

## Lab Assignment 6

*sender*

```
#include<stdio.h>
#include<string.h>
#include<arpa/inet.h>
#include<unistd.h>
#include<sys/socket.h>
int main()
{
    char str[100],ans[10];
    struct sockaddr_in serv;
    int broad=1,sock;
    sock=socket(AF_INET,SOCK_DGRAM,0);
    bzero(&serv,sizeof(serv));
    bzero(ans,10);
    setsockopt(sock,SOL_SOCKET,SO_BROADCAST,&broad,sizeof(broad));
    serv.sin_family=AF_INET;
    serv.sin_port=htons(12345);
    serv.sin_addr.s_addr=inet_addr("255.255.255.255");
    while(1)
    {
        printf("Do you want to broadcast?");
        fgets(ans,10,stdin);
        if(strncmp(ans,"yes",3)==0)
        {
            bzero(str,100);
            printf("Enter the data");
            fgets(str,100,stdin);
            sendto(sock,str,strlen(str),0,(struct sockaddr *)&serv,sizeof(serv));
        }
    }

    close(sock);
}
```

*receiver*

```

#include<stdio.h>
#include<string.h>
#include<arpa/inet.h>
#include<unistd.h>
#include<sys/socket.h>
int main()
{
    char str[100];
    struct sockaddr_in serv;
    int reuse=1, sock, len;
    sock=socket(AF_INET, SOCK_DGRAM, 0);
    bzero(&serv, sizeof(serv));
    setsockopt(sock, SOL_SOCKET, SO_REUSEADDR, &reuse, sizeof(reuse));
    serv.sin_family=AF_INET;
    serv.sin_port=htons(12345);
    serv.sin_addr.s_addr=INADDR_ANY;
    len=sizeof(serv);
    bind(sock, (struct sockaddr *)&serv, sizeof(serv));
    while(1)
    {
        bzero(str, 100);
        recvfrom(sock, str, 100, 0, (struct sockaddr *)&serv, &len);
        printf("%s", str);
    }
    close(sock);
}

```

## Lab Assignment 7

*sender*

```

/* Multicast Sender */
#include <sys/types.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <netinet/in.h>
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int sck;

```

```

struct sockaddr_in grpSck;
struct in_addr lclInterface;
char buf[1024];
int blen;

sck=socket(AF_INET,SOCK_DGRAM,0);
bzero((char *) &grpSck, sizeof(grpSck));
grpSck.sin_family = AF_INET;
grpSck.sin_addr.s_addr = inet_addr("226.1.2.3");
grpSck.sin_port = htons(5435);

lclInterface.s_addr =htonl(INADDR_ANY);

setsockopt(sck, IPPROTO_IP, IP_MULTICAST_IF, (char *)&lclInterface,

while(1)
{
    blen=sizeof(buf);
    bzero(buf,blen);

    printf("\nEnter multicast message :");
    fgets(buf,blen,stdin);

    sendto(sck,buf,blen,0, (struct sockaddr*)&grpSck, sizeof(grpSck)
}
return 0;
}

```

## *receiver*

```

/* Multicast Receiver */
#include <sys/types.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <netinet/in.h>
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int sck;
    struct sockaddr_in lclSck;
    struct ip_mreq grp;

```

```

struct in_addr lclInterface;
char buf[1024];
int reuse = 1;
int blen;

sck=socket(AF_INET,SOCK_DGRAM,0);

setsockopt(sck, SOL_SOCKET, SO_REUSEADDR, (char *)&reuse, sizeof(reu

bzero((char *)&lclSck, sizeof(lclSck));
lclSck.sin_family = AF_INET;
lclSck.sin_port = htons(5435);
lclSck.sin_addr.s_addr = INADDR_ANY;

bind(sck, (struct sockaddr*)&lclSck, sizeof(lclSck)))

grp.imr_multiaddr.s_addr = inet_addr("226.1.2.3");
grp.imr_interface.s_addr = htonl(INADDR_ANY);

setsockopt(sck, IPPROTO_IP, IP_ADD_MEMBERSHIP, (char *)&grp, sizeof(
while(1)
{
    blen=sizeof(buf);
    bzero(buf,blen);
    read(sck,buf,blen)
    printf("\nMessage from multicast sender is: %s",buf);
}

}
return 0;
}

```

## Lab Assignment 8

1. Config the router
  1. make vlans and give name
  2. add port types to the fa ports
  3. add access to pc ports
  4. add trunk to router port

```
enable
config terminal
vlan 10
names sales
vlan 20
name it

int fa 0/1
switchport mode access
switchport access vlan 10
## similar for the rest port

int fa 0/5
switchport mode trunk
```

2. Assign ip address and Dns to all the systems
3. config router

```
enable
config terminal
int fa 0/0
no shutdown
int fa 0/0.10
encapsulation dot1q 10
ip add 192.168.1.1 255.255.255.0

int fa 0/0.20
encapsulation dot2q 20
ip add 192.168.2.1 255.255.255.0
```

router is 2811

DNS me wo padega jo router me ip add hui hai

## Lab Assignment 9

Switched virtual interfaces (SVI)

## 1. config layer 3 switch

```
vlan 10
name sales
vlan 20
name it

int vlan 10
ip add 192.168.10.1 255.255.255.0
no shut
exit

int vlan 20
ip add 192.168.20.1 255.255.255.0
no shut
exit
```

## 2. add pc mode access

## 3. add pc ip address and dns

| DNS will the ip address added in the switch.

```
ip routing
```