```c
1   /*
2       Simple udp server
3   */
4
5   #include<stdio.h> //printf
6   #include<string.h> //memset
7   #include<stdlib.h> //exit(0);
8   #include<arpa/inet.h>
9   #include<sys/socket.h>
10  #include<unistd.h>
11  #include<time.h>
12  #include<sys/time.h>
13  #include<sys/select.h>
14
15  #define BUFLEN 512  //Max length of buffer
16  #define PORT 8882   //The port on which to listen for incoming data
17
18  void die(char *s)
19  {
20      perror(s);
21      exit(1);
22  }
23
24  typedef struct packet1{
25      int sq_no;
26  }ACK_PKT;
27
28  typedef struct packet2{
29      int sq_no;
30      char data[BUFLEN];
31  }DATA_PKT;
32
33  int main(void)
34  {
35      struct sockaddr_in si_me, si_other;
36
37      time_t t;
38      srand((unsigned) time(&t));
39
40      int s, i, slen = sizeof(si_other) , recv_len;
41      int FLAG=1;
42      DATA_PKT rcv_pkt;
43      ACK_PKT  ack_pkt;
44
45      //create a UDP socket
46      if ((s=socket(AF_INET, SOCK_DGRAM, IPPROTO_UDP)) == -1)
47      {
48          die("socket");
49      }
50
51      // struct timeval timer;
52      // timer.tv_sec = 5;
53      // timer.tv_usec = 0;
54      // fd_set listen_for;
55      // FD_ZERO(&listen_for);
56      // FD_SET(s, &listen_for);
57
58      // zero out the structure
59      memset((char *) &si_me, 0, sizeof(si_me));
60
61      si_me.sin_family = AF_INET;
62      si_me.sin_port = htons(PORT);
63      si_me.sin_addr.s_addr = htonl(INADDR_ANY);
64
65      //bind socket to port
66      if( bind(s , (struct sockaddr*)&si_me, sizeof(si_me) ) == -1)
67      {
68          die("bind");
69      }
```

```c
70
71      int state =0;
72      while(1)
73      {
74
75      switch(state)
76      {  case 0:
77          {   printf("Waiting for packet 0 from sender...\n");
78                      fflush(stdout);
79                      //try to receive some data, this is a blocking call
80                      if ((recv_len = recvfrom(s, &rcv_pkt, BUFLEN, 0, (struct
        sockaddr *) &si_other, &slen)) == -1)
81                          {
82                              die("recvfrom()");
83                          }
84                       if(rand()%100 == 0){
85                          state = 0;
86                          break;
87                      }
88                      if (rcv_pkt.sq_no==0)
89                      {  printf("Packet received with seq. no. %d and Packet
        content is = %s\n",rcv_pkt.sq_no, rcv_pkt.data);
90                          ack_pkt.sq_no = 0;
91                          if (sendto(s, &ack_pkt, recv_len, 0, (struct
        sockaddr*) &si_other, slen) == -1)
92                          {
93                              die("sendto()");
94                          }
95                          state = 1;
96                          break;
97                      }
98                       if(rcv_pkt.sq_no == 1){
99                          printf("Received packet 1.\n");
100                         ack_pkt.sq_no = 1;
101                         if (sendto(s, &ack_pkt, recv_len, 0, (struct
        sockaddr*) &si_other, slen) == -1)
102                         {
103                             die("sendto()");
104                         }
105                         state = 0;
106                         break;
107                     }
108
109         }
110         case 1:
111                     printf("Waiting for packet 1 from sender...\n");
112                         fflush(stdout);
113
114                     //try to receive some data, this is a blocking call
115                     if ((recv_len = recvfrom(s, &rcv_pkt, BUFLEN, 0, (struct
        sockaddr *) &si_other, &slen)) == -1)
116                         {
117                             die("recvfrom()");
118                         }
119                      if(rand()%100 == 0){
120                         state = 1;
121                         break;
122                     }
123                     if (rcv_pkt.sq_no==1){
124                         printf("Packet received with seq. no.= %d and Packet
        content is= %s\n",rcv_pkt.sq_no, rcv_pkt.data);
125                          ack_pkt.sq_no = 1;
126                          if (sendto(s, &ack_pkt, recv_len, 0, (struct
        sockaddr*) &si_other, slen) == -1){
127                              die("sendto()");
128                         }
129                         state = 0;
130                         break;
131                     }
```

```c
132                        if(rcv_pkt.sq_no == 0){
133                            printf("Received packet 0.\n");
134                            ack_pkt.sq_no = 0;
135                            if (sendto(s, &ack_pkt, recv_len, 0, (struct
     sockaddr*) &si_other, slen) == -1)
136                            {
137                                die("sendto()");
138                            }
139                            state = 1;
140                            break;
141                        }
142            }

144        }

146        close(s);
147        return 0;
148    }
```