DEPARTMENT OF COMPUTER & SOFTWARE ENGINEERING

COLLEGE OF E&ME, NUST, RAWALPINDI

Microprocessor and Microcontroller Based Design

Lab 01

SUBMITTED TO:
Dr Taimoor Zahid

SUBMITTED BY:
Muawiz Umer
Reg # 00000335806
DE-42 (C&SE)-A

Submission Date: 24/10/2022

Tasks:
CODE:
SERVER.CPP

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

#include <sys/socket.h>
#include <netinet/in.h>

#define SERVER_PORT_NO 80
 using namespace
std;
 int main()
{
     int fd = socket(AF_INET, SOCK_STREAM,
0);     if (fd == -1) {
perror("Socket Creation failed\n");
return -1;
    }

    struct sockaddr_in addr;
     addr.sin_addr.s_addr    = INADDR_ANY;
addr.sin_family = AF_INET;     addr.sin_port
= htons(SERVER_PORT_NO);
     if (bind(fd, (struct sockaddr*) &addr, sizeof(addr)) == -1)
{         perror("Bind failed on socket\n");         return -1;
    }

    int backlog = 10;
    if (listen(fd, backlog) == -1) {
perror("Listen Failed on server: \n");
return -1;
    }

    int connfd;     struct sockaddr_in cliaddr;     socklen_t
cliaddr_len;     connfd = accept(fd, (struct sockaddr *) &cliaddr,
&cliaddr_len);
```

```
    if (connfd <=0 ) {
perror("accept failed on socket: ");
    }          string
send_data = "";     char
buffer[100];


    while(send_data != "1"){
// Recieve Data From Client
bzero(buffer, 100);
        recv(connfd, buffer, 100, 0);
                int len_in_data =
strlen(buffer);         send_data = "";
for(int i = 0; i < len_in_data; i++){
send_data += buffer[i];
        }         if(send_data == "1"){
cout << "Client left !!!\n";
break;
        }
         cout<<"Client: "<< buffer <<
endl;         cout << "Me:     ";

        // Return Msg
        getline(cin,send_data);

        // Check length
        int len = send_data.length();
// Convert string to char         char
buff[len +1];
        strcpy(buff, send_data.c_str());
        // Send data         send(connfd,
buff, strlen(buff), 0);

