DA-IICT

Course: IT215, Systems Software.
Instructor: Prof. Sanjay Chaudhary.

Group Members:
Raghuvir Songhela (200901194)
Kaushik Shelat (200901215)

Project: Air Traffic Control

Goal: To control the schedules of flights taking off and landing on any particular airport.

A. Project Definition: To simulate a small model of Air Traffic Control at a given Air-Port, using basic concepts of system software and Inter Process Communication (IPC) Mechanisms.

B. List of Programs:
We have broken our project in various programs, where in we declare and define the various function. Each program consists of a single function which has a particular task. The following programs have been made.

1. **send_request.c** : In this program, the air-flight sends a request to the terminal that it wants to land. The flight number (thread ID) is passed as a parameter, using the pthread_self() function.
2. **send_response.c:** This is essentially the main function of our project. The response of the terminal in case of a landing request is defined here. This function first validates the flight, by processing the flight number, and then checks if the runway is empty. If the runway is empty, it asks the flight to land, and if it is busy, it queues the flight in an array. It also diverts the flights to nearby terminals .
3. **emergency_request.c:** The flight can send an emergency landing request using this function. It operates in the same way as send_request.c.
4. **emergency_response.c:** The terminal asks the flight to land on the emergency runway as soon as the request is sent. In this case, it will not validate the flight, since it is a case of emergency. One separate runway is maintained for emergency landing.
5. **takeoff.c:** In this program, we sequentially, allow all the flights that had previously landed to take-off. The flight numbers of the flights that had landed on the airport, had been stored in an array, and now the terminal will sequentially ask the flights to take-off in the order of their landing.
6. **testrun.c :** This is the program containing the main function. In this program, we create an array of threads (flights), which will ask for permission to land and then perform the above mentioned functions on them. Besides, we also generate three separate threads that will generate an emergency request and the terminal will respond accordingly for them.
7. **header.h :** This is the file containing all header files. It also does the declaration of some of the global and extern variables.

C. Sample input used to test your programs:

1. We have generated 30 flights (threads) which is the normal case. In addition to that, 3 more flights (threads) are created which is the case of emergency. All these flights will send landing request to terminal. The ones which are allowed to land will be asked to take off by terminal.

INPUT –

```
#include "header.h"

int
main ()
{
 int j;
 pthread_t t[30], te[3];

 printf
  ("\n This is Airport Terminal !! \n 30 Flights are going to send their request. \n In addition to that, 3 emergency landing will be shown. \n");
 sleep (2);
 printf
  (" \n After sucessful execution of Landing, the flights will be asked to take off, one by one at the interval of 2 sec.! \n");
 sleep (2);

 for (i = 0; i < 30; i++)     //WE GENERATE 30 THREADS , WHICH ARE SENDING A REQUEST TO LAND .
  {

   pthread_create (&t[i], NULL, send_request, NULL);
  }
 for (j = 0; j < 3; j++)       //THESE 3 THREADS ARE SENDING A REQUEST FOR EMERGENCY LANDING.
  {

   pthread_create (&te[j], NULL, send_emergencyrequest, NULL);

}

 sleep (15);
 printf ("\n The landing is finished sucessfully !\n\n\n");
 sleep (2);
 takeoff ();                          //ALL THE FLIGHTS WHICH HAD LANDED WILL NOW TAKEOFF.

 for (i = 0; i < 30; i++)
  {
   pthread_join (t[i], NULL);       // MAIN WAITS UNTIL ALL 30 THREADS ARE TERMINATED !
  }

 for (j = 0; j < 3; j++)
  {
   pthread_join (te[j], NULL);
  }
 printf
  ("\n All the flights have taken off.\n The Airport Terminalis free now!!\n");
 return 0;
}
```

