ttl=64 time=0.942 ms
64 bytes from 192.168.100.4: icmp_seq=3 ttl=64 time=1.31 ms
64 bytes from 192.168.100.4: icmp_seq=4 ttl=64 time=0.660 ms
64 bytes from 192.168.100.4: icmp_seq=5 ttl=64 time=1.99 ms
64 bytes from 192.168.100.4: icmp_seq=6 ttl=64 time=0.901 ms
64 bytes from 192.168.100.4: icmp_seq=7 ttl=64 time=1.45 ms
64 bytes from 192.168.100.4: icmp_seq=8 ttl=64 time=1.14 ms
```

☑ Enable Network Adapter

Attached to: NAT Network

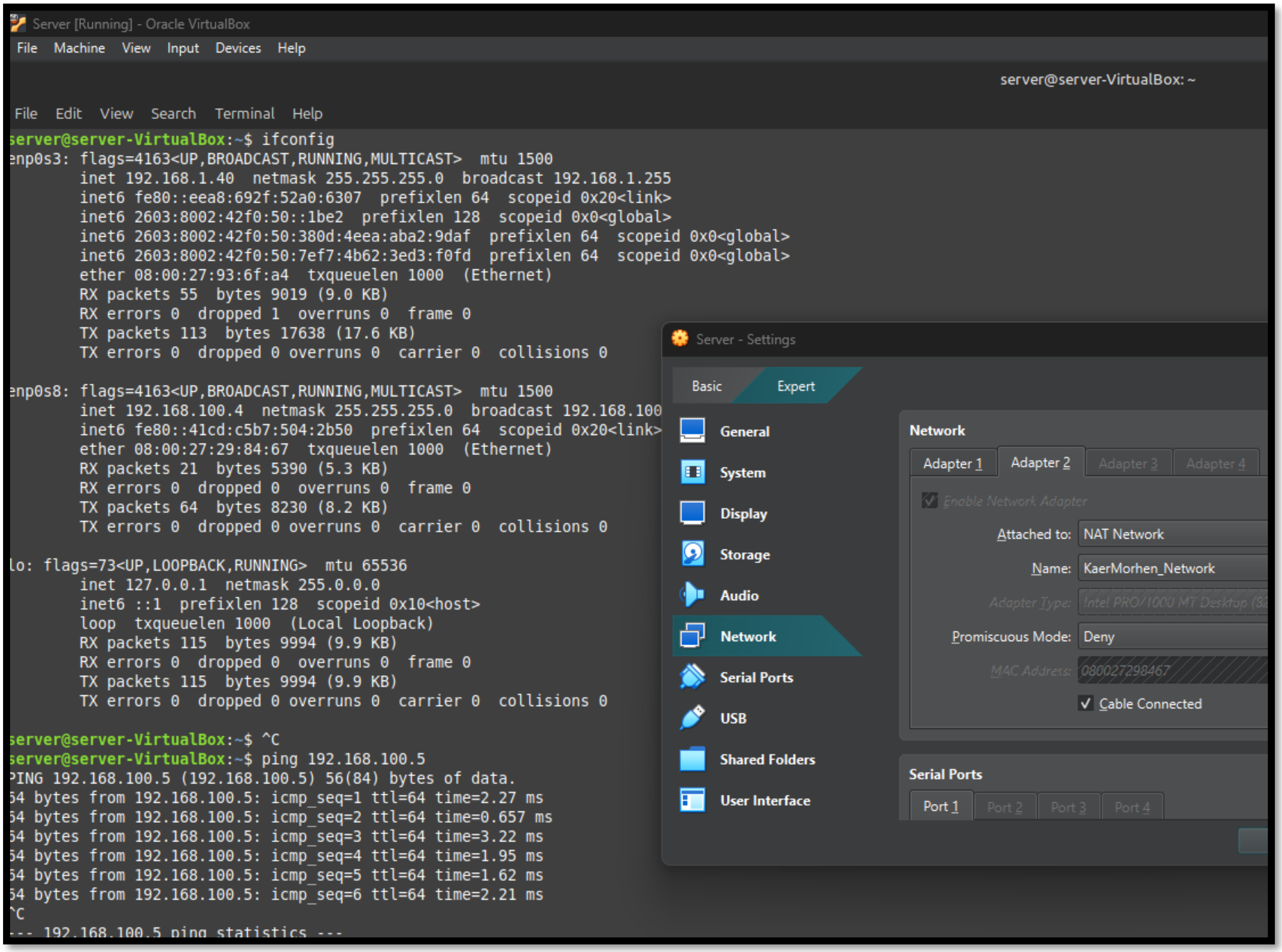
Name: KaerMorhen_Network

Adapter Type: Intel PRO/1000 MT Desktop (82540EM)

Promiscuous Mode: Deny

MAC Address: 08002724AC0F

☑ Cable Connected



Server [Running] - Oracle VirtualBox
File Machine View Input Devices Help
server@server-VirtualBox: ~
File Edit View Search Terminal Help

```
server@server-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.40  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::eea8:692f:52a0:6307  prefixlen 64  scopeid 0x20<link>
        inet6 2603:8002:42f0:50::1be2  prefixlen 128  scopeid 0x0<global>
        inet6 2603:8002:42f0:50:380d:4eea:aba2:9daf  prefixlen 64  scopeid 0x0<global>
        inet6 2603:8002:42f0:50:7ef7:4b62:3ed3:f0fd  prefixlen 64  scopeid 0x0<global>
        ether 08:00:27:93:6f:a4  txqueuelen 1000  (Ethernet)
        RX packets 55  bytes 9019 (9.0 KB)
        RX errors 0  dropped 1  overruns 0  frame 0
        TX packets 113  bytes 17638 (17.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.100.4  netmask 255.255.255.0  broadcast 192.168.100
        inet6 fe80::41cd:c5b7:504:2b50  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:29:84:67  txqueuelen 1000  (Ethernet)
        RX packets 21  bytes 5390 (5.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 64  bytes 8230 (8.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 115  bytes 9994 (9.9 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 115  bytes 9994 (9.9 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

