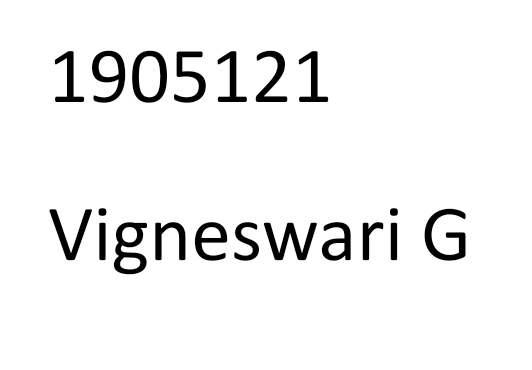
**Multicast Application**

**Aim:**

To develop UDP Multicast server/Client for Automated Question paper distribution in a university

Maintain three departments as a multicast group (CSE, IT, ECE) in the server.

Consider 10 students as clients and allow the students to become the members of one of the multicast group available in the server.

Once all the 10 clients join the group the server should distribute the respective Question papers to the students.

**Code**:

Server.c

#include<stdio.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<string.h>

#include<sys/types.h>

#include<arpa/inet.h>

#include<net/if.h>

#include<time.h>

void main()

{

int s,k,len;

const int on=1;

struct sockaddr\_in servaddr, cliaddr1, cliaddr2, cliaddr3;

char cse\_qp[10]="cse\_qp.txt";

char it\_qp[10]="it\_qp.txt";

char ece\_qp[10]="ece\_qp.txt";

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

perror("Socket status\n");

bzero(&servaddr, sizeof(servaddr));

servaddr.sin\_port=htons(5000);

servaddr.sin\_family=AF\_INET;

inet\_pton(AF\_INET, "192.168.1.6",&servaddr.sin\_addr.s\_addr);

bzero(&cliaddr1, sizeof(cliaddr1));

cliaddr1.sin\_port=htons(5000);

cliaddr1.sin\_family=AF\_INET;

inet\_pton(AF\_INET, "227.0.0.2",&cliaddr1.sin\_addr.s\_addr);

len=sizeof(struct sockaddr);

sendto(s,cse\_qp,sizeof(cse\_qp),0,(struct sockaddr\*)&cliaddr1,len);

bzero(&cliaddr2, sizeof(cliaddr2));

cliaddr2.sin\_port=htons(5000);

cliaddr2.sin\_family=AF\_INET;

inet\_pton(AF\_INET, "225.0.0.16",&cliaddr2.sin\_addr.s\_addr);

sendto(s,it\_qp,sizeof(it\_qp),0,(struct sockaddr\*)&cliaddr2,len);

bzero(&cliaddr2, sizeof(cliaddr2));

cliaddr2.sin\_port=htons(5000);

cliaddr2.sin\_family=AF\_INET;

inet\_pton(AF\_INET, "226.0.0.15",&cliaddr2.sin\_addr.s\_addr);

sendto(s,ece\_qp,sizeof(ece\_qp),0,(struct sockaddr\*)&cliaddr2,len);

perror("status");

}

Multicast group 1:

#include<stdio.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<string.h>

#include<sys/types.h>

#include<arpa/inet.h>

#include<net/if.h>

#include<time.h>

void main()

{

int s,mlen,on=1;

char buff1[200], msg[300];

struct sockaddr\_in servaddr, cliaddr;

struct ip\_mreq mreq;

s = socket(PF\_INET, SOCK\_DGRAM,0);

perror("Socket status\n");

setsockopt(s,SOL\_SOCKET, SO\_REUSEADDR,&on, sizeof(on));

bzero((void \*)&servaddr, sizeof(servaddr));

servaddr.sin\_port=htons(5000);

servaddr.sin\_family=AF\_INET;

inet\_pton(AF\_INET, "192.168.1.6",&servaddr.sin\_addr.s\_addr);

bzero((void \*)&cliaddr, sizeof(cliaddr));

cliaddr.sin\_port=htons(5000);

cliaddr.sin\_family=AF\_INET;

inet\_pton(AF\_INET, "227.0.0.2",&cliaddr.sin\_addr.s\_addr);

mlen=sizeof(struct sockaddr\_in);

bind(s,(struct sockaddr\* )&cliaddr, sizeof(struct sockaddr));

perror("Bind status\n");

inet\_pton(AF\_INET, "227.0.0.2",(struct inaddr\*)&mreq.imr\_multiaddr);

inet\_pton(AF\_INET, "192.168.1.6",(struct inaddr\*)&mreq.imr\_interface);

setsockopt(s,IPPROTO\_IP,IP\_ADD\_MEMBERSHIP,&mreq,sizeof(mreq));

setsockopt(s,IPPROTO\_IP,IP\_MULTICAST\_LOOP,&mlen,sizeof(mlen));

recvfrom(s,buff1,sizeof(buff1),0,(struct sockaddr \*) & servaddr, &mlen);

int c,k=0;

buff1[strcspn(buff1,"\r\n")]=0;

