**Assignment No : A 3 (a)**

**TCP**

**SOURCE CODE:**

TCP SERVER:

#include <stdio.h>

#include <netdb.h>

#include <netinet/in.h>

#include <stdlib.h>

#include <string.h>

#include <sys/socket.h>

#include <sys/types.h>

#define MAX 80

#define PORT 8080

#define SA struct sockaddr

//function designed for chat between client and server

void chat(int connfd){

char buff[MAX];

int n;

//infinite loop for chat

for(;;){

bzero(buff,MAX);

//read the message from client and copy it in buffer

read(connfd,buff,sizeof(buff));

//print buffer which contains the client contents

printf("From client: %s\t To client :",buff);

bzero(buff,MAX);

n=0;

//copy server message in the buffer

while ((buff[n++]=getchar())!='\n');

//and send that buffer to client

write(connfd,buff,sizeof(buff));

//if msg contains"EXIT" hen server exit and chat ended.

if (strncmp("exit",buff,4)==0) {

printf("Server exit...\n");

break;

}

}

}

//Driver function

int main()

{

int sockfd,connfd,len;

struct sockaddr\_in servaddr,cli;

//socket create and verification

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

if(sockfd== -1)

{

printf("socket creation failed...\n");

exit(0);

}

else

printf("Socket successfully created..\n");

bzero(&servaddr, sizeof(servaddr));

//assign IP,PORT

servaddr.sin\_family=AF\_INET;

servaddr.sin\_addr.s\_addr=htonl(INADDR\_ANY);

servaddr.sin\_port=htons(PORT);

//Binding newly created socket to given IP and verification

if((bind(sockfd, (SA\*)&servaddr, sizeof(servaddr))) !=0) {

printf("socket bind failed...\n");

exit(0);

}

else

printf("Socket successfully binded...\n");

//now server is ready to listen and verification

if((listen(sockfd,5)) !=0)

{

printf("Listen failed...\n");

exit(0);

}

else

printf("Server listening...\n");

len = sizeof(cli);

//accept the data packet from client and verification

connfd = accept(sockfd, (SA\*)&cli, &len);

if(connfd<0)

{

printf("server accept failed..\n");

exit(0);

}

else

printf("server accept the client...\n");

//function for chatting between client and server

chat(connfd);

//after chatting close the socket

close(sockfd);

}

TCP CLIENT:

#include <stdio.h>

#include <netdb.h>

#include <sys/types.h>

#include <stdlib.h>

#include <string.h>

#include <sys/socket.h>

#define MAX 80

#define PORT 8080

#define SA struct sockaddr

void chat(int sockfd)

{

char buff[MAX];

int n;

for(;;)

{

bzero(buff, sizeof(buff));

printf("Enter the string:");

n=0;

while ((buff[n++] = getchar()) !='\n');

write(sockfd, buff, sizeof(buff));

read(sockfd, buff, sizeof(buff));

printf("From server: %s", buff);

if ((strncmp(buff, "exit",4)) == 0)

{

printf("Client exit..\n");

break;

}

}

}

int main()

{

int sockfd, connfd;

struct sockaddr\_in servaddr, cli;

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

if(sockfd == -1)

{

printf("socket creation failed...\n");

exit(0);

}

else

printf("socket successfully created..\n");

bzero(&servaddr, sizeof(servaddr));

servaddr.sin\_family = AF\_INET;

servaddr.sin\_addr.s\_addr=inet\_addr("127.0.0.1");

servaddr.sin\_port = htons(PORT);

if(connect(sockfd, (SA\*)&servaddr, sizeof(servaddr)) !=0)

{

printf("connection with the server failed...\n");

exit(0);

}

else

printf("connected to the server..\n");

chat(sockfd);

close(sockfd);

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*OUTPUT\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