server@server-VirtualBox:~$ ^C
server@server-VirtualBox:~$ ping 192.168.100.5
PING 192.168.100.5 (192.168.100.5) 56(84) bytes of data.
64 bytes from 192.168.100.5: icmp_seq=1 ttl=64 time=2.27 ms
64 bytes from 192.168.100.5: icmp_seq=2 ttl=64 time=0.657 ms
64 bytes from 192.168.100.5: icmp_seq=3 ttl=64 time=3.22 ms
64 bytes from 192.168.100.5: icmp_seq=4 ttl=64 time=1.95 ms
64 bytes from 192.168.100.5: icmp_seq=5 ttl=64 time=1.62 ms
64 bytes from 192.168.100.5: icmp_seq=6 ttl=64 time=2.21 ms
^C
--- 192.168.100.5 ping statistics ---
```

Network

Adapter 1 | Adapter 2 | Adapter 3 | Adapter 4

☑ Enable Network Adapter

Attached to: NAT Network

Name: KaerMorhen_Network

Adapter Type: Intel PRO/1000 MT Desktop (82

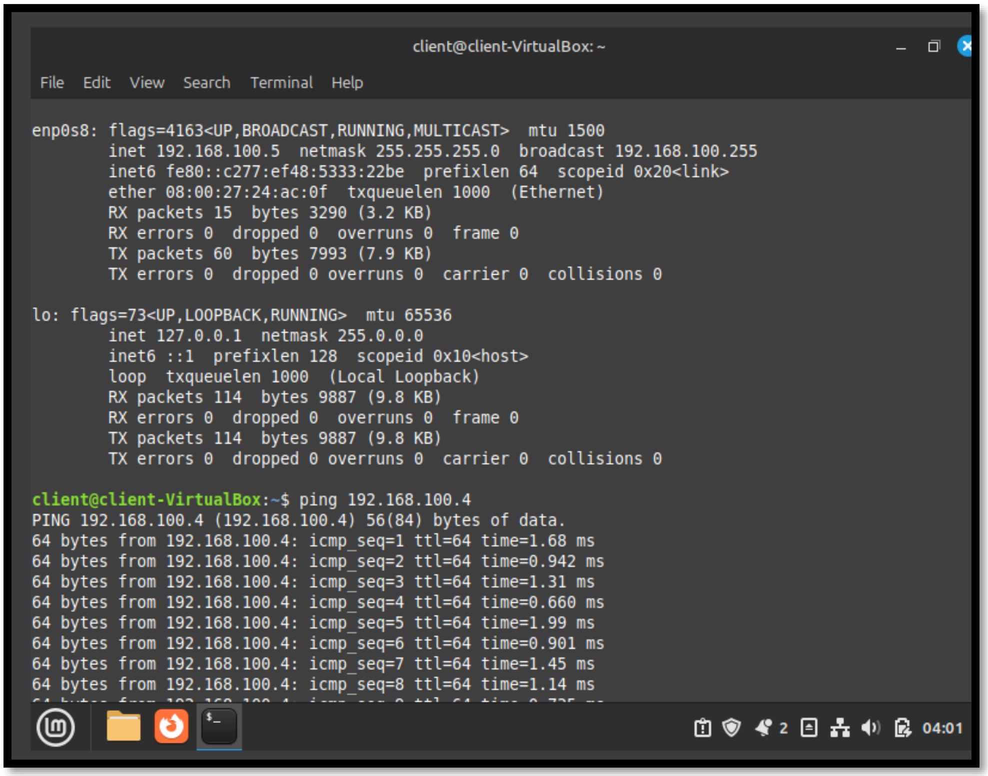
Promiscuous Mode: Deny

MAC Address: 080027298467

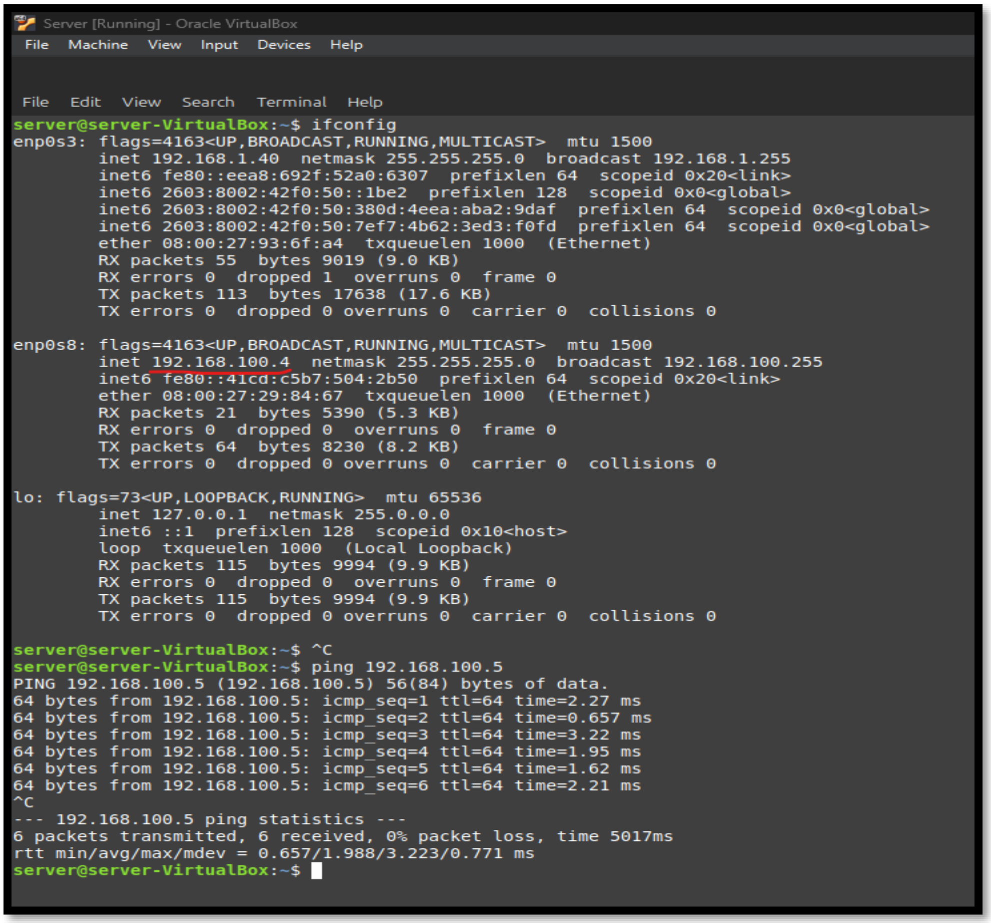
☑ Cable Connected

Serial Ports

Next step was to make sure the server and the client are communicating. For this, I figured out the IP addresses of the machines and made them ping using the ID addresses. The bottom screenshots show them communicating.



