

Assignment 3_stream output:

1. ONE-ONE mode:

All clients connect one by one and the server asks client 1 to start sending the message while other clients wait to receive a message.

```
aryan@pop-os: ~/assignment/socket
aryan@pop-os:~/assignment/socket$ gcc -pthread 201801015_server_stream.c
aryan@pop-os:~/assignment/socket$ ./a.out
Socket created...
Binding complete...
Server started listening on port 8080.
Client 1 connected.
Client 2 connected.
Client 3 connected.

aryan@pop-os:~/assignment/socket
aryan@pop-os:~/assignment/socket$ gcc -pthread 201801015_client_stream.c
aryan@pop-os:~/assignment/socket$ ./a.out
Socket created...Connection established...
start
Enter Message: |

aryan@pop-os:~/assignment/socket
aryan@pop-os:~/assignment/socket$ gcc -pthread 201801015_client_stream.c
aryan@pop-os:~/assignment/socket$ ./a.out
Socket created...Connection established...

aryan@pop-os:~/assignment/socket
aryan@pop-os:~/assignment/socket$ gcc -pthread 201801015_client_stream.c
aryan@pop-os:~/assignment/socket$ ./a.out
Socket created...Connection established...
```

Client 1 sends the message to the server and the server acknowledges and forwards the message to client 2 as shown by the arrows. Now client 2 is ready to send a message while other clients wait to receive a message.

```
aryan@pop-os:~/assignment/socket
aryan@pop-os:~/assignment/socket$ gcc -pthread 201801015_server_stream.c
aryan@pop-os:~/assignment/socket$ ./a.out
Socket created...
Binding complete...
Server started listening on port 8080.
Client 1 connected.
Client 2 connected.
Client 3 connected.
recieved from client 1: hello1

aryan@pop-os:~/assignment/socket
aryan@pop-os:~/assignment/socket$ gcc -pthread 201801015_client_stream.c
aryan@pop-os:~/assignment/socket$ ./a.out
Socket created...Connection established...
start
Enter Message: hello1
Sever got your message

aryan@pop-os:~/assignment/socket
aryan@pop-os:~/assignment/socket$ gcc -pthread 201801015_client_stream.c
aryan@pop-os:~/assignment/socket$ ./a.out
Socket created...Connection established...
hello1
Enter Message: |

aryan@pop-os:~/assignment/socket
aryan@pop-os:~/assignment/socket$ gcc -pthread 201801015_client_stream.c
aryan@pop-os:~/assignment/socket$ ./a.out
Socket created...Connection established...
```

Client 2 sends the message to the server which is acknowledged and client 3 gets the same message from the server as shown by the arrows while other clients wait to receive the message.

```
aryan@pop-os:~/assignment/socket$ gcc -pthread 201801015_server_stream.c
aryan@pop-os:~/assignment/socket$ ./a.out
Socket created...
Binding complete...
Server started listening on port 8080.
Client 1 connected.
Client 2 connected.
Client 3 connected.
received from client 1: hello1
received from client 2: hello2

aryan@pop-os:~/assignment/socket$ gcc -pthread 201801015_client_stream.c
aryan@pop-os:~/assignment/socket$ ./a.out
Socket created...Connection established...
start
Enter Message: hello1
Sever got your message

aryan@pop-os:~/assignment/socket$ ./a.out
Socket created...Connection established...
hello2
Enter Message: |

aryan@pop-os:~/assignment/socket$ ./a.out
Socket created...Connection established...
hello1
Enter Message: hello2
Sever got your message
```

Similarly, client 3 sends the message to the server which is acknowledged and received by client 1 completing 1 iteration of messages as shown by the arrows.

