## Grado de Ingegneria en informatica

Programacion Multi-Hilo

# Sistemas Operativos

### Pratica 3

Davide Mercanti NIA: 100390177 100390177@alumnos.uc3m.es

Lorenzo Melis NIA: 100390168 100390168@alumnos.uc3m.es

# INDICE

Nota Inicial pag 3

Description del codigo pag 3

Bateria de Pruebas pag 4

Comentarios pag 5

# *Nota Inicial:*

*Esta memoria fue escrita de la mano de dos estudiantes Erasmus italianos, entonces nos disculpamos para todo los errores de gramática que habrá.*

# Description del codigo

El codigo trata basicamente de crear tres hilos que tienen roles diferentes y que comparten el mismo buffer de memoria:

* un Jefe de Pista que produce n aviones que despuegan;
* un radar que produce m aviones que atterizan;
* una torre de control que “consuma” los n+m aviones.

Entonces, es basicamente un problema productor/consumidor con dos productores y un consumidor que comparten el dicho buffer.

Por primero vamos a describir el contenido del fichero arcport.c.

Aquí tenemos que crear los tres hilos con los tres roles mencionados.

En el main encontramos por primero un control de los argumentos y la llamada a *queue\_init*, luego la creacion de los hilos con la utilizacion de *pthread\_create.* Despues esperamos por la terminacion de los threads *pthread\_join* y al final liberemos la memoria.

Los hilos creatos ejecutan las siguientes funciones:

* *jefe\_de\_pista*

Maneja los aviones que deben despuegar. Por primero asigna la memoria para la estructura *plane* y pasa el apuntador a *queue\_put* (las funciones de gestion del bufer solo trabajan con apuntadores). Los aviones se almacenan en el heap entonces siguen existendo hasta el free corespondiente.

* *radar*

Hace lo mismo con los aviones que deben atterizar.

* *torre\_de\_control*

Empieza llamando *a queue\_get* para obtener un avion y simplemente ejecuta sleep para simular la accion del avion. Esto se hace en un while; cuando se detecta que el avion es el ultimo el thread termina.

El fichero *queue.c* crea y maneja el buffer; tienes 6 funciones para las varias tareas:

* *queue\_init;*
* *queue\_put*
* *queue\_get*
* *queue\_empty*
* *queue\_full*
* *queue\_destroy*

# Bateria de Pruebas

### Function base:

El funcionamento de base es muy simple, tenemos 2 comandos de base:

* El primero solamente llama “arcport”
* El segundo llama “arcport” mas 5 argumentos que tiengon que ser interos

### Casos Extremos:

Errores:

# Conclusiones

La parte de major dificultade fue la sincronizacion de los hilos con los problemas de concurrencia