Network Assignment 8

Name: Avijit Mandal (18CS30010)
Name: Sumit Kumar Yadav (18CS30042)

Data Structures:

```
int serverfd;
int ID;
char user[MAX_USER][MAX_SIZE];
struct sockaddr_in user_addresses[MAX_USER];
int client_fd[MAX_USER];
long long last_communication[MAX_USER];
struct timeval tv;
fd_set fd_read;
```

→ `serverfd` saves current server file descriptor
→ `ID` stores information about the server where program is running
→ `user` stores the name of the socket which is later used to make trace out user ID and from `ID` all other details retrieved
→ `user_addresses` stores the port number information for each user
→ `client_fd` stores information about socket fd where to send message
→ `last_communication` is array that stores the last communicated time, which later used for TIMEOUT checking
→ `tv` is used for making the select call free after a certain time interval, so that the programme can check for timeout even when nothing is changed in read_fd
→ `fd_set` stores the current file descriptor

Algorithms:

>> populating the table with pre loaded user information
>> creating the socket
>> first all the client_fd and last_communication is made -1
>> inserting serverfd and STDIN in fd_set
>> inserting all client_fd[i] for i = 0 to 5 in fd_set if there are not -1 { some connection established and not yet TIMEOUT happened }

```
>> if(client_fd[i] != -1)
                    absolute_time = time(NULL)-last_communication[i];
```

Here is absolute time is calculated and and if it is greater than TIMEOUT then it give a timeout message and again make cliend_fd[i] = -1

>> for those when client_fd[i] is not -1 max_fd is calculated in max variable so using in select call
>> take input and tokenize it in client_name and input_message
>> k = findUserByName(client_name, user); // extracts the ID for client using findUserByName function

```
>>          if(k >= MAX_USER)
             {
                  error((char *)"ERROR! User Not Found");
                  goto DONE;
             }
```

Here if user not found then it send to DONE, here DONE starts just before retrieving the message
>> then if client_fd[k] != -1 i.e connection not established, it creates a new connection and sends ID from where connect call is made

```
            char id[2];
            id[0] = ID+'0';
            id[1] = '\0';
            send(client_fd[k], id, strlen(id)+1, 0);
```

>> updating the last_communication[k] = time(NULL) after establishment of connection
>> sends the input_message
>> After the server checks for every place where connection is established and then retrieves if any message is coming, if any non empty message is coming from any connected peer it printing it and updating the last_communication[i] = time(NULL)

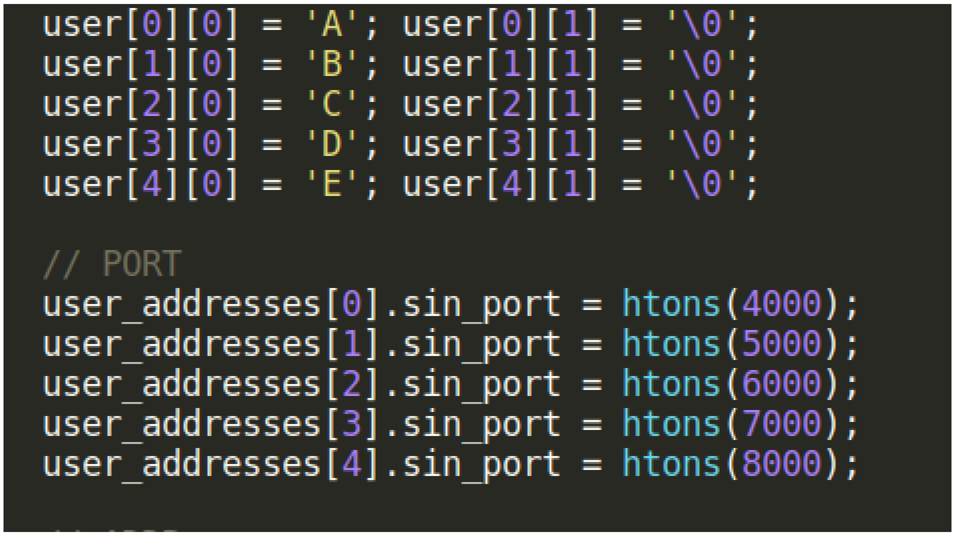
Compilation and Running

```
>> unzip Assignment_8_18CS30010_18CS30042.zip
>> cd Assignment_8_18CS30010_18CS30042
>> make
```

