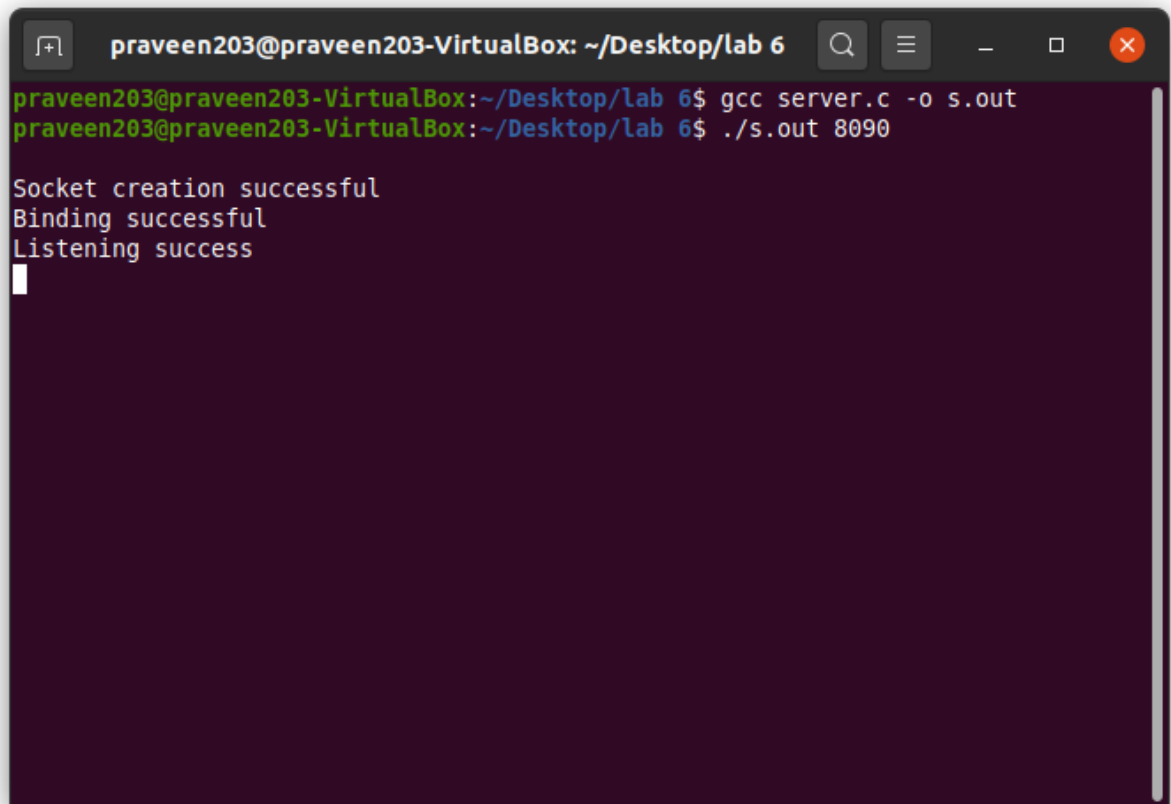


Computer networks lab 6

1. Server creation and waiting for clients:



```
praveen203@praveen203-VirtualBox: ~/Desktop/lab 6
praveen203@praveen203-VirtualBox:~/Desktop/lab 6$ gcc server.c -o s.out
praveen203@praveen203-VirtualBox:~/Desktop/lab 6$ ./s.out 8090

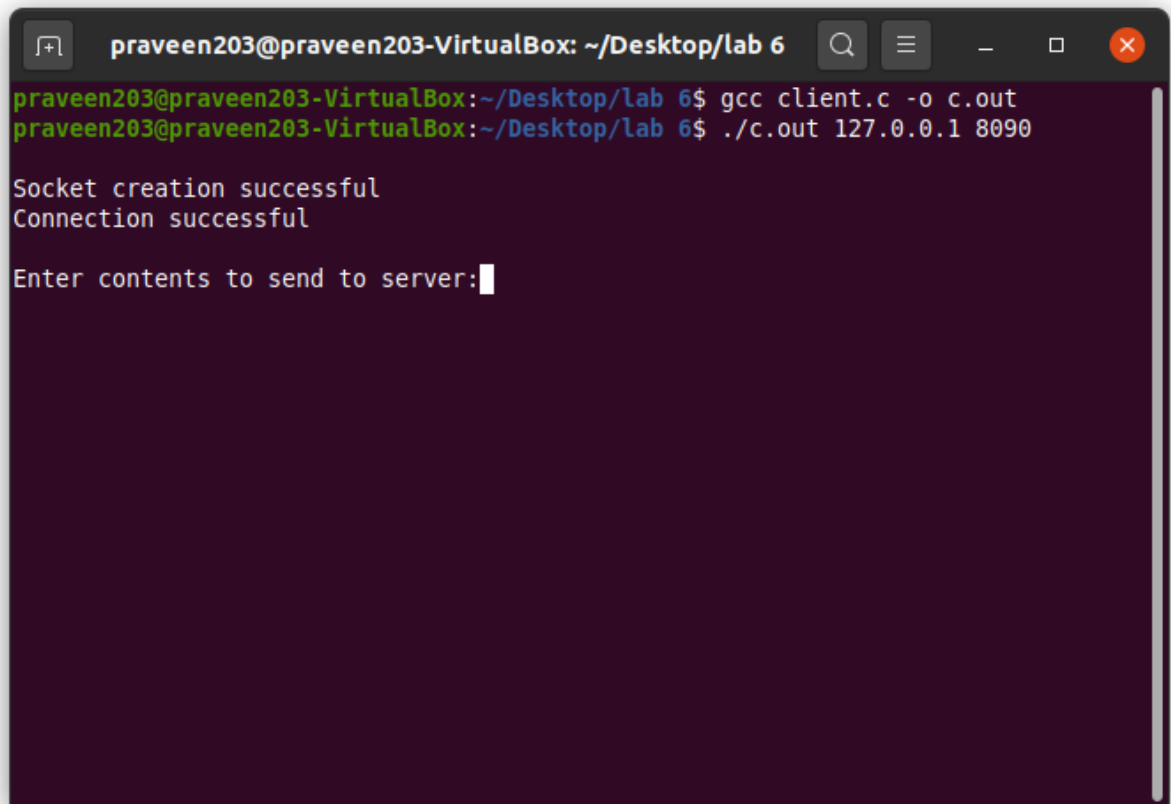
Socket creation successful
Binding successful
Listening success
█
```

The image shows a terminal window titled "praveen203@praveen203-VirtualBox: ~/Desktop/lab 6". The terminal displays the following commands and output:

- `gcc server.c -o s.out`
- `./s.out 8090`
- Output: `Socket creation successful`
- Output: `Binding successful`
- Output: `Listening success`

A cursor is visible on the line following the last output message.

2. Client creation and joining server:



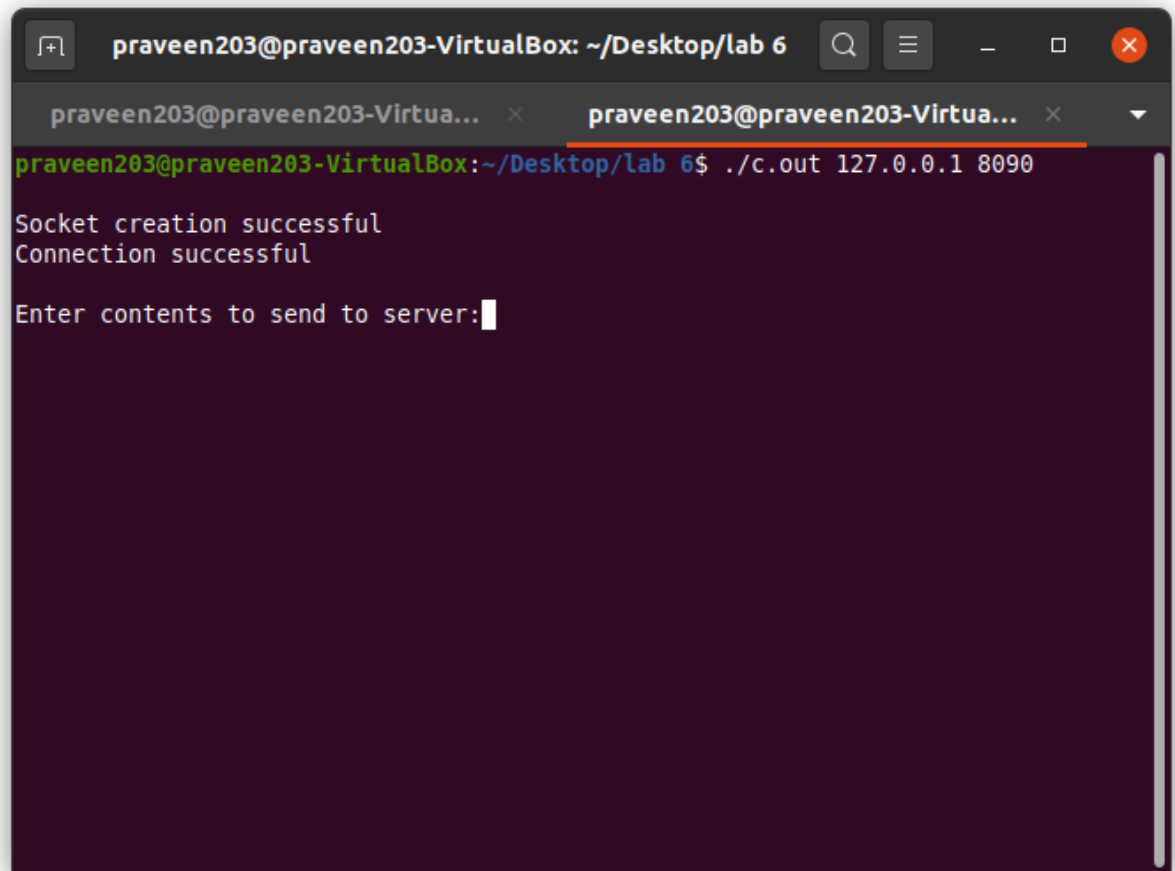
```
praveen203@praveen203-VirtualBox: ~/Desktop/lab 6
praveen203@praveen203-VirtualBox:~/Desktop/lab 6$ gcc client.c -o c.out
praveen203@praveen203-VirtualBox:~/Desktop/lab 6$ ./c.out 127.0.0.1 8090

Socket creation successful
Connection successful

Enter contents to send to server:
```

