

```

1  udpserver.c
2  =====
3  #include <stdio.h>
4  #include <stdlib.h>
5  #include <string.h>
6  #include <unistd.h>
7  #include <arpa/inet.h>
8
9  #define PORT 12345
10 #define BUFFER_SIZE 1024
11
12 int main() {
13     int sockfd;
14     char buffer[BUFFER_SIZE];
15     struct sockaddr_in server_addr, client_addr;
16     socklen_t addr_len = sizeof(client_addr);
17     ssize_t recv_len;
18
19     // Create UDP socket
20     if ((sockfd = socket(AF_INET, SOCK_DGRAM, 0)) < 0) {
21         perror("socket creation failed");
22         exit(EXIT_FAILURE);
23     }
24
25     // Set up the server address struct
26     memset(&server_addr, 0, sizeof(server_addr));
27     server_addr.sin_family = AF_INET;
28     server_addr.sin_addr.s_addr = INADDR_ANY;
29     server_addr.sin_port = htons(PORT);
30
31     // Bind the socket to the server address
32     if (bind(sockfd, (struct sockaddr *)&server_addr, sizeof(server_addr)) < 0) {
33         perror("bind failed");
34         close(sockfd);
35         exit(EXIT_FAILURE);
36     }
37
38     printf("UDP server listening on port %d\n", PORT);
39
40     // Receive and print messages from clients
41     while (1) {
42         recv_len = recvfrom(sockfd, buffer, BUFFER_SIZE - 1, 0,
43                             (struct sockaddr *)&client_addr, &addr_len);
44         if (recv_len < 0) {
45             perror("recvfrom failed");
46             close(sockfd);
47             exit(EXIT_FAILURE);
48         }
49
50         buffer[recv_len] = '\0';
51         printf("Received message: %s\n", buffer);
52     }
53
54     close(sockfd);
55     return 0;
56 }
57
58 udpclient.c
59 =====
60 #include <stdio.h>
61 #include <stdlib.h>
62 #include <string.h>
63 #include <unistd.h>
64 #include <arpa/inet.h>
65
66 #define PORT 12345
67 #define BUFFER_SIZE 1024
68
69 int main() {

```

```

70     int sockfd;
71     struct sockaddr_in server_addr;
72     char buffer[BUFFER_SIZE];
73
74     // Create UDP socket
75     if ((sockfd = socket(AF_INET, SOCK_DGRAM, 0)) < 0) {
76         perror("socket creation failed");
77         exit(EXIT_FAILURE);
78     }
79
80     // Set up the server address struct
81     memset(&server_addr, 0, sizeof(server_addr));
82     server_addr.sin_family = AF_INET;
83     server_addr.sin_addr.s_addr = inet_addr("127.0.0.1"); // Replace with server IP if
different
84     server_addr.sin_port = htons(PORT);
85
86     printf("Enter message to send to server: ");
87     fgets(buffer, BUFFER_SIZE, stdin);
88
89     // Send message to server
90     if (sendto(sockfd, buffer, strlen(buffer), 0,
91         (struct sockaddr *)&server_addr, sizeof(server_addr)) < 0) {
92         perror("sendto failed");
93         close(sockfd);
94         exit(EXIT_FAILURE);
95     }
96
97     printf("Message sent to server: %s", buffer);
98
99     close(sockfd);
100    return 0;
101 }
102
103

```