**CONNECTION**

**AIM:**

TCP Connection

**PROGRAM:**

**SERVER:**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<sys/socket.h>

#include<sys/types.h>

#include<netdb.h>

#include<netinet/in.h>

#define PORT 8080

#define MAX 1024

#define SA struct sockaddr

void func(int sockfd){

char buff[MAX];

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

printf("\n\n%s", buff);

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

}

int main(){

int sockfd, connfd, len;

struct sockaddr\_in servaddr, cli;

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

perror("Socket status :");

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);

bind(sockfd, (SA\*)&servaddr, sizeof(servaddr));

perror("Bind status :");

listen(sockfd, 5);

perror("Listening :");

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

perror("Connection accept status :");

func(connfd);

close(sockfd);

}

**CLIENT:**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<sys/socket.h>

#include<sys/types.h>

#include<netdb.h>

#include<netinet/in.h>

#define PORT 8080

#define MAX 1024

#define SA struct sockaddr

void func(int sockfd){

char buff[MAX];

int n=0;

printf("Enter the string :");

scanf("%s",buff);

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

}

int main(){

int sockfd, connfd;

struct sockaddr\_in servaddr, cli;

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

perror("Socket status :");

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);

connect(sockfd, (SA\*)&servaddr, sizeof(servaddr));

perror("Connect status :");

func(sockfd);

close(sockfd);

}

**AIM:**

UDP Connection

**PROGRAM:**

**SERVER:**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<sys/socket.h>

#include<sys/types.h>

#include<netdb.h>

#include<netinet/in.h>

#define PORT 8080

#define SA struct sockaddr

int main(){

int sockfd,connfd;

struct sockaddr\_in servaddr, cli\_addr;

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

perror("Socket status :");

servaddr.sin\_family = AF\_INET;

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

servaddr.sin\_port = htons(PORT);

bind(sockfd,(SA\*)&servaddr,sizeof(servaddr));

perror("Bind :");

char buff[100];

int len = sizeof(cli\_addr);

recvfrom(sockfd,buff,sizeof(buff),0,(SA\*)&cli\_addr,&len);

perror("recv");

printf("\n%s",buff);

close(sockfd);

}

**CLIENT:**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<sys/socket.h>

#include<sys/types.h>

#include<netdb.h>

#include<netinet/in.h>

#define PORT 8080

#define SA struct sockaddr

int main(){

int sockfd,connfd;

struct sockaddr\_in servaddr, cli\_addr;

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

perror("Socket status :");

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);

char buff[]="vijay";

int len = sizeof(servaddr);

sendto(sockfd,buff,sizeof(buff),0,(SA\*)&servaddr,len);

perror("send");

close(sockfd);

}