The image shows a terminal window titled "praveen203@praveen203-VirtualBox: ~/Desktop/lab 6". The user has compiled a C program "client.c" into "c.out" using "gcc". Then, they have executed the program with the command "./c.out 127.0.0.1 8090". The output shows "Socket creation successful" and "Connection successful". The prompt "Enter contents to send to server:" is displayed, indicating the program is waiting for input.

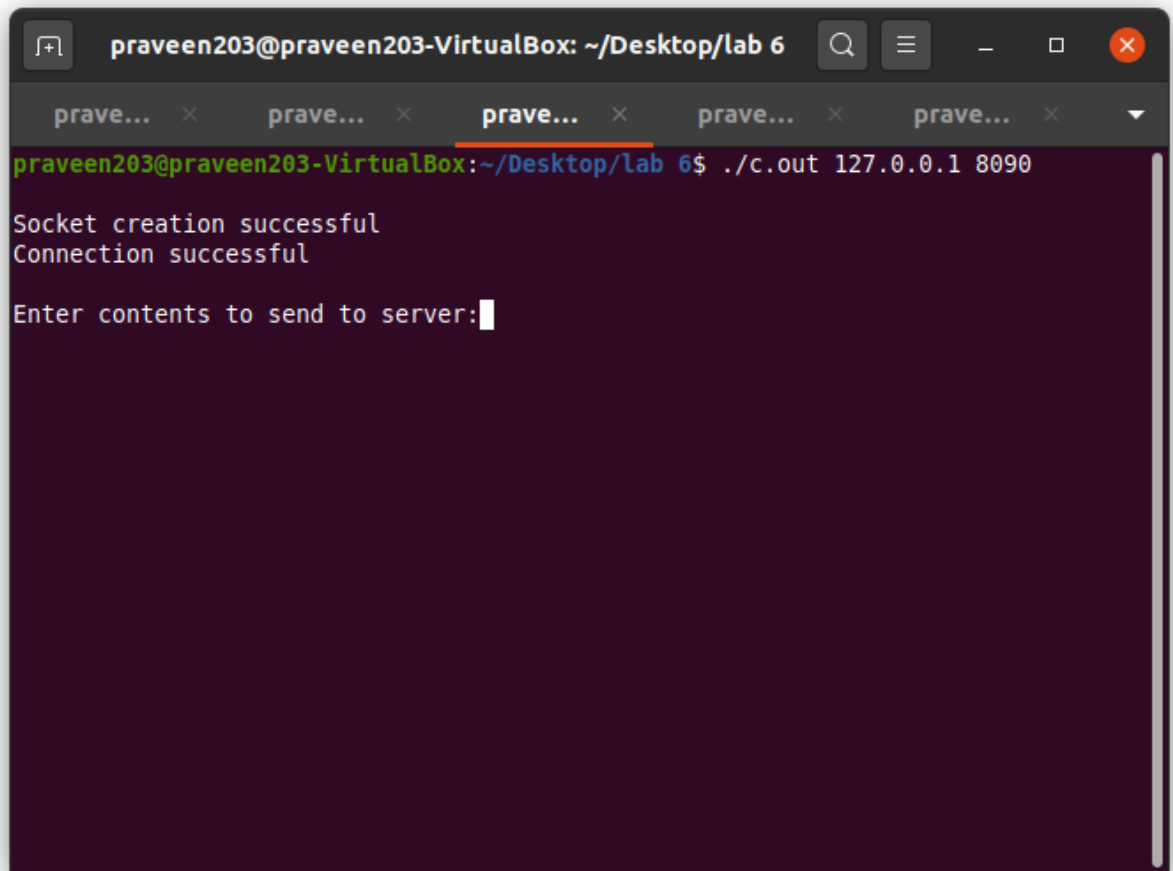
3. Client 2 joining server:



A terminal window titled "praveen203@praveen203-VirtualBox: ~/Desktop/lab 6" with standard window controls. The terminal shows the execution of a program that successfully creates a socket and connects to a server at 127.0.0.1 on port 8090. It then prompts the user to enter content to send to the server.

```
praveen203@praveen203-VirtualBox: ~/Desktop/lab 6$ ./c.out 127.0.0.1 8090
Socket creation successful
Connection successful
Enter contents to send to server: 
```

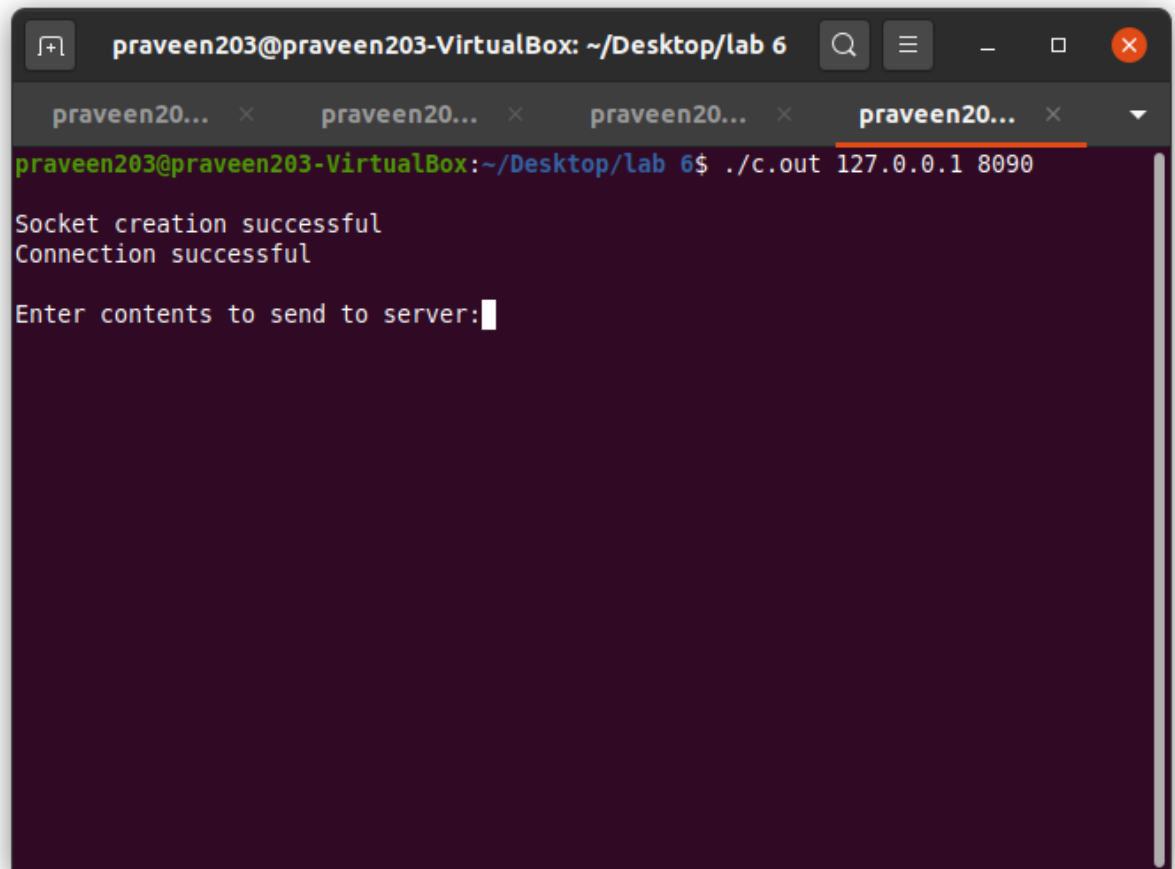
4. Client 3 joining server:



A terminal window titled "praveen203@praveen203-VirtualBox: ~/Desktop/lab 6" with five tabs labeled "prave...". The terminal shows the command `./c.out 127.0.0.1 8090` being executed, followed by the output "Socket creation successful" and "Connection successful". The prompt "Enter contents to send to server:" is displayed with a cursor.