```
enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.100.5  netmask 255.255.255.0  broadcast 192.168.100.255
        inet6 fe80::c277:ef48:5333:22be  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:24:ac:0f  txqueuelen 1000  (Ethernet)
        RX packets 15  bytes 3290 (3.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 60  bytes 7993 (7.9 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 114  bytes 9887 (9.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 114  bytes 9887 (9.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

client@client-VirtualBox:~$ ping 192.168.100.4
PING 192.168.100.4 (192.168.100.4) 56(84) bytes of data.
64 bytes from 192.168.100.4: icmp_seq=1 ttl=64 time=1.68 ms
64 bytes from 192.168.100.4: icmp_seq=2 ttl=64 time=0.942 ms
64 bytes from 192.168.100.4: icmp_seq=3 ttl=64 time=1.31 ms
64 bytes from 192.168.100.4: icmp_seq=4 ttl=64 time=0.660 ms
64 bytes from 192.168.100.4: icmp_seq=5 ttl=64 time=1.99 ms
64 bytes from 192.168.100.4: icmp_seq=6 ttl=64 time=0.901 ms
64 bytes from 192.168.100.4: icmp_seq=7 ttl=64 time=1.45 ms
64 bytes from 192.168.100.4: icmp_seq=8 ttl=64 time=1.14 ms
```

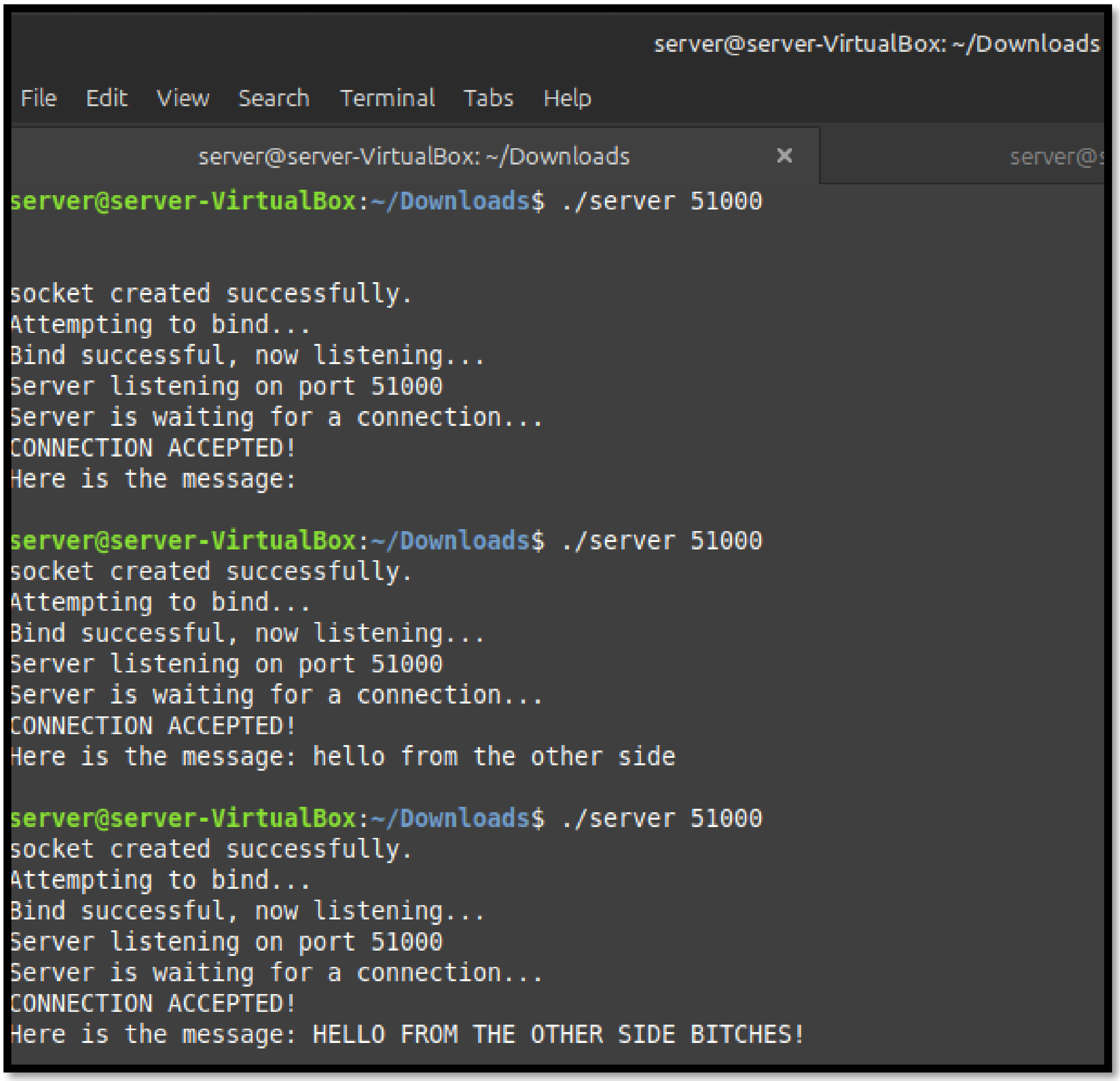


```
server@server-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.40  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::eea8:692f:52a0:6307  prefixlen 64  scopeid 0x20<link>
        inet6 2603:8002:42f0:50::1be2  prefixlen 128  scopeid 0x0<global>
        inet6 2603:8002:42f0:50:380d:4eea:aba2:9daf  prefixlen 64  scopeid 0x0<global>
        inet6 2603:8002:42f0:50:7ef7:4b62:3ed3:f0fd  prefixlen 64  scopeid 0x0<global>
        ether 08:00:27:93:6f:a4  txqueuelen 1000  (Ethernet)
        RX packets 55  bytes 9019 (9.0 KB)
        RX errors 0  dropped 1  overruns 0  frame 0
        TX packets 113  bytes 17638 (17.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.100.4  netmask 255.255.255.0  broadcast 192.168.100.255
        inet6 fe80::41cd:c5b7:504:2b50  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:29:84:67  txqueuelen 1000  (Ethernet)
        RX packets 21  bytes 5390 (5.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 64  bytes 8230 (8.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 115  bytes 9994 (9.9 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 115  bytes 9994 (9.9 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

