NATIONAL INSTITUTE OF TECHNOLOGY SILCHAR

Cachar, Assam

B.Tech. Vth Sem

Subject Code: CS-311

Subject Name: Computer Network Laboratory

Submitted By:

Name : Subhojit Ghimire

Sch. Id. : 1912160

Branch : CSE – B

Q.5. Write a program for "Connectionless Iterative Service" in which the server finds the factorial of a number sent by the client and sends it back.

[redacted]

AIM: TO IMPLEMENT "CONNECTIONLESS ITERATIVE SERVICE" TO FIND THE FACTORIAL OF A NUMBER SENT BY THE CLIENT USING UDP IN CPP.

THEORY: 1. UDP CLIENT SERVER: In UDP, the client does not form a connection with the server (hence, connectionless) like in TCP. Instead, the client just sends a datagram. Similarly, the server does not need to accept a connection and just waits for datagrams to arrive. Datagrams upon arrival contain the address of sender which the server uses to send data to the correct client.

2. CONNECTIONLESS ITERATIVE SERVER: In this model, the server receives a request packet from UDP, processes the request and gives response to the UDP to send to the client. The packets are stored in queue and processed in order of arrival.

3. CONNECTIONLESS SERVICE: It is a technique that is used in data communications to send or transfer data or message at layer 4, i.e., Transport layer of Open System Interconnection model.

4. ITERATIVE SERVER: An iterative server processes request from clients in a serial manner; one connection is served and responded to before the server accepts a new client connection.

The entire operation can be broken down as follows:

CONNECTIONLESS ITERATIVE SERVICE SERVER:

(i) A socket is created and binded to an advertised port number.

(ii) An infinite loop is started to process the client requests for connections.

(iii) The process receives a number from the client using recvfrom() function and calculates the factorial for the received number, and sends it back to the client using sendto() function.

CONNECTIONLESS ITERATIVE SERVICE CLIENT:

(i) A socket is created and binded.

(ii) The number, whose factorial is to be found, is sent as a message from the user (client) using sendto() function.

(iii) The factorial value from the server is received using recvfrom() function and displayed.

CODE :

```
// CIS SERVER

# include <iostream>
# include <cstdlib.>
# include <cstring>
# include <unistd.h>
# include <sys/types.h>
# include <sys/socket.h>
# include <arpa/inet.h>
# include <netinet/in.h>

# define PORT 8080
# define MAXLINE 1024
char fact[MAXLINE];

using namespace std;

int multiply (int xx, int res[], int resSize) {
        int carry = 0;
        for (int ii = 0; ii < resSize; ++ii) {
            int prod = res[ii] * xx + carry;
            res[ii] = prod % 10;
            carry = prod / 10;
        }
```

```
        while (carry) {
                res [resSize] = carry % 10;
                carry = carry / 10;
                ++resSize;
        }
        return resSize;
}

void factorial (int number) {
        int res [MAXLINE];
        char numToTxt [5];
        res [0] = 1;
        int resSize = 1;
        for (int xx = 2; xx <= number; ++xx)
                resSize = multiply (xx, res, resSize);
        for (int ii = resSize - 1; ii >= 0; --ii) {
                sprintf (numToTxt, "%d", res [ii]);
                strcat (fact, numToTxt);
        }
}

int main () {
        int sockfd;
        struct sockaddr_in servaddr, cliaddr;

        if ((sockfd = socket (AF_INET, SOCK_DGRAM, 0)) < 0) {
                perror ("FAIL: SOCKET CREATION \n");
                exit (EXIT_FAILURE);
        }
```

