DHRUV GARG 16 BCE 1190

NETWORK LAB

AIRPLANE PROBLEM

CODE: CLIENT SIDE

```
/* tcpclient.c */
#include<time.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <netdb.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>

int main()
{
  time_t current_time;
  char* c_time_string;
    int sock, bytes_recieved;
    char send_data[1024],recv_data[1024];
    struct hostent *host;
    struct sockaddr_in server_addr;

    host = gethostbyname("127.0.0.1");

    if ((sock = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
      perror("Socket");
      exit(1);
    }

    server_addr.sin_family = AF_INET;
    server_addr.sin_port = htons(5000);
    server_addr.sin_addr = *((struct in_addr *)host->h_addr);
    bzero(&(server_addr.sin_zero),8);
    int i = 0;

    if (connect(sock, (struct sockaddr *)&server_addr,sizeof(struct sockaddr)) == -1)
    {
      perror("Connect");
      exit(1);
    }

    int currAltitude = 0;
    int prevAltitude = 0;
    while(1)
    {  current_time = time(NULL);
     c_time_string = ctime(&current_time);
     bytes_recieved=recv(sock,recv_data,1024,0);
     recv_data[bytes_recieved] = '\0';

     if (strcmp(recv_data , "q") == 0 || strcmp(recv_data , "Q") == 0)
     {
      printf("\nConnection closed from server\n");
      close(sock);
      break;
     }

     else
     {
      {printf("\nRecieved data = %s " , recv_data);
     printf("\nCurrent time is %s", c_time_string);
      }
        char *str[] = {"0", "2148", "11840", "17263", "22827", "28623", "34501", "35248", "35871", "35800",
"33987", "28163", "23748", "19273", "14082", "10454", "7832", "4216", "2023", "1056", "343", "0"};
        strncpy(send_data, str[i], 30);
        printf("\nSEND (q or Q to quit) : %s\n\n", send_data);
        char *temp;
        strncpy(temp, send_data, 10);
        currAltitude = atoi(temp);
        if(i == 0)
         prevAltitude = 0;
        //gets(send_data);
        if((prevAltitude - currAltitude >= 1000)||(currAltitude - prevAltitude >= 1000))
         {
          strcat(send_data, " ft");
          strcat(send_data,"\n\t ALERT!!! Altitude changed by 1000 ft in the last 1 min\n");
         }
        if (strcmp(send_data , "q") != 0 && strcmp(send_data , "Q") != 0)
        {
         strcat(send_data, " ft");
         send(sock,send_data,strlen(send_data), 0);
        }

        else
        {
         send(sock,send_data,strlen(send_data), 0);
         close(sock);
         break;
        }
        i++;
        prevAltitude = currAltitude;
        sleep(60);
     }
```

```
        }
    return 0;
    }
```

// sleep(60)
// Has been used to make 1 min intervals at which altitude data is sent to the server.