server@server-VirtualBox:~$ ^C
server@server-VirtualBox:~$ ping 192.168.100.5
PING 192.168.100.5 (192.168.100.5) 56(84) bytes of data.
64 bytes from 192.168.100.5: icmp_seq=1 ttl=64 time=2.27 ms
64 bytes from 192.168.100.5: icmp_seq=2 ttl=64 time=0.657 ms
64 bytes from 192.168.100.5: icmp_seq=3 ttl=64 time=3.22 ms
64 bytes from 192.168.100.5: icmp_seq=4 ttl=64 time=1.95 ms
64 bytes from 192.168.100.5: icmp_seq=5 ttl=64 time=1.62 ms
64 bytes from 192.168.100.5: icmp_seq=6 ttl=64 time=2.21 ms
^C
--- 192.168.100.5 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5017ms
rtt min/avg/max/mdev = 0.657/1.988/3.223/0.771 ms
server@server-VirtualBox:~$ 
```

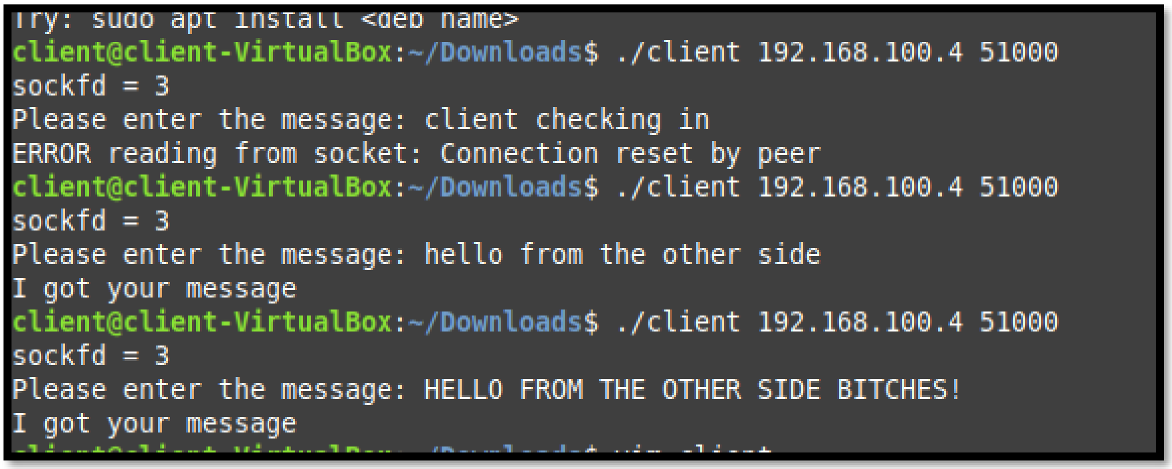
PART 3:

Next step was to actually run the executables, run the server and wait for it to listen after the binding to the address gets done.

Then we run the client, which basically makes the server go, “CONNECTION ACCEPTED!”. Finally, we send a message to the server by typing it in the client VM and see it live on the server (as shown in the images below).


server@server-VirtualBox: ~/Downloads
File
Edit
View
Search
Terminal
Tabs
Help
server@server-VirtualBox: ~/Downloads
server@s

```
server@server-VirtualBox:~/Downloads$ ./server 51000


socket created successfully.
Attempting to bind...
Bind successful, now listening...
Server listening on port 51000
Server is waiting for a connection...
CONNECTION ACCEPTED!
Here is the message:

server@server-VirtualBox:~/Downloads$ ./server 51000
socket created successfully.
Attempting to bind...
Bind successful, now listening...
Server listening on port 51000
Server is waiting for a connection...
CONNECTION ACCEPTED!
Here is the message: hello from the other side

server@server-VirtualBox:~/Downloads$ ./server 51000
socket created successfully.
Attempting to bind...
Bind successful, now listening...
Server listening on port 51000
Server is waiting for a connection...
CONNECTION ACCEPTED!
Here is the message: HELLO FROM THE OTHER SIDE BITCHES!
```

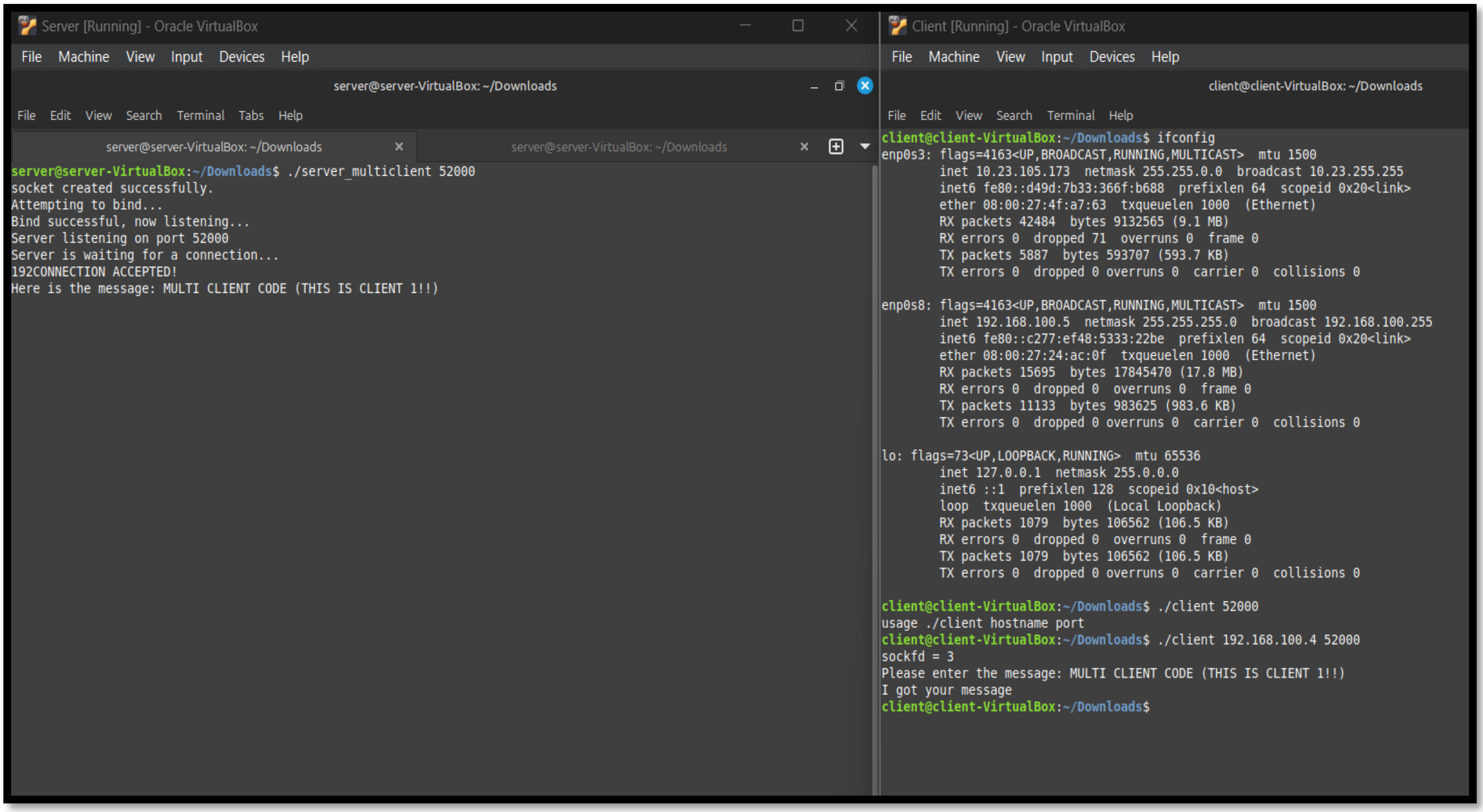


