```c
/*
    Simple udp server
*/

#include<stdio.h> //printf
#include<string.h> //memset
#include<stdlib.h> //exit(0);
#include<arpa/inet.h>
#include<sys/socket.h>
#include<unistd.h>
#include<time.h>
#include<sys/time.h>
#include<sys/select.h>

#define BUFLEN 512  //Max length of buffer
#define PORT 8882   //The port on which to listen for incoming data

void die(char *s)
{
    perror(s);
    exit(1);
}

typedef struct packet1{
    int sq_no;
}ACK_PKT;

typedef struct packet2{
    int sq_no;
    char data[BUFLEN];
}DATA_PKT;

int main(void)
{
    struct sockaddr_in si_me, si_other;

    time_t t;
    srand((unsigned) time(&t));

    int s, i, slen = sizeof(si_other) , recv_len;
    int FLAG=1;
    DATA_PKT rcv_pkt;
    ACK_PKT  ack_pkt;

    //create a UDP socket
    if ((s=socket(AF_INET, SOCK_DGRAM, IPPROTO_UDP)) == -1)
    {
        die("socket");
    }

    // struct timeval timer;
    // timer.tv_sec = 5;
    // timer.tv_usec = 0;
    // fd_set listen_for;
    // FD_ZERO(&listen_for);
    // FD_SET(s, &listen_for);

    // zero out the structure
    memset((char *) &si_me, 0, sizeof(si_me));

    si_me.sin_family = AF_INET;
    si_me.sin_port = htons(PORT);
    si_me.sin_addr.s_addr = htonl(INADDR_ANY);

    //bind socket to port
    if( bind(s , (struct sockaddr*)&si_me, sizeof(si_me) ) == -1)
    {
        die("bind");
    }
```

```
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                      }
```

```
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       }
143
144       }
145
146       close(s);
147       return 0;
148   }
149
```