OUTPUT –

```
Script started on Thu 14 Apr 2011 09:13:41 AM PDT
]0;kaushik@ubuntu: ~/Desktop/mainkaushik@ubuntu:~/Desktop/main$ ./a.out

This is Airport Terminal !!
20 Flights are going to send their request.
In addition to that, 2 emergency landing will be shown.

After sucessful execution of Landing, the flights will be asked to take off, one by one at the interval of 2 sec.!

THIS IS FLIGHT NO. 7168 !
 Please ALLOW ME TO LAND !
FLIGHT NO. 7168
        PERMISSION DENIED... Please Go to HYDERABAD !



THIS IS FLIGHT NO. 4464 !
 Please ALLOW ME TO LAND !
FLIGHT NO. 4464
                PERMISSION DENIED... Please Go to KOLKATA !



THIS IS FLIGHT NO. 1760 !
 Please ALLOW ME TO LAND !
FLIGHT NO. 1760         PLEASE LAND .......

THIS IS FLIGHT NO. 9056 !
 Please ALLOW ME TO LAND !
FLIGHT NO. 9056
        PERMISSION DENIED... Please Go to MUMBAI !



THIS IS FLIGHT NO. 6352 !
 Please ALLOW ME TO LAND !
FLIGHT NO. 6352
                PERMISSION DENIED... Please Go to DELHI !



THIS IS FLIGHT NO. 3648 !
 Please ALLOW ME TO LAND !
FLIGHT NO. 3648
        PERMISSION DENIED... Please Go to HYDERABAD !
```

THIS IS FLIGHT NO. 944 !
Please ALLOW ME TO LAND !
FLIGHT NO. 944
PERMISSION DENIED... Please Go to KOLKATA !

THIS IS FLIGHT NO. 8240 !
Please ALLOW ME TO LAND !
FLIGHT NO. 8240
runway is busy ... wait for a while..

THIS IS FLIGHT NO. 5536 !
Please ALLOW ME TO LAND !
FLIGHT NO. 5536
PERMISSION DENIED... Please Go to MUMBAI !

THIS IS FLIGHT NO. 2832 !
Please ALLOW ME TO LAND !
FLIGHT NO. 2832
PERMISSION DENIED... Please Go to DELHI !

THIS IS FLIGHT NO. 128 !
Please ALLOW ME TO LAND !
FLIGHT NO. 128
PERMISSION DENIED... Please Go to HYDERABAD !

THIS IS FLIGHT NO. 7424 !
Please ALLOW ME TO LAND !
FLIGHT NO. 7424
PERMISSION DENIED... Please Go to KOLKATA !

THIS IS FLIGHT NO. 4720 !
Please ALLOW ME TO LAND !
FLIGHT NO. 4720
runway is busy ... wait for a while..

THIS IS FLIGHT NO. 2016 !
Please ALLOW ME TO LAND !
FLIGHT NO. 2016
PERMISSION DENIED... Please Go to MUMBAI !

THIS IS FLIGHT NO. 9312 !
Please ALLOW ME TO LAND !
FLIGHT NO. 9312
PERMISSION DENIED... Please Go to DELHI !

THIS IS FLIGHT NO. 6608 !
Please ALLOW ME TO LAND !
FLIGHT NO. 6608
PERMISSION DENIED... Please Go to HYDERABAD !

THIS IS FLIGHT NO. 3904 !
Please ALLOW ME TO LAND !
FLIGHT NO. 3904
PERMISSION DENIED... Please Go to KOLKATA !

THIS IS FLIGHT NO. 1200 !
Please ALLOW ME TO LAND !
FLIGHT NO. 1200
runway is busy ... wait for a while..

THIS IS FLIGHT NO. 8496 !
Please ALLOW ME TO LAND !
FLIGHT NO. 8496
PERMISSION DENIED... Please Go to MUMBAI !

THIS IS FLIGHT NO. 5792 !
Please ALLOW ME TO LAND !
FLIGHT NO. 5792
PERMISSION DENIED... Please Go to DELHI !

THIS IS FLIGHT NO. 3088 !
Please ALLOW ME TO LAND !
FLIGHT NO. 3088
PERMISSION DENIED... Please Go to HYDERABAD !

THIS IS FLIGHT NO. 384 !
Please ALLOW ME TO LAND !
FLIGHT NO. 384
PERMISSION DENIED... Please Go to KOLKATA !

THIS IS FLIGHT NO. 7680 !
Please ALLOW ME TO LAND !
FLIGHT NO. 7680
runway is busy ... wait for a while..

THIS IS FLIGHT NO. 4976 !
Please ALLOW ME TO LAND !
FLIGHT NO. 4976
PERMISSION DENIED... Please Go to MUMBAI !

THIS IS FLIGHT NO. 2272 !
Please ALLOW ME TO LAND !
FLIGHT NO. 2272
PERMISSION DENIED... Please Go to DELHI !

