

Ex7-Full Duplex Communication Using TCP/IP

MAYANK SINHA RA191100301086

SERVER CODE:

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

#include <sys/socket.h>

#include <netinet/in.h>

#include <arpa/inet.h>

#include <netdb.h>

#include <stdio.h>

#include <unistd.h>

#include <strings.h>

#include <string.h>

#define MAX_MSG 100

#define SERVER_ADDR "127.0.0.1"

#define SERVER_PORT 3786


main ( ) {

    int sd, newSd, cliLen, n,num1,num2,sum;

    struct sockaddr_in cliAddr, servAddr;
```

```
char line[MAX_MSG],line1[MAX_MSG],line2[MAX_MSG];

bzero((char *)&servAddr, sizeof(servAddr));

servAddr.sin_family = AF_INET;

servAddr.sin_addr.s_addr = inet_addr(SERVER_ADDR);

servAddr.sin_port = htons(SERVER_PORT);

sd = socket(AF_INET, SOCK_STREAM, 0);

printf("successfully created stream socket \n");

bind(sd, (struct sockaddr *) &servAddr, sizeof(servAddr));

printf("bound local port successfully\n");

listen(sd,5);


while(1) {

    printf("waiting for client connection on port TCP %u\n",SERVER_PORT);

    cliLen = sizeof(cliAddr);

    newSd = accept(sd, (struct sockaddr *) &cliAddr, &cliLen);

    printf("received connection from host [IP %s ,TCP port %d]\n",

           inet_ntoa(cliAddr.sin_addr), ntohs(cliAddr.sin_port));

    do{

        memset(line,0x0,MAX_MSG);
```

```
n=recv(newSd, line, MAX_MSG, 0);
```

```
num1=atoi(line);
```

```
n=recv(newSd, line, MAX_MSG, 0);
```

```
num2=atoi(line);
```

```
sum=num1+num2;
```

```
sprintf(line1,"%d",sum);
```

```
printf("received from host [IP %s ,TCP port %d] : %s\n",
```

```
    inet_ntoa(cliAddr.sin_addr), ntohs(cliAddr.sin_port), line1);
```

```
send(newSd, line1, strlen(line1) + 1, 0);
```

```
}while(abs(strcmp(line, "quit")));
```

```
printf("closing connection with host [IP %s ,TCP port %d]\n",
```

```
    inet_ntoa(cliAddr.sin_addr), ntohs(cliAddr.sin_port));
```

```
close(newSd);
```

```
}  
  
}
```

CLIENT CODE:

```
#include <sys/types.h>  
  
#include <sys/socket.h>  
  
#include <netinet/in.h>  
  
#include <arpa/inet.h>  
  
#include <netdb.h>  
  
#include <stdio.h>  
  
#include <unistd.h>  
  
#include <string.h>  
  
#include <strings.h>  
  
#define MAX_MSG 100  
  
#define SERVER_ADDR "127.0.0.1"  
  
#define CLIENT_ADDR "127.0.0.1"  
  
#define SERVER_PORT 3786  
  
#define CLIENT_PORT 8229  
  
main () {  
  
    int sd, rc, i,n;  
  
    struct sockaddr_in clientAddr, servAddr;  
  
    char line[MAX_MSG];  
  
    bzero((char *)&servAddr, sizeof(servAddr));
```

```
servAddr.sin_family = AF_INET;

servAddr.sin_addr.s_addr = inet_addr(SERVER_ADDR);

servAddr.sin_port = htons(SERVER_PORT);
bzero((char *)&clientAddr, sizeof(clientAddr));

clientAddr.sin_family = AF_INET;

clientAddr.sin_addr.s_addr = INADDR_ANY;

clientAddr.sin_port = htons(0);

sd = socket(AF_INET, SOCK_STREAM, 0);

printf("successfully created stream socket \n");
bind(sd, (struct sockaddr *) &clientAddr, sizeof(clientAddr));

printf("bound local port successfully\n");
connect(sd, (struct sockaddr *) &servAddr, sizeof(servAddr));

printf("connected to server successfully\n");

do{

    printf("Enter 1st number : ");

    scanf("%s", line);

    send(sd, line, strlen(line) + 1, 0);

    printf("data sent (%s)\n", line);

    printf("Enter 2nd number : ");

    scanf("%s", line);

    send(sd, line, strlen(line) + 1, 0);

    printf("data sent (%s)\n", line);
```

```

n=recv(sd, line, MAX_MSG, 0);

printf("received from server %s\n", line);

}while(strcmp(line, "quit"));

printf("closing connection with the server\n");

close(sd);

}

```

OUTPUT:

The screenshot shows the AWS Cloud9 IDE interface. The main editor displays the code for `client_fullduplex.c`. The code is as follows:

```

82
83
84     n=recv(sd, line, MAX_MSG, 0);
85
86     printf("received from server %s\n", line);
87
88 }while(strcmp(line, "quit"));
89
90
91
92
93
94     printf("closing connection with the server\n");
95
96     close(sd);
97
98 }

```

The output window at the bottom shows the following text:

```

server/386/server_fullduplex.c
server/386/client_fullduplex.c
[Stop] [Run] Command: server/386/client_fullduplex.c Runner: C CWD ENV
Enter 2nd number : 6
data sent (6)
received from server 11
Enter 1st number : 7
data sent (7)
Enter 2nd number : 7
data sent (7)
received from server 14
Enter 1st number : 

```

Ex7-Full Duplex Communication | x Meet - hcv-fqng-pzu x (26) WhatsApp x 18CSC302J Batch 1/DAY 2/HOU x +

us-west-1.console.aws.amazon.com/cloud9/ide/151c4768c25240cdadda0c7bedf1f13c7#

Apps Google Classroom i... Career Developmen... Welcome to smist... GitHub - Ebazhano... willywonka32 (May... Dashboard | Hacker... 18CSC302J Batch 1/... Reading list

File Edit Find View Go Run Tools Window Support Preview Run

Go to Anything (Ctrl-P)

- client1.c
- client1.c.o
- client2.c
- client2.c.o
- client7.c
- client7.c.o
- client_fullduplex.c
- client_fullduplex.c.o
- ex4 server.c
- ex5server.c
- ex5server.c.o
- exp6client.c
- exp6client.c.o
- exp6server.c
- exp6server.c.o
- server.c
- server.c.o
- server1.c
- server1.c.o
- server2.c
- server2.c.o
- server7.c

server_fullduplex.c x client_fullduplex.c x

```
82
83
84     n=recv(sd, line, MAX_MSG, 0);
85
86     printf("received from server %s\n", line);
87
88     }while(strcmp(line, "quit"));
89
90
91
92
93
94     printf("closing connection with the server\n");
95
96     close(sd);
97
98 }
```

98:2 C and C++ Spaces: 4

server386/server_fullduplex.c server386/client_fullduplex.c

Stop Command: server386/server_fullduplex.c Runner: C CWD ENV

```
}while(abs(strcmp(line, "quit")));
successfully created stream socket
bound local port successfully
waiting for client connection on port TCP 3786
received connection from host [IP 127.0.0.1 ,TCP port 57645]
received from host [IP 127.0.0.1 ,TCP port 57645] : 11
received from host [IP 127.0.0.1 ,TCP port 57645] : 14
```

AWS (not connected) 14:23 18-09-2021

30°C Partly sunny