

18CSC302J (Computer Networks Lab)

Lab session - Full Duplex Chat Using TCP/IP

Name :- Puneet Sharma

Reg. No. :- RA1911003010331

Class :-CSE F1

SERVER CODE:

```
#include<sys/types.h>
#include<sys/socket.h>
#include<stdio.h>
#include<unistd.h>
#include<netdb.h>
#include<arpa/inet.h>
#include<netinet/in.h>
#include<string.h>
int main(int argc,char *argv[])
{
int clientSocketDescriptor,socketDescriptor;
struct sockaddr_in serverAddress,clientAddress;
socklen_t clientLength;
char recvBuffer[1000],sendBuffer[1000];
pid_t cpid;
bzero(&serverAddress,sizeof(serverAddress));
serverAddress.sin_family=AF_INET;
serverAddress.sin_addr.s_addr=htonl(INADDR_ANY);
serverAddress.sin_port=htons(5531);
socketDescriptor=socket(AF_INET,SOCK_STREAM,0);
bind(socketDescriptor,(struct sockaddr*)&serverAddress,sizeof(serverAddress));
listen(socketDescriptor,5);
printf("%s\n","Server is running ...");
clientSocketDescriptor=accept(socketDescriptor,(struct sockaddr*)&clientAddress,&clientLength);
cpid=fork();
if(cpid==0)
```

```
{
while(1)
{
bzero(&recvBuffer,sizeof(recvBuffer));
recv(clientSocketDescriptor,recvBuffer,sizeof(recvBuffer),0);
printf("\nCLIENT : %s\n",recvBuffer);
}
}
else
{
while(1)
{
bzero(&sendBuffer,sizeof(sendBuffer));
printf("\nEnter Message: ");
fgets(sendBuffer,10000,stdin);
send(clientSocketDescriptor,sendBuffer,strlen(sendBuffer)+1,0);
printf("\nMessage sent !\n");
}
}
return 0;
}
```

```
Full_duplex_server.c x  Ftp_server.c x  data.txt x +
1  #include<sys/types.h>
2  #include<sys/socket.h>
3  #include<stdio.h>
4  #include<unistd.h>
5  #include<netdb.h>
6  #include<arpa/inet.h>
7  #include<netinet/in.h>
8  #include<string.h>
9  int main(int argc, char *argv[])
10 {
11     int clientSocketDescriptor, socketDescriptor;
12     struct sockaddr_in serverAddress, clientAddress;
13     socklen_t clientLength;
14     char recvBuffer[1000], sendBuffer[1000];
15     pid_t cpid;
16     bzero(&serverAddress, sizeof(serverAddress));
17     serverAddress.sin_family = AF_INET;
18     serverAddress.sin_addr.s_addr = htonl(INADDR_ANY);
19     serverAddress.sin_port = htons(5531);
20     socketDescriptor = socket(AF_INET, SOCK_STREAM, 0);
21     bind(socketDescriptor, (struct sockaddr*)&serverAddress, sizeof(serverAddress));
22     listen(socketDescriptor, 5);
23     printf("%s\n", "Server is running ...");
24     clientSocketDescriptor = accept(socketDescriptor, (struct sockaddr*)&clientAddress, &cli
25     cpid = fork();
26     if(cpid == 0)
27     {
28         while(1)
29         {
30             bzero(&recvBuffer, sizeof(recvBuffer));
31             recv(clientSocketDescriptor, recvBuffer, sizeof(recvBuffer), 0);
32             printf("\nCLIENT : %s\n", recvBuffer);
33         }
34     }
35     else
```

```
331/Full_duplex_ser 331/Ftp_server.c - S x +
Stop Run Config Com 331/Full_duplex_server.c Runner: C CWD ENV
Enter Message: Puneet
Message sent !
Enter Message:
CLIENT : 331
Hello
Message sent !
Enter Message:
CLIENT : bye

```

CLIENT CODE

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

int main()
{
    int socketDescriptor;
    struct sockaddr_in serverAddress;
    char sendBuffer[1000],recvBuffer[1000];
    pid_t cpid;

    bzero(&serverAddress,sizeof(serverAddress));
    serverAddress.sin_family=AF_INET;
    serverAddress.sin_addr.s_addr=inet_addr("127.0.0.1");
    serverAddress.sin_port=htons(5531);
    socketDescriptor=socket(AF_INET,SOCK_STREAM,0);
    connect(socketDescriptor,(struct sockaddr*)&serverAddress,sizeof(serverAddress));
    cpid=fork();
    if(cpid==0)
    {
        while(1)
        {
            bzero(&sendBuffer,sizeof(sendBuffer));
            printf("\nEnter Message: ");
            fgets(sendBuffer,10000,stdin);
            send(socketDescriptor,sendBuffer,strlen(sendBuffer)+1,0);
            printf("\nMessage sent !\n");
        }
    }
}
```