```
Try: sudo apt install <deb name>
client@client-VirtualBox:~/Downloads$ ./client 192.168.100.4 51000
sockfd = 3
Please enter the message: client checking in
ERROR reading from socket: Connection reset by peer
client@client-VirtualBox:~/Downloads$ ./client 192.168.100.4 51000
sockfd = 3
Please enter the message: hello from the other side
I got your message
client@client-VirtualBox:~/Downloads$ ./client 192.168.100.4 51000
sockfd = 3
Please enter the message: HELLO FROM THE OTHER SIDE BITCHES!
I got your message
client@client-VirtualBox:~/Downloads$ vim client
```

PART 4

For the multi-client connections, I made changes to the code as asked by the professor in the document and some of my own. Ran the executables as above and see the output. Only this time, the server didn't drop the connection after receiving message from the client. It was waiting for some other connection to take place (evident by the blinking cursor)

I did not make any changes to the client-side code for this part.



Server [Running] - Oracle VirtualBox
File Machine View Input Devices Help
server@server-VirtualBox: ~/Downloads
File Edit View Search Terminal Tabs Help
server@server-VirtualBox: ~/Downloads
server@server-VirtualBox: ~/Downloads
Client [Running] - Oracle VirtualBox
File Machine View Input Devices Help
client@client-VirtualBox: ~/Downloads
File Edit View Search Terminal Help

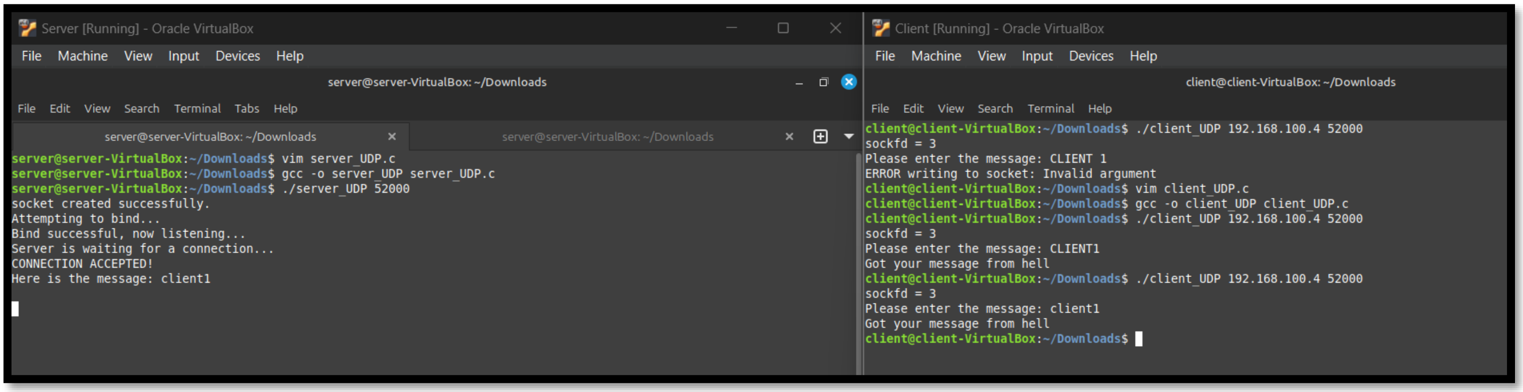
```
server@server-VirtualBox:~/Downloads$ ./server_multiclient 52000
socket created successfully.
Attempting to bind...
Bind successful, now listening...
Server listening on port 52000
Server is waiting for a connection...
192CONNECTION ACCEPTED!
Here is the message: MULTI CLIENT CODE (THIS IS CLIENT 1!!)
```

```
client@client-VirtualBox:~/Downloads$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.23.105.173  netmask 255.255.0.0  broadcast 10.23.255.255
        inet6 fe80::d49d:7b33:366f:b688  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:4f:a7:63  txqueuelen 1000  (Ethernet)
        RX packets 42484  bytes 9132565 (9.1 MB)
        RX errors 0  dropped 71  overruns 0  frame 0
        TX packets 5887  bytes 593707 (593.7 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.100.5  netmask 255.255.255.0  broadcast 192.168.100.255
        inet6 fe80::c277:ef48:5333:22be  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:24:ac:0f  txqueuelen 1000  (Ethernet)
        RX packets 15695  bytes 17845470 (17.8 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 11133  bytes 983625 (983.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 1079  bytes 106562 (106.5 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1079  bytes 106562 (106.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

client@client-VirtualBox:~/Downloads$ ./client 52000
usage ./client hostname port
client@client-VirtualBox:~/Downloads$ ./client 192.168.100.4 52000
sockfd = 3
Please enter the message: MULTI CLIENT CODE (THIS IS CLIENT 1!!)
I got your message
client@client-VirtualBox:~/Downloads$
```

PART 5

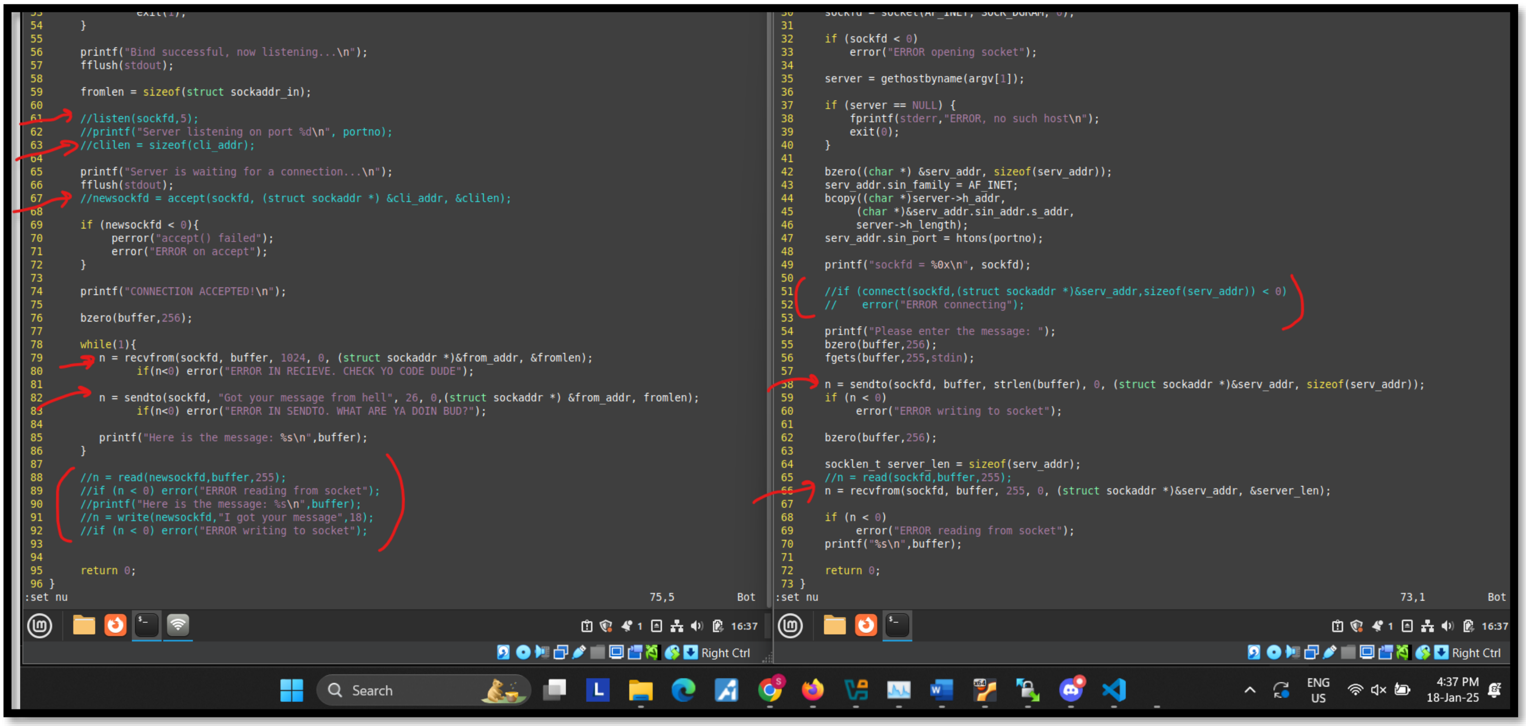
Finally, we needed to run and play around with datagram sockets and UDP connections.

For this we made wholesale changes to the code. After compiling, we ran the code and were FINALLY able to communicate using UDP sockets. See the images below.


Server [Running] - Oracle VirtualBox
File Machine View Input Devices Help
server@server-VirtualBox: ~/Downloads
File Edit View Search Terminal Tabs Help
server@server-VirtualBox: ~/Downloads
server@server-VirtualBox: ~/Downloads
Client [Running] - Oracle VirtualBox
File Machine View Input Devices Help
client@client-VirtualBox: ~/Downloads
File Edit View Search Terminal Help