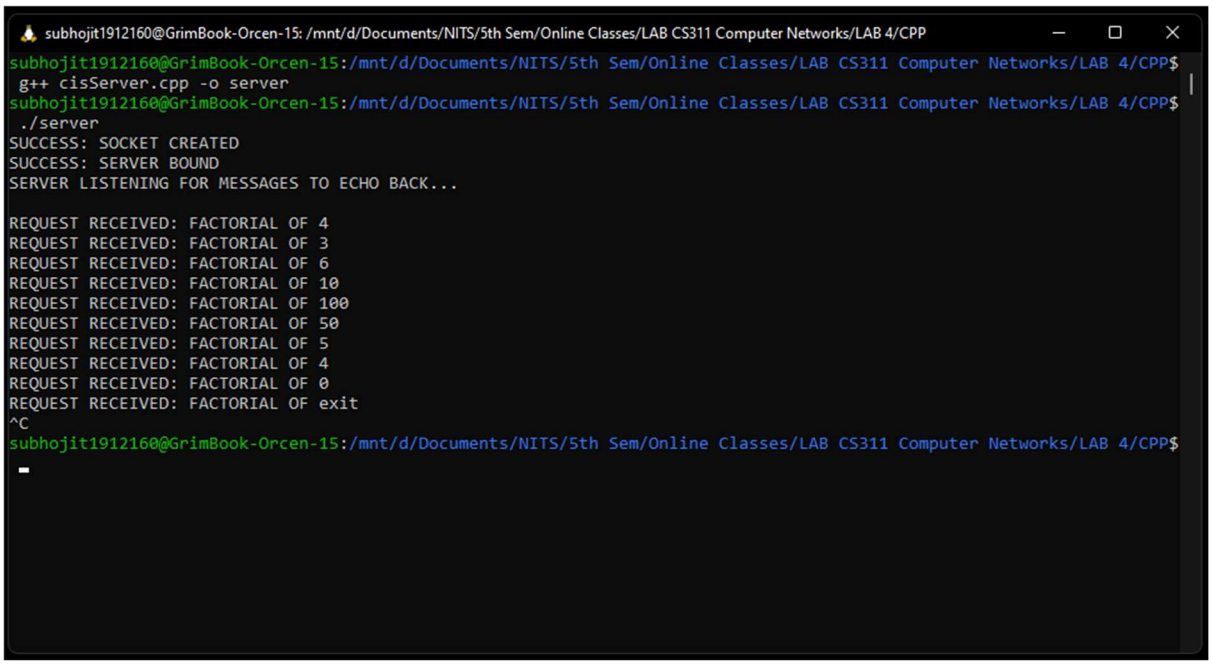
```
cout << " SUCCESS: SOCKET CREATED \n" ;

memset (& servaddr , 0 , sizeof (servaddr));
memset (& cliaddr , 0 , sizeof ( cliaddr ));

servaddr. sin_family = AF_INET ;
servaddr. sin_addr. s_addr = INADDR_ANY;
servaddr. sin_port = htons (PORT) ;

if (bind (sockfd, (const struct sockaddr *) & servaddr,
                    sizeof (servaddr)) < 0 ) {
        perror ("FAIL: SERVER BIND \n");
        exit (EXIT_FAILURE);
}
cout << "SUCCESS: SERVER BOUND \n" ;

cout <<" SERVER LISTENING FOR MESSAGES... \n\n";
char buffer [MAXLINE];
unsigned int len , nn;
len = sizeof (cliaddr) ;
while (1) {
        memset (buffer , 0 , MAXLINE) ;
        bzero (fact , MAXLINE);
        nn = recvfrom (sockfd, (char *) buffer,
                MAXLINE, MSG_WAITALL, (struct sockaddr*)
                & cliaddr , & len);
        buffer [nn] = '\0' ;
        cout << " REQUEST RECEIVED : FACTORIAL of " << buffer
        int num = atoi (buffer) ;
```

```
            factorial (num);
            bzero (buffer, MAXLINE);
            strcpy (buffer, fact);
            sendto (sockfd, (const char *) buffer, strlen (buffer),
                    MSG_CONFIRM, (const struct sockaddr *) &cliaddr,
                    len);
        }
        close (sockfd);
        return 0;
}

// CIS CLIENT

# include <iostream>
# include <cstdlib>
# include <unistd.h>
# include <cstring>
# include <sys/types.h>
# include <sys/socket.h>
# include <arpa/inet.h>
# include <netinet/in.h>

# define PORT 8080
# define MAXLINE 1024

using namespace std;

int main () {
        int sockfd;
```

