Ex No: 3 Pranit Tandon

Date: 02/08/2021 RA1911028010014 (CSE – H2) SIMPLE TCP/IP CLIENT SERVER COMMUNICATION

# GIVEN REQUIREMENTS:

There are two hosts, Client and Server. The Client accepts the message from the user and sends it to the Server. The Server receives the message and prints it.

# TECHNICAL OBJECTIVE:

To implement a simple TCP Client-Server application , where the Client on establishing a connection with the Server, sends a string to the Server. The Server reads the String and prints it.

# METHODOLOGY:

**Server:**

* Include the necessary header files.
* Create a socket using socket function with family AF\_INET, type as SOCK\_STREAM.
* Initialize server address to 0 using the bzero function.
* Assign the sin\_family to AF\_INET, sin\_addr to INADDR\_ANY, sin\_port to a dynamically assigned port number.
* Bind the local host address to socket using the bind function.
* Listen on the socket for connection request from the client.
* Accept connection request from the client using accept function.
* Within an infinite loop, using the recv function receive message from the client and print it on the console.

# Client:

* Include the necessary header files.
* Create a socket using socket function with family AF\_INET, type as SOCK\_STREAM.
* Initialize server address to 0 using the bzero function.
* Assign the sin\_family to AF\_INET.
* Get the server IP address and port number from the console.
* Using gethostbyname function assign it to a hostent structure, and assign it to sin\_addr of the server address structure.
* Request a connection from the server using the connect function.
* Within an infinite loop, read message from the console and send the message to the server using the send function.

# CODING:

**Server: tcpserver.c** #include<sys/types.h> #include<sys/socket.h> #include<netinet/in.h> #include<netdb.h> #include<arpa/inet.h> #include<string.h>

int main(int asrgc,char\*argv[])

{

int bd,sd,ad; char buff[1024];

struct sockaddr\_in cliaddr,servaddr; socklen\_t clilen; clilen=sizeof(cliaddr); bzero(&servaddr,sizeof(servaddr));

**/\*Socket address structure\*/** servaddr.sin\_family=AF\_INET; servaddr.sin\_addr.s\_addr=htonl(INADDR\_ANY); servaddr.sin\_port=htons(1999);

# /\*TCP socket is created, an Internet socket address structure is filled with wildcard address & server’s well known port\*/

sd=socket(AF\_INET,SOCK\_STREAM,0);

# /\*Bind function assigns a local protocol address to the socket\*/

bd=bind(sd,(struct sockaddr\*)&servaddr,sizeof(servaddr));

# /\*Listen function specifies the maximum number of connections that kernel should queue for this socket\*/

listen(sd,5);

printf("Server is running….\n");

# /\*The server to return the next completed connection from the front of the completed connection Queue calls it\*/

ad=accept(sd,(struct sockaddr\*)&cliaddr,&clilen); while(1)

{

bzero(&buff,sizeof(buff));

# /\*Receiving the request message from the client\*/

recv(ad,buff,sizeof(buff),0); printf("Message received is %s\n",buff);

}

}

**Client: tcpclient.c** #include<stdio.h> #include<string.h> #include<sys/socket.h> #include<sys/types.h> #include<unistd.h> #include<netinet/in.h> #include<netdb.h> #include<arpa/inet.h>

int main(int argc,char \* argv[])

{

int cd,sd,ad; char buff[1024];

struct sockaddr\_in cliaddr,servaddr;

struct hostent \*h;

# /\*This function looks up a hostname and it returns a pointer to a hostent structure that contains all the IPV4 address\*/

h=gethostbyname(argv[1]); bzero(&servaddr,sizeof(servaddr));

# /\*Socket address structure\*/

servaddr.sin\_family=AF\_INET;

memcpy((char \*)&servaddr.sin\_addr.s\_addr,h->h\_addr\_list[0],h->h\_length); servaddr.sin\_port = htons(1999);