CODE: SERVER SIDE

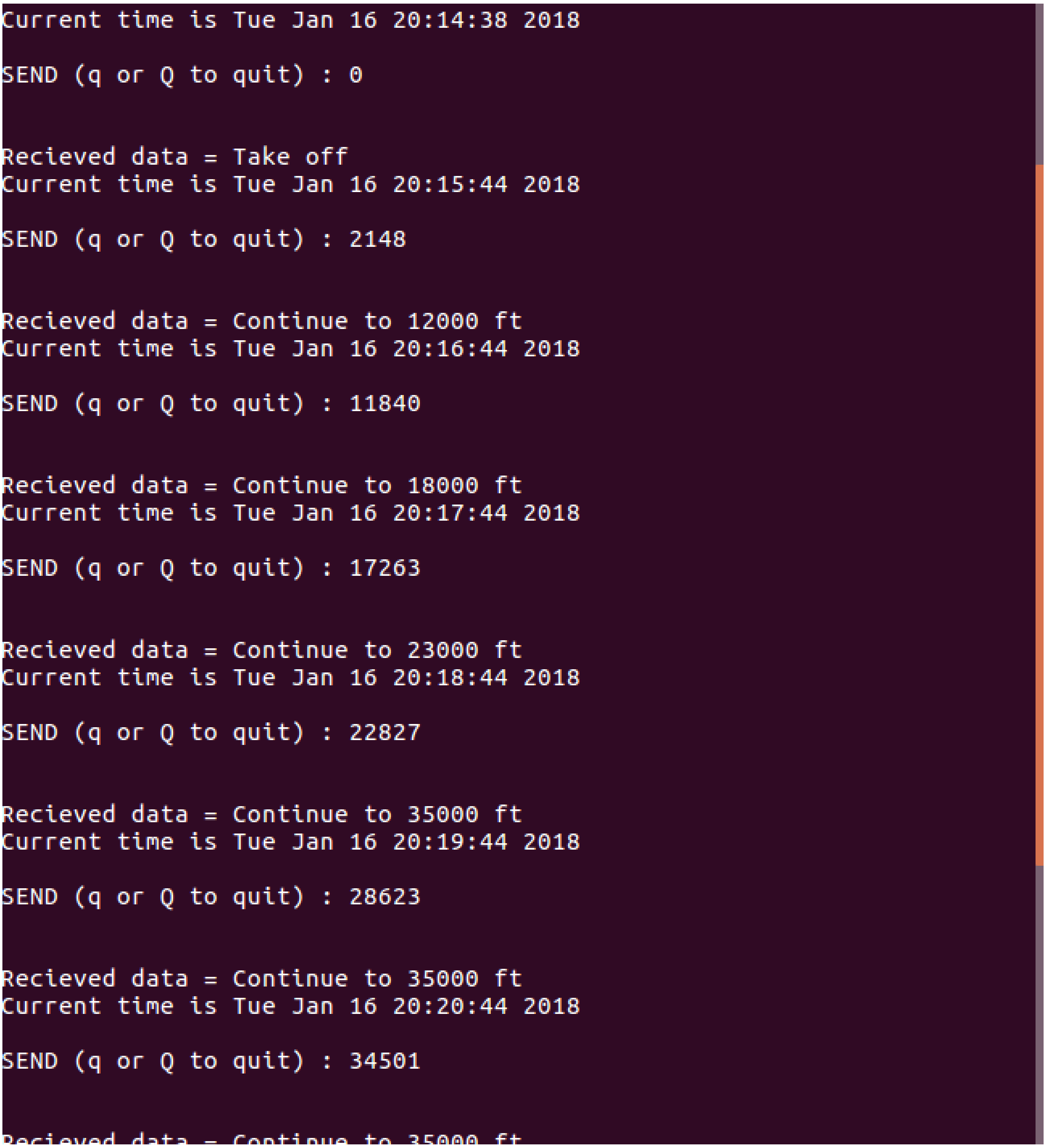
```
/* tcpserver.c */
#include<time.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>


int main()
{
   time_t current_time;
   char* c_time_string;

      int sock, connected, bytes_recieved , true = 1;
      char send_data [1024] , recv_data[1024];

      struct sockaddr_in server_addr,client_addr;
      int sin_size;

      if ((sock = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
         perror("Socket");
         exit(1);
      }

      if (setsockopt(sock,SOL_SOCKET,SO_REUSEADDR,&true,sizeof(int)) == -1) {
         perror("Setsockopt");
         exit(1);
      }

      server_addr.sin_family = AF_INET;
      server_addr.sin_port = htons(5000);
      server_addr.sin_addr.s_addr = INADDR_ANY;
      bzero(&(server_addr.sin_zero),8);

      if (bind(sock, (struct sockaddr *)&server_addr, sizeof(struct sockaddr))
                                                         == -1) {
         perror("Unable to bind");
         exit(1);
      }
```

```
     if (listen(sock, 5) == -1) {
        perror("Listen");
        exit(1);
     }
          printf("\nTCPServer Waiting for client on port 5000");
     fflush(stdout);


     while(1)
     {

        sin_size = sizeof(struct sockaddr_in);

        connected = accept(sock, (struct sockaddr *)&client_addr,&sin_size);

        printf("\n I got a connection from (%s , %d)",
             inet_ntoa(client_addr.sin_addr),ntohs(client_addr.sin_port));

        while (1)
        {current_time = time(NULL);

         c_time_string = ctime(&current_time);
         printf("\n SEND (q or Q to quit) : ");
         gets(send_data);

         if (strcmp(send_data , "q") == 0 || strcmp(send_data , "Q") == 0)
         {
           send(connected, send_data,strlen(send_data), 0);
           close(connected);
           return 0;
         }

         else
            send(connected, send_data,strlen(send_data), 0);

         bytes_recieved = recv(connected,recv_data,1024,0);

         recv_data[bytes_recieved] = '\0';

         if (strcmp(recv_data , "q") == 0 || strcmp(recv_data , "Q") == 0)
         {
           close(connected);
           break;
         }

         else
         printf("\n RECIEVED DATA = %s " , recv_data);
         printf("\n Current time is %s", c_time_string);
         fflush(stdout);
        }
     }

    close(sock);
    return 0;
}
```