THIS IS FLIGHT NO. 9568 !
Please ALLOW ME TO LAND !
FLIGHT NO. 9568
PERMISSION DENIED... Please Go to HYDERABAD !

THIS IS FLIGHT NO. 6864 !
Please ALLOW ME TO LAND !
FLIGHT NO. 6864
PERMISSION DENIED... Please Go to KOLKATA !

THIS IS FLIGHT NO. 4160 !
Please ALLOW ME TO LAND !
FLIGHT NO. 4160
runway is busy ... wait for a while..

THIS IS FLIGHT NO. 1456 !
Please ALLOW ME TO LAND !
FLIGHT NO. 1456
PERMISSION DENIED... Please Go to MUMBAI !

THIS IS FLIGHT NO. 8752 !
Please ALLOW ME TO LAND !
FLIGHT NO. 8752
PERMISSION DENIED... Please Go to DELHI !

THIS IS FLIGHT NO. 6048!
There is an EMERGENCY ....

Please ALLOW ME TO LAND !
FLIGHT NO.6048
PLEASE LAND

THIS IS FLIGHT NO. 3344!
There is an EMERGENCY ....

Please ALLOW ME TO LAND !
FLIGHT NO.3344
PLEASE LAND

THIS IS FLIGHT NO. 640!
There is an EMERGENCY ....

Please ALLOW ME TO LAND !
FLIGHT NO.640
PLEASE LAND

FLIGHT NO.8240

PLEASE LAND !

count = 4

FLIGHT NO. 4720 PLEASE WAIT !!

FLIGHT NO. 1200 PLEASE WAIT !!

FLIGHT NO. 7680 PLEASE WAIT !!

FLIGHT NO. 4160 PLEASE WAIT !!

FLIGHT NO.4720
PLEASE LAND !

count = 3

FLIGHT NO. 1200 PLEASE WAIT !!

FLIGHT NO. 7680 PLEASE WAIT !!

FLIGHT NO. 4160 PLEASE WAIT !!

FLIGHT NO.1200
PLEASE LAND !

count = 2

FLIGHT NO. 7680 PLEASE WAIT !!

FLIGHT NO. 4160 PLEASE WAIT !!

FLIGHT NO.7680
PLEASE LAND !

count = 1

FLIGHT NO. 4160 PLEASE WAIT !!

FLIGHT NO.4160
PLEASE LAND !

count = 0

The landing is finished sucessfully !

Flight no. 1760 ..
Please Takeoff..

Flight no. - 8240 ... Please wait for takeoff !!

Flight no. - 4720 ... Please wait for takeoff !!

Flight no. - 1200 ... Please wait for takeoff !!

Flight no. - 7680 ... Please wait for takeoff !!

Flight no. - 4160 ... Please wait for takeoff !!

Flight no. 8240 ..
Please Takeoff..

Flight no. - 4720 ... Please wait for takeoff !!

Flight no. - 1200 ... Please wait for takeoff !!

Flight no. - 7680 ... Please wait for takeoff !!

Flight no. - 4160 ... Please wait for takeoff !!

Flight no. 4720 ..
Please Takeoff..

Flight no. - 1200 ... Please wait for takeoff !!

Flight no. - 7680 ... Please wait for takeoff !!

Flight no. - 4160 ... Please wait for takeoff !!

Flight no. 1200 ..
Please Takeoff..

Flight no. - 7680 ... Please wait for takeoff !!

Flight no. - 4160 ... Please wait for takeoff !!

Flight no. 7680 ..
Please Takeoff..

Flight no. - 4160 ... Please wait for takeoff !!

Flight no. 4160 ..
Please Takeoff..

All the flights have taken off.
The Airport Terminal is free now!!
]0;kaushik@ubuntu: ~/Desktop/mainkaushik@ubuntu:~/Desktop/main$ exit
exit

Script done on Thu 14 Apr 2011 09:14:21 AM PDT

2. We have generated 20 flights (threads) which is the normal case. In addition to that, 4 more flights (threads) are created which is the case of emergency. All these flights will send landing request to terminal. The ones which are allowed to land will be asked to take off by terminal.