Open terminal and execute

```
./a.out <portno> 127.0.0.1
```

portno can be looked up in user data structure



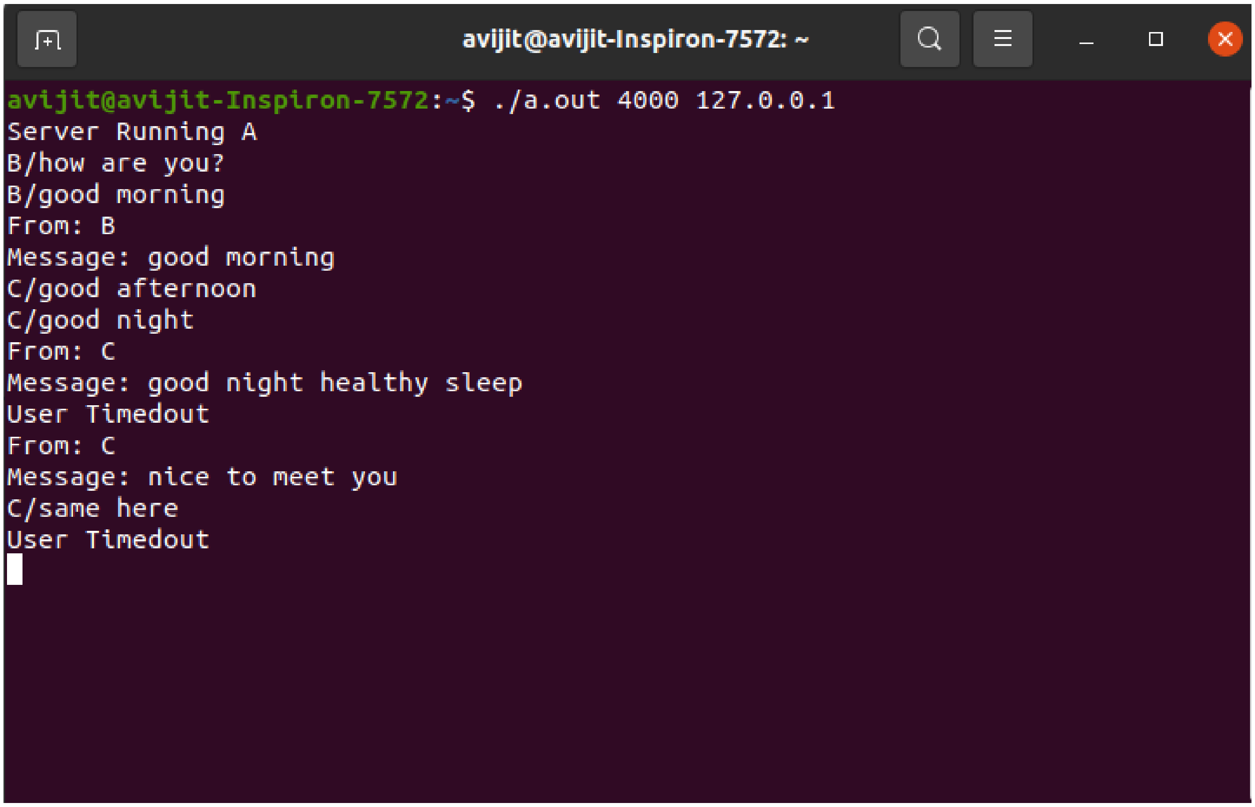
```
user[0][0] = 'A'; user[0][1] = '\0';
user[1][0] = 'B'; user[1][1] = '\0';
user[2][0] = 'C'; user[2][1] = '\0';
user[3][0] = 'D'; user[3][1] = '\0';
user[4][0] = 'E'; user[4][1] = '\0';

// PORT
user_addresses[0].sin_port = htons(4000);
user_addresses[1].sin_port = htons(5000);
user_addresses[2].sin_port = htons(6000);
user_addresses[3].sin_port = htons(7000);
user_addresses[4].sin_port = htons(8000);
```

