Name: Preyash   Date: 06-01-2022

Registration Number: 20BPS1022

LAB-01

Socket Programming

AIM: To print the day and time in the client side using the socket functions.

ALGORITHM

Server-side Algorithm

1. Start.

2. Declare variables.

3. Take the port number as input from the user.

4. Using socket(), create TCP socket for server.

5. Using bind(), bind the socket to server address.

6. Using listen(), put the server socket in a passive mode, where it waits for the client to approach the server to make a connection

 7. Using accept(), at this point, connection is established between client and server, and they are ready to transfer data.

Client-Side Algorithm

1. Start.

2. Declare variables.

3. Take port as input from user.

4. Using socket(), create TCP socket for client.

5. Using connect(),connect the client to server to exchange information.

6. Using recv() ,receive the data from server side.

7. If the connection will be successful data will be revived.

8. The date and time received from server side is printed at client side.

Server Program Source Code:

```c
#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:\n");
        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("Socket 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:

```
preyash-20bps1022@Preyash-20BPS1022:~/Netcom1022/SocketPrograming$ gedit day_server.c
preyash-20bps1022@Preyash-20BPS1022:~/Netcom1022/SocketPrograming$ gedit day_server.c
preyash-20bps1022@Preyash-20BPS1022:~/Netcom1022/SocketPrograming$ gcc day_server.c
preyash-20bps1022@Preyash-20BPS1022:~/Netcom1022/SocketPrograming$ ./a.out
Enter the server port:
2906
2906Socket is created
 Binded
Accepted
```

Client Program Source Code:

```c
#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 \n");
        scanf("%d",&cport);
        printf("%d",cport);
        csd=socket(AF_INET,SOCK_STREAM,0);
        if(csd<0)
        printf("Can't create\n");
        else
        printf("Socket 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:

```
preyash-20bps1022@Preyash-20BPS1022:~/Netcom1022/SocketPrograming$ gedit day_client.c
preyash-20bps1022@Preyash-20BPS1022:~/Netcom1022/SocketPrograming$ gcc day_client.c
preyash-20bps1022@Preyash-20BPS1022:~/Netcom1022/SocketPrograming$ ./a.out
Enter the port
2906
2906Socket is created
Connected sucessfully
Message read Wed Jan 26 18:19:42 2022
```

Result: A successful connection between the client and server is established.