

NAME: MOUNVI PODAPATI
REG.NO: 19BCE0396
COURSE CODE: CSE1004
FACULTY: SATHIYA KUMAR C

LAB ASSIGNMENT

Aim: To implement TCP using socket programming:-

SERVER CODE:

```
#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <string.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <arpa/inet.h>

int main(int argc, char * argv[]) {

    int sockfd, newsockfd, portno, n;

    struct sockaddr_in server_address, client_address;

    char buffer[1024];

    socklen_t len;

    if(argc < 2){

        printf("Input format: ./server (port number)");

        exit(1);

    }

    portno = atoi(argv[1]);

    sockfd = socket(AF_INET, SOCK_STREAM, 0);
```

```
if(sockfd<0){
printf("Error in opening socket");
exit(1);
}

server_address.sin_family=AF_INET;
server_address.sin_addr.s_addr=INADDR_ANY;
server_address.sin_port=htons(portno);

if(bind(sockfd,(struct sockaddr *) &server_address,sizeof(server_address))<0)
printf("Error in binding");

listen(sockfd,5);

len=sizeof(client_address);

newsockfd=accept(sockfd,(struct sockaddr *) &client_address,&len);

if(newsockfd<0){
printf("Error in accepting");
exit(1);
}

while(1){
bzero(buffer,1024);
n=read(newsockfd,buffer,1024);
if(n<0){
printf("Error in reading from buffer");
exit(1);
}
printf("Client message:%s",buffer);
```

```

        bzero(buffer,1024);

        fgets(buffer,1024,stdin);
        n=write(newsockfd,buffer,1024);
        if(n<0){
            printf("Error in writing to buffer");
            exit(1);
        }

        if(!strncmp("bye",buffer,3))
        {
            break;
        }
    }

    close(newsockfd);
    close(sockfd);
    return 0;
}

```

CLIENT CODE:

```

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <unistd.h>
#include <netdb.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <netinet/in.h>

int main(int argc, char *argv[])

```

```

{

    int sockfd,portno,n;

    struct sockaddr_in serv_addr;

    char buffer[1024];


    if(argc<3){
        printf("Input format: ./client (loopback adress) (port number)");
        exit(1);
    }


    portno=atoi(argv[2]);
    sockfd=socket(AF_INET,SOCK_STREAM,0);


    if(sockfd<0){
        printf("Error in opening c");
        exit(1);
    }

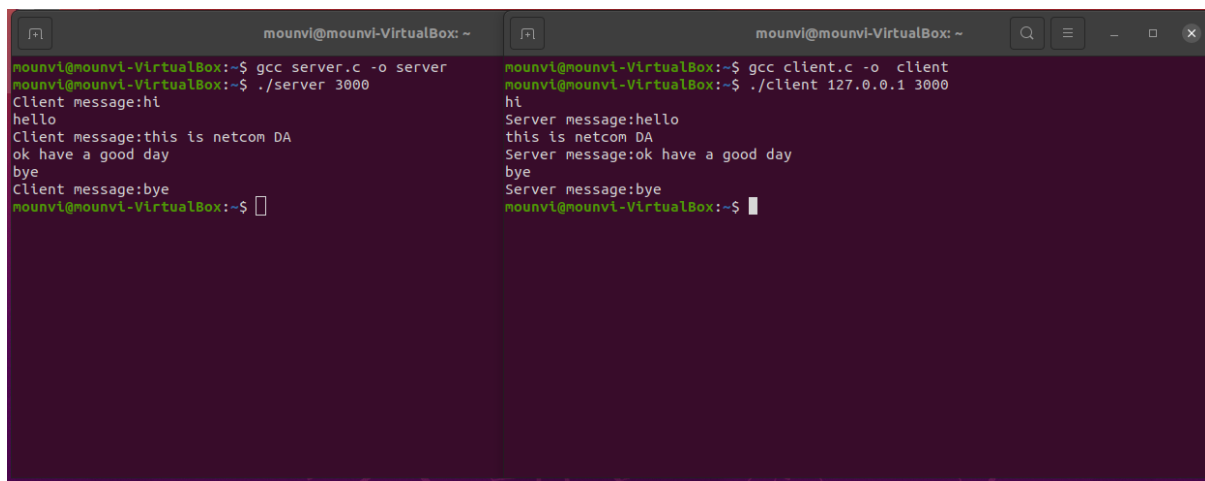

    serv_addr.sin_family=AF_INET;
    serv_addr.sin_port=htons(portno);
    serv_addr.sin_addr.s_addr=inet_addr(argv[1]);


    if(connect(sockfd,(struct sockaddr *) &serv_addr,sizeof(serv_addr))<0){
        printf("Connection failed");
        exit(1);
    }
    while(1){
        bzero(buffer,1024);
        fgets(buffer,1024,stdin);
        n=write(sockfd,buffer,strlen(buffer));
        if(n<0){

```

```
    printf("Error in writing on buffer");  
    exit(1);  
}  
bzero(buffer,1024);  
n=read(sockfd,buffer,1024);  
if(n<0){  
    printf("error in reading from buffer");  
    exit(1);  
}  
printf("Server message:%s",buffer);  
if(!strncmp("bye",buffer,3))  
    break;  
}  
close(sockfd);  
return 0;  
}
```

OUTPUT:



The image shows two terminal windows side-by-side, both titled 'mounvi@mounvi-VirtualBox: ~'. The left window shows the compilation and execution of a server program. The right window shows the compilation and execution of a client program. Both programs are written in C and use sockets for communication.

```
mounvi@mounvi-VirtualBox: ~  
mounvi@mounvi-VirtualBox:~$ gcc server.c -o server  
mounvi@mounvi-VirtualBox:~$ ./server 3000  
Client message:hi  
hello  
Client message:this is netcom DA  
ok have a good day  
bye  
Client message:bye  
mounvi@mounvi-VirtualBox:~$
```

```
mounvi@mounvi-VirtualBox: ~  
mounvi@mounvi-VirtualBox:~$ gcc client.c -o client  
mounvi@mounvi-VirtualBox:~$ ./client 127.0.0.1 3000  
hi  
Server message:hello  
this is netcom DA  
Server message:ok have a good day  
bye  
Server message:bye  
mounvi@mounvi-VirtualBox:~$
```