OUTPUT –

```
This is Airport Terminal !!
20 Flights are going to send their request.
In addition to that, 2 emergency landing will be shown.

After sucessful execution of Landing, the flights will be asked to take off, one by one at the interval of 2 sec.!

THIS IS FLIGHT NO. 2416 !
 Please ALLOW ME TO LAND !
FLIGHT NO. 2416
        PERMISSION DENIED... Please Go to MUMBAI !



THIS IS FLIGHT NO. 9712 !
 Please ALLOW ME TO LAND !
FLIGHT NO. 9712
                PERMISSION DENIED... Please Go to DELHI !



THIS IS FLIGHT NO. 7008 !
 Please ALLOW ME TO LAND !
FLIGHT NO. 7008
        PERMISSION DENIED... Please Go to HYDERABAD !



THIS IS FLIGHT NO. 4304 !
 Please ALLOW ME TO LAND !
FLIGHT NO. 4304
                PERMISSION DENIED... Please Go to KOLKATA !



THIS IS FLIGHT NO. 1600 !
 Please ALLOW ME TO LAND !
FLIGHT NO. 1600         PLEASE LAND .....

THIS IS FLIGHT NO. 8896 !
```

Please ALLOW ME TO LAND !
FLIGHT NO. 8896
PERMISSION DENIED... Please Go to MUMBAI !

THIS IS FLIGHT NO. 6192 !
Please ALLOW ME TO LAND !
FLIGHT NO. 6192
PERMISSION DENIED... Please Go to DELHI !

THIS IS FLIGHT NO. 3488 !
Please ALLOW ME TO LAND !
FLIGHT NO. 3488
PERMISSION DENIED... Please Go to HYDERABAD !

THIS IS FLIGHT NO. 784 !
Please ALLOW ME TO LAND !
FLIGHT NO. 784
PERMISSION DENIED... Please Go to KOLKATA !

THIS IS FLIGHT NO. 8080 !
Please ALLOW ME TO LAND !
FLIGHT NO. 8080
runway is busy ... wait for a while..

THIS IS FLIGHT NO. 5376 !
Please ALLOW ME TO LAND !
FLIGHT NO. 5376
PERMISSION DENIED... Please Go to MUMBAI !

THIS IS FLIGHT NO. 2672 !
Please ALLOW ME TO LAND !
FLIGHT NO. 2672
PERMISSION DENIED... Please Go to DELHI !

THIS IS FLIGHT NO. 9968 !

Please ALLOW ME TO LAND !
FLIGHT NO. 9968
PERMISSION DENIED... Please Go to HYDERABAD !

THIS IS FLIGHT NO. 7264 !
Please ALLOW ME TO LAND !
FLIGHT NO. 7264
PERMISSION DENIED... Please Go to KOLKATA !

THIS IS FLIGHT NO. 4560 !
Please ALLOW ME TO LAND !
FLIGHT NO. 4560
runway is busy ... wait for a while..

THIS IS FLIGHT NO. 1856 !
Please ALLOW ME TO LAND !
FLIGHT NO. 1856
PERMISSION DENIED... Please Go to MUMBAI !

THIS IS FLIGHT NO. 9152 !
Please ALLOW ME TO LAND !
FLIGHT NO. 9152
PERMISSION DENIED... Please Go to DELHI !

THIS IS FLIGHT NO. 3744!
There is an EMERGENCY ....

Please ALLOW ME TO LAND !
FLIGHT NO.3744
PLEASE LAND

THIS IS FLIGHT NO. 6448!
There is an EMERGENCY ....

Please ALLOW ME TO LAND !
FLIGHT NO.6448
PLEASE LAND

THIS IS FLIGHT NO. 5120 !
Please ALLOW ME TO LAND !
FLIGHT NO. 5120
runway is busy ... wait for a while..

THIS IS FLIGHT NO. 7824 !
Please ALLOW ME TO LAND !
FLIGHT NO. 7824
PERMISSION DENIED... Please Go to KOLKATA !

THIS IS FLIGHT NO. 528 !
Please ALLOW ME TO LAND !
FLIGHT NO. 528
PERMISSION DENIED... Please Go to HYDERABAD !

FLIGHT NO.8080
PLEASE LAND !

count = 2

FLIGHT NO. 4560 PLEASE WAIT !!

FLIGHT NO. 5120 PLEASE WAIT !!

FLIGHT NO.4560
PLEASE LAND !

count = 1

FLIGHT NO. 5120 PLEASE WAIT !!

FLIGHT NO.5120
PLEASE LAND !

count = 0

The landing is finished sucessfully !

Flight no. 1600 ..
Please Takeoff..