# /\*Creating a socket, assigning IP address and port number for that socket\*/

sd = socket(AF\_INET,SOCK\_STREAM,0);

# /\*Connect establishes connection with the server using server IP address\*/

cd=connect(sd,(struct sockaddr\*)&servaddr,sizeof(servaddr)); while(1)

{

printf("Enter the message: \n");

# /\*Reads the message from standard input\*/

fgets(buff,100,stdin);

# /\*Send function is used on client side to send data given by user on client side to the server\*/

send(sd,buff,sizeof(buff)+1,0); printf("\n Data Sent ");

//recv(sd,buff,strlen(buff)+1,0); printf("%s",buff);

}

}

# SAMPLE OUTPUT:

**Server**:

# (Host Name:Root1)

[root@localhost 4ita33]# vi tcpserver.c [root@localhost 4ita33]# cc tcpserver.c [root@localhost 4ita33]# ./a.out

Server is running…. Message received is hi

Message received is hi

# Client:

**(Host Name:Root2)**

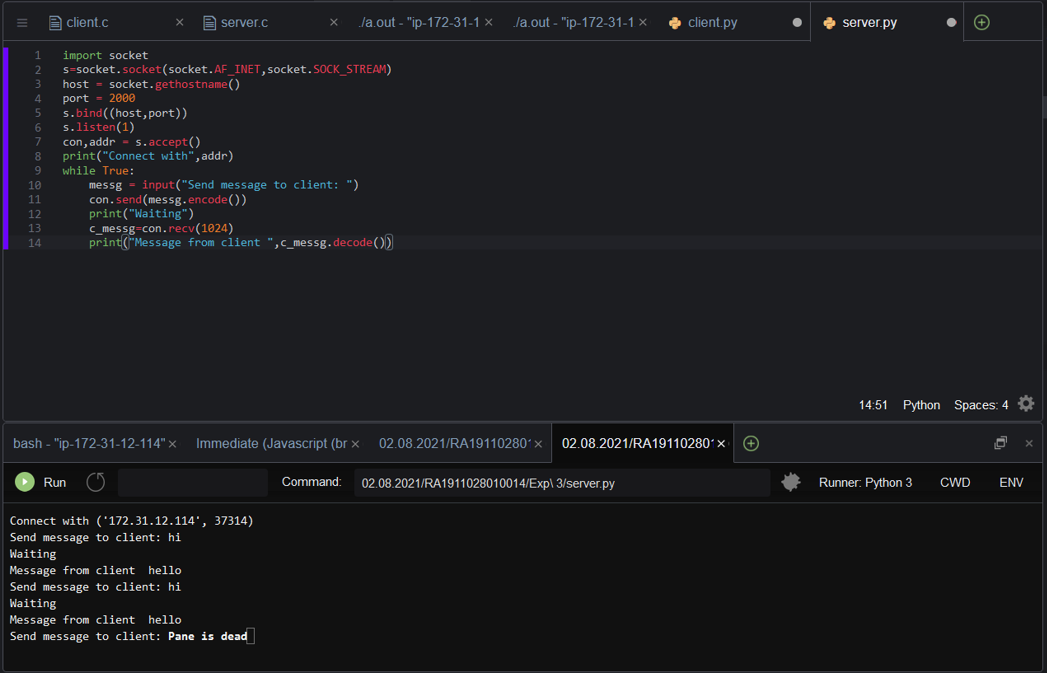
[root@localhost 4ita33]# vi tcpclient.c [root@localhost 4ita33]# cc tcpclient.c [root@localhost 4ita33]# ./a.out 127.0.0.1 Enter the message:

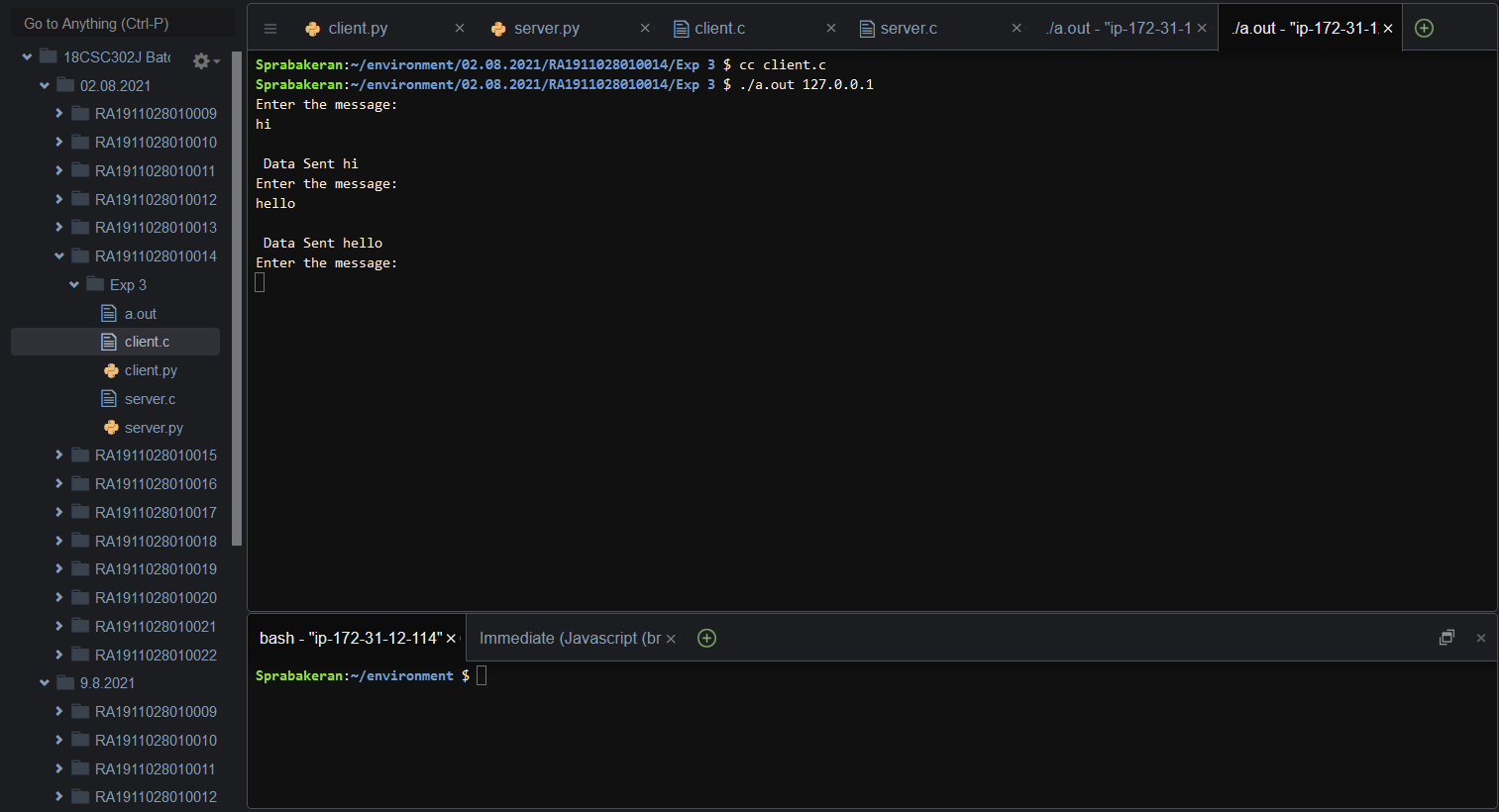
hi

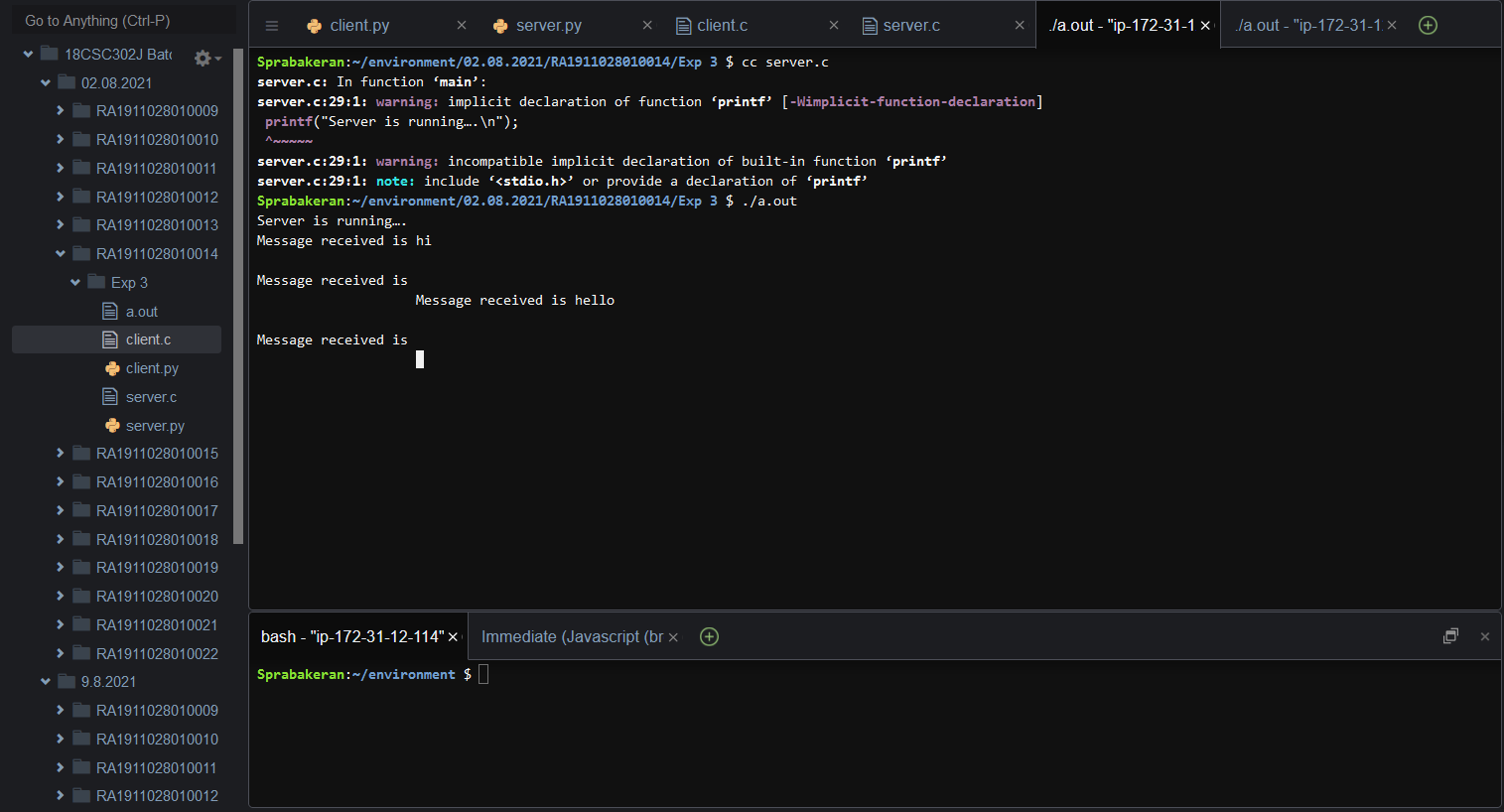
Data Sent hi Enter the message: how r u

Data Sent how r u Enter the message:

**OUTPUT:**

****

****

****

# INFERENCE:

Thus, a program to perform simple communication between client and server using TCP/IP was implemented.