SCREENSHOTS

CLIENT SIDE 1: AIRPLANE TAKING OFF

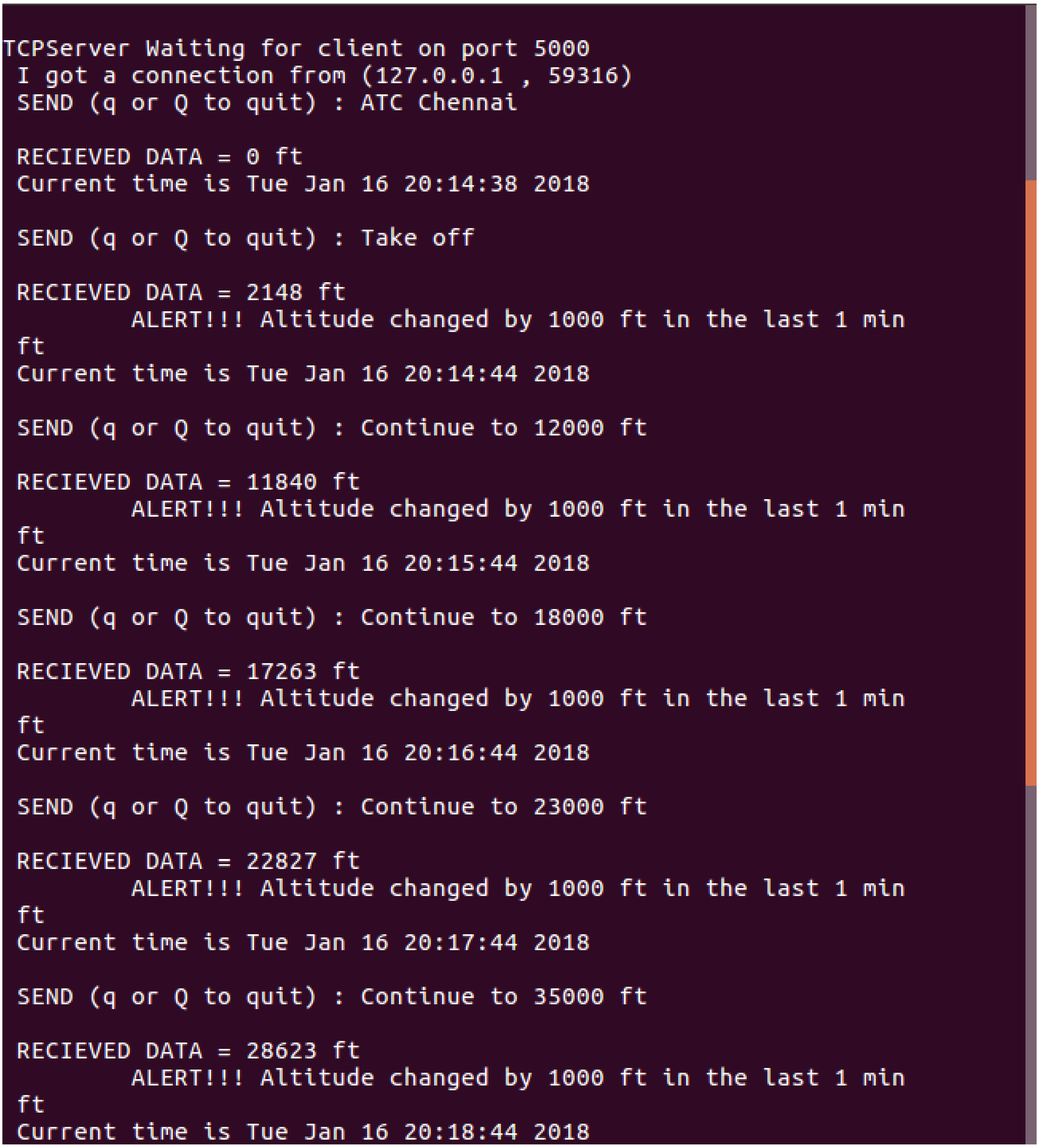
Messages sent from client side and the responses from the server side are shown. Time stamp is also shown. Client side sends the altitude at regular intervals of 1 min.



```
Current time is Tue Jan 16 20:14:38 2018

SEND (q or Q to quit) : 0


Recieved data = Take off
Current time is Tue Jan 16 20:15:44 2018

SEND (q or Q to quit) : 2148


Recieved data = Continue to 12000 ft
Current time is Tue Jan 16 20:16:44 2018

SEND (q or Q to quit) : 11840


Recieved data = Continue to 18000 ft
Current time is Tue Jan 16 20:17:44 2018

SEND (q or Q to quit) : 17263


Recieved data = Continue to 23000 ft
Current time is Tue Jan 16 20:18:44 2018

SEND (q or Q to quit) : 22827


Recieved data = Continue to 35000 ft
Current time is Tue Jan 16 20:19:44 2018

SEND (q or Q to quit) : 28623


Recieved data = Continue to 35000 ft
Current time is Tue Jan 16 20:20:44 2018

SEND (q or Q to quit) : 34501


Recieved data = Continue to 35000 ft
```