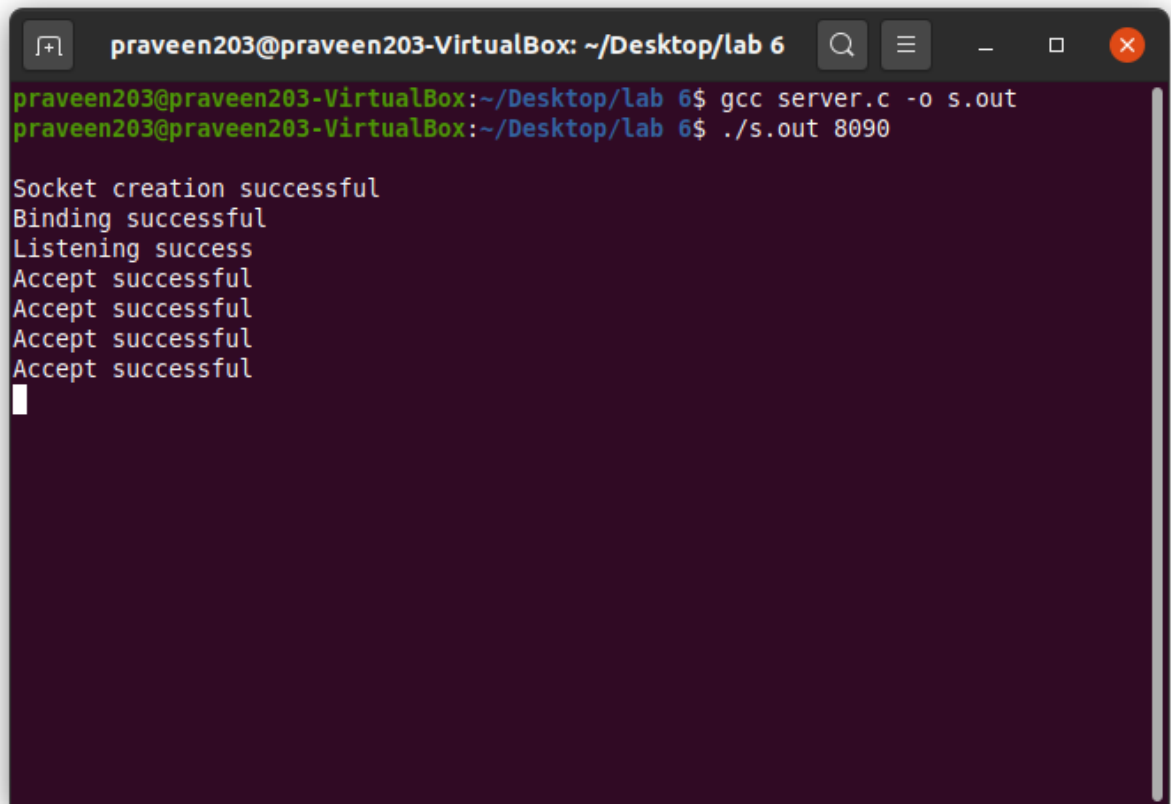
```
praveen203@praveen203-VirtualBox: ~/Desktop/lab 6$ ./c.out 127.0.0.1 8090
Socket creation successful
Connection successful
Enter contents to send to server: 
```

5. Client 4 joining server:

A terminal window titled 'praveen203@praveen203-VirtualBox: ~/Desktop/lab 6' with four tabs. The active tab shows the command './c.out 127.0.0.1 8090' being executed. The output is 'Socket creation successful' and 'Connection successful'. The prompt 'Enter contents to send to server:' is displayed with a cursor.

```
praveen203@praveen203-VirtualBox: ~/Desktop/lab 6$ ./c.out 127.0.0.1 8090
Socket creation successful
Connection successful
Enter contents to send to server: 
```

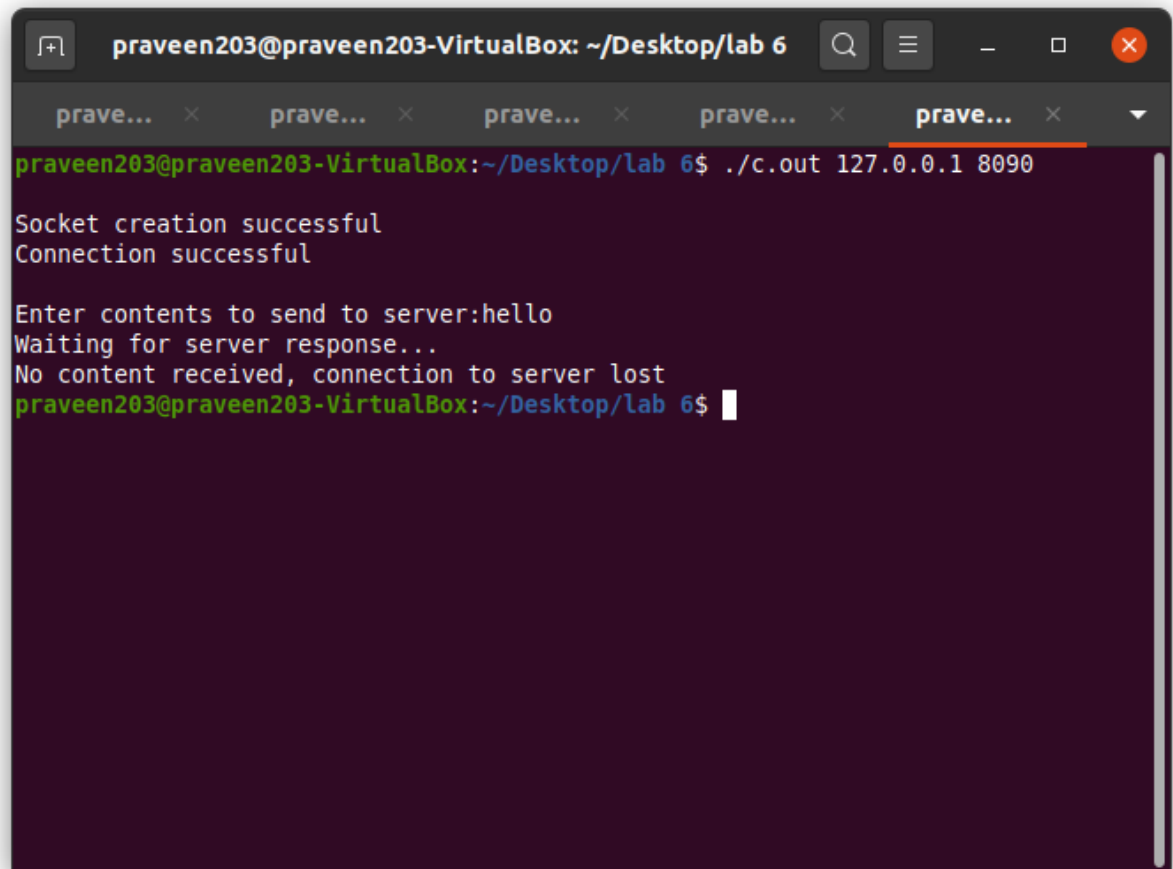
6. Server accepting all clients (because number of clients less than 4):

A terminal window titled "praveen203@praveen203-VirtualBox: ~/Desktop/lab 6" with standard window controls. The terminal shows the compilation of a C program and its execution. The output indicates successful socket creation, binding, and listening, followed by four successful client acceptances.

```
praveen203@praveen203-VirtualBox:~/Desktop/lab 6$ gcc server.c -o s.out
praveen203@praveen203-VirtualBox:~/Desktop/lab 6$ ./s.out 8090

Socket creation successful
Binding successful
Listening success
Accept successful
Accept successful
Accept successful
Accept successful
█
```

7. 5th client trying to join server and getting rejected:



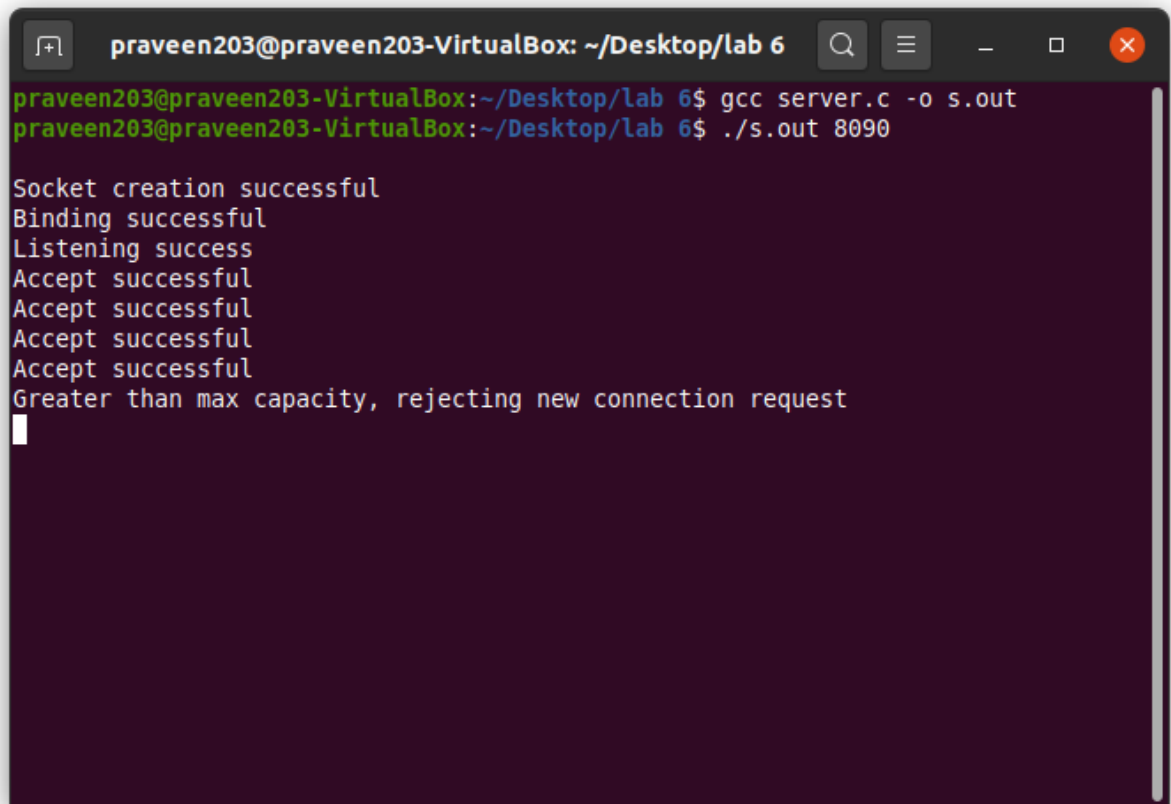
```
praveen203@praveen203-VirtualBox: ~/Desktop/lab 6
prave... x prave... x prave... x prave... x prave... x
praveen203@praveen203-VirtualBox:~/Desktop/lab 6$ ./c.out 127.0.0.1 8090

Socket creation successful
Connection successful

Enter contents to send to server:hello
Waiting for server response...
No content received, connection to server lost
praveen203@praveen203-VirtualBox:~/Desktop/lab 6$
```

The image shows a terminal window titled "praveen203@praveen203-VirtualBox: ~/Desktop/lab 6". The terminal has five tabs, all labeled "prave...". The active tab shows the following output: a command prompt followed by the command `./c.out 127.0.0.1 8090`, which results in "Socket creation successful" and "Connection successful". Then, the user enters "hello" at the prompt "Enter contents to send to server:", followed by "Waiting for server response...". Finally, the message "No content received, connection to server lost" is displayed, and the prompt returns to `praveen203@praveen203-VirtualBox:~/Desktop/lab 6$`.

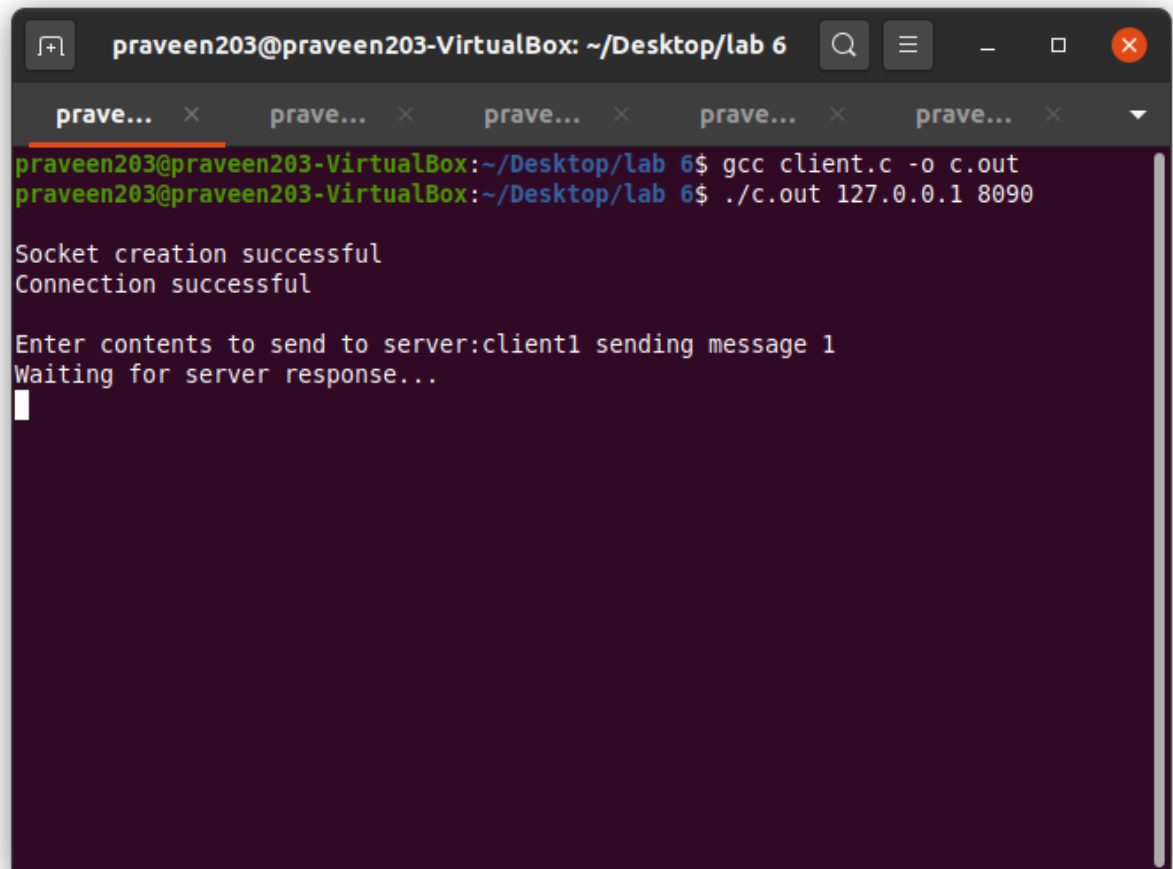
8. Server rejecting the 5th client:

A terminal window titled "praveen203@praveen203-VirtualBox: ~/Desktop/lab 6" with standard window controls. The terminal shows the compilation of a C program and its execution. The output indicates that the server successfully created a socket, bound it, and listened. It then accepted four connections successfully. The fifth connection attempt results in a rejection because the server's capacity has been reached.

```
praveen203@praveen203-VirtualBox:~/Desktop/lab 6$ gcc server.c -o s.out
praveen203@praveen203-VirtualBox:~/Desktop/lab 6$ ./s.out 8090

Socket creation successful
Binding successful
Listening success
Accept successful
Accept successful
Accept successful
Accept successful
Greater than max capacity, rejecting new connection request
█
```


9. Client 1 sending a message to the server:

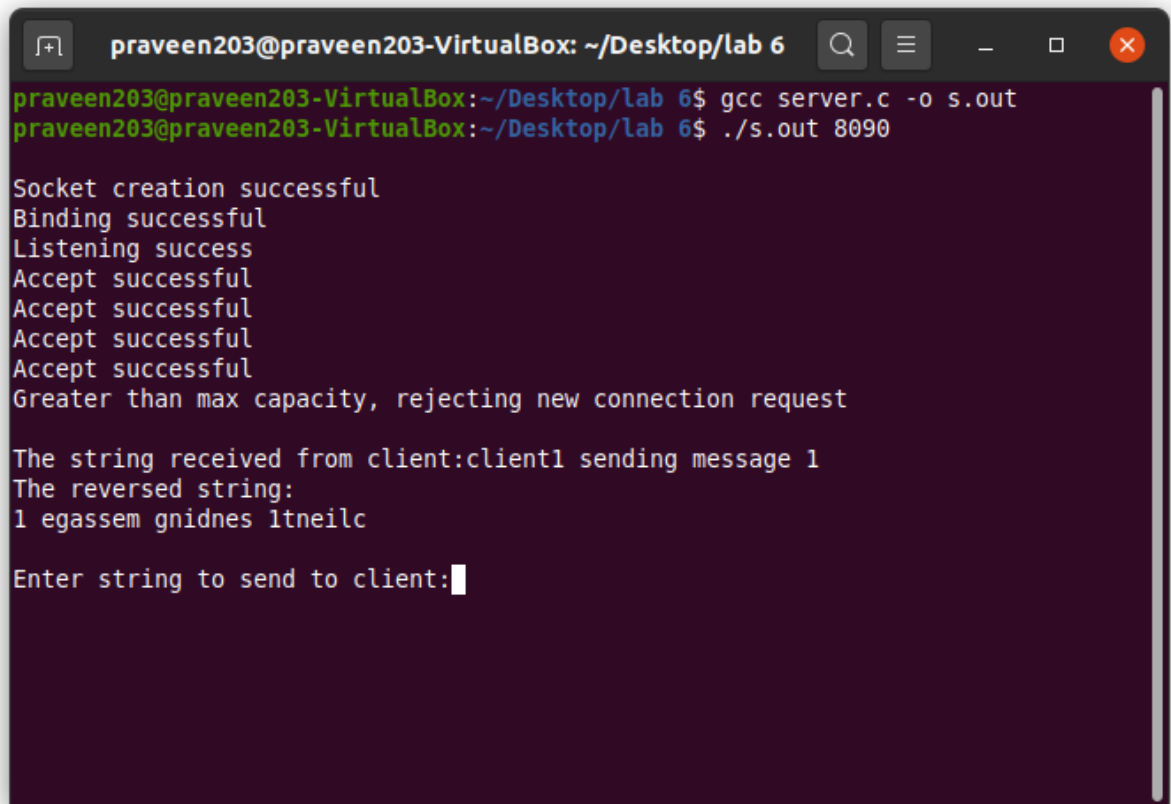
A terminal window titled 'praveen203@praveen203-VirtualBox: ~/Desktop/lab 6'. The window has a dark background with light-colored text. The prompt is 'praveen203@praveen203-VirtualBox:~/Desktop/lab 6\$'. The first command is 'gcc client.c -o c.out' and the second is './c.out 127.0.0.1 8090'. The output shows 'Socket creation successful' and 'Connection successful'. It then prompts 'Enter contents to send to server:' and the user has entered 'client1 sending message 1'. The program is now 'Waiting for server response...'.

```
praveen203@praveen203-VirtualBox:~/Desktop/lab 6$ gcc client.c -o c.out
praveen203@praveen203-VirtualBox:~/Desktop/lab 6$ ./c.out 127.0.0.1 8090

Socket creation successful
Connection successful

Enter contents to send to server:client1 sending message 1
Waiting for server response...
█
```

