

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <unistd.h>
4  #include <string.h>
5  #include <sys/socket.h>
6  #include <netinet/in.h>
7  #include <arpa/inet.h>
8  #define SIZE 1024
9
10 int main(int argc, char *argv[]){
11
12     if (argc<2){
13         printf("Usage: message c/s [addr] port\n");
14         exit(EXIT_FAILURE);
15     }
16
17     if (strcmp(argv[1], "s")==0){
18         int fd, new_socket, addrlen, n;
19         char buffer[SIZE];
20         struct sockaddr_in address;
21
22         if (argc!=3){
23             printf("Usage: chat s port\n");
24             exit(EXIT_FAILURE);
25         }
26
27         printf("Server mode (receiver)\n");
28
29         if ((fd = socket(AF_INET, SOCK_STREAM, 0)) < 0) {
30             perror("socket failed");
31             exit(EXIT_FAILURE);
32         }
33
34         if (setsockopt(fd, SOL_SOCKET, SO_REUSEADDR, &(int){1}, sizeof(int)) == -1){
35             perror("setsockopt failed");
36             exit(EXIT_FAILURE);
37         }
38
39         address.sin_family = AF_INET;
40         address.sin_addr.s_addr = INADDR_ANY;
41         address.sin_port = htons(atoi(argv[2]));
42
43         if (bind(fd, (struct sockaddr*)&address, sizeof(address)) < 0) {
44             perror("bind failed");
45             exit(EXIT_FAILURE);
46         }
47
48         if (listen(fd, 3) < 0) {
49             perror("listen failed");
50             exit(EXIT_FAILURE);
51         }
52
53         printf("Listening on port: %s\n", argv[2]);
54
55         addrlen = sizeof(address);
56         while (1){
57
58             if ((new_socket = accept(fd, (struct sockaddr*)&address, (socklen_t*)&addrlen)) < 0) {
59                 perror("accept failed");
60                 exit(EXIT_FAILURE);
61             }
62
63             printf("New connection from: %s\n", inet_ntoa(address.sin_addr));
64
65             while ( (n = read(new_socket, buffer, SIZE))>0 )
66                 write(1, buffer, n);
67
68             printf("Connection closed\n");
69             close(new_socket);
70         }
71
72     } else if (strcmp(argv[1], "c")==0){
73         int fd, new_socket, n;
74         char buffer[SIZE];
75         struct sockaddr_in address;
76
77         if (argc!=4){
78             printf("Usage: chat c address port\n");
79             exit(EXIT_FAILURE);
80         }
81
82         printf("Client mode (sender)\n");
83
84         if ((fd = socket(AF_INET, SOCK_STREAM, 0)) < 0) {
85             perror("socket failed");
86             exit(EXIT_FAILURE);
87         }
88
89         address.sin_family = AF_INET;

```

```
90     address.sin_port = htons(atoi(argv[3]));
91     if (inet_aton(argv[2], &address.sin_addr) <=0){
92         perror("convert server ip failed");
93         exit(EXIT_FAILURE);
94     }
95
96     if ((connect(fd, (struct sockaddr*)&address,sizeof(address)))< 0){
97         perror("connect failed");
98         exit(EXIT_FAILURE);
99     }
100
101     printf("Connected to %s:%s\n", argv[2], argv[3]);
102     while ( (n = read(0, buffer, SIZE))>0 )
103         write(fd, buffer, n);
104 }
105 else{
106     printf("Usage: message c/s [addr] port\n");
107     exit(EXIT_FAILURE);
108 }
109 return 0;
110 }
```