SERVER SIDE 1: AIRPLANE TAKING OFF

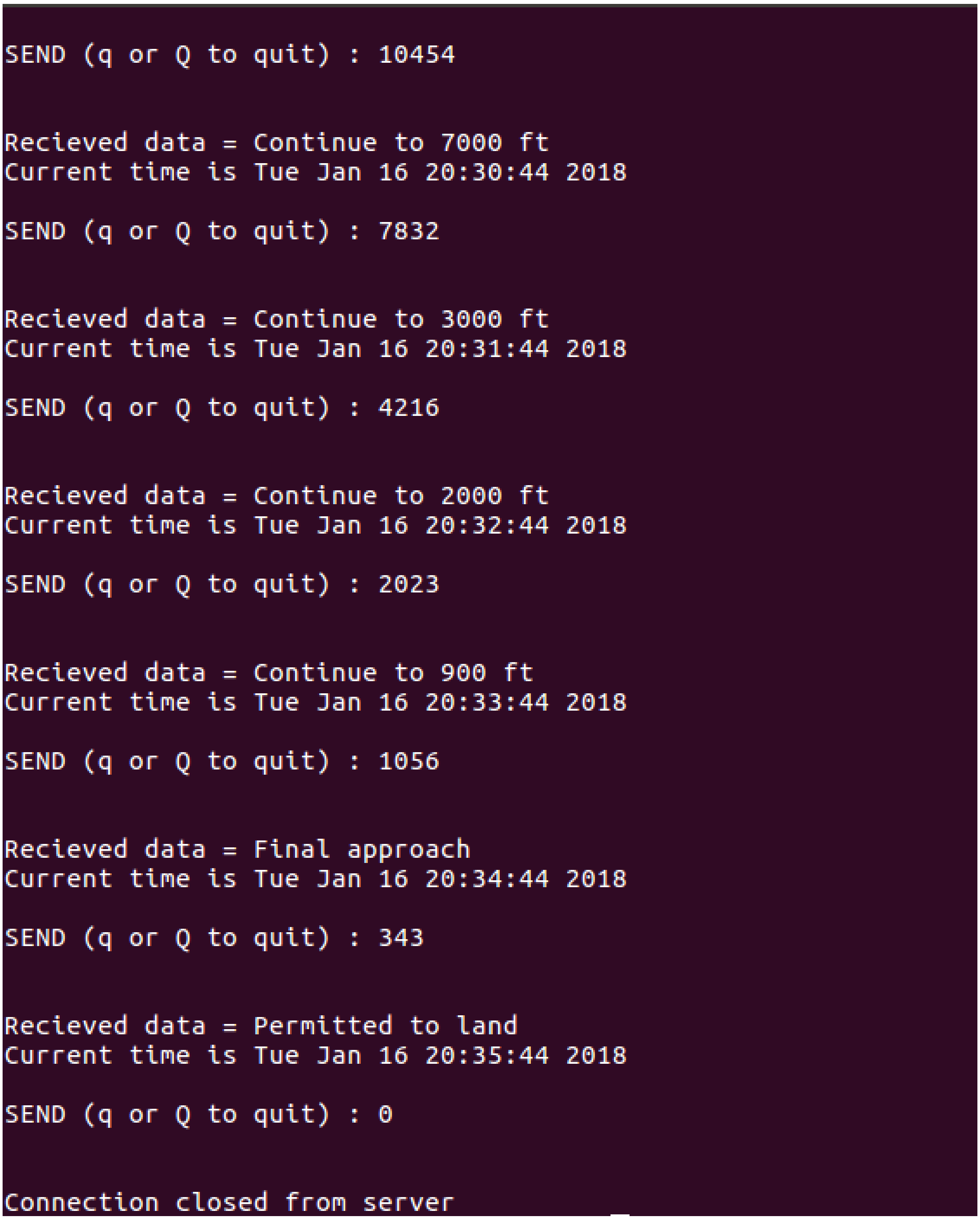
Messages received by the server from client side along with its response. Time-stamp is also given. The client side sends a message every 1 min. Alert messages are displayed for change in altitude of the plane by 1000 ft.



```
TCPServer Waiting for client on port 5000
 I got a connection from (127.0.0.1 , 59316)
 SEND (q or Q to quit) : ATC Chennai

 RECIEVED DATA = 0 ft
 Current time is Tue Jan 16 20:14:38 2018

 SEND (q or Q to quit) : Take off

 RECIEVED DATA = 2148 ft
         ALERT!!! Altitude changed by 1000 ft in the last 1 min
 ft
 Current time is Tue Jan 16 20:14:44 2018

 SEND (q or Q to quit) : Continue to 12000 ft

 RECIEVED DATA = 11840 ft
         ALERT!!! Altitude changed by 1000 ft in the last 1 min
 ft
 Current time is Tue Jan 16 20:15:44 2018

 SEND (q or Q to quit) : Continue to 18000 ft

 RECIEVED DATA = 17263 ft
         ALERT!!! Altitude changed by 1000 ft in the last 1 min
 ft
 Current time is Tue Jan 16 20:16:44 2018

 SEND (q or Q to quit) : Continue to 23000 ft

 RECIEVED DATA = 22827 ft
         ALERT!!! Altitude changed by 1000 ft in the last 1 min
 ft
 Current time is Tue Jan 16 20:17:44 2018

 SEND (q or Q to quit) : Continue to 35000 ft

 RECIEVED DATA = 28623 ft
         ALERT!!! Altitude changed by 1000 ft in the last 1 min
 ft
 Current time is Tue Jan 16 20:18:44 2018
```

