

19CSE301

## Computer Networks Lab

### Lab Sheet 5 - TCP Socket Programming

*S Abhishek*

*AM.EN.U4CSE19147*

**Implement a Client-Server chat application using TCP.**

Steps for execution

- Save clients and server programs as two separate C files
- Open two terminals
- Compile the server program first and then execute it in one terminal.
- Compile the Client program second and then execute it in second terminal.
- Send the message from the program in which the write statement is given first.

Requirement

- Client and Server should continuously chat till the client sends a terminating character/message to server.

## Server.C

```
#include<sys/types.h>
```

```
#include<sys/socket.h>
```

```
#include<string.h>
```

```
#include<arpa/inet.h>
```

```
#include<netinet/in.h>
```

```
#include<unistd.h>
```

```
#include<stdlib.h>
```

```
#include<stdio.h>
```

```
int main()
```

```
{
```

```
    struct sockaddr_in serv_addr, client_addr;
```

```
    int sockfd, newsockfd, clientlen = sizeof (client_addr), childpid;
```

```
    char msg[100] = {0};
```

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

```
    bzero ((char*)&serv_addr, sizeof (serv_addr));
```

```
    serv_addr.sin_family      = AF_INET;
```

```
    serv_addr.sin_addr.s_addr = inet_addr ("127.0.0.1");
```

```
    serv_addr.sin_port        = htons (2345);
```

```
bind (sockfd,(struct sockaddr*) &serv_addr, sizeof (serv_addr));

listen (sockfd, 5);

for (;;)
{
    clientlen = sizeof (client_addr);
    newsockfd = accept (sockfd, (struct sockaddr*)&client_addr,
&clientlen);

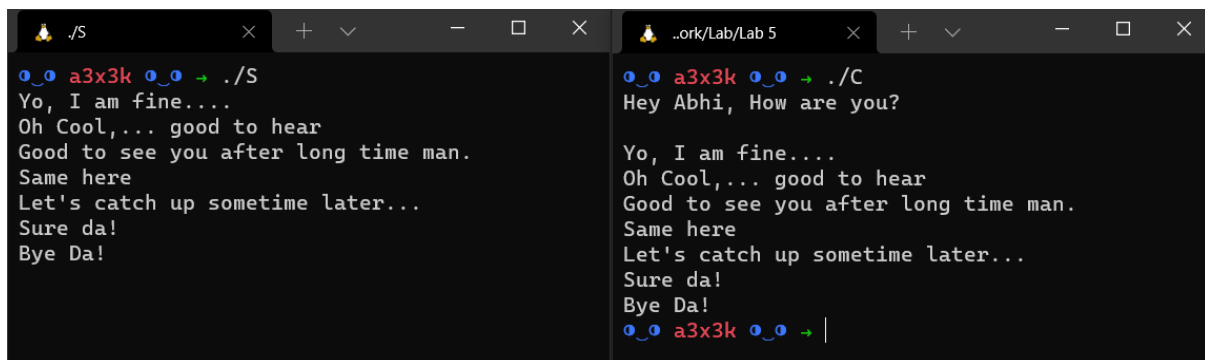
    if ((childpid = fork ()) == 0)
    {
        close (sockfd);
        write (newsockfd, "Hey Abhi, How are you?\n", 100);
        while(1){
            read (newsockfd, msg, 100);
            printf ("%s\n", msg);
            if(strcmp(msg,"Bye Da!")==0){
                break;
                exit (0);
            }
            char data[100];
            scanf("%[^\n]%*c",data);
            write (newsockfd, data, 100);
        }
        exit (0); //free the resources
    }
}
```

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

## **Client.C**

```
#include<netinet/in.h>  
#include<stdlib.h>  
#include<string.h>  
#include<arpa/inet.h>  
#include<sys/socket.h>  
#include<stdio.h>  
#include<unistd.h>  
int main()  
{  
    int sockfd;  
    struct sockaddr_in serv_addr;  
  
    bzero ((char*)&serv_addr, sizeof (serv_addr));  
    serv_addr.sin_family    = AF_INET;  
    serv_addr.sin_addr.s_addr    = inet_addr ("127.0.0.1");  
    serv_addr.sin_port        = htons (2345);  
  
    sockfd = socket (AF_INET, SOCK_STREAM, IPPROTO_TCP);  
  
    connect (sockfd, (struct sockaddr*)&serv_addr, sizeof (serv_addr));
```

```
while(1){  
    char data[100] = {0};  
    read (sockfd, data, 100);  
    printf ("%s\n", data);  
    char msg[100];  
    scanf("%[^\\n]%*c",msg);  
    write (sockfd, msg, 100);  
    if(strcmp(msg,"Bye Da!")==0){  
        break;  
    }  
}  
close (sockfd);  
exit (0);  
return 0;  
}
```



The image shows two terminal windows side-by-side. The left window, titled `./S`, shows the output of the program: `Yo, I am fine....`, `Oh Cool,... good to hear`, `Good to see you after long time man.`, `Same here`, `Let's catch up sometime later...`, `Sure da!`, and `Bye Da!`. The right window, titled `..ork/Lab/Lab 5`, shows the input to the program: `Hey Abhi, How are you?`, followed by the same output as the left window: `Yo, I am fine....`, `Oh Cool,... good to hear`, `Good to see you after long time man.`, `Same here`, `Let's catch up sometime later...`, `Sure da!`, and `Bye Da!`. Both windows have a dark background and light-colored text.

*Thankyou!!*