10. Server receiving the message and printing it reversed:

A terminal window titled 'praveen203@praveen203-VirtualBox: ~/Desktop/lab 6'. The prompt is 'praveen203@praveen203-VirtualBox:~/Desktop/lab 6\$'. The first command is 'gcc server.c -o s.out'. The second command is './s.out 8090'. The output shows the server starting successfully, listening, and accepting connections. It then receives a message from 'client1' and prints the reversed string 'l egassem gnidnes ltneilc'. The prompt 'Enter string to send to client:' is shown at the bottom.

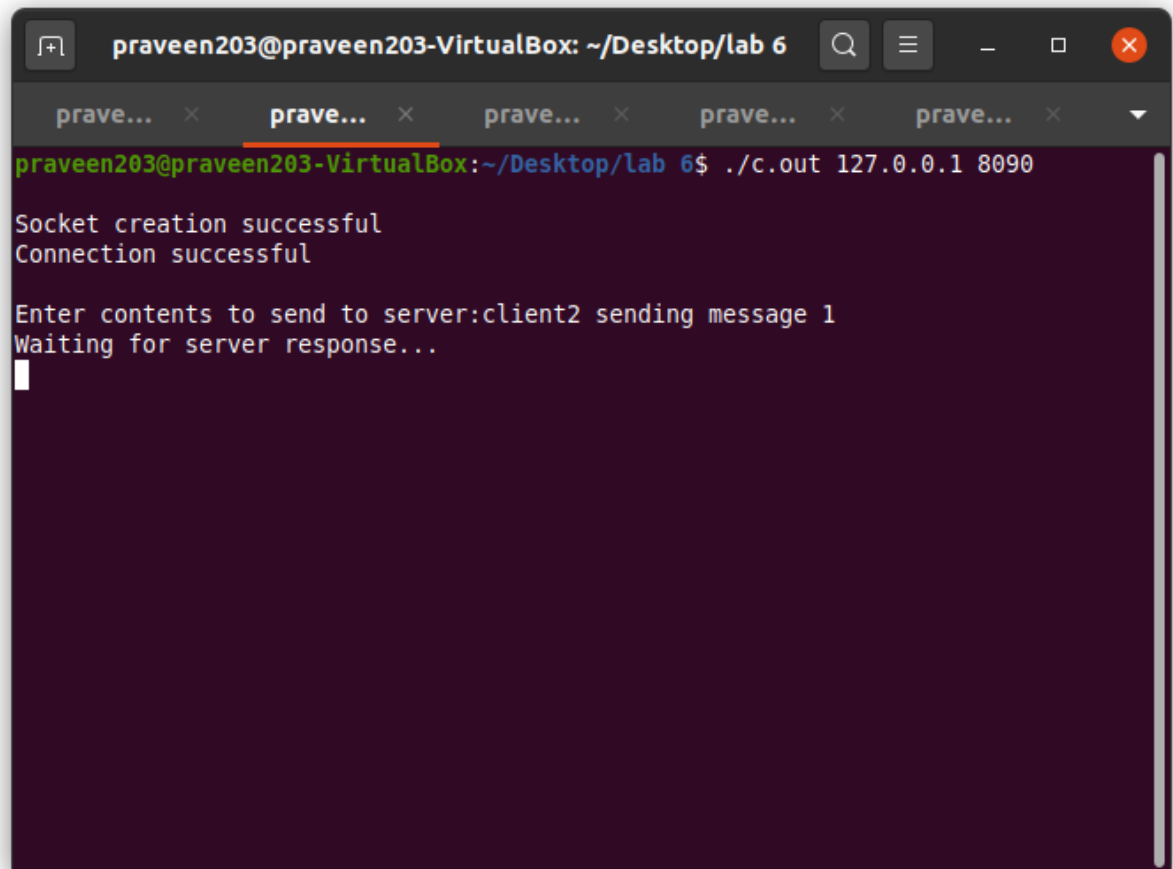
```
praveen203@praveen203-VirtualBox:~/Desktop/lab 6$ gcc server.c -o s.out
praveen203@praveen203-VirtualBox:~/Desktop/lab 6$ ./s.out 8090

Socket creation successful
Binding successful
Listening success
Accept successful
Accept successful
Accept successful
Accept successful
Greater than max capacity, rejecting new connection request

The string received from client:client1 sending message 1
The reversed string:
l egassem gnidnes ltneilc

Enter string to send to client:
```

11. Client 2 sending a message to the server:



A terminal window titled "praveen203@praveen203-VirtualBox: ~/Desktop/lab 6" with five tabs. The terminal output shows the execution of a program that successfully creates a socket and connects to 127.0.0.1:8090. It then prompts for input, and the user enters "client2 sending message 1". The program then displays "Waiting for server response..." and a cursor is visible on the next line.

```
praveen203@praveen203-VirtualBox:~/Desktop/lab 6$ ./c.out 127.0.0.1 8090
Socket creation successful
Connection successful
Enter contents to send to server:client2 sending message 1
Waiting for server response...
█
```

12. Server receiving the message from client 2 and printing it reversed:

```
praveen203@praveen203-VirtualBox: ~/Desktop/lab 6
praveen203@praveen203-VirtualBox:~/Desktop/lab 6$ gcc server.c -o s.out
praveen203@praveen203-VirtualBox:~/Desktop/lab 6$ ./s.out 8090

Socket creation successful
Binding successful
Listening success
Accept successful
Accept successful
Accept successful
Accept successful
Greater than max capacity, rejecting new connection request

The string received from client:client1 sending message 1
The reversed string:
1 egassem gnidnes 1tneilc

Enter string to send to client:
The string received from client:client2 sending message 1
The reversed string:
1 egassem gnidnes 2tneilc

Enter string to send to client:█
```

13. Server sending response to client 1(serves clients in order of receiving message):

```
praveen203@praveen203-VirtualBox: ~/Desktop/lab 6
praveen203@praveen203-VirtualBox:~/Desktop/lab 6$ gcc server.c -o s.out
praveen203@praveen203-VirtualBox:~/Desktop/lab 6$ ./s.out 8090

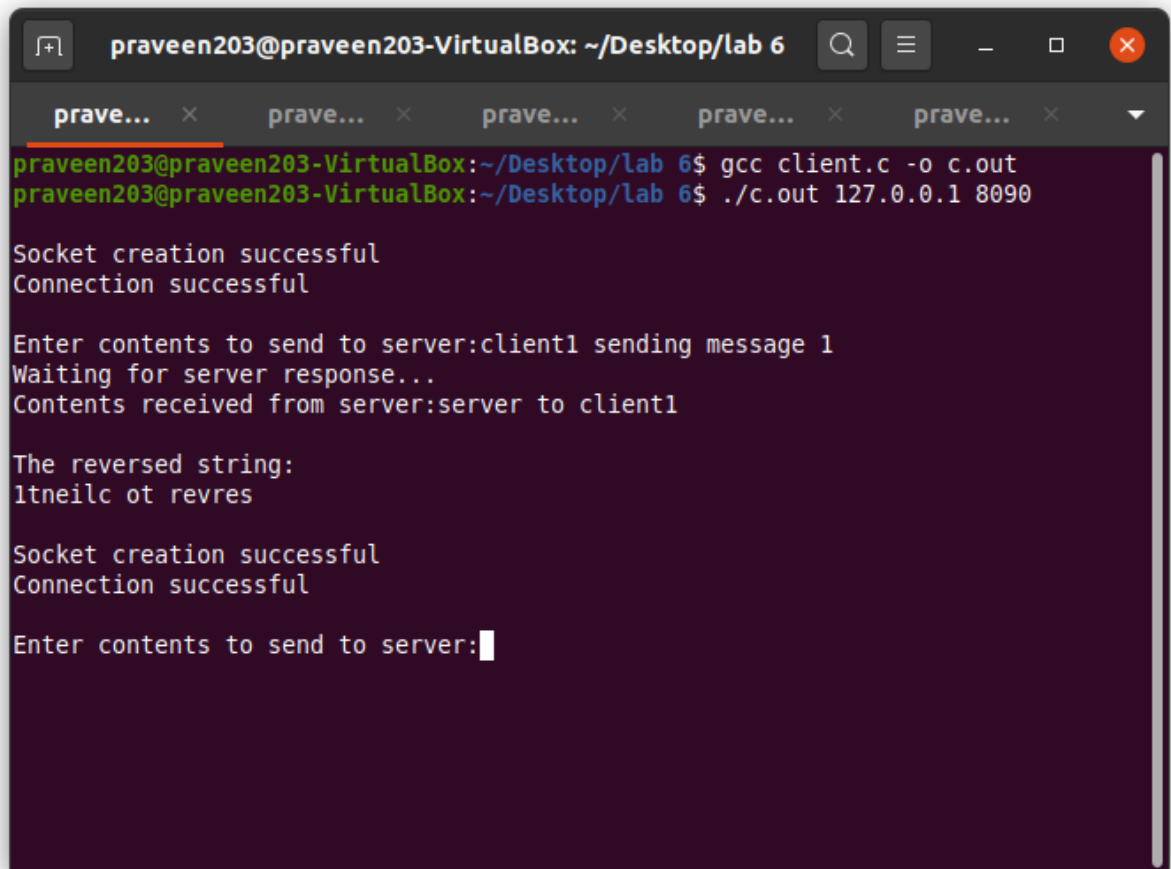
Socket creation successful
Binding successful
Listening success
Accept successful
Accept successful
Accept successful
Accept successful
Greater than max capacity, rejecting new connection request

The string received from client:client1 sending message 1
The reversed string:
1 egassem gnidnes 1tneilc

Enter string to send to client:
The string received from client:client2 sending message 1
The reversed string:
1 egassem gnidnes 2tneilc

Enter string to send to client:server to client1
Accept successful
█
```