```

else
{
while(1)
{
bzero(&recvBuffer,sizeof(recvBuffer));
recv(socketDescriptor,recvBuffer,sizeof(recvBuffer),0);
printf("\nSERVER : %s\n",recvBuffer);
}
}
return 0;
}

```

```

Full_duplex_clint.c  x  Ftp_clint.c  x  +
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #include <sys/types.h>
5  #include <sys/socket.h>
6  #include <netinet/in.h>
7  #include <unistd.h>
8  #include <netdb.h>
9  #include <arpa/inet.h>
10 int main()
11 {
12     int socketDescriptor;
13     struct sockaddr_in serverAddress;
14     char sendBuffer[1000],recvBuffer[1000];
15     pid_t cpid;
16     bzero(&serverAddress,sizeof(serverAddress));
17     serverAddress.sin_family=AF_INET;
18     serverAddress.sin_addr.s_addr=inet_addr("127.0.0.1");
19     serverAddress.sin_port=htons(5531);
20     socketDescriptor=socket(AF_INET,SOCK_STREAM,0);
21     connect(socketDescriptor,(struct sockaddr*)&serverAddress,sizeof(serverAddress));
22     cpid=fork();
23     if(cpid==0)
24     {
25         while(1)
26         {
27             bzero(&sendBuffer,sizeof(sendBuffer));
28             printf("\nEnter Message: ");
29             fgets(sendBuffer,10000,stdin);
30             send(socketDescriptor,sendBuffer,strlen(sendBuffer)+1,0);
31             printf("\nMessage sent !\n");
32         }
33     }
34     else
35     {

```

6:24 C and C++ Spaces: 4

```
331/Full_duplex_clin 331/Ftp_clint.c - Stoj x +
Stop Run Config Com 331/Full_duplex_clint.c Runner: C CWD ENV

Enter Message:
SERVER : Puneet

331

Message sent !

Enter Message:
SERVER : Hello

bye

Message sent !

Enter Message: 
```

```
AWS Cloud9 File Edit Find View Go Run Tools Window Support Preview Run Last
Go to Anything (Ctrl-P)
Full_duplex_server.c Ftp_server.c data.txt Full_duplex_clint.c Ftp_clint.c
#include <sys/types.h>
#include <sys/socket.h>
#include <stdio.h>
#include <unistd.h>
#include <netdb.h>
#include <arpa/inet.h>
#include <netinet/in.h>
#include <string.h>
int main(int argc, char *argv[])
{
    int clientSocketDescriptor, socketDescriptor;
    struct sockaddr_in serverAddress, clientAddress;
    socklen_t clientLength;
    char recvBuffer[1000], sendBuffer[1000];
    pid_t cpid;
    bzero(&serverAddress, sizeof(serverAddress));
    serverAddress.sin_family = AF_INET;
    serverAddress.sin_addr.s_addr = htonl(INADDR_ANY);
    serverAddress.sin_port = htons(5531);
    socketDescriptor = socket(AF_INET, SOCK_STREAM, 0);
    bind(socketDescriptor, (struct sockaddr*)&serverAddress, sizeof(serverAddress));
    listen(socketDescriptor, 5);
    printf("Server is running...\n");
    clientSocketDescriptor = accept(socketDescriptor, (struct sockaddr*)&clientAddress, &cli-
    cpid = fork();
    if(cpid == 0)
    {
        1:22 C and C++ Spaces: 4

331/Full_duplex_ser 331/Ftp_server.c - S x +
Stop Run Config Com 331/Full_duplex_server.c Runner: C CWD ENV
Enter Message: 331
Message sent !
Enter Message:
CLIENT : 331
CLIENT : puneet
Enter Message:

331/Full_duplex_clin 331/Ftp_clint.c - Stoj x +
Stop Run Config Com 331/Full_duplex_clint.c Runner: C CWD ENV
SERVER : 331
331
Message sent !
Enter Message: puneet
Message sent !
Enter Message: 
```