EX NO: 8

09/12/2021

**MULTICAST APPLICATION**

**AIM:**

To create a UDP program to develop UDP Multicast server/client for automated Question paper distribution in a university.

**ALGORITHM:**

**SERVER:**

1. Include header files, initialize the required variables and specify the family, protocol, IP address and port number.
2. Create a socket using socket() function.
3. Observe the response of the various clients.
4. Based on the response from the clients, multicast the question papers’ message to the respective clients.
5. Verify the status of multicasting the messages.

**CLIENT:**

1. Include header files, initialize the required variables and specify the family, protocol, IP address and port number.
2. Create a socket using socket() function.
3. Call the setsocketopt() function.
4. Get the respective inputs from the user.
5. Based on the choice of user, display the respective question paper.
6. Close the socket.

**PROGRAM:**

**SERVER:**

**CLIENT:**

#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>

#include<netdb.h>

#include<stdlib.h>

#include<unistd.h>

void main(){

int sockfd,k,mlen,on=1,roll\_number,choice;

char buff1[100], msg[200],id[34];

struct sockaddr\_in servaddr, client\_address;

struct ip\_mreq mreq;

printf("\nEnter your roll number:");

scanf("%d",&roll\_number);

printf("\n\nEnter the respective choice for your department:\nCSE -- 1\nECE -- 2\nIT -- 3\n\n");

scanf("%d",&choice);

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

perror("\nSocket status :");

setsockopt(sockfd,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, "10.0.2.15",&servaddr.sin\_addr.s\_addr);

mlen=sizeof(struct sockaddr\_in);

if(choice==1){

bzero((void \*)&client\_address, sizeof(client\_address));

client\_address.sin\_port=htons(5000);

client\_address.sin\_family=AF\_INET;

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

bind(sockfd,(struct sockaddr\* )&client\_address, sizeof(struct sockaddr));

perror("\nBind status :");

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

inet\_pton(AF\_INET, "10.0.2.15",(struct inaddr\*)&mreq.imr\_interface);

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

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

}

if (choice==2){

bzero((void \*)&client\_address, sizeof(client\_address));

client\_address.sin\_port=htons(5000);

client\_address.sin\_family=AF\_INET;

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

bind(sockfd,(struct sockaddr\* )&client\_address, sizeof(struct sockaddr));

perror("\nBind status :");

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

inet\_pton(AF\_INET, "10.0.2.15",(struct inaddr\*)&mreq.imr\_interface);

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

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

}

if (choice==3){

bzero((void \*)&client\_address, sizeof(client\_address));

client\_address.sin\_port=htons(5000);

client\_address.sin\_family=AF\_INET;

inet\_pton(AF\_INET, "226.0.0.4",&client\_address.sin\_addr.s\_addr);

bind(sockfd,(struct sockaddr\* )&client\_address, sizeof(struct sockaddr));

perror("\nBind status :");

inet\_pton(AF\_INET, "226.0.0.4",(struct inaddr\*)&mreq.imr\_multiaddr);

inet\_pton(AF\_INET, "10.0.2.15",(struct inaddr\*)&mreq.imr\_interface);

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

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

}

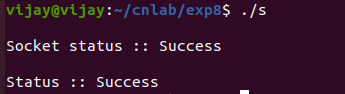
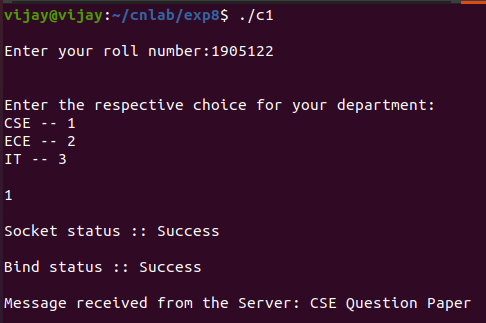
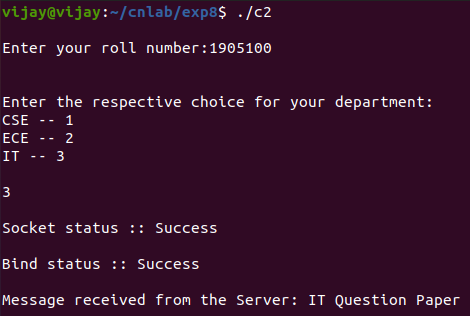
recvfrom(sockfd,buff1,sizeof(buff1),0,(struct sockaddr \*) & servaddr, &mlen);

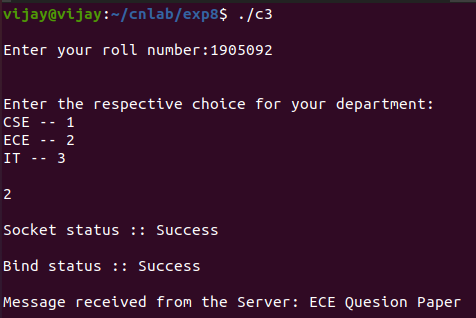
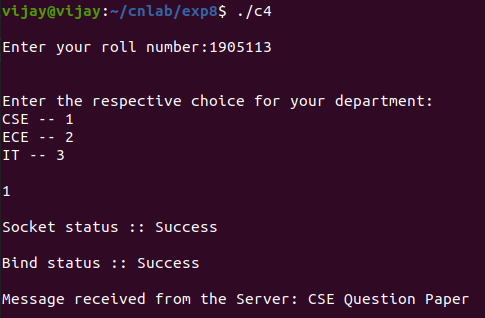
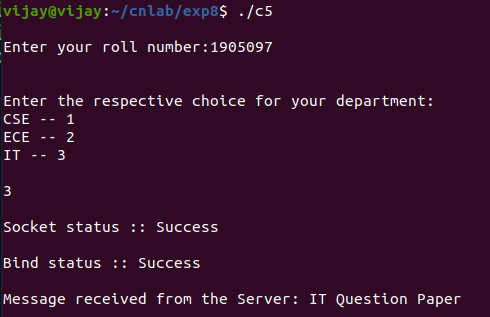
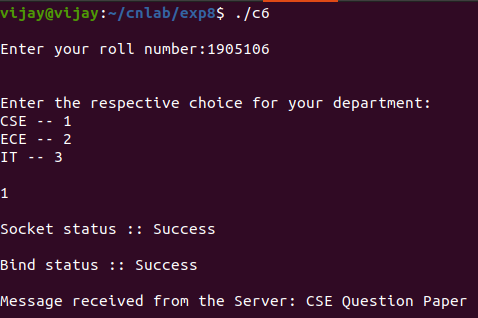
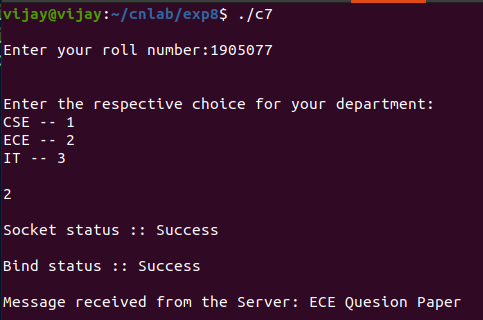
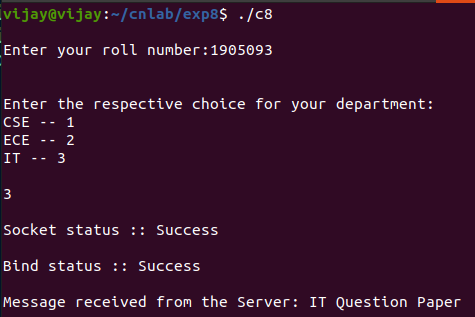
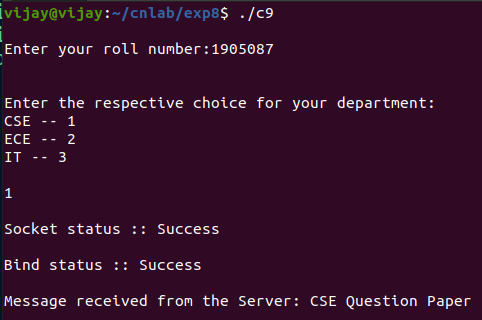
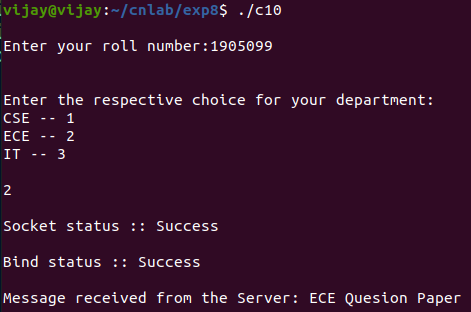
printf("\nMessage received from the Server: %s\n\n",buff1);

close(sockfd);

}

**SAMPLE OUTPUT:**



****

**RESULT:**

Hence a UDP program is created to develop UDP Multicast server/client for automated Question paper distribution in a university.