```
server@server-VirtualBox:~/Downloads$ vim server_UDP.c
server@server-VirtualBox:~/Downloads$ gcc -o server_UDP server_UDP.c
server@server-VirtualBox:~/Downloads$ ./server_UDP 52000
socket created successfully.
Attempting to bind...
Bind successful, now listening...
Server is waiting for a connection...
CONNECTION ACCEPTED!
Here is the message: client1
```

```
client@client-VirtualBox:~/Downloads$ ./client_UDP 192.168.100.4 52000
sockfd = 3
Please enter the message: CLIENT 1
ERROR writing to socket: Invalid argument
client@client-VirtualBox:~/Downloads$ vim client_UDP.c
client@client-VirtualBox:~/Downloads$ gcc -o client_UDP client_UDP.c
client@client-VirtualBox:~/Downloads$ ./client_UDP 192.168.100.4 52000
sockfd = 3
Please enter the message: CLIENT1
Got your message from hell
client@client-VirtualBox:~/Downloads$ ./client_UDP 192.168.100.4 52000
sockfd = 3
Please enter the message: client1
Got your message from hell
client@client-VirtualBox:~/Downloads$
```

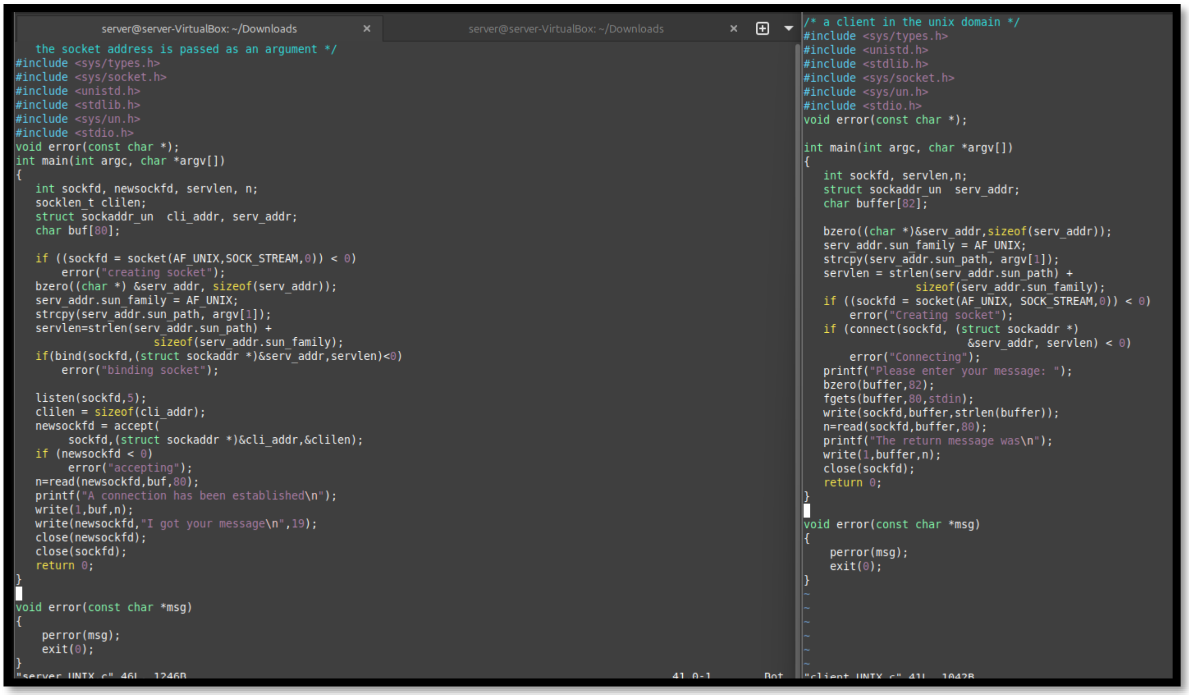


```
            exit(1);
    }

    printf("Bind successful, now listening...\n");
    fflush(stdout);

    fromlen = sizeof(struct sockaddr_in);

    //listen(sockfd,5);
    //printf("Server listening on port %d\n", portno);
    //clilen = sizeof(cli_addr);

    printf("Server is waiting for a connection...\n");
    fflush(stdout);
    //newsockfd = accept(sockfd, (struct sockaddr *) &cli_addr, &clilen);

    if (newsockfd < 0){
        perror("accept() failed");
        error("ERROR on accept");
    }

    printf("CONNECTION ACCEPTED!\n");

    bzero(buffer,256);

    while(1){
      n = recvfrom(sockfd, buffer, 1024, 0, (struct sockaddr *)&from_addr, &fromlen);
            if(n<0) error("ERROR IN RECIEVE. CHECK YO CODE DUDE");

      n = sendto(sockfd, "Got your message from hell", 26, 0,(struct sockaddr *) &from_addr, fromlen);
            if(n<0) error("ERROR IN SENDTO. WHAT ARE YA DOIN BUD?");

      printf("Here is the message: %s\n",buffer);
    }

    //n = read(newsockfd,buffer,255);
    //if (n < 0) error("ERROR reading from socket");
    //printf("Here is the message: %s\n",buffer);
    //n = write(newsockfd,"I got your message",18);
    //if (n < 0) error("ERROR writing to socket");


    return 0;
}
:set nu                                                     75,5          Bot
```