CLIENT SIDE 2: AIRPLANE LANDING

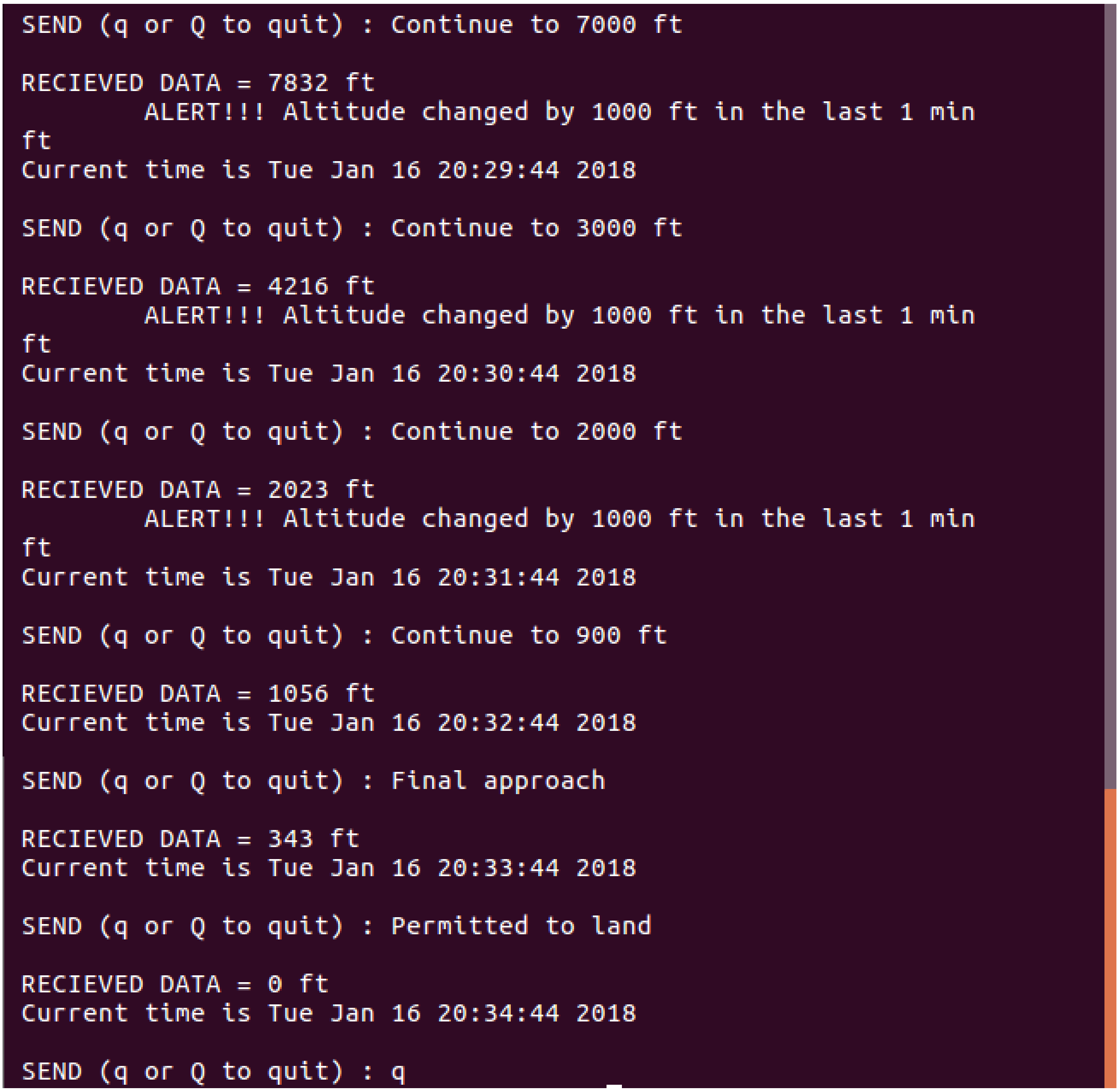
Messages sent from client side and the responses from the server side are shown. Time stamp is also shown. Client side sends the altitude at regular intervals of 1 min.



```
SEND (q or Q to quit) : 10454


Recieved data = Continue to 7000 ft
Current time is Tue Jan 16 20:30:44 2018

SEND (q or Q to quit) : 7832


Recieved data = Continue to 3000 ft
Current time is Tue Jan 16 20:31:44 2018

SEND (q or Q to quit) : 4216


Recieved data = Continue to 2000 ft
Current time is Tue Jan 16 20:32:44 2018

SEND (q or Q to quit) : 2023


Recieved data = Continue to 900 ft
Current time is Tue Jan 16 20:33:44 2018

SEND (q or Q to quit) : 1056


Recieved data = Final approach
Current time is Tue Jan 16 20:34:44 2018

SEND (q or Q to quit) : 343


Recieved data = Permitted to land
Current time is Tue Jan 16 20:35:44 2018

SEND (q or Q to quit) : 0


Connection closed from server
```

SERVER SIDE 2: AIRPLANE LANDING