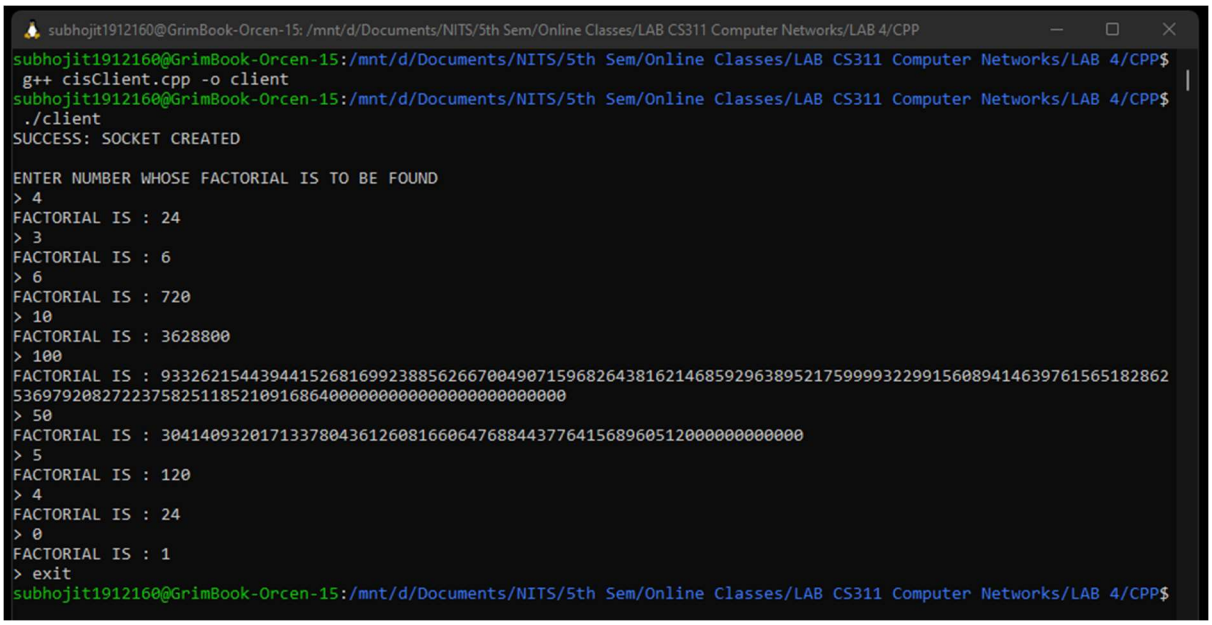
```
struct sockaddr_in servaddr;
if ((sockfd = socket (AF_INET, SOCK_DGRAM, 0)) < 0) {
        perror ("FAIL: SOCKET CREATION \n");
        exit (EXIT_FAILURE);
}
cout << "SUCCESS: SOCKET CREATED \n\n";
memset (&servaddr, 0, sizeof (servaddr));
servaddr.sin_family = AF_INET;
servaddr.sin_port = htons (PORT);
servaddr.sin_addr.s_addr = INADDR_ANY;
unsigned int nn, len;
char buffer [MAXLINE], msg[MAXLINE];
cout << " ENTER NUMBER WHOSE FACTORIAL IS TO BE FOUND ";
while (1) {
    memset (msg, 0, MAXLINE);
    memset (buffer, 0, MAXLINE);
    cout << "\n> ";
    fgets (msg, MAXLINE, stdin);
    sendto (sockfd, (const char *)msg, strlen (msg), MSG_CONFIRM,
            (const struct sockaddr *)&servaddr, sizeof (servaddr));
    nn = recvform (sockfd, (char *)buffer, MAXLINE, MSG_WAITALL,
         (struct sockaddr *) &servaddr, &len);
    buffer [nn] = '\0';
    if (strncmp (msg, "exit", 4) == 0)
        break;
    cout << "FACTORIAL IS: " <<buffer;
}
close (sockfd);
return 0;
}
```

OUTPUT:

// CONNECTIONLESS ITERATIVE SERVICE SERVER



// CONNECTIONLESS ITERATIVE SERVICE CLIENT



Output Explanation:

Firstly, the CIS Server is compiled and run, making it open to receive all types of connections. Then the CIS Client is compiled and run, connecting it to the CIS Server. Then, in the Client Terminal, the number for which the factorial is to be found is entered. The Server Terminal receives the request and calculates the factorial for the requested number. After the factorial is calculated, the server echoes back the calculated value (which is the factorial value) to the client, which is then displayed on the Client Terminal.