FILE \*f = fopen(buff1,"r");

while((c = fgetc(f)) != "\0") {

if(feof(f))

break;

msg[k++]=c;

}

printf("%s",msg);

close(s);

}

Multicast group 2:

#include<stdio.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<string.h>

#include<sys/types.h>

#include<arpa/inet.h>

#include<net/if.h>

#include<time.h>

void main()

{

int s,mlen,on=1;

char buff1[200], msg[300];

struct sockaddr\_in servaddr, cliaddr;

struct ip\_mreq mreq;

s = socket(PF\_INET, SOCK\_DGRAM,0);

perror("Socket status\n");

setsockopt(s,SOL\_SOCKET, SO\_REUSEADDR,&on, sizeof(on));

bzero((void \*)&servaddr, sizeof(servaddr));

servaddr.sin\_port=htons(5000);

servaddr.sin\_family=AF\_INET;

inet\_pton(AF\_INET, "192.168.1.6",&servaddr.sin\_addr.s\_addr);

bzero((void \*)&cliaddr, sizeof(cliaddr));

cliaddr.sin\_port=htons(5000);

cliaddr.sin\_family=AF\_INET;

inet\_pton(AF\_INET, "225.0.0.16",&cliaddr.sin\_addr.s\_addr);

mlen=sizeof(struct sockaddr\_in);

bind(s,(struct sockaddr\* )&cliaddr, sizeof(struct sockaddr));

perror("Bind status\n");

inet\_pton(AF\_INET, "225.0.0.16",(struct inaddr\*)&mreq.imr\_multiaddr);

inet\_pton(AF\_INET, "192.168.1.6",(struct inaddr\*)&mreq.imr\_interface);

setsockopt(s,IPPROTO\_IP,IP\_ADD\_MEMBERSHIP,&mreq,sizeof(mreq));

setsockopt(s,IPPROTO\_IP,IP\_MULTICAST\_LOOP,&mlen,sizeof(mlen));

recvfrom(s,buff1,sizeof(buff1),0,(struct sockaddr \*) & servaddr, &mlen);

int c,k=0;

buff1[strcspn(buff1,"\r\n")]=0;

FILE \*f = fopen(buff1,"r");

while((c = fgetc(f)) != "\0") {

if(feof(f))

break;

msg[k++]=c;

}

printf("%s",msg);

close(s);

}

Multicast group 3:

#include<stdio.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<string.h>

#include<sys/types.h>

#include<arpa/inet.h>

#include<net/if.h>

#include<time.h>

void main()

{

int s,mlen,on=1;

char buff1[100], msg[200];

struct sockaddr\_in servaddr, cliaddr;

struct ip\_mreq mreq;

s = socket(PF\_INET, SOCK\_DGRAM,0);

perror("Socket status\n");

setsockopt(s,SOL\_SOCKET, SO\_REUSEADDR,&on, sizeof(on));

bzero((void \*)&servaddr, sizeof(servaddr));

servaddr.sin\_port=htons(5000);

servaddr.sin\_family=AF\_INET;

inet\_pton(AF\_INET, "192.168.1.6",&servaddr.sin\_addr.s\_addr);

bzero((void \*)&cliaddr, sizeof(cliaddr));

cliaddr.sin\_port=htons(5000);

cliaddr.sin\_family=AF\_INET;

inet\_pton(AF\_INET, "226.0.0.15",&cliaddr.sin\_addr.s\_addr);

mlen=sizeof(struct sockaddr\_in);

bind(s,(struct sockaddr\* )&cliaddr, sizeof(struct sockaddr));

perror("Bind status\n");

inet\_pton(AF\_INET, "226.0.0.15",(struct inaddr\*)&mreq.imr\_multiaddr);

inet\_pton(AF\_INET, "192.168.1.6",(struct inaddr\*)&mreq.imr\_interface);

setsockopt(s,IPPROTO\_IP,IP\_ADD\_MEMBERSHIP,&mreq,sizeof(mreq));

setsockopt(s,IPPROTO\_IP,IP\_MULTICAST\_LOOP,&mlen,sizeof(mlen));

recvfrom(s,buff1,sizeof(buff1),0,(struct sockaddr \*) & servaddr, &mlen);

int c,k=0;

buff1[strcspn(buff1,"\r\n")]=0;

FILE \*f = fopen(buff1,"r");

while((c = fgetc(f)) != "\0") {

if(feof(f))

break;

msg[k++]=c;

}

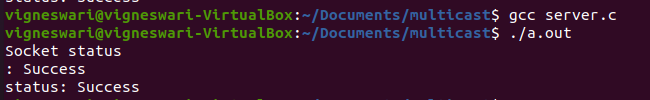
printf("%s",msg);

close(s);

