

```

1  /*
2      Simple udp client
3  */
4  #include<stdio.h> //printf
5  #include<string.h> //memset
6  #include<stdlib.h> //exit(0);
7  #include<arpa/inet.h>
8  #include<sys/socket.h>
9
10
11 #define BUFLen 512 //Max length of buffer
12 #define PORT 8888 //The port on which to send data
13
14 void die(char *s)
15 {
16     perror(s);
17     exit(1);
18 }
19
20 int main(void)
21 {
22     struct sockaddr_in si_other;
23     int s, i, slen=sizeof(si_other);
24     char buf[BUFLen];
25     char message[BUFLen];
26
27     if ( (s=socket(AF_INET, SOCK_DGRAM, IPPROTO_UDP)) == -1)
28     {
29         die("socket");
30     }
31
32     memset((char *) &si_other, 0, sizeof(si_other));
33     si_other.sin_family = AF_INET;
34     si_other.sin_port = htons(PORT);
35     si_other.sin_addr.s_addr = inet_addr("127.0.0.1");
36
37     while(1)
38     {
39         printf("Enter message : ");
40         fgets(message, BUFLen, stdin);
41
42         //send the message
43         if (sendto(s, message, strlen(message) , 0 , (struct sockaddr *)
44 &si_other, slen)==-1)
45         {
46             die("sendto()");
47         }
48
49         //receive a reply and print it
50         //clear the buffer by filling null, it might have previously received
51         data
52         memset(buf, '\0', BUFLen);
53         //try to receive some data, this is a blocking call
54         if (recvfrom(s, buf, BUFLen, 0, (struct sockaddr *) &si_other, &slen)
55 == -1)
56         {
57             die("recvfrom()");
58         }
59
60         puts(buf);
61
62     }
63
64     close(s);
65     return 0;
66 }

```