18CSC302J (Computer Networks Lab) Lab session - Full Duplex Chat Using TCP/IP

Name:- Puneet Sharma

Reg. No.:- RA1911003010331

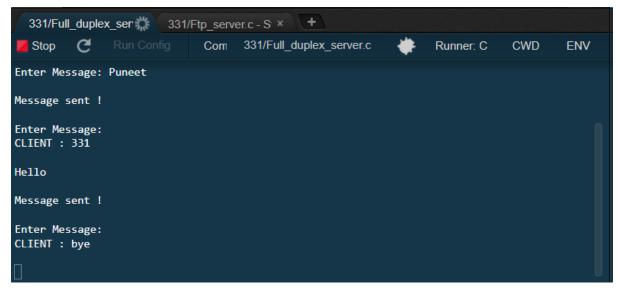
Class:-CSE F1

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 ×
                     Ftp_server.c
                                            data.txt
   #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;
   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,&cli
   cpid=fork();
   if(cpid==0)
   while(1)
   bzero(&recvBuffer,sizeof(recvBuffer));
   recv(clientSocketDescriptor,recvBuffer,sizeof(recvBuffer),0);
   printf("\nCLIENT : %s\n",recvBuffer);
                                                          8:19 C and C++ Spaces: 4
   else
```



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");
server Address. 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 ×
                      Ftp_clint.c
   #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");
                                                             6:24 C and C++ Spaces: 4
```

