**21/09/2021 CN LAB 3 2019103573**

**CONNECTING MULTIPLE CLIENTS WITH A SINGLE SERVER USING SOCKETS**

**SERVER.C**

#include<stdio.h>

#include<stdlib.h>

#include<unistd.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<string.h>

#define PORT 3554

#define SIZE 256

int main()

{

    int serverSocket,newSocket;

    struct sockaddr\_in server\_address,new\_address;

    pid\_t childpid;

    socklen\_t serverLen = sizeof(server\_address);

    socklen\_t clientLen = sizeof(new\_address);

    char msg[SIZE],buf[SIZE];

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

    if(serverSocket<0)

    {

        perror("Can't create Socket..\n");

        return 1;

    }

    printf("Server socket created..\n");

    memset(&server\_address,'\0',sizeof(server\_address));

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

    server\_address.sin\_port = htons(PORT);

    server\_address.sin\_addr.s\_addr = INADDR\_ANY;

    int b = bind(serverSocket,(struct sockaddr \*)&server\_address,serverLen);

    if(b<0)

    {

        perror("Binding error..\n");

        return 1;

    }

    if(listen(serverSocket,2)==0)

    {

        printf("Listening..\n");

    }

    else

    {

        perror("Can't listen to requests..\n");

        return 1;

    }

    while(1)

    {

        newSocket = accept(serverSocket,(struct sockadddr\*)&new\_address,&clientLen);

        if(newSocket<0)

        {

            perror("Can't accept requests..\n");

            return 1;

        }

        printf("Connection accepted %d:%d",inet\_ntoa(new\_address.sin\_addr),ntohs(new\_address.sin\_port));

if((childpid=fork())==0)

        {

            close(serverSocket);

            while(1)

            {

                recv(newSocket,msg,sizeof(msg),0);

                if(strcmp(msg,"exit")==0)

                {

                    printf("Client %d disconnected from server..\n",inet\_ntoa(new\_address.sin\_addr));

                    return 1;

                }

                else

                {

                    printf("Client %d: %s\n",ntohs(new\_address.sin\_port),msg);

                    sprintf(buf,"Server: %s\n",msg);

                    send(newSocket, buf, strlen(buf), 0);

                    bzero(buf,sizeof(buf));

                }

            }

        }

    }

    return 0;

}

**CLIENT.C**

#include<stdio.h>

#include<stdlib.h>

#include<unistd.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<string.h>

#define PORT 3554

#define SIZE 256

int main()

{

    int clientSocket;

    struct sockaddr\_in server\_address;

    char msg[SIZE],buf[SIZE];

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

    if(clientSocket<0)

    {

        perror("Can't create Socket..\n");

        return 1;

    }

    printf("Client socket created..\n");

    memset(&server\_address,'\0',sizeof(server\_address));

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

    server\_address.sin\_port = htons(PORT);

    server\_address.sin\_addr.s\_addr = INADDR\_ANY;

    int c = connect(clientSocket,(struct sockaddr\*)&server\_address,sizeof(server\_address));

    if(c<0)

    {

        perror("Can't connect to the server..\n");

        return 1;

    }

    printf("Connected to the server..\n");

    while(1)

    {

        printf("Enter your message and press enter..\n");

        scanf("%s",buf);

        send(clientSocket,buf,sizeof(buf),0);

        if(strcmp(buf,"exit")==0)

        {

            close(clientSocket);

            printf("Disconnected from the server..\n");

            return 1;

        }

        if(recv(clientSocket,msg,sizeof(msg),0)<0)

        {

            perror("Can't receive message..\n");

            return 1;

        }

        printf("%s\n",msg);

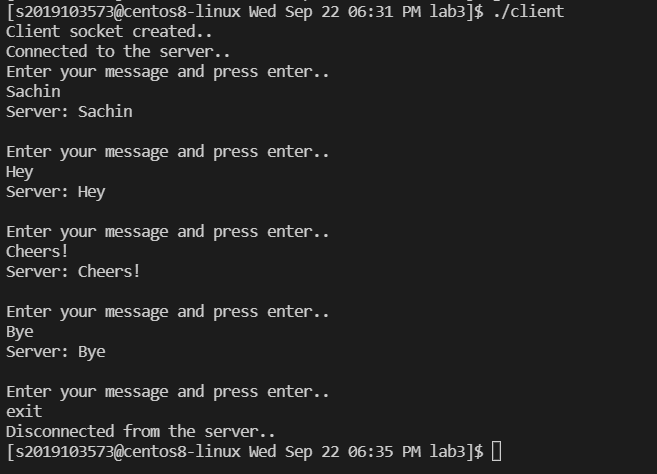
        bzero(msg,sizeof(msg));

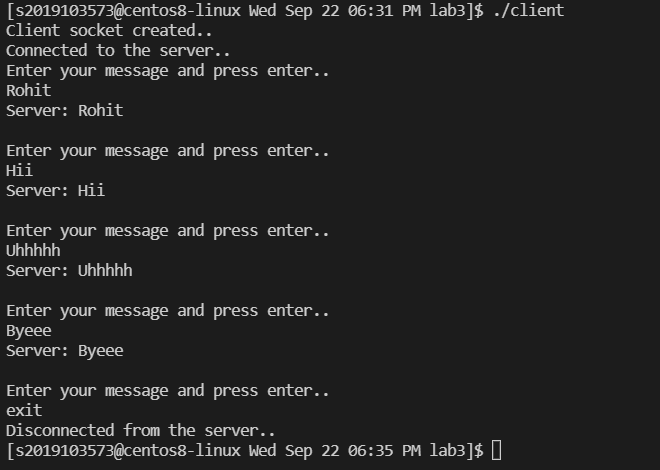
    }

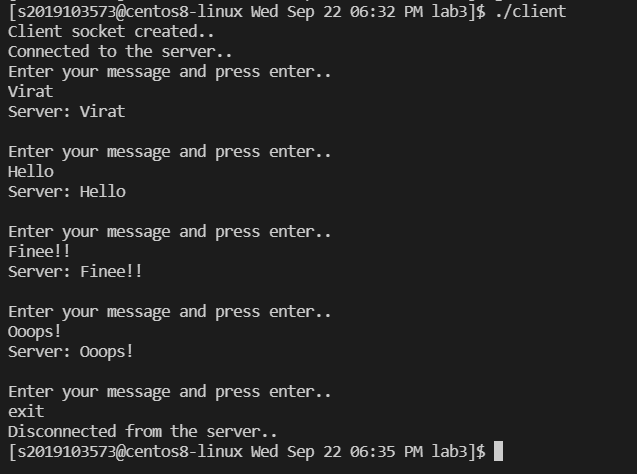
    return 0;

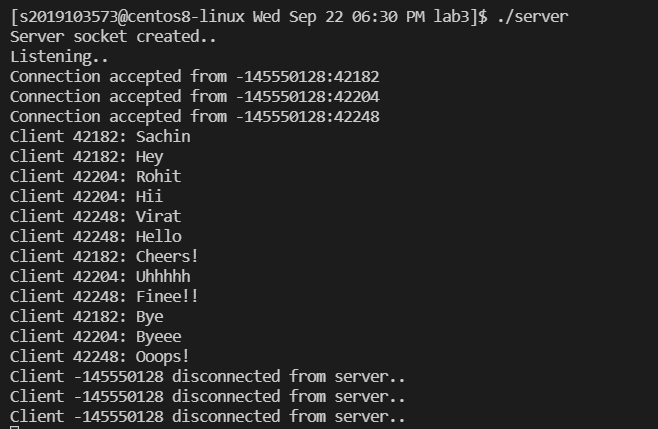
}

**OUTPUT**

**CLIENT 1 :-**

**CLIENT 2 :-**

**CLIENT 3 :-**

**SERVER :-**