INDIAN INSTITUTE OF TECHNOLOGY, KHARAGPUR

CS39006: NETWORKS LABORATORY

ASSIGNMENT 8 REPORT

GROUP NO: 31

Compilation:

- First get the executable ‘p2p’.
  Command: $> make
- Open ‘n’ terminals, where n stands for number of users.
  Command: $> ./p2p <port>
  Here, enter the port number of users.

Implementation(For each user):

- A static user_info table is made available to all users in the form of an input file.
- Format for input file:

  n
  <user_1> <ip_1> <port_1>
  .
  .
  <user_n> <ip_n> <port_n>

  where ‘n’ stands for number of users.
- A typedef ‘peer’ is defined which stores name, IPv4 address and a port for each user and a variable prev_conntime for timeout. A global variable **peer *p** is defined.
- First, the user_info from the input file is read and stored in **p;**
- The port number given as command line argument is verfied, whether the corresponding user info is present in user_info table or not.
- If present, then a server socket named ‘sock_s’ is created by setting the paramters accordingly, which listens for incoming connection requests.
- A fd collection ‘rd’ is initiated and the bits corresponding to STDIN and ‘sock_s’ are set.
- Then, A timeout value is defined which is to be used by select().
- Now, a while loop runs until user explicitly enters an exit message*.
- In the loop, first, select is called to check if any fds are ready. If the return value is zero(timeout occured), then check whether the user wants to exit. A value is <0 indicates an error in selecting and a positive value indicates that some fds are ready.
- If some fds are ready,

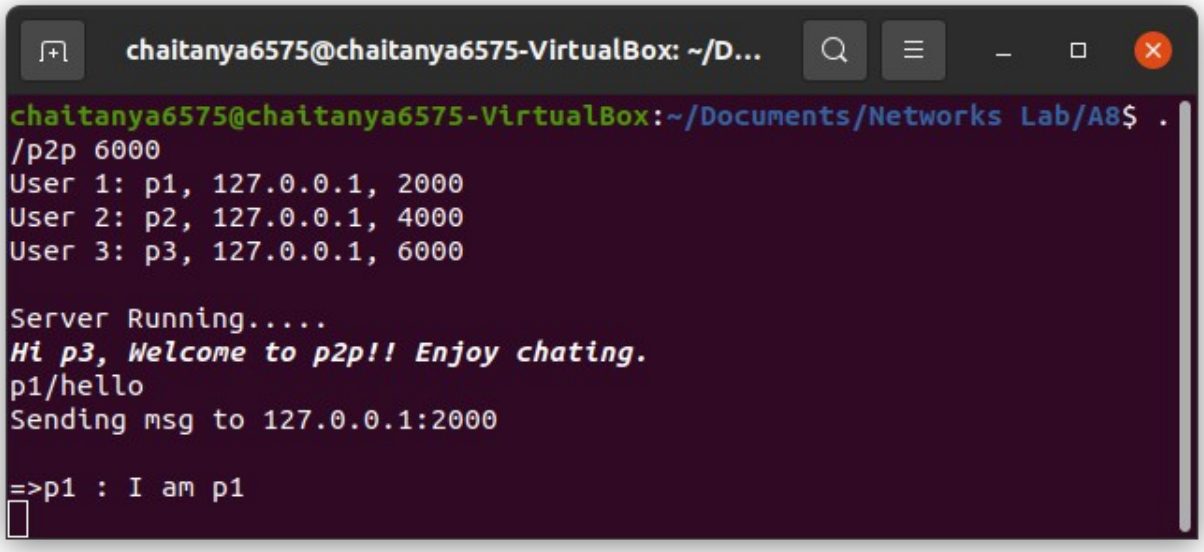
i) check whether the bit for sock_s is set in the fdset 'temp' i.e, check if the user has some connection to accept. If set, then accept the connection
ii) Also, if bit corresponding to STDIN is set in the fdset, i.e, the user is ready to send a msg to some other user.