14. Client1 receiving the message and ready to send next message:

A terminal window titled 'praveen203@praveen203-VirtualBox: ~/Desktop/lab 6'. The terminal shows the compilation of 'client.c' into 'c.out' and its execution with arguments '127.0.0.1 8090'. The program output indicates successful socket creation and connection, followed by a message exchange where the client sends 'client1 sending message 1' and receives 'server to client1'. It then displays the reversed string '1tneilc ot revres' and prompts for the next message.

```
praveen203@praveen203-VirtualBox:~/Desktop/lab 6$ gcc client.c -o c.out
praveen203@praveen203-VirtualBox:~/Desktop/lab 6$ ./c.out 127.0.0.1 8090

Socket creation successful
Connection successful

Enter contents to send to server:client1 sending message 1
Waiting for server response...
Contents received from server:server to client1

The reversed string:
1tneilc ot revres

Socket creation successful
Connection successful

Enter contents to send to server:
```

15. Server sending response to client 2:

```
praveen203@praveen203-VirtualBox: ~/Desktop/lab 6
praveen203@praveen203-VirtualBox:~/Desktop/lab 6$ gcc server.c -o s.out
praveen203@praveen203-VirtualBox:~/Desktop/lab 6$ ./s.out 8090

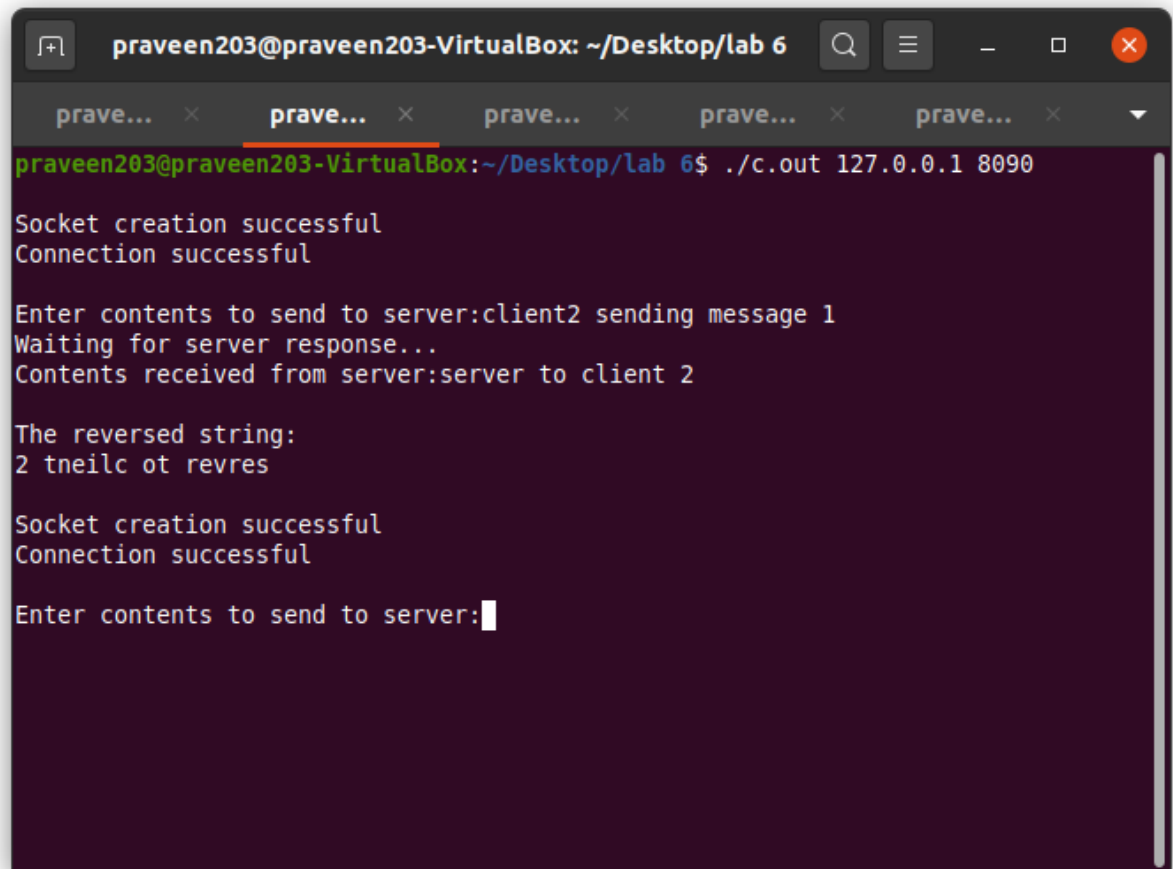
Socket creation successful
Binding successful
Listening success
Accept successful
Accept successful
Accept successful
Accept successful
Greater than max capacity, rejecting new connection request

The string received from client:client1 sending message 1
The reversed string:
1 egassem gnidnes 1tneilc

Enter string to send to client:
The string received from client:client2 sending message 1
The reversed string:
1 egassem gnidnes 2tneilc

Enter string to send to client:server to client1
Accept successful
server to client 2
Accept successful
Client exit
█
```

16. Client2 receiving the message and ready to send the next message:



```
praveen203@praveen203-VirtualBox: ~/Desktop/lab 6
prave... x prave... x prave... x prave... x prave... x
praveen203@praveen203-VirtualBox:~/Desktop/lab 6$ ./c.out 127.0.0.1 8090

Socket creation successful
Connection successful

Enter contents to send to server:client2 sending message 1
Waiting for server response...
Contents received from server:server to client 2

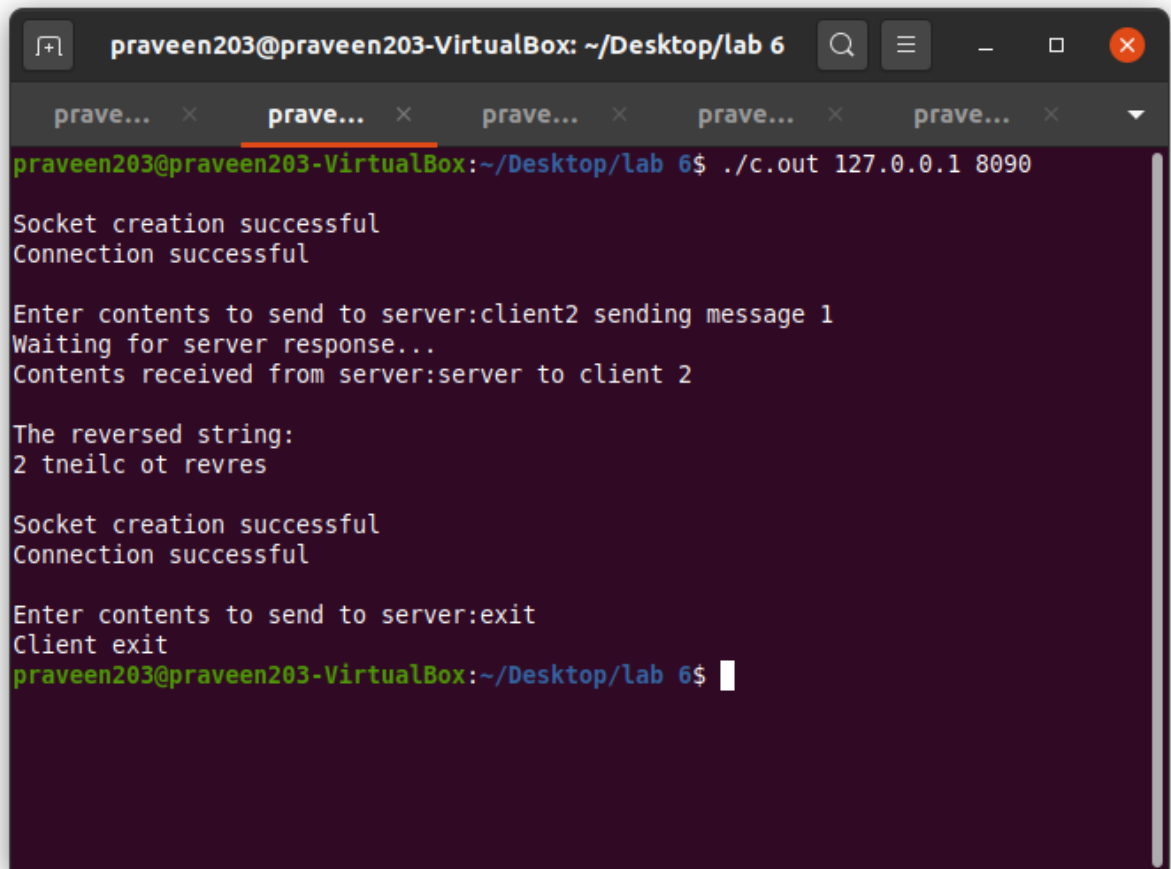
The reversed string:
2 tneilc ot revres

Socket creation successful
Connection successful

Enter contents to send to server:█
```