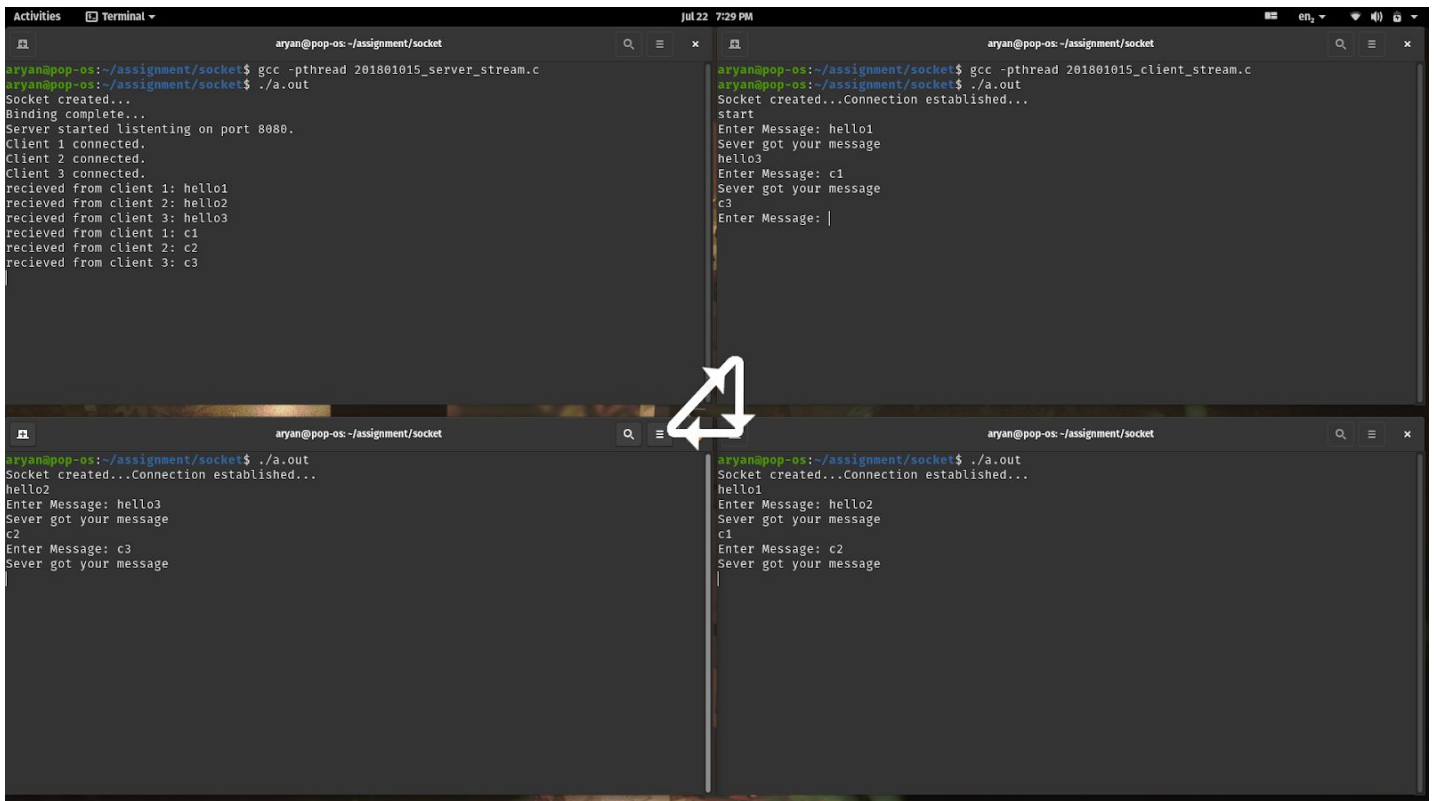
```
aryan@pop-os:~/assignment/socket$ gcc -pthread 201801015_server_stream.c
aryan@pop-os:~/assignment/socket$ ./a.out
Socket created...
Binding complete...
Server started listening on port 8080.
Client 1 connected.
Client 2 connected.
Client 3 connected.
received from client 1: hello1
received from client 2: hello2
received from client 3: hello3

aryan@pop-os:~/assignment/socket$ gcc -pthread 201801015_client_stream.c
aryan@pop-os:~/assignment/socket$ ./a.out
Socket created...Connection established...
start
Enter Message: hello1
Sever got your message
hello3
Enter Message: |

aryan@pop-os:~/assignment/socket$ ./a.out
Socket created...Connection established...
hello2
Enter Message: hello3
Sever got your message

aryan@pop-os:~/assignment/socket$ ./a.out
Socket created...Connection established...
hello1
Enter Message: hello2
Sever got your message
```