        if(send_data == "1"){
break;
        }
    }     cout<<"Server
Exiting!\n";
```

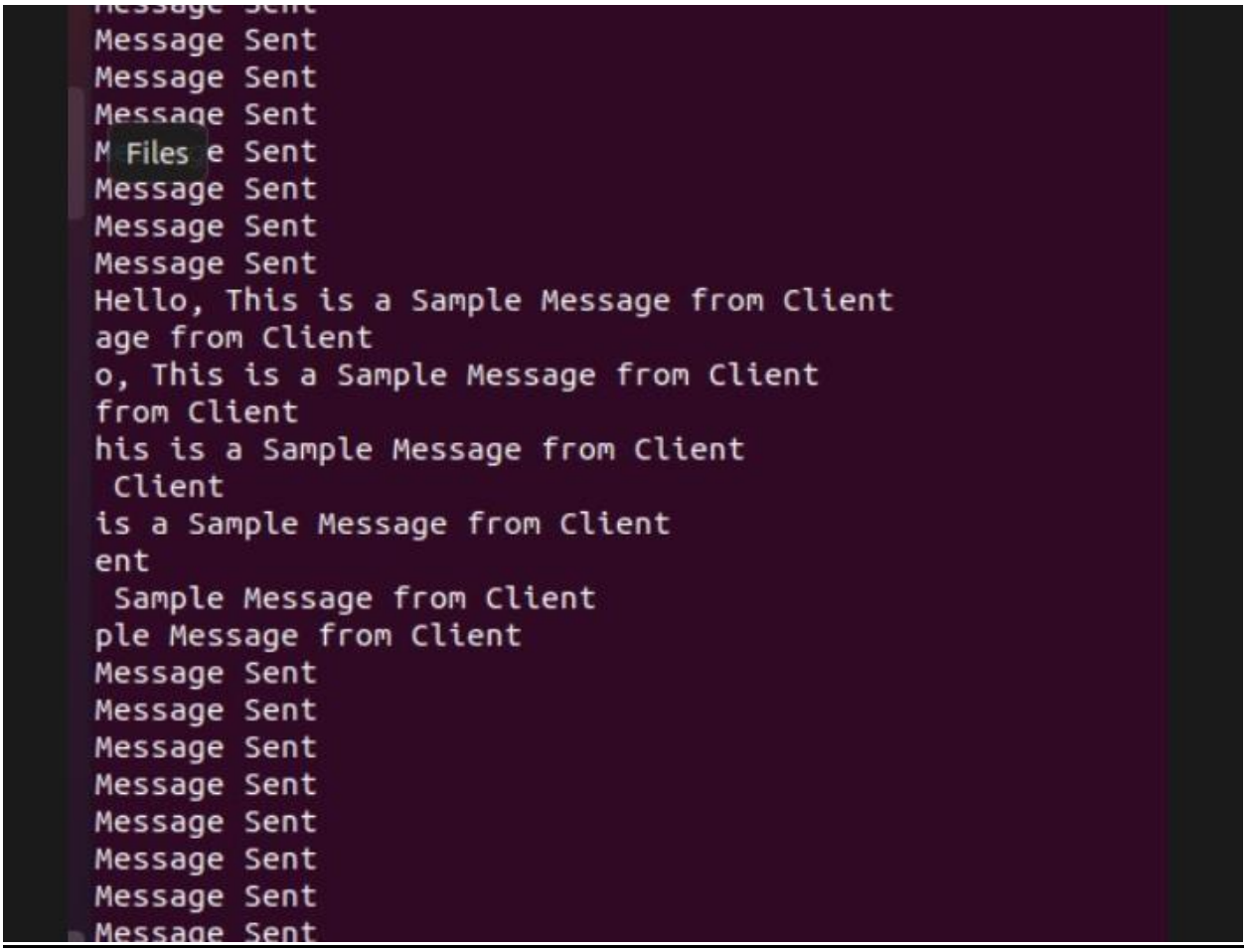
```
    shutdown(fd,SHUT_WR);
shutdown(connfd,SHUT_RDWR);     return
0;
}
```

CLIENT.CPP

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

#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>

#define SERVER_PORT_NO 80
 using namespace
std;
 int main()
{
     int fd = socket(AF_INET, SOCK_STREAM,
0);     if (fd == -1) {
perror("Socket Creation failed\n");
return -1;
    }

    struct sockaddr_in s_addr;
     s_addr.sin_family   =
AF_INET;     s_addr.sin_port =
htons(80);
    inet_aton("127.0.0.1", &s_addr.sin_addr);
    if (connect(fd, (struct sockaddr *) &s_addr, sizeof(s_addr)) == -1)
{         perror("Socket Connect failed\n");         return -1;
    }      string
send_data;     char
buff[100];     int
len_income_data;
while(send_data != "1")
    {
        // Send data to Server
cout << "Me:     ";
```

```
        // Take Input from User
getline(cin, send_data);

        // Convert String to char data
type          int n = send_data.length();
char buffer[ n + 1];
strcpy(buffer, send_data.c_str());

        // Now send to server
send(fd, buffer, strlen(buffer), 0);

        // Check if you want to break
loop          if(send_data == "1"){
break;
        }

        // Start recieving data;
bzero(buff, 100);          recv(fd,
buff, 100, 0);
         send_data =
"";
        // Check length of characters
len_income_data = strlen(buff);

        // Store in String          for (int i =
0; i < len_income_data; i++)
        {              send_data
+= buff[i];
        }
        // Check if server left or not
if(send_data == "1")
        {
            cout << "Server left!!! \n";
break;
        }

        // Display server msg
        cout << "Server: " << buff << endl;
    }     cout<<"Client
Exiting!"<<endl;
shutdown(fd,SHUT_RDWR);
return 0;
}
```

OUTPUT:


Message Sent
Message Sent
Message Sent
Files
Message Sent
Message Sent
Message Sent
Hello, This is a Sample Message from Client
age from Client
o, This is a Sample Message from Client
from Client
his is a Sample Message from Client
Client
is a Sample Message from Client
ent
Sample Message from Client
ple Message from Client
Message Sent
Message Sent
Message Sent
Message Sent
Message Sent
Message Sent
Message Sent
Message Sent

```
 Sample Message from Server
ple Message from Server
Message from Server
Hello, This is a Sample Message from Server
age from Server
o, This is a Sample Message from Server
from Server
his is a Sample Message from Server
 Server
is a Sample Message from Server
ver
 Sample Message from Server
ple Message from Server
Message Sent


 Sent
Message Sent


 Sent
Message Sent


 Sent
```