**Name: Brijesh Rohit**

**Admission no.: U19CS009**

**DISTRIBUTED SYSTEMS - ASSIGNMENT - 4**

**Implement echo client-server message passing application. Message sent from client should be displayed on server and then program should terminate.**

**1. Write a server (TCP) C Program that opens a listening socket and waits to serve client.**

**CODE=>**

|  |
| --- |
| //U19CS009  //Brijesh Rohit  #include<stdio.h>  #include<stdlib.h>  #include<sys/types.h>  #include<sys/socket.h>  #include<unistd.h>  #include<netinet/in.h>  int main() {  //creating a socket for server  int server\_socket, client\_socket;  server\_socket = socket(AF\_INET, SOCK\_STREAM, 0);  if (server\_socket < 0)  {  perror("--> Server error !!!\n");  exit(1);  }  //define server\_address and client\_address  struct sockaddr\_in server\_address, client\_address;  server\_address.sin\_family = AF\_INET;  server\_address.sin\_port = htons(9002);  //passing port number 9002  server\_address.sin\_addr.s\_addr = INADDR\_ANY;  //specifing local machine address  //binding socket with specific IP and port number  int bind\_connection = bind(server\_socket, (struct sockaddr \*)&server\_address, sizeof(server\_address));  printf("Bind to the port number : 9002");  //listening to connection  listen(server\_socket, 4);  printf("\nListening...........\n");  //accept a connection  int size\_client\_addr = sizeof(client\_address);  client\_socket = accept(server\_socket, (struct sockaddr \*)&client\_address, &size\_client\_addr);  printf(".....Client is connected.....\n");  //recive data from client  char response\_client[256];  recv(client\_socket, &response\_client, sizeof(response\_client), 0);  //last parameter is optional so putting 0  //print out the data received from client  printf("Message from client : ");  printf("%s", response\_client);  printf("\n");  //close the socket  close(server\_socket);  printf("\nClient disconnected…………..\n");  return 0;  } |

**2. Write a client (TCP) C Program that connects with the server program knowing IP address and port number.**

**CODE=>**

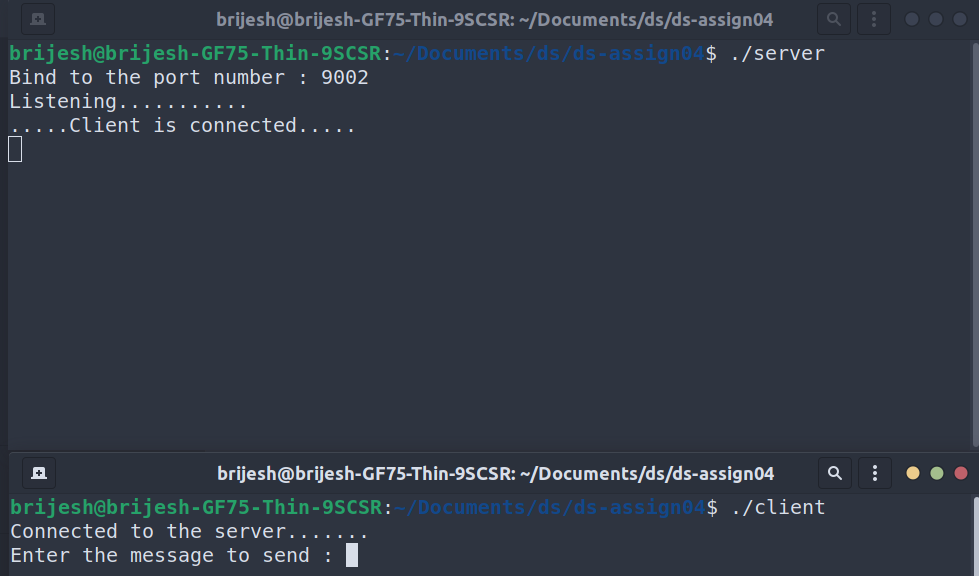
|  |
| --- |
| //U19CS009  //Brijesh Rohit  #include<stdio.h>  #include<stdlib.h>  #include<sys/types.h>  #include<sys/socket.h>  #include<unistd.h>  #include<netinet/in.h>  int main() {  //Creating a socket for client  int no\_socket;  no\_socket = socket(AF\_INET, SOCK\_STREAM, 0);  //specifing address for client socket  struct sockaddr\_in client\_address;  client\_address.sin\_family = AF\_INET;  client\_address.sin\_port = htons(9002);  //passing port number 9002  client\_address.sin\_addr.s\_addr = INADDR\_ANY;  //specifing local machine address  int connection\_status = connect(no\_socket, (struct sockaddr \*)&client\_address, sizeof(client\_address));  // 0 OK -1 error  //Checking whether there is an error in connection  if (connection\_status < 0)  {  perror("--->There was an error making connection with the remote socket\n\n");  exit(1);  }  printf("Connected to the server.......\n");  //send data to server  char send\_server[256];  printf("Enter the message to send : ");  scanf("%s", send\_server);  send(no\_socket, send\_server, sizeof(send\_server), 0);  printf("\n");  //close the connection  close(no\_socket);  printf("\n Disconnected from server.......\n");  return 0;  } |

**3. Get the input string from console on client and send it to server, server displays the same string.**

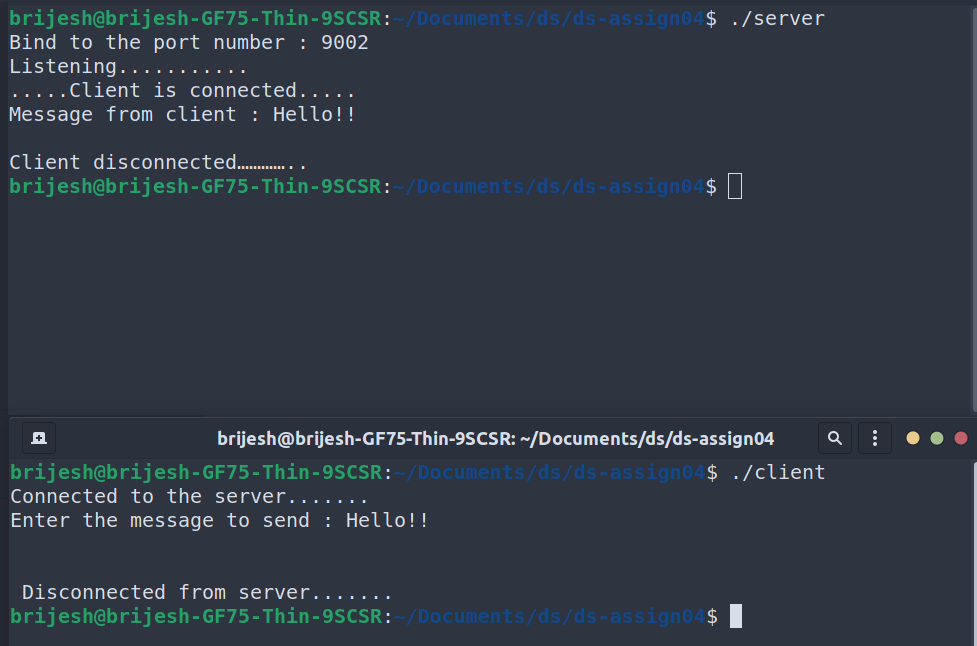
**Waiting for client to respond**



**Client connected and connection established**



**Requested completed and message exchanged, and network ended.**



**In case of error on client side, connection automatically ends and nothing is passed.**

