

MIT WORLD PEACE UNIVERSITY

Computer Networks
Second Year B. Tech, Semester 3

UDP SOCKET PROGRAMMING

PRACTICAL REPORT
ASSIGNMENT 9

Prepared By
Krishnaraj Thadesar
Cyber Security and Forensics
Batch A1, PA 20
November 29, 2022

Contents

1	Aim and Objectives	1
2	Problem Statement	1
3	Platform	1
4	Code	1
5	Output	3

1 Aim and Objectives

To understand Concept of UDP Socket programming.

2 Problem Statement

Write a C Program for wired network using udp socket to perform Reversing of a String.

3 Platform

Operating System: Arch Linux x86-64

IDEs or Text Editors Used: Visual Studio Code

Programs Used: Cisco Packet Tracer v6.0.1

4 Code

```
1 #include <stdio.h>
2 #include <string.h>
3 #include <sys/socket.h>
4 #include <arpa/inet.h>
5
6 void reverse_string(char *str)
7 {
8     int i, j;
9     char temp;
10    for (i = 0, j = strlen(str) - 1; i < j; i++, j--)
11    {
12        temp = str[i];
13        str[i] = str[j];
14        str[j] = temp;
15    }
16 }
17
18 int main(void)
19 {
20     int socket_desc;
21     struct sockaddr_in server_addr, client_addr;
22     char server_message[2000], client_message[2000];
23     int client_struct_length = sizeof(client_addr);
24
25     // Clean buffers:
26     memset(server_message, '\0', sizeof(server_message));
27     memset(client_message, '\0', sizeof(client_message));
28
29     // Create UDP socket:
30     socket_desc = socket(AF_INET, SOCK_DGRAM, IPPROTO_UDP);
31
32     if (socket_desc < 0)
33     {
34         printf("Error while creating socket\n");
35         return -1;
36     }
37     printf("Socket created successfully\n");
38 }
```

```
39 // Set port and IP:
40 server_addr.sin_family = AF_INET;
41 server_addr.sin_port = htons(2000);
42 server_addr.sin_addr.s_addr = inet_addr("127.0.0.1");
43
44 // Bind to the set port and IP:
45 if (bind(socket_desc, (struct sockaddr *)&server_addr, sizeof(server_addr)) <
46 0)
47 {
48     printf("Couldn't bind to the port\n");
49     return -1;
50 }
51 printf("Done with binding\n");
52 printf("Listening for incoming messages...\n\n");
53
54 // Receive client's message:
55 if (recvfrom(socket_desc, client_message, sizeof(client_message), 0,
56             (struct sockaddr *)&client_addr, &client_struct_length) < 0)
57 {
58     printf("Couldn't receive\n");
59     return -1;
60 }
61 printf("Received message from IP: %s and port: %i\n",
62        inet_ntoa(client_addr.sin_addr), ntohs(client_addr.sin_port));
63
64 reverse_string(client_message);
65 printf("Msg from client: %s\n", client_message);
66
67 // Respond to client:
68 strcpy(server_message, client_message);
69
70 if (sendto(socket_desc, server_message, strlen(server_message), 0,
71           (struct sockaddr *)&client_addr, client_struct_length) < 0)
72 {
73     printf("Can't send\n");
74     return -1;
75 }
76
77 // Close the socket:
78 close(socket_desc);
79
80 return 0;
81 }
```

Listing 1: Server

```
1 #include <stdio.h>
2 #include <string.h>
3 #include <sys/socket.h>
4 #include <arpa/inet.h>
5
6 int main(void)
7 {
8     int socket_desc;
9     struct sockaddr_in server_addr;
10    char server_message[2000], client_message[2000];
11    int server_struct_length = sizeof(server_addr);
12 }
```

```
13 // Clean buffers:
14 memset(server_message, '\0', sizeof(server_message));
15 memset(client_message, '\0', sizeof(client_message));
16
17 // Create socket:
18 socket_desc = socket(AF_INET, SOCK_DGRAM, IPPROTO_UDP);
19
20 if (socket_desc < 0)
21 {
22     printf("Error while creating socket\n");
23     return -1;
24 }
25 printf("Socket created successfully\n");
26
27 // Set port and IP:
28 server_addr.sin_family = AF_INET;
29 server_addr.sin_port = htons(2000);
30 server_addr.sin_addr.s_addr = inet_addr("127.0.0.1");
31
32 // Get input from the user:
33 printf("Enter message: ");
34 gets(client_message);
35
36 // Send the message to server:
37 if (sendto(socket_desc, client_message, strlen(client_message), 0,
38           (struct sockaddr *)&server_addr, server_struct_length) < 0)
39 {
40     printf("Unable to send message\n");
41     return -1;
42 }
43
44 // Receive the server's response:
45 if (recvfrom(socket_desc, server_message, sizeof(server_message), 0,
46             (struct sockaddr *)&server_addr, &server_struct_length) < 0)
47 {
48     printf("Error while receiving server's msg\n");
49     return -1;
50 }
51
52 printf("Server's response: %s\n", server_message);
53
54 // Close the socket:
55 close(socket_desc);
56
57 return 0;
58 }
```

Listing 2: Client

5 Output

SERVER

```
Socket created successfully
Done with binding
Listening for incoming messages...
```

Computer Networks Assignment 8

Received message from IP: 127.0.0.1 and port: 34067
Msg from client: olleh

CLIENT

Socket created successfully
Enter message: hello
Server's response: olleh