Flight no. - 8080 ... Please wait for takeoff !!

Flight no. - 4560 ... Please wait for takeoff !!

Flight no. - 5120 ... Please wait for takeoff !!

Flight no. 8080 ..
Please Takeoff..

Flight no. - 4560 ... Please wait for takeoff !!

Flight no. - 5120 ... Please wait for takeoff !!

Flight no. 4560 ..
Please Takeoff..

Flight no. - 5120 ... Please wait for takeoff !!

Flight no. 5120 ..
Please Takeoff..

All the flights have taken off.
The Airport Terminal is free now!!

Appendix – A

1. send_request.c –

```
#include "header.h"
void send_response (pthread_t tid);


int *
send_request (void *arg)
{
  void handle ();
  pthread_t t = pthread_self ();
  unsigned int w = t % 10000;
  static struct sigaction respond;
  respond.sa_handler = handle;
  sigfillset (&(respond.sa_mask));

  sigaction (SIGUSR1, &respond, NULL);

  printf ("\n THIS IS FLIGHT NO. %u ! \n  Please ALLOW ME TO LAND !\n ", w);
  pthread_kill (pthread_self (), SIGUSR1);
  return 0;
}

void
handle ()
{
  send_response (pthread_self ());
}
```

2. send_response.c –

```
#include "header.h"

int i, k;
pthread_mutex_t msg_mutex = PTHREAD_MUTEX_INITIALIZER;

int log_count = 0;                    //THIS WILL MAINTAIN THE COUNT OF ALL THEE FLIGHTS THAT LAND ON THE AIRPORT
void
send_response (pthread_t tid)
{
  unsigned int k = (int) tid;         //CHANGING THE FLIGHT NUMBER IN A UNIFORM FORMAT
  int j;
```

```
 unsigned int flight_number = k % 10000;          //THIS IS OUR FLIGHT NUMBER (which is randomly generated by the
system in this case)
/* NOW WE VALIDATE THE FLIGHTS.
  ACCORDING TO THE LOGIC FOLLOWED , IF A FLIGHT NUMBER IS A MULTIPLE OF 5,
  IT IS SUPPOSED TO LAND AT THE AIRPORT .IF IT IS NOT A MULTIPLE OF 5, THEN IT
  IS DIVERTED TO SOME ANOTHER CITY. */
 if (k % 5 == 1)
  {
    printf
          ("FLIGHT NO. %u \n\t PERMISSION DENIED... Please Go to MUMBAI ! \n\n\n",
           flight_number);

  }

 else if (k % 5 == 2)
  {
    printf
          ("FLIGHT NO. %u \n \t\t PERMISSION DENIED... Please Go to DELHI ! \n\n\n",
           flight_number);

  }
 else if (k % 5 == 3)
  {
    printf
          ("FLIGHT NO. %u \n \t PERMISSION DENIED... Please Go to HYDERABAD ! \n\n\n",
           flight_number);

  }

 else if (k % 5 == 4)
  {
    printf
          ("FLIGHT NO. %u \n \t\t PERMISSION DENIED... Please Go to KOLKATA !\n\n\n",
           flight_number);

  }

 else
  {
  r:
    if (runway == 0 && count != 0)          //THE RUNWAY IS FREE BUT THE WAITING QUEUE IS NOT EMPTY.
          {

            pthread_mutex_lock (&msg_mutex);    //LOCKING THE THREAD_MUTEX
            runway = 1;
            printf ("FLIGHT NO.%u \n PLEASE LAND !\n\n\n", pending[0]);
            logg[log_count] = pending[0];
            log_count++;
for (i = 0; i < (count - 1); i++)
```

```
            pending[i] = pending[i + 1];
          count--;
          printf ("\n\ncount = %d\n", count);
          if (count != 0)
           {
             for (i = 0; i < count; i++)
                  printf ("\n\nFLIGHT NO. %u PLEASE WAIT !! \n\n", pending[i]);
           }
          sleep (2);
          runway = 0;
          pthread_mutex_unlock (&msg_mutex);          //UNLOCKING THE THREAD_MUTEX

        }

    else if (runway == 1) //THE RUNWAY IS BUSY
        {

          printf
           ("FLIGHT NO. %u\nrunway is busy ... wait for a while..\n\n\n",
            flight_number);
          pending[count] = flight_number;
          count++;
          sleep (2);
          runway = 0;
          goto r;

        }

    else                        //THE RUNWAY IS FREE AND THE QUEUE IS EMPTY
        {
          pthread_mutex_lock (&msg_mutex);    //LOCKING THE THREAD_MUTEX
          runway = 1;
          printf ("FLIGHT NO. %u \tPLEASE LAND ..... \n", flight_number);
          logg[log_count] = flight_number;
          log_count++;
          sleep (2);
          runway = 0;
          pthread_mutex_unlock (&msg_mutex);          //UNLOCKING THE THREAD_MUTEX
        }
   }
}
```