```
    sockfd = socket(AF_INET, SOCK_DGRAM, 0);

    if (sockfd < 0)
        error("ERROR opening socket");

    server = gethostbyname(argv[1]);

    if (server == NULL) {
        fprintf(stderr,"ERROR, no such host\n");
        exit(0);
    }

    bzero((char *) &serv_addr, sizeof(serv_addr));
    serv_addr.sin_family = AF_INET;
    bcopy((char *)server->h_addr,
         (char *)&serv_addr.sin_addr.s_addr,
         server->h_length);
    serv_addr.sin_port = htons(portno);

    printf("sockfd = %0x\n", sockfd);

    //if (connect(sockfd,(struct sockaddr *)&serv_addr,sizeof(serv_addr)) < 0)
    //    error("ERROR connecting");

    printf("Please enter the message: ");
    bzero(buffer,256);
    fgets(buffer,255,stdin);

    n = sendto(sockfd, buffer, strlen(buffer), 0, (struct sockaddr *)&serv_addr, sizeof(serv_addr));
    if (n < 0)
         error("ERROR writing to socket");

    bzero(buffer,256);

    socklen_t server_len = sizeof(serv_addr);
    //n = read(sockfd,buffer,255);
    n = recvfrom(sockfd, buffer, 255, 0, (struct sockaddr *)&serv_addr, &server_len);

    if (n < 0)
         error("ERROR reading from socket");
    printf("%s\n",buffer);

    return 0;
}
:set nu                                                     73,1          Bot
```

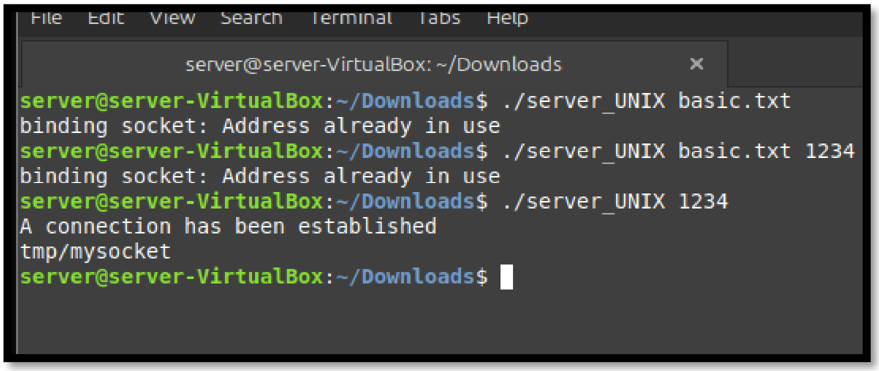
PART 6

I tried working on the UNIX based server-client system, and I opened two different tabs of the same VM. Ran the server code and then the client code in a different tab(SAME VM) and got the socket to accept connection and respond

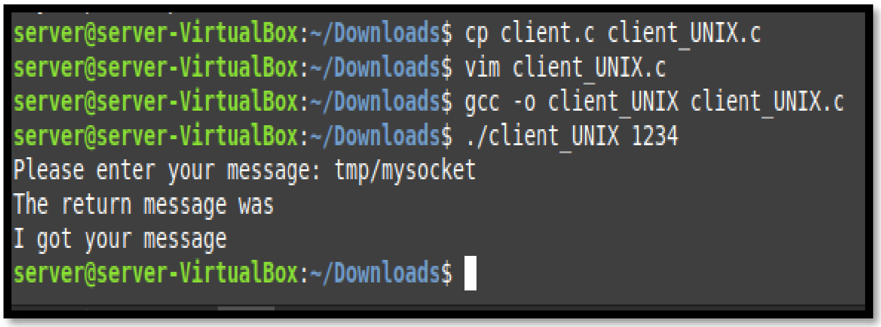

server@server-VirtualBox: ~/Downloads
server@server-VirtualBox: ~/Downloads

```
   the socket address is passed as an argument */
#include <sys/types.h>
#include <sys/socket.h>
#include <unistd.h>
#include <stdlib.h>
#include <sys/un.h>
#include <stdio.h>
void error(const char *);
int main(int argc, char *argv[])
{
   int sockfd, newsockfd, servlen, n;
   socklen_t clilen;
   struct sockaddr_un  cli_addr, serv_addr;
   char buf[80];

   if ((sockfd = socket(AF_UNIX,SOCK_STREAM,0)) < 0)
       error("creating socket");
   bzero((char *) &serv_addr, sizeof(serv_addr));
   serv_addr.sun_family = AF_UNIX;
   strcpy(serv_addr.sun_path, argv[1]);
   servlen=strlen(serv_addr.sun_path) +
                     sizeof(serv_addr.sun_family);
   if(bind(sockfd,(struct sockaddr *)&serv_addr,servlen)<0)
       error("binding socket");

   listen(sockfd,5);
   clilen = sizeof(cli_addr);
   newsockfd = accept(
        sockfd,(struct sockaddr *)&cli_addr,&clilen);
   if (newsockfd < 0)
        error("accepting");
   n=read(newsockfd,buf,80);
   printf("A connection has been established\n");
   write(1,buf,n);
   write(newsockfd,"I got your message\n",19);
   close(newsockfd);
   close(sockfd);
   return 0;
}

void error(const char *msg)
{
    perror(msg);
    exit(0);
}
"server_UNIX.c" 46L, 1246B                                    41,0-1        Bot
```

```
/* a client in the unix domain */
#include <sys/types.h>
#include <unistd.h>
#include <stdlib.h>
#include <sys/socket.h>
#include <sys/un.h>
#include <stdio.h>
void error(const char *);

int main(int argc, char *argv[])
{
   int sockfd, servlen,n;
   struct sockaddr_un  serv_addr;
   char buffer[82];

   bzero((char *)&serv_addr,sizeof(serv_addr));
   serv_addr.sun_family = AF_UNIX;
   strcpy(serv_addr.sun_path, argv[1]);
   servlen = strlen(serv_addr.sun_path) +
                 sizeof(serv_addr.sun_family);
   if ((sockfd = socket(AF_UNIX, SOCK_STREAM,0)) < 0)
       error("Creating socket");
   if (connect(sockfd, (struct sockaddr *)
                         &serv_addr, servlen) < 0)
       error("Connecting");
   printf("Please enter your message: ");
   bzero(buffer,82);
   fgets(buffer,80,stdin);
   write(sockfd,buffer,strlen(buffer));
   n=read(sockfd,buffer,80);
   printf("The return message was\n");
   write(1,buffer,n);
   close(sockfd);
   return 0;
}

