```
SERVER TCP

/*This server program is concurrent*/
#include <sys/types.h>
#include <sys/socket.h>
#include <stdio.h>
#include <netinet/in.h>
#define MAXLINESIZE 100
#define SERV_PORT 5555
int listensd,clientsd;
char buffer[MAXLINESIZE+1];
struct sockaddr_in servaddr;
int noBytesRead=0;


/*this function will server client that connects*/
void processClient(int);

int main()
{
    /*Create socket*/
    if((listensd=socket(AF_INET,SOCK_STREAM,0))<0)
    {
        fprintf(stderr,"Cannot create socket\n");
        exit(-1);
    }

    /*Initialize socket address structure*/
    bzero(&servaddr,sizeof(servaddr));
    servaddr.sin_family=AF_INET;
    servaddr.sin_port=htons(SERV_PORT);
    servaddr.sin_addr.s_addr=htonl(INADDR_ANY);

    /*bind socket address to the socket*/
    if(bind(listensd,(struct sockaddr*)&servaddr,sizeof(servaddr))<0)
    {
        fprintf(stderr,"Error in bind\n");
        exit(-1);
    }

    /*Make the socket listening socket*/
    if(listen(listensd,5)<0)
    {
        fprintf(stderr,"Error in listen\n");
        exit(-1);
    }

    for(;;)
    {
        /*wait for client connection*/
        clientsd=accept(listensd,(struct sockaddr*)NULL,NULL);
        if(fork()==0)
        {
            /*close listening socket in child.
            So that reference count remains one.
            The child serves the client.
            It does not need listening socket to do this. */
            close(listensd);

            /*server client*/
            processClient(clientsd);
            /*close connected socket*/
            close(clientsd);
            exit(0);
        }
        /*close connected socket in parent so that reference count remains one. */
        close(clientsd);
    }
    return 0;
}

void processClient(int clientsd)
{
```

```
    /*read message from client and send back*/
    while((noBytesRead=read(clientsd,buffer,sizeof(buffer)))>0)
    write(clientsd,buffer,noBytesRead);
}



Client

#include <sys/types.h>
#include <sys/socket.h>
#include <stdio.h>
#include <netinet/in.h>
#include <string.h>
#define MAXLINESIZE 100
#define SERV_PORT 5555

int main(int argc,char** argv)
{
    int connectsd;
    char sendBuffer[MAXLINESIZE+1];
    char recvBuffer[MAXLINESIZE+1];
    struct sockaddr_in servaddr;
    int noBytesRead=0;

    if(argc!=2)
    {
        fprintf(stderr,"Usage: %s IP-Address\n",argv[0]);
        exit(-1);
    }

    /*Create socket*/
    if((connectsd=socket(AF_INET,SOCK_STREAM,0))<0)
    {
        fprintf(stderr,"Cannot create socket\n");
        exit(-1);
    }

    /*Initialize socket address structure*/
    bzero(&servaddr,sizeof(servaddr));
    servaddr.sin_family=AF_INET;
    servaddr.sin_port=htons(SERV_PORT);

    /*assign server address in socket address structure*/
    if(inet_pton(PF_INET,argv[1],&servaddr.sin_addr)<=0)
    {
        fprintf(stderr,"Error in inet_pton\n");
        exit(-1);
    }

    /*Get connected with the server*/
    if(connect(connectsd,(struct sockaddr*)&servaddr,sizeof(servaddr))<0)
    {
        fprintf(stderr,"Error in connect\n");
        exit(-1);
    }

    /*Read message from user through keyboard*/
    for(;gets(sendBuffer)!=NULL;)
    {
        /*Send the message to the server*/
        write(connectsd,sendBuffer,strlen(sendBuffer)+1);
        if(noBytesRead=read(connectsd,recvBuffer,sizeof(recvBuffer))<0)
            exit(0);

        /*Display what the server sent in reply*/
        fprintf(stdout,"%s\n",recvBuffer);
    }
    return 0;
}
```