3. emergency_request.c :

```
#include "header.h"
void send_emergencyresponse (pthread_t tid);

int *
```

```
send_emergencyrequest (void *arg)
{
  void handle2 ();
  pthread_t tr = pthread_self ();
  unsigned int wr = tr % 10000;
  static struct sigaction respond;
  respond.sa_handler = handle2;
  sigfillset (&(respond.sa_mask));

  sigaction (SIGUSR1, &respond, NULL);

  printf
    ("\n THIS IS FLIGHT NO. %u! \n There is an EMERGENCY ....\n\nPlease ALLOW ME TO LAND !\n ",
     wr);
  pthread_kill (pthread_self (), SIGUSR1);
  return 0;
}

void
handle2 ()
{
  send_emergencyresponse (pthread_self ());
}
```

4. emergency_response.c –

```
#include "header.h"

void
send_emergencyresponse (pthread_t tid)
{
  unsigned int q = tid % 10000;
  printf ("FLIGHT NO.%u \n PLEASE LAND\n\n\n", q);
  sleep (2);


}
```

5.takeoff.c –

```
#include "header.h"

takeoff ()
{
  int c = 0;
```

```
  while (logg[c] != 0)
    {
      int z = c + 1;
      printf ("\tFlight no. %d ..\n \tPlease Takeoff..\n\n\n", logg[c]);

      while (logg[z] != 0)
            {
              printf ("\t\t\t  Flight no. - %d ... Please wait for takeoff !! \n\n", logg[z]);
              z++;
            }

      c++;
      sleep (2);

    }


}
```

6.testrun.c –

```
#include "header.h"

int
main ()
{
  int j;
  pthread_t t[20], te[2];

  printf
    ("\n This is Airport Terminal !! \n 20 Flights are going to send their request. \n In addition to that, 2 emergency
landing will be shown. \n");
  sleep (2);
  printf
    (" \n After sucessful execution of Landing, the flights will be asked to take off, one by one at the interval of 2 sec.!
\n");
  sleep (2);

  for (i = 0; i < 20; i++)     //WE GENERATE 30 THREADS , WHICH ARE SENDING A REQUEST TO LAND .
    {

      pthread_create (&t[i], NULL, send_request, NULL);
    }
  for (j = 0; j < 2; j++)       //THESE 3 THREADS ARE SENDING A REQUEST FOR EMERGENCY LANDING.
    {

      pthread_create (&te[j], NULL, send_emergencyrequest, NULL);
```

```
  }


 sleep (15);
 printf ("\n The landing is finished sucessfully !\n\n\n");
 sleep (2);
 takeoff ();                          //ALL THE FLIGHTS WHICH HAD LANDED WILL NOW TAKEOFF.

 for (i = 0; i < 20; i++)
  {
    pthread_join (t[i], NULL);       // MAIN WAITS UNTIL ALL 30 THREADS ARE TERMINATED !
  }

 for (j = 0; j < 2; j++)
  {
    pthread_join (te[j], NULL);
  }
 printf
  ("\n All the flights have taken off.\n The Airport Terminal is free now!!\n");
 return 0;
}
```

7.header.h –

```
#include<errno.h>
#include<stdio.h>
#include<sys/wait.h>
#include<sys/types.h>
#include<unistd.h>
#include<stdlib.h>
#include<string.h>
#include<sys/time.h>
#include<signal.h>
#include<pthread.h>
#include<math.h>

int *send_request (void *arg);
int *send_emergencyrequest (void *arg);
extern int i, k, j;
static int runway = 0;
static int count = 0;
pthread_t pending[10];
pthread_t logg[30];
```

CONCLUSION

We may conclude that the use of threads and user defined signals is very efficient in real world and can be used to simulate many real time applications like Air Traffic Control.