The image shows a terminal window titled "praveen203@praveen203-VirtualBox: ~/Desktop/lab 6". The terminal displays the execution of a program named "c.out" with arguments "127.0.0.1 8090". The output shows successful socket creation and connection. The user enters "client2 sending message 1", and the program receives "server to client 2" from the server. It then prints the reversed string "2 tneilc ot revres". Finally, it shows another successful socket creation and connection, and prompts the user to "Enter contents to send to server:".

17. Client2 typing exit and exiting:

A terminal window titled 'praveen203@praveen203-VirtualBox: ~/Desktop/lab 6'. The window has several tabs, with the active one highlighted in orange. The terminal output shows the execution of a program that connects to a server at 127.0.0.1:8090. It displays messages for successful socket creation and connection, followed by sending a message and receiving a reversed response. Finally, it shows the client sending 'exit' and then exiting the program.

```
praveen203@praveen203-VirtualBox: ~/Desktop/lab 6$ ./c.out 127.0.0.1 8090
Socket creation successful
Connection successful

Enter contents to send to server:client2 sending message 1
Waiting for server response...
Contents received from server:server to client 2

The reversed string:
2 tneilc ot revres

Socket creation successful
Connection successful

Enter contents to send to server:exit
Client exit
praveen203@praveen203-VirtualBox: ~/Desktop/lab 6$
```

18. Server ready to accept a new connection because client 2 has left:

```
praveen203@praveen203-VirtualBox: ~/Desktop/lab 6
praveen203@praveen203-VirtualBox:~/Desktop/lab 6$ gcc server.c -o s.out
praveen203@praveen203-VirtualBox:~/Desktop/lab 6$ ./s.out 8090

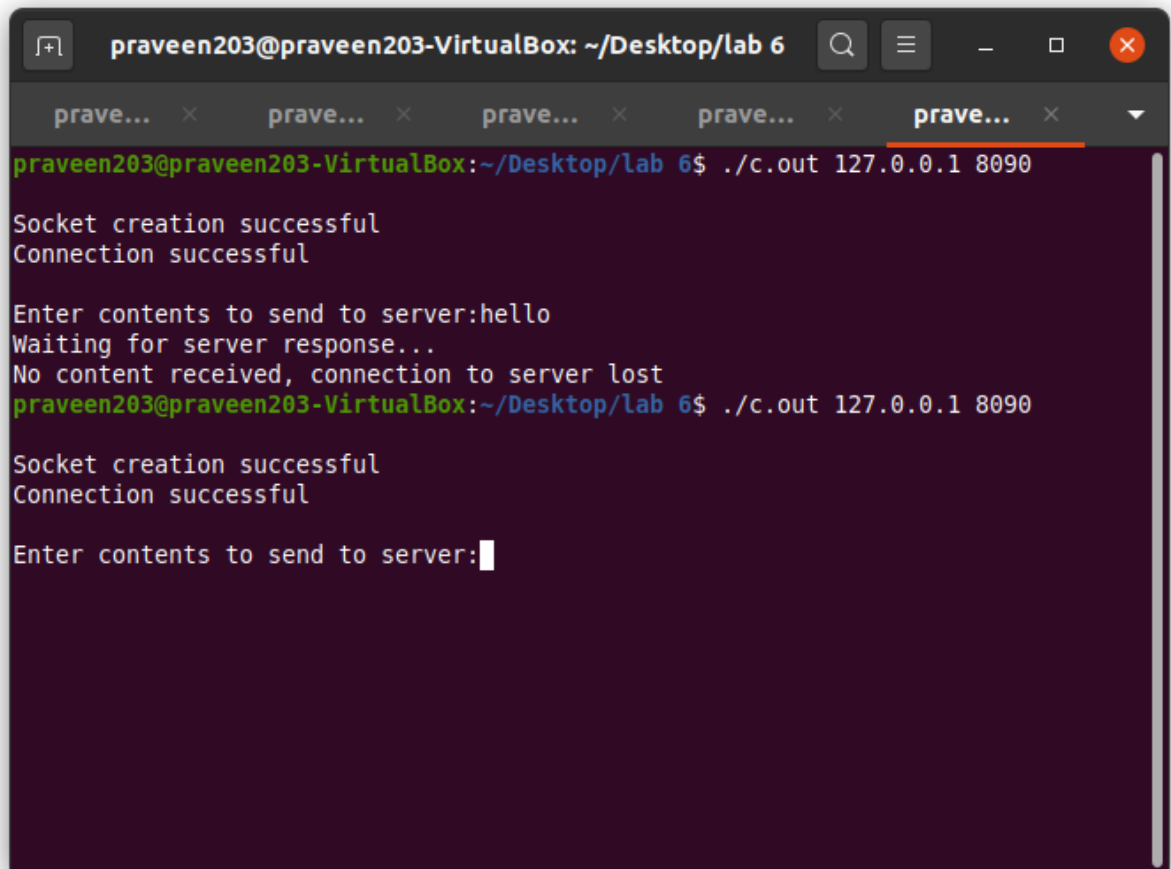
Socket creation successful
Binding successful
Listening success
Accept successful
Accept successful
Accept successful
Accept successful
Greater than max capacity, rejecting new connection request

The string received from client:client1 sending message 1
The reversed string:
1 egassem gnidnes 1tneilc

Enter string to send to client:
The string received from client:client2 sending message 1
The reversed string:
1 egassem gnidnes 2tneilc

Enter string to send to client:server to client1
Accept successful
server to client 2
Accept successful
Client exit
█
```

19. New client 5 joining the server:



A terminal window titled "praveen203@praveen203-VirtualBox: ~/Desktop/lab 6" with standard window controls. The terminal shows a sequence of commands and outputs. The first command is `./c.out 127.0.0.1 8090`, which results in "Socket creation successful" and "Connection successful". The user then enters "hello", followed by "Waiting for server response...". The next output is "No content received, connection to server lost". The user then runs the same command again, `./c.out 127.0.0.1 8090`, which again shows "Socket creation successful" and "Connection successful". The prompt "Enter contents to send to server:" is displayed with a cursor, and the terminal window is currently empty of further input or output.

```
praveen203@praveen203-VirtualBox: ~/Desktop/lab 6$ ./c.out 127.0.0.1 8090
Socket creation successful
Connection successful

Enter contents to send to server:hello
Waiting for server response...
No content received, connection to server lost
praveen203@praveen203-VirtualBox:~/Desktop/lab 6$ ./c.out 127.0.0.1 8090
Socket creation successful
Connection successful

Enter contents to send to server:
```