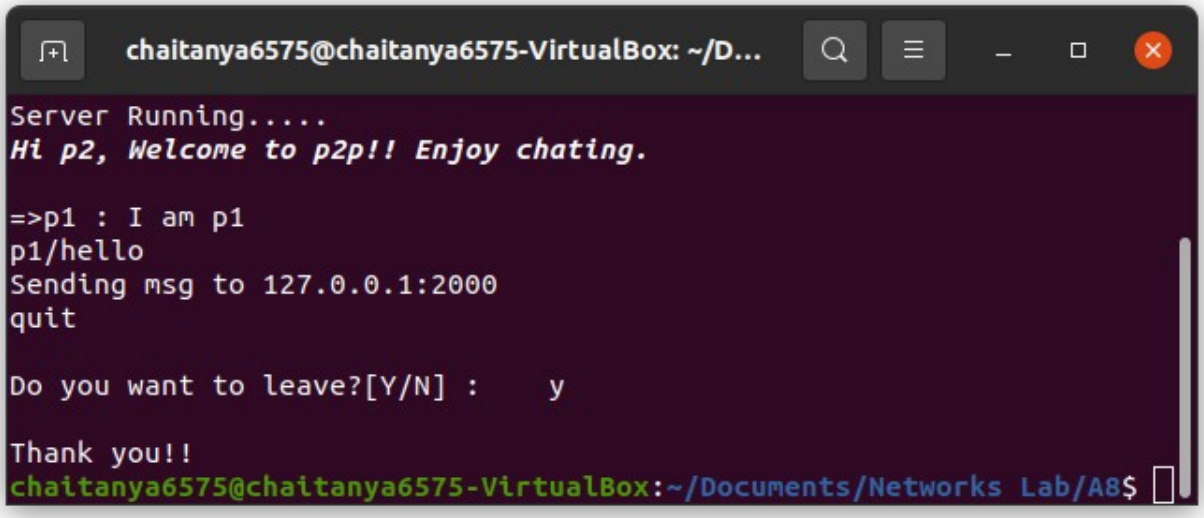
- After reading the msg to be sent by the user to peer(if any) in the form of <receiver>/<port>, split the input string into receiver and port which can be used to identify the information of corresponding receiver.
- For this purpose, a typedef ‘msg’ is defined which contains name, txt corresponding to each message.
- Now, the port number and ip of the receiver is found out and then checks whether client socket exists or not. If not present, create a socket.
- send the message to the receiver and the format would be <sender>/<port>.
- Now, receive the messages from other users, from all sockets available and print them.
- Timeout is implemented using pthread library, When a user sends a msg and receiver doesnt get reply in TIMEOUT time, then the user closes connection.