Messages received by the server from client side along with its response. Time-stamp is also given. The client side sends a message every 1 min. Alert messages are displayed for change in altitude of the plane by 1000 ft.



```
SEND (q or Q to quit) : Continue to 7000 ft

RECIEVED DATA = 7832 ft
        ALERT!!! Altitude changed by 1000 ft in the last 1 min
ft
Current time is Tue Jan 16 20:29:44 2018

SEND (q or Q to quit) : Continue to 3000 ft

RECIEVED DATA = 4216 ft
        ALERT!!! Altitude changed by 1000 ft in the last 1 min
ft
Current time is Tue Jan 16 20:30:44 2018

SEND (q or Q to quit) : Continue to 2000 ft

RECIEVED DATA = 2023 ft
        ALERT!!! Altitude changed by 1000 ft in the last 1 min
ft
Current time is Tue Jan 16 20:31:44 2018

SEND (q or Q to quit) : Continue to 900 ft

RECIEVED DATA = 1056 ft
Current time is Tue Jan 16 20:32:44 2018

SEND (q or Q to quit) : Final approach

RECIEVED DATA = 343 ft
Current time is Tue Jan 16 20:33:44 2018

SEND (q or Q to quit) : Permitted to land

RECIEVED DATA = 0 ft
Current time is Tue Jan 16 20:34:44 2018

SEND (q or Q to quit) : q
```