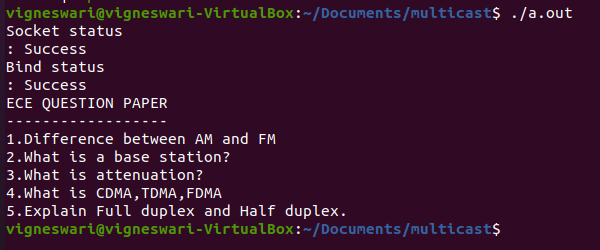
}

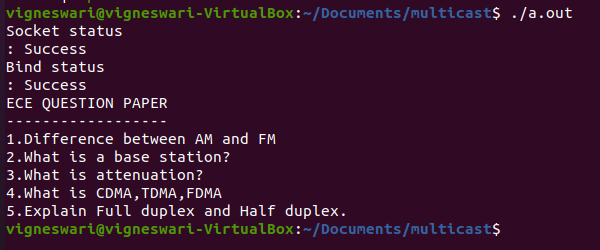
**Output:**

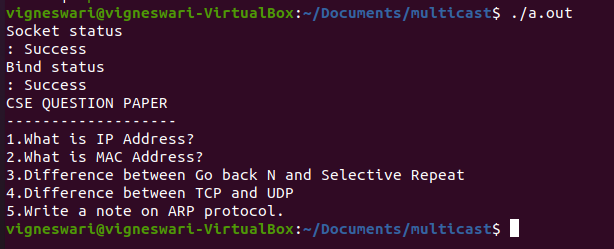
**Server:**

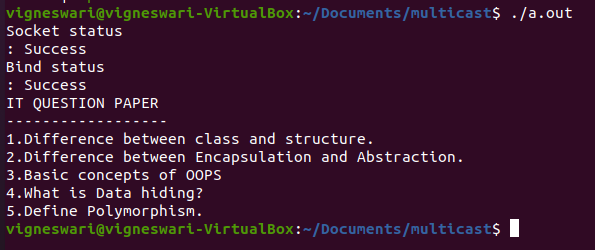
****

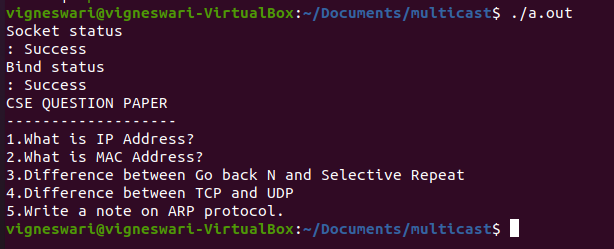
**Clients:**

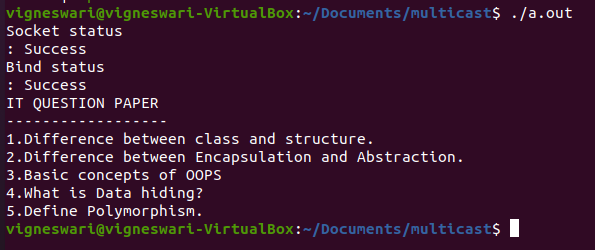


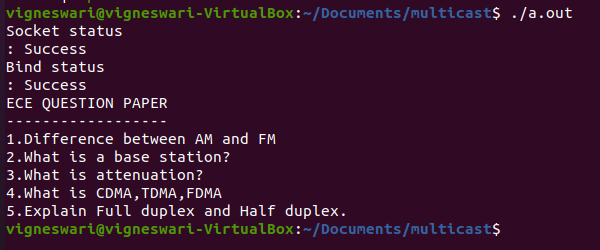


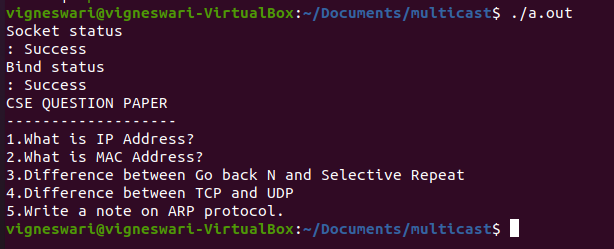
****

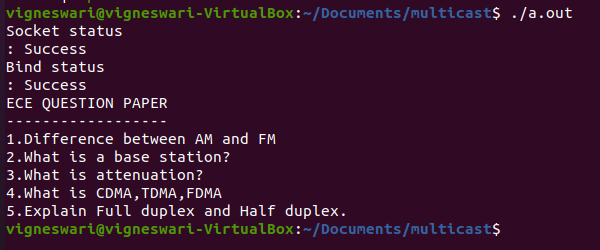
****

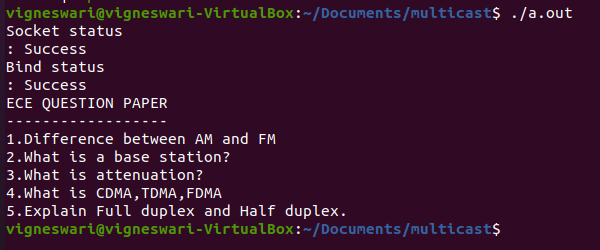
****

****

****

****

****

****

**Result**:

The above program to design a UDP multicast client-server is successfully executed.