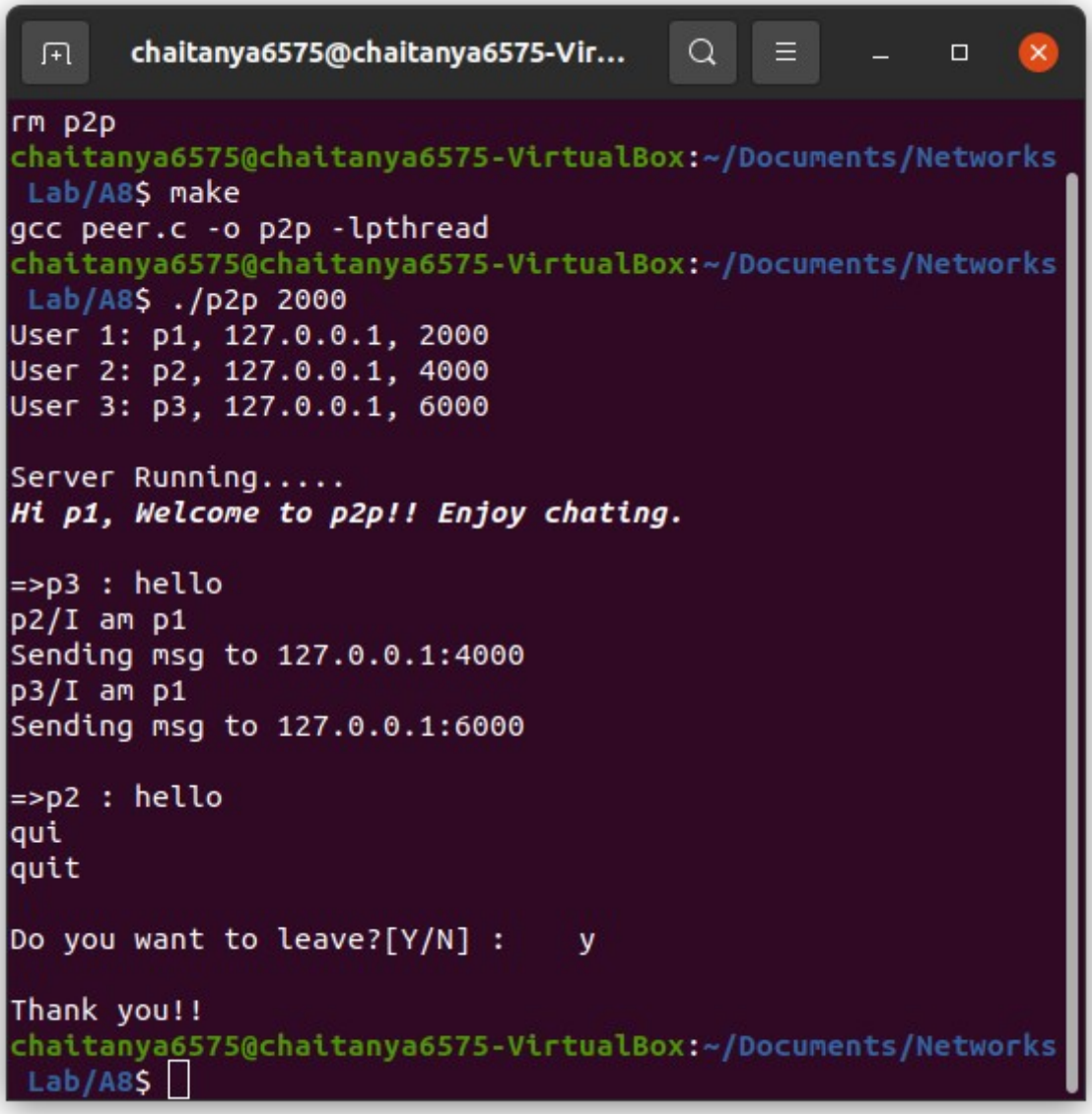
Sample Input:

3
p1 127.0.0.1 2000
p2 127.0.0.1 4000
p3 127.0.0.1 6000

Sample Output:






chaitanya6575@chaitanya6575-Vir...
rm p2p
chaitanya6575@chaitanya6575-VirtualBox:~/Documents/Networks Lab/A8$ make
gcc peer.c -o p2p -lpthread
chaitanya6575@chaitanya6575-VirtualBox:~/Documents/Networks Lab/A8$ ./p2p 2000
User 1: p1, 127.0.0.1, 2000
User 2: p2, 127.0.0.1, 4000
User 3: p3, 127.0.0.1, 6000
Server Running.....
Hi p1, Welcome to p2p!! Enjoy chating.
=>p3 : hello
p2/I am p1
Sending msg to 127.0.0.1:4000
p3/I am p1
Sending msg to 127.0.0.1:6000
=>p2 : hello
qui
quit
Do you want to leave?[Y/N] : y
Thank you!!
chaitanya6575@chaitanya6575-VirtualBox:~/Documents/Networks Lab/A8$