The next snap is of complete iteration 2 of the same procedure as indicated by the arrows. We start with client 1 again and follow the same cycle as before.



```
aryan@pop-os: ~/assignment/socket
aryan@pop-os:~/assignment/socket$ gcc -pthread 201801015_server_stream.c
aryan@pop-os:~/assignment/socket$ ./a.out
Socket created...
Binding complete...
Server started listening on port 8080.
Client 1 connected.
Client 2 connected.
Client 3 connected.
recieved from client 1: hello1
recieved from client 2: hello2
recieved from client 3: hello3
recieved from client 1: c1
recieved from client 2: c2
recieved from client 3: c3

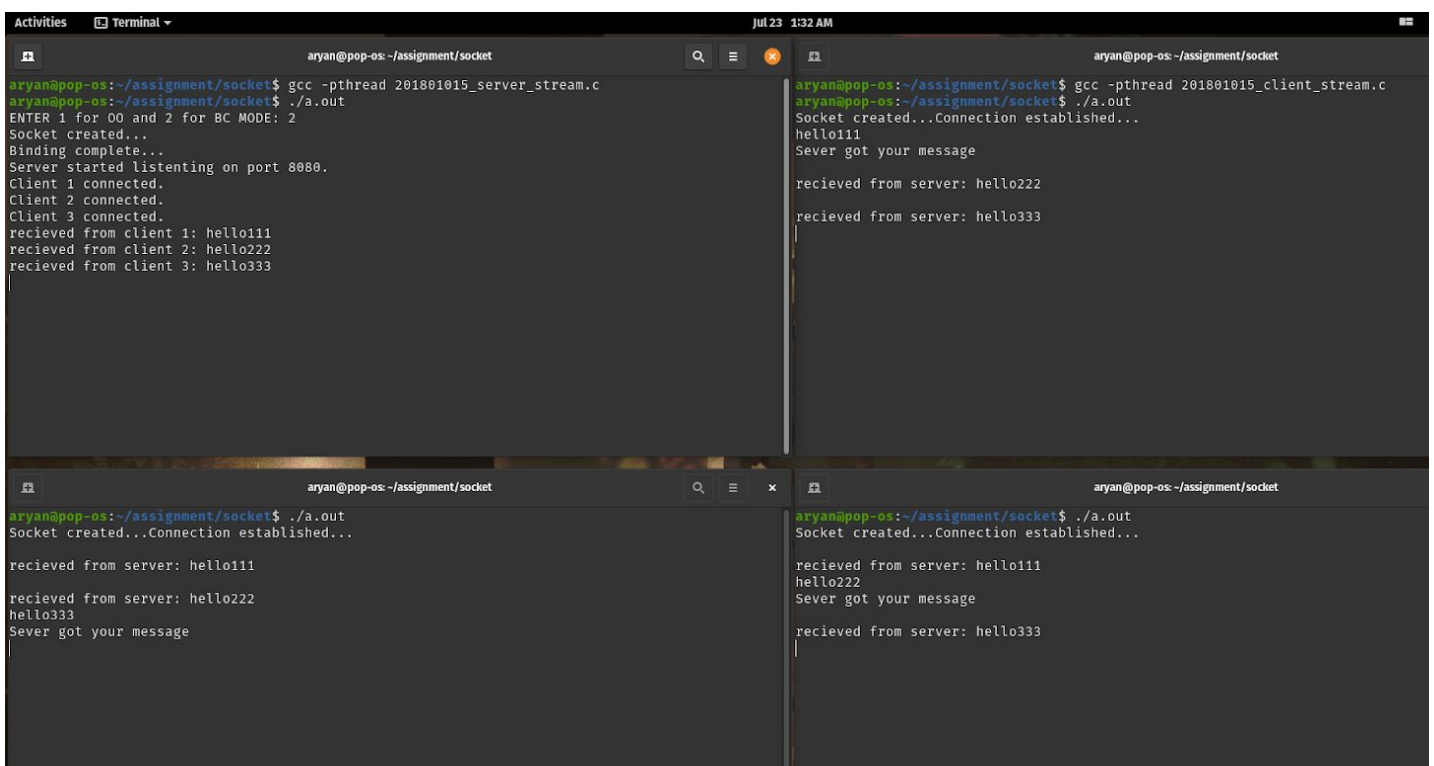
aryan@pop-os:~/assignment/socket$ gcc -pthread 201801015_client_stream.c
aryan@pop-os:~/assignment/socket$ ./a.out
Socket created...Connection established...
start
Enter Message: hello1
Sever got your message
hello3
Enter Message: c1
Sever got your message
c3
Enter Message: |

aryan@pop-os:~/assignment/socket$ ./a.out
Socket created...Connection established...
hello2
Enter Message: hello3
Sever got your message
c2
Enter Message: c3
Sever got your message

aryan@pop-os:~/assignment/socket$ ./a.out
Socket created...Connection established...
hello1
Enter Message: hello2
Sever got your message
c1
Enter Message: c2
Sever got your message
```

2. BROADCAST MODE:

Here the message sent by one client is received by the server and then sent to the remaining clients following the same cycle as previous screenshots. Client 1(top right) sends hello1 which is acknowledged by the server and forwarded to the remaining clients as shown. And the cyclic procedure repeats when client 2(bottom right) sends hello2 and then client 3(bottom left) sends hello 3 as shown in the snap.



```
aryan@pop-os:~/assignment/socket
aryan@pop-os:~/assignment/socket$ gcc -pthread 201801015_server_stream.c
aryan@pop-os:~/assignment/socket$ ./a.out
ENTER 1 for OO and 2 for BC MODE: 2
Socket created...
Binding complete...
Server started listening on port 8080.
Client 1 connected.
Client 2 connected.
Client 3 connected.
recieved from client 1: hello111
recieved from client 2: hello222
recieved from client 3: hello333

aryan@pop-os:~/assignment/socket$ gcc -pthread 201801015_client_stream.c
aryan@pop-os:~/assignment/socket$ ./a.out
Socket created...Connection established...
hello111
Sever got your message
recieved from server: hello222
recieved from server: hello333

aryan@pop-os:~/assignment/socket$ ./a.out
Socket created...Connection established...
recieved from server: hello111
recieved from server: hello222
hello333
Sever got your message
recieved from server: hello333

aryan@pop-os:~/assignment/socket$ ./a.out
Socket created...Connection established...
recieved from server: hello111
hello222
Sever got your message
recieved from server: hello333
```