CSE1004 NAC LAB
ASSIGNMENT NO. – 1

SOCKET PROGRAMMING

DATE – 06/01/22

NAME – AYUSHI TRIVEDI

REGISTRATION NUMBER – 20BPS1135

CLIENT:

ALGORITHM:

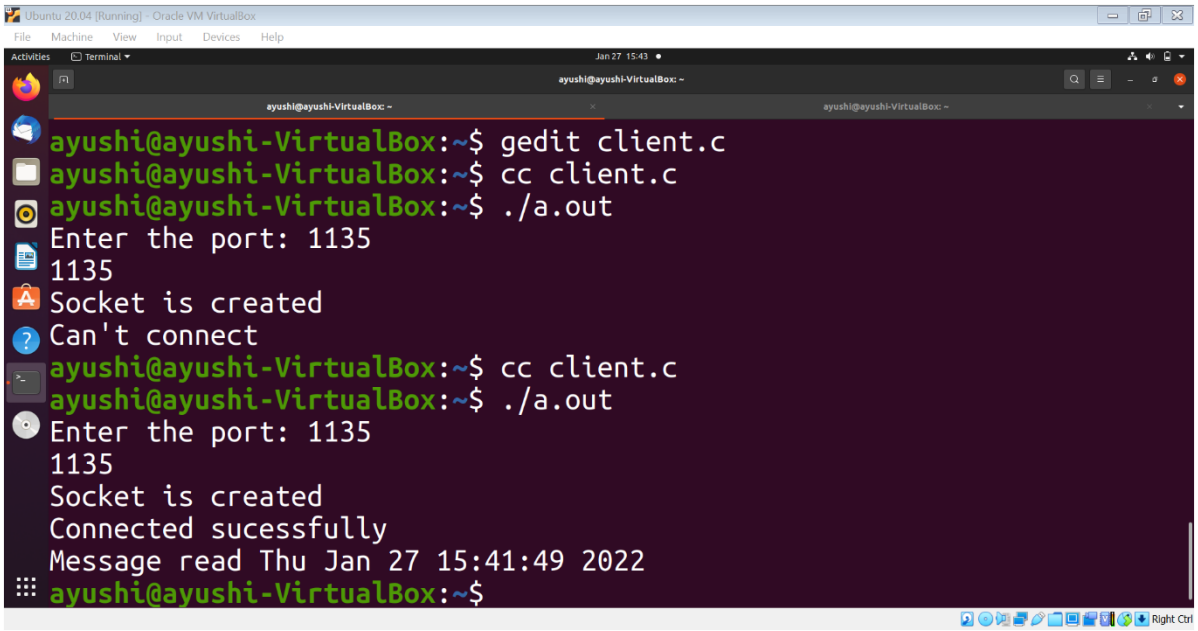
1. Take the port number from the user.
2. Start the program by creating the socket using socket() function, hold it in variable named sockid.
3. Now use connect() function, to connect your client to the server, so that they can exchange information among them
4. Now use recv() to receive the data packet from sever side and store it into the char array buffer.
5. Then close the connection, to stop receiving the data packet from server.

CODE:

```
#include<stdio.h>
#include<sys/types.h>
#include<netinet/in.h>
#include<string.h>
int main(){
int csd,cport,len;
char sendmsg[100],revmsg[100];
struct sockaddr_in servaddr;
printf("Enter the port: ");
```

```
scanf("%d",&cport);
printf("%d",cport);
csd=socket(AF_INET,SOCK_STREAM,0);
if(csd<0)
printf("Can't create\n");
else
printf("\nSocket is created\n");
servaddr.sin_family=AF_INET;
servaddr.sin_addr.s_addr=htonl(INADDR_ANY);
servaddr.sin_port=htons(cport);
if(connect(csd,(struct sockaddr *)&servaddr,sizeof(servaddr))<0)
printf("Can't connect\n");
else
printf("Connected sucessfully\n");
recv(csd,revmsg,100,0);
printf("Message read %s",revmsg);
}
```

OUTPUT:



SERVER:

ALGORITHM:

1. Take the port number from the user.
2. Start the program by creating the socket using socket() function, hold it in variable named sockid.
3. Then use bind() function to bind the current sockid to the program
4. Then use listen() function to check whether server is willing for communication.
5. Now using accept() function to make a synchronization point from client() side
6. Once done, make the sever to send the data packet to client side.
7. Once received, printf() the data packet received.

CODE:

```
#include<stdio.h>
#include<sys/types.h>
#include<netinet/in.h>
#include<string.h>
#include<time.h>
int main(){
int sd,sd2,nsd,clilen,sport,len;
int port;
time_t ticks;
char sendmsg[100],rcvmsg[100];
struct sockaddr_in servaddr,cliaddr;
printf("Enter the server port: ");
scanf("%d",&sport);
printf("%d",sport);
sd=socket(AF_INET,SOCK_STREAM,0);
ticks=time(NULL);
```

```
strcpy(sendmsg,ctime(&ticks));
if(sd<0)
        printf("Can't create \n");
else
        printf("\nSocket is created\n");
        servaddr.sin_family=AF_INET;
        servaddr.sin_addr.s_addr=htonl(INADDR_ANY);
        servaddr.sin_port=htons(sport);
        sd2=bind(sd,(struct sockaddr*) &servaddr,sizeof(servaddr));
if(sd2<0)
        printf("Can't bind\n");
else
        printf("Binded \n");
        listen(sd,5);
        clilen=sizeof(cliaddr);
        nsd=accept(sd,(struct sockaddr *)&cliaddr,&clilen);
if(nsd<0)
        printf("Can't accept\n");
else
        printf("Accepted\n");
        send(nsd,sendmsg,100,0);
}
```

OUTPUT: