

Name: Preyash

Date: 10-02-2022

Registration Number: 20BPS1022

## **LAB-06**

### **UDP Socket Programming**

**AIM:** To compute the sum of the first 'n' even numbers, write a UDP socket application and run it. The server computes the sum using the 'n' value provided by the client, and the result should be shown on the client side.

#### **ALGORITHM**

##### **Server-side Algorithm:**

- ✓ Start
- ✓ Include the `sys/socket.h` and `arpa/inet.h` header files.
- ✓ Using the `socket()` method, create a socket that returns a socket descriptor.
- ✓ Set the server address based on the port and IP.
- ✓ Bind the socket to the server address with `bind()`.
- ✓ Using the `recvfrom()` method, get the value of "n" from the client. Then, using  $n*(n+1)$ , calculate the value of the sum of "n" even integers.
- ✓ Send the result back to the client.
- ✓ End.

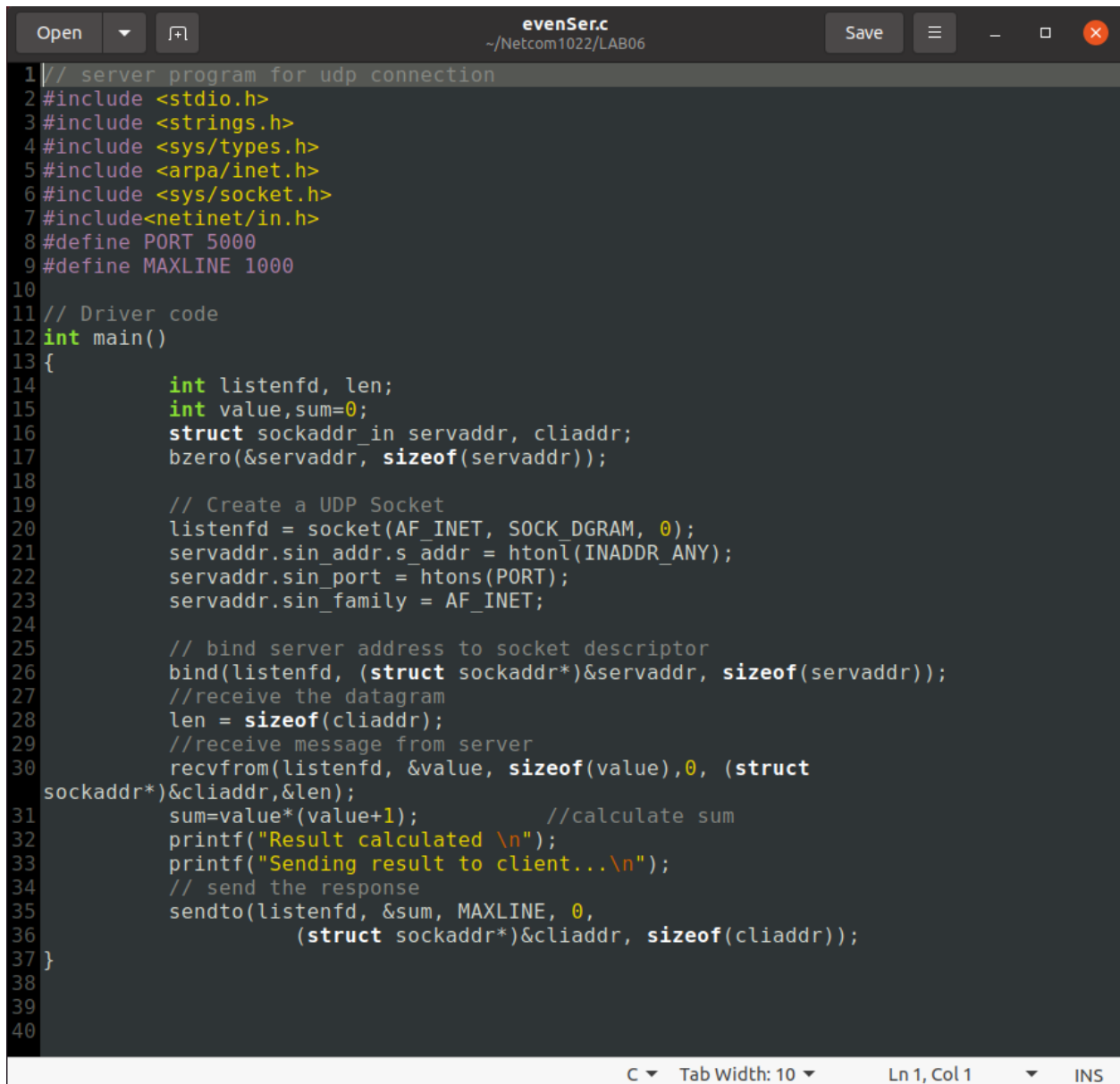
##### **Client-Side Algorithm:**

- ✓ Start
- ✓ Include the `sys/socket.h` and `arpa/inet.h` header files.
- ✓ Create a datagram socket and use the `connect()` function to connect to the server.
- ✓ Take the user's value for "n."
- ✓ If you're connected, send the value to the server.

- ✓ Using the recvfrom() method, wait for a response from the server.
- ✓ Obtain the server's result and print it.
- ✓ To cease the conversation, close the socket.
- ✓

### Server Program Source Code:

Code window:



```
1 // server program for udp connection
2 #include <stdio.h>
3 #include <strings.h>
4 #include <sys/types.h>
5 #include <arpa/inet.h>
6 #include <sys/socket.h>
7 #include <netinet/in.h>
8 #define PORT 5000
9 #define MAXLINE 1000
10
11 // Driver code
12 int main()
13 {
14     int listenfd, len;
15     int value, sum=0;
16     struct sockaddr_in servaddr, cliaddr;
17     bzero(&servaddr, sizeof(servaddr));
18
19     // Create a UDP Socket
20     listenfd = socket(AF_INET, SOCK_DGRAM, 0);
21     servaddr.sin_addr.s_addr = htonl(INADDR_ANY);
22     servaddr.sin_port = htons(PORT);
23     servaddr.sin_family = AF_INET;
24
25     // bind server address to socket descriptor
26     bind(listenfd, (struct sockaddr*)&servaddr, sizeof(servaddr));
27     //receive the datagram
28     len = sizeof(cliaddr);
29     //receive message from server
30     recvfrom(listenfd, &value, sizeof(value), 0, (struct
sockaddr*)&cliaddr, &len);
31     sum=value*(value+1); //calculate sum
32     printf("Result calculated \n");
33     printf("Sending result to client...\n");
34     // send the response
35     sendto(listenfd, &sum, MAXLINE, 0,
(struct sockaddr*)&cliaddr, sizeof(cliaddr));
36 }
37
38
39
40
```

C Tab Width: 10 Ln 1, Col 1 INS

**Code:**

```
// server program for udp connection

#include <stdio.h>

#include <strings.h>

#include <sys/types.h>

#include <arpa/inet.h>

#include <sys/socket.h>

#include <netinet/in.h>

#define PORT 5000

#define MAXLINE 1000

// Driver code

int main()

{

    int listenfd, len;

    int value,sum=0;

    struct sockaddr_in servaddr, cliaddr;

    bzero(&servaddr, sizeof(servaddr));

    // Create a UDP Socket

    listenfd = socket(AF_INET, SOCK_DGRAM, 0);

    servaddr.sin_addr.s_addr = htonl(INADDR_ANY);

    servaddr.sin_port = htons(PORT);

    servaddr.sin_family = AF_INET;

    // bind server address to socket descriptor

    bind(listenfd, (struct sockaddr*)&servaddr, sizeof(servaddr));
```

```
//receive the datagram
len = sizeof(cliaddr);

//receive message from server
recvfrom(listenfd, &value, sizeof(value),0, (struct sockaddr*)&cliaddr,&len);

sum=value*(value+1);    //calculate sum

printf("Result calculated \n");

printf("Sending result to client...\n");

// send the response

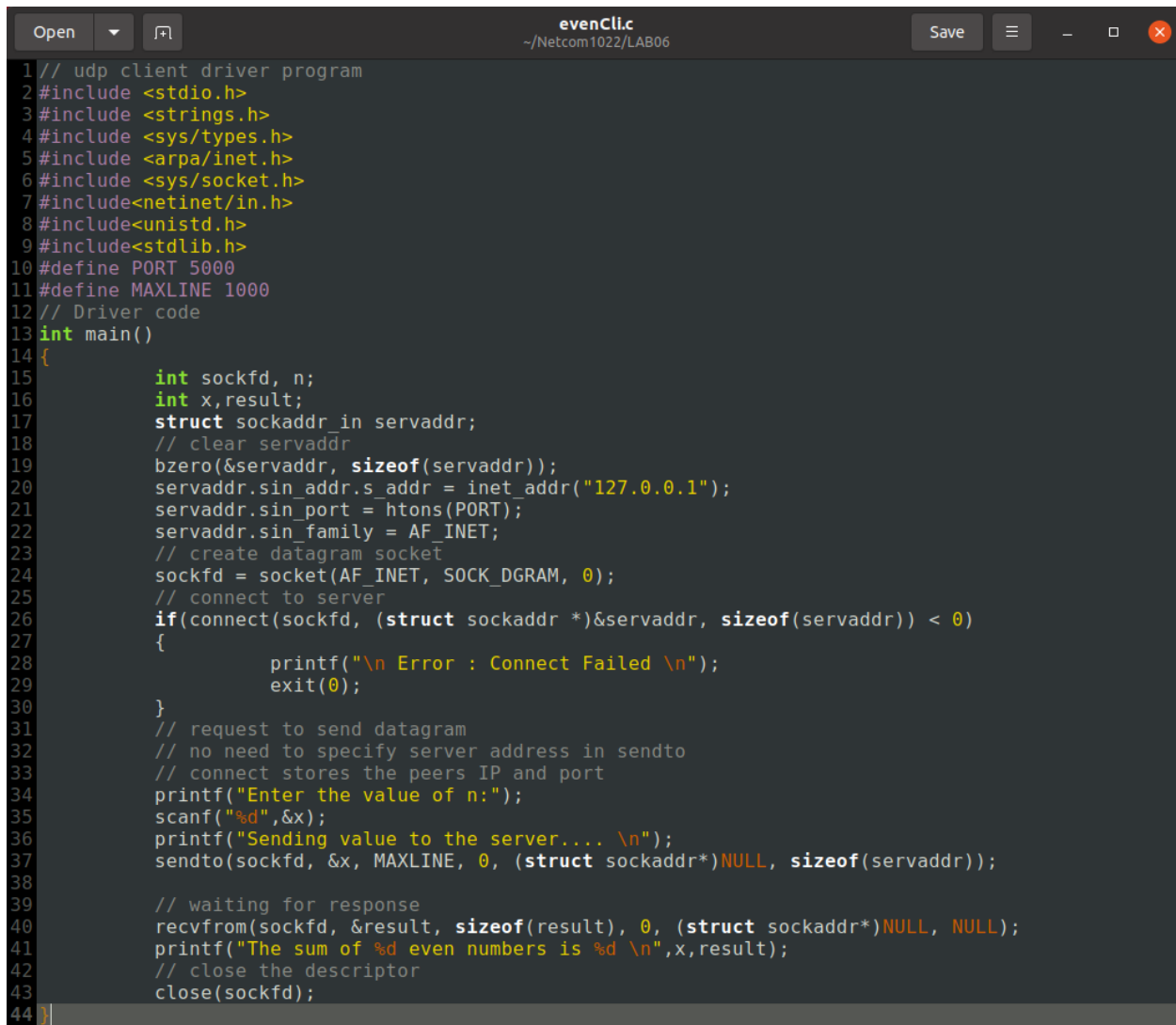
sendto(listenfd, &sum, MAXLINE, 0,
        (struct sockaddr*)&cliaddr, sizeof(cliaddr));
}
```

### Output:

```
preyash-20bps1022@Preyash-20BPS1022:~/Netcom1022/LAB06$ gedit evenSer.c
preyash-20bps1022@Preyash-20BPS1022:~/Netcom1022/LAB06$ gcc evenSer.c
preyash-20bps1022@Preyash-20BPS1022:~/Netcom1022/LAB06$ ./a.out
Result calculated
Sending result to client...
preyash-20bps1022@Preyash-20BPS1022:~/Netcom1022/LAB06$
```

## Client Program Source Code:

### Code window:



```
1 // udp client driver program
2 #include <stdio.h>
3 #include <strings.h>
4 #include <sys/types.h>
5 #include <arpa/inet.h>
6 #include <sys/socket.h>
7 #include <netinet/in.h>
8 #include <unistd.h>
9 #include <stdlib.h>
10 #define PORT 5000
11 #define MAXLINE 1000
12 // Driver code
13 int main()
14 {
15     int sockfd, n;
16     int x,result;
17     struct sockaddr_in servaddr;
18     // clear servaddr
19     bzero(&servaddr, sizeof(servaddr));
20     servaddr.sin_addr.s_addr = inet_addr("127.0.0.1");
21     servaddr.sin_port = htons(PORT);
22     servaddr.sin_family = AF_INET;
23     // create datagram socket
24     sockfd = socket(AF_INET, SOCK_DGRAM, 0);
25     // connect to server
26     if(connect(sockfd, (struct sockaddr *)&servaddr, sizeof(servaddr)) < 0)
27     {
28         printf("\n Error : Connect Failed \n");
29         exit(0);
30     }
31     // request to send datagram
32     // no need to specify server address in sendto
33     // connect stores the peers IP and port
34     printf("Enter the value of n:");
35     scanf("%d",&x);
36     printf("Sending value to the server.... \n");
37     sendto(sockfd, &x, MAXLINE, 0, (struct sockaddr *)NULL, sizeof(servaddr));
38
39     // waiting for response
40     recvfrom(sockfd, &result, sizeof(result), 0, (struct sockaddr *)NULL, NULL);
41     printf("The sum of %d even numbers is %d \n",x,result);
42     // close the descriptor
43     close(sockfd);
44 }
```

### Code:

// udp client driver program

#include <stdio.h>

#include <strings.h>

#include <sys/types.h>

#include <arpa/inet.h>

#include <sys/socket.h>

```
#include<netinet/in.h>
#include<unistd.h>
#include<stdlib.h>
#define PORT 5000
#define MAXLINE 1000
// Driver code
int main()
{
    int sockfd, n;
    int x,result;
    struct sockaddr_in servaddr;
    // clear servaddr
    bzero(&servaddr, sizeof(servaddr));
    servaddr.sin_addr.s_addr = inet_addr("127.0.0.1");
    servaddr.sin_port = htons(PORT);
    servaddr.sin_family = AF_INET;
    // create datagram socket
    sockfd = socket(AF_INET, SOCK_DGRAM, 0);
    // connect to server
    if(connect(sockfd, (struct sockaddr *)&servaddr, sizeof(servaddr)) < 0)
    {
        printf("\n Error : Connect Failed \n");
        exit(0);
    }
    // request to send datagram
    // no need to specify server address in sendto
```

```
// connect stores the peers IP and port
printf("Enter the value of n:");
scanf("%d",&x);
printf("Sending value to the server.... \n");
sendto(sockfd, &x, MAXLINE, 0, (struct sockaddr*)NULL, sizeof(servaddr));

// waiting for response
recvfrom(sockfd, &result, sizeof(result), 0, (struct sockaddr*)NULL, NULL);
printf("The sum of %d even numbers is %d \n",x,result);

// close the descriptor
close(sockfd);
}
```

### Output:

```
preyash-20bps1022@Preyash-20BPS1022:~/Netcom1022/LAB06$ gcc evenCli.c
preyash-20bps1022@Preyash-20BPS1022:~/Netcom1022/LAB06$ ./a.out
Enter the value of n:6
Sending value to the server....
The sum of 6 even numbers is 42
preyash-20bps1022@Preyash-20BPS1022:~/Netcom1022/LAB06$
```

**Result:** We successfully created a program to perform the required output.