void error(const char *msg)
{
    perror(msg);
    exit(0);
}
~
~
~
~
~
~
"client_UNIX.c" 41L, 1042B
```

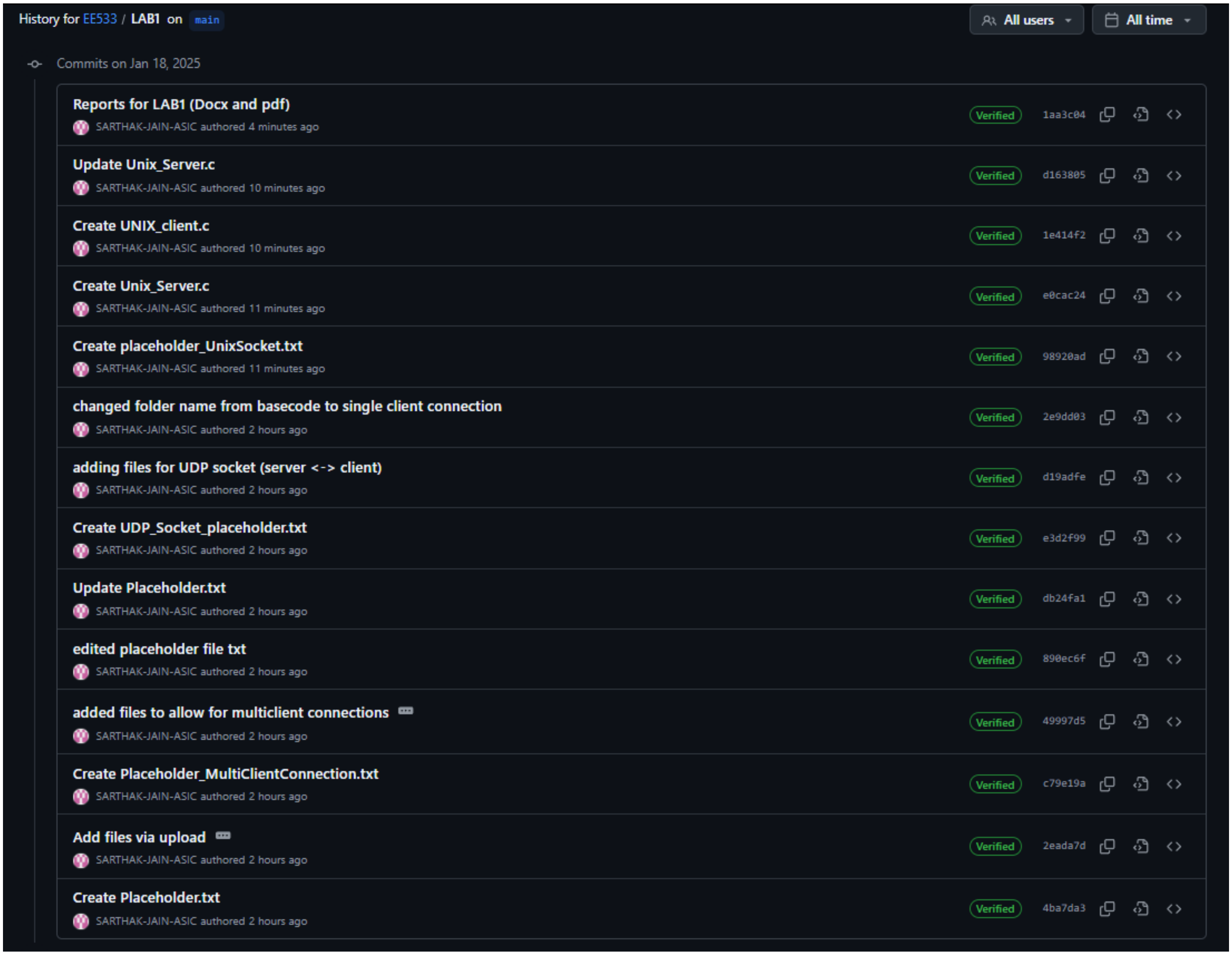


```
server@server-VirtualBox:~/Downloads$ ./server_UNIX basic.txt
binding socket: Address already in use
server@server-VirtualBox:~/Downloads$ ./server_UNIX basic.txt 1234
binding socket: Address already in use
server@server-VirtualBox:~/Downloads$ ./server_UNIX 1234
A connection has been established
tmp/mysocket
server@server-VirtualBox:~/Downloads$ 
```



```
server@server-VirtualBox:~/Downloads$ cp client.c client_UNIX.c
server@server-VirtualBox:~/Downloads$ vim client_UNIX.c
server@server-VirtualBox:~/Downloads$ gcc -o client_UNIX client_UNIX.c
server@server-VirtualBox:~/Downloads$ ./client_UNIX 1234
Please enter your message: tmp/mysocket
The return message was
I got your message
server@server-VirtualBox:~/Downloads$ 
```

GITHUB COMMIT HISTORY -



History for EE533 / LAB1 on main

Commits on Jan 18, 2025

Reports for LAB1 (Docx and pdf)
SARTHAK-JAIN-ASIC authored 4 minutes ago — Verified 1aa3c04

Update Unix_Server.c
SARTHAK-JAIN-ASIC authored 10 minutes ago — Verified d163805

Create UNIX_client.c
SARTHAK-JAIN-ASIC authored 10 minutes ago — Verified 1e414f2

Create Unix_Server.c
SARTHAK-JAIN-ASIC authored 11 minutes ago — Verified e0cac24

Create placeholder_UnixSocket.txt
SARTHAK-JAIN-ASIC authored 11 minutes ago — Verified 98920ad

changed folder name from basecode to single client connection
SARTHAK-JAIN-ASIC authored 2 hours ago — Verified 2e9dd03

adding files for UDP socket (server <-> client)
SARTHAK-JAIN-ASIC authored 2 hours ago — Verified d19adfe

Create UDP_Socket_placeholder.txt
SARTHAK-JAIN-ASIC authored 2 hours ago — Verified e3d2f99

Update Placeholder.txt
SARTHAK-JAIN-ASIC authored 2 hours ago — Verified db24fa1

edited placeholder file txt
SARTHAK-JAIN-ASIC authored 2 hours ago — Verified 890ec6f

added files to allow for multiclient connections …
SARTHAK-JAIN-ASIC authored 2 hours ago — Verified 49997d5

Create Placeholder_MultiClientConnection.txt
SARTHAK-JAIN-ASIC authored 2 hours ago — Verified c79e19a

Add files via upload …
SARTHAK-JAIN-ASIC authored 2 hours ago — Verified 2eada7d

Create Placeholder.txt
SARTHAK-JAIN-ASIC authored 2 hours ago — Verified 4ba7da3