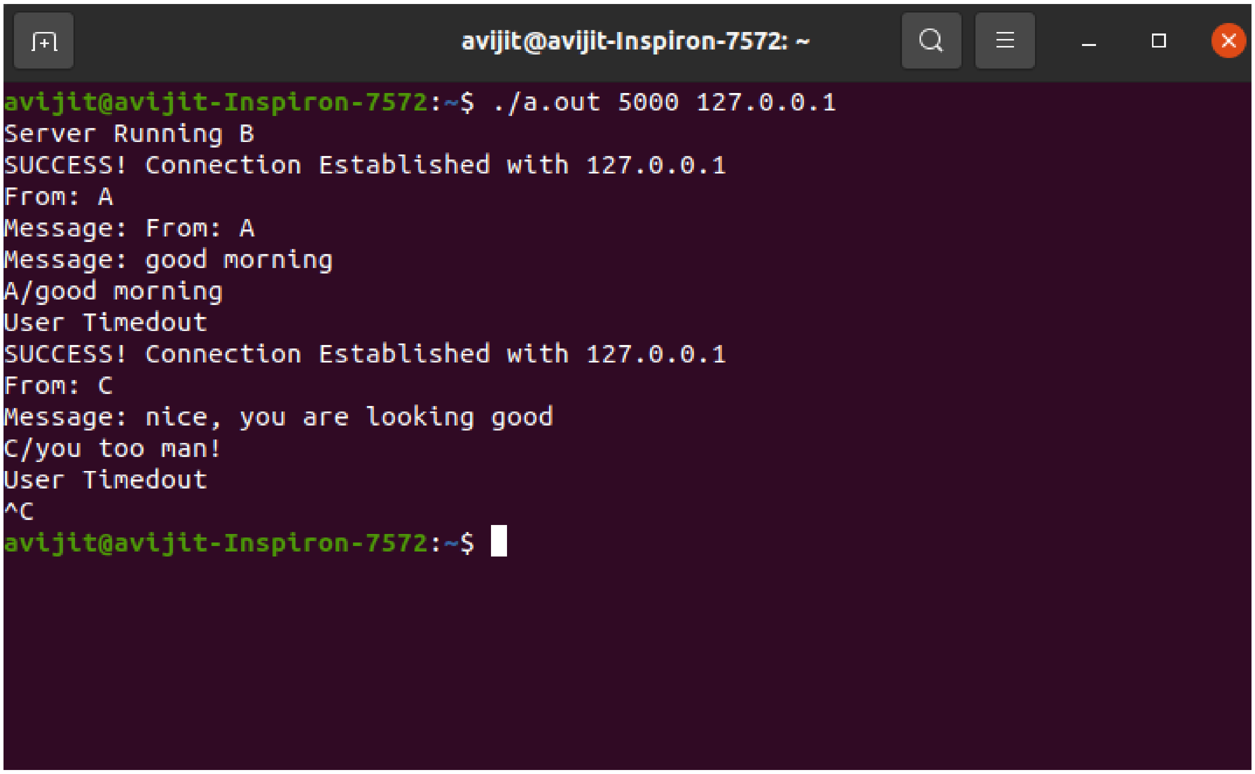
ex: ./a.out 5000 127.0.0.1

Glitch:
In our code the first message after making the connection is not coming

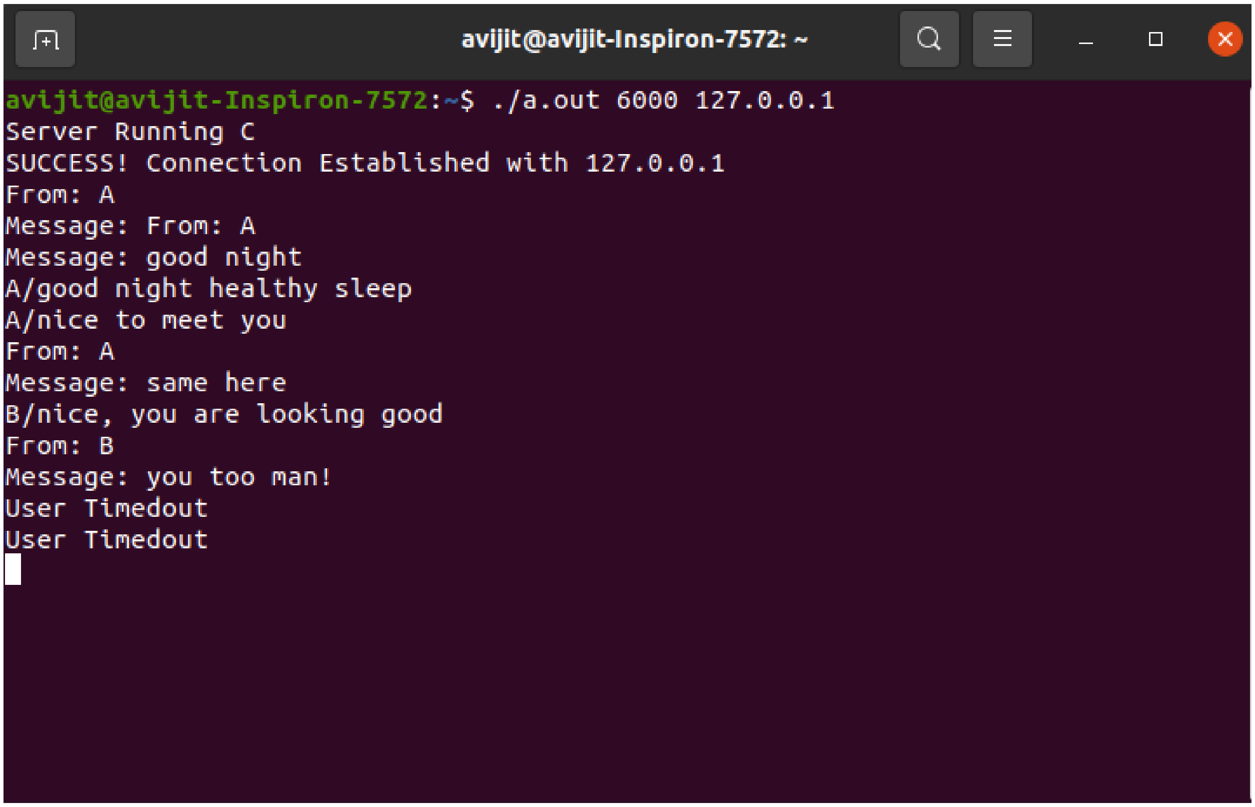
Sample Output



```
avijit@avijit-Inspiron-7572:~$ ./a.out 4000 127.0.0.1
Server Running A
B/how are you?
B/good morning
From: B
Message: good morning
C/good afternoon
C/good night
From: C
Message: good night healthy sleep
User Timedout
From: C
Message: nice to meet you
C/same here
User Timedout
```



```
avijit@avijit-Inspiron-7572:~$ ./a.out 5000 127.0.0.1
Server Running B
SUCCESS! Connection Established with 127.0.0.1
From: A
Message: From: A
Message: good morning
A/good morning
User Timedout
SUCCESS! Connection Established with 127.0.0.1
From: C
Message: nice, you are looking good
C/you too man!
User Timedout
^C
avijit@avijit-Inspiron-7572:~$ 
```



```
avijit@avijit-Inspiron-7572:~$ ./a.out 6000 127.0.0.1
Server Running C
SUCCESS! Connection Established with 127.0.0.1
From: A
Message: From: A
Message: good night
A/good night healthy sleep
A/nice to meet you
From: A
Message: same here
B/nice, you are looking good
From: B
Message: you too man!
User Timedout
User Timedout
```