20. Server accepting the new client:

```
praveen203@praveen203-VirtualBox: ~/Desktop/lab 6
praveen203@praveen203-VirtualBox:~/Desktop/lab 6$ gcc server.c -o s.out
praveen203@praveen203-VirtualBox:~/Desktop/lab 6$ ./s.out 8090

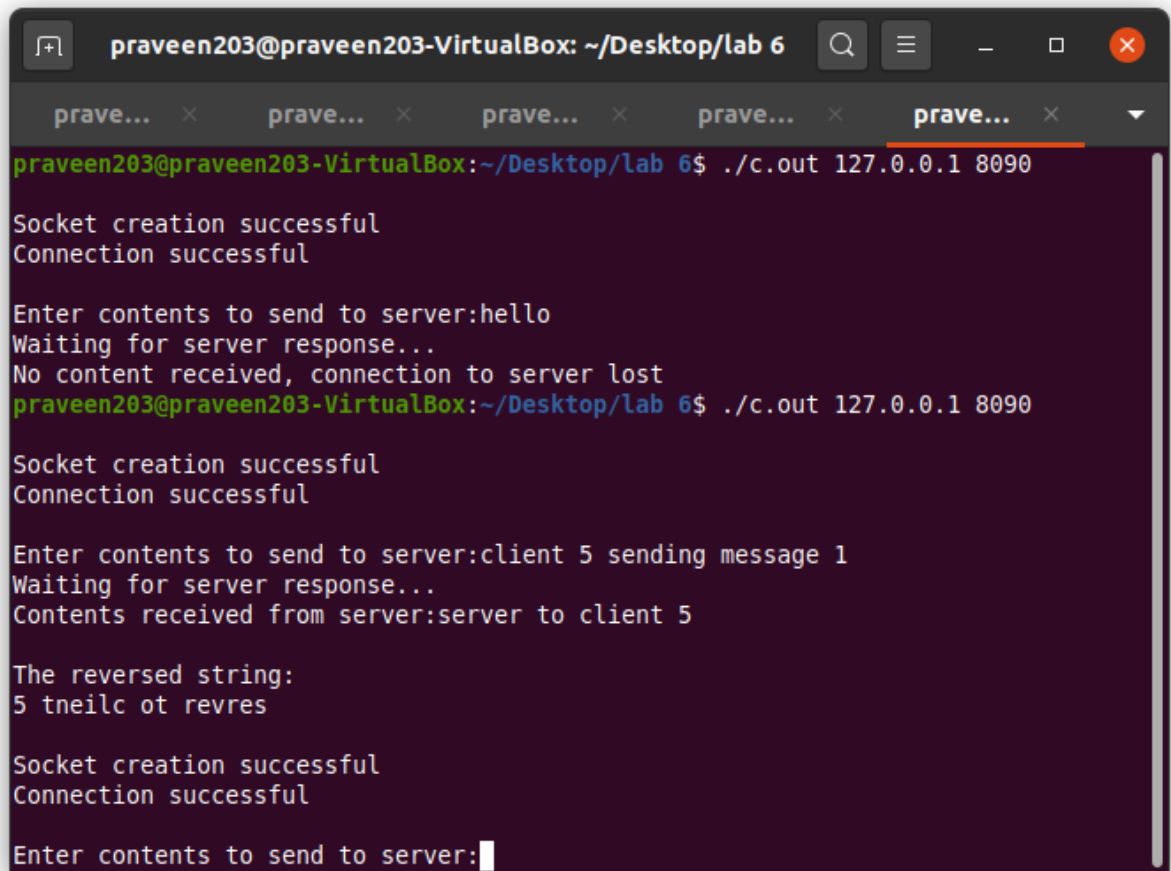
Socket creation successful
Binding successful
Listening success
Accept successful
Accept successful
Accept successful
Accept successful
Greater than max capacity, rejecting new connection request

The string received from client:client1 sending message 1
The reversed string:
1 egassem gnidnes 1tneilc

Enter string to send to client:
The string received from client:client2 sending message 1
The reversed string:
1 egassem gnidnes 2tneilc

Enter string to send to client:server to client1
Accept successful
server to client 2
Accept successful
Client exit
Accept successful
█
```

21. Client 5 sending message and receiving response from the server:



```
praveen203@praveen203-VirtualBox: ~/Desktop/lab 6
prave... x prave... x prave... x prave... x prave... x
praveen203@praveen203-VirtualBox:~/Desktop/lab 6$ ./c.out 127.0.0.1 8090

Socket creation successful
Connection successful

Enter contents to send to server:hello
Waiting for server response...
No content received, connection to server lost
praveen203@praveen203-VirtualBox:~/Desktop/lab 6$ ./c.out 127.0.0.1 8090

Socket creation successful
Connection successful

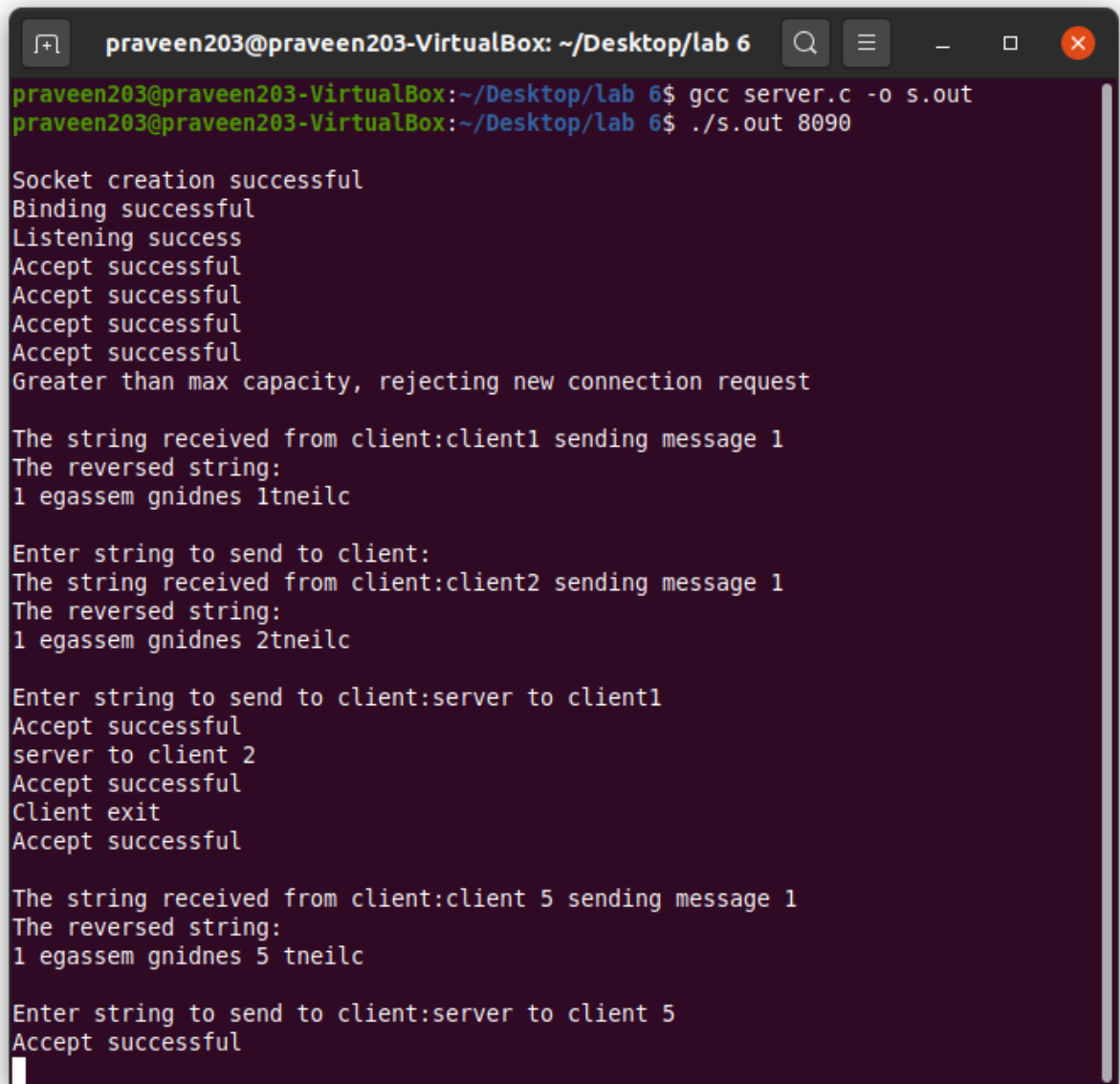
Enter contents to send to server:client 5 sending message 1
Waiting for server response...
Contents received from server:server to client 5

The reversed string:
5 tneilc ot revres

Socket creation successful
Connection successful

Enter contents to send to server:█
```

22. Server receiving message from client 5 and sending response back:

A terminal window titled 'praveen203@praveen203-VirtualBox: ~/Desktop/lab 6'. The prompt is 'praveen203@praveen203-VirtualBox:~/Desktop/lab 6\$'. The user enters 'gcc server.c -o s.out' and then './s.out 8090'. The program output shows successful socket creation, binding, and listening. It then shows four successful 'Accept' operations. The first connection is rejected because it exceeds the max capacity. The second connection (client1) sends 'message 1', which is reversed to '1 egassem gnidnes 1tneilc'. The user enters 'server to client1', which is sent to client2. The third connection (client2) sends 'message 1', which is reversed to '1 egassem gnidnes 2tneilc'. The user enters 'server to client 2', which is sent to client5. The fourth connection (client5) sends 'message 1', which is reversed to '1 egassem gnidnes 5 tneilc'. The user enters 'server to client 5', which is sent to client5. The prompt is now 'praveen203@praveen203-VirtualBox:~/Desktop/lab 6\$' with a cursor on the next line.

```
praveen203@praveen203-VirtualBox: ~/Desktop/lab 6$ gcc server.c -o s.out
praveen203@praveen203-VirtualBox:~/Desktop/lab 6$ ./s.out 8090

Socket creation successful
Binding successful
Listening success
Accept successful
Accept successful
Accept successful
Accept successful
Greater than max capacity, rejecting new connection request

The string received from client:client1 sending message 1
The reversed string:
1 egassem gnidnes 1tneilc

Enter string to send to client:
The string received from client:client2 sending message 1
The reversed string:
1 egassem gnidnes 2tneilc

Enter string to send to client:server to client1
Accept successful
server to client 2
Accept successful
Client exit
Accept successful

The string received from client:client 5 sending message 1
The reversed string:
1 egassem gnidnes 5 tneilc

Enter string to send to client:server to client 5
Accept successful
praveen203@praveen203-VirtualBox:~/Desktop/lab 6$
```

Praveen Sridhar
2018A7PS0166G