# 2. Batería de Pruebas

Hemos realizado las pruebas en el servidor de la universidad guernika.lab.inf.uc3m.es. Para las pruebas suponemos que estamos trabajando en el directorio ~/Documentos/Practica3\_SSOO/ donde se encuentra tanto el código realizado como el ejecutable *arcport*.

La siguiente prueba de integración es usada para probar la funcionalidad de los diferentes componentes del programa. Para esta se usa el modo de ejecución con parámetros <n\_planes\_takeoff> <time\_takeoff> <n\_planes\_land> <time\_landing> <size>. Es resultado en consola es enumerado para facilitar la explicación del funcionamiento

a0389461@guernika-2018:~/Documentos/Practica3\_SSOO$ ./arcport 4 2 2 1 6

1. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
2. Welcome to ARCPORT - The ARCOS AIRPORT.
3. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
4. [QUEUE] Buffer initialized
5. [CONTROL] Waiting for planes in empty queue
6. [TRACKBOSS] Plane with id 0 checked
7. [QUEUE] Storing plane with id 0
8. [TRACKBOSS] Plane with id 0 ready to take off
9. [TRACKBOSS] Plane with id 1 checked
10. [RADAR] Plane with id 2 detected!
11. [QUEUE] Getting plane with id 0
12. [CONTROL] Putting plane with id 0 in track
13. [QUEUE] Storing plane with id 1
14. [TRACKBOSS] Plane with id 1 ready to take off
15. [TRACKBOSS] Plane with id 3 checked
16. [QUEUE] Storing plane with id 2
17. [RADAR] Plane with id 2 ready to land
18. [RADAR] Plane with id 4 detected!
19. [QUEUE] Storing plane with id 3
20. [TRACKBOSS] Plane with id 3 ready to take off
21. [TRACKBOSS] Plane with id 5 checked
22. [QUEUE] Storing plane with id 4
23. [RADAR] Plane with id 4 ready to land
24. [QUEUE] Storing plane with id 5
25. [TRACKBOSS] Plane with id 5 ready to take off
26. [CONTROL] Plane 0 took off after 2 seconds
27. [QUEUE] Getting plane with id 1
28. [CONTROL] Putting plane with id 1 in track
29. [CONTROL] Plane 1 took off after 2 seconds
30. [QUEUE] Getting plane with id 2
31. [CONTROL] Track is free for plane with id 2
32. [CONTROL] Plane 2 landed in 1 seconds
33. [QUEUE] Getting plane with id 3
34. [CONTROL] Putting plane with id 3 in track
35. [CONTROL] Plane 3 took off after 2 seconds
36. [QUEUE] Getting plane with id 4
37. [CONTROL] Track is free for plane with id 4
38. [CONTROL] Plane 4 landed in 1 seconds
39. [QUEUE] Getting plane with id 5
40. [CONTROL] Putting plane with id 5 in track
41. [CONTROL] After plane with id 5 the airport will be closed
42. [CONTROL] Plane 5 took off after 2 seconds
43. Airport Closed!
44. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
45. ---> Thanks for your trust in us <---
46. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

a0389461@guernika-2018:~/Documentos/Practica3\_SSOO$ more resume.air

1. Total number of planes processed: 22
2. Number of planes landed: 2
3. Number of planes taken off: 20

## 2.1 Jefe de Pista

Dado el ejemplo anterior, se espera que 4 aviones despeguen, por la tanto se espera la siguiente secuencia 4 veces:

[TRACKBOSS] Plane with id <id> checked

[TRACKBOSS] Plane with id <id> ready to take off

En la línea 6 se muestra un mensaje indicando como el TRACKBOSS crea el primer avión y consecuentemente en la línea 8 después de poner el avión en la cola compartida, se retorna mensaje indicando que el avión con id 0 esta listo para despegar. La línea 12 nos demuestra que el CONTROL exitosamente fue notificado por el TRACKBOSS de que había un avión listo para despegar.

El TRACKBOSS repite este ciclo por cada uno de los 4 aviones supuestos a despegar. Se puede observar una cantidad correcta de aviones recibidos y alistados para despegar por el TRACKBOSS en líneas 6 y 8, 9 y 14, 15 y 20, 21 y 25. Se observan dos mensajes por cada uno de los 4 aviones como es esperado.

Es importante notar la secuencia numérica en el id de los aviones procesados es la esperada, y encaja con el procesamiento concurrente de los aviones que intentan aterrizar.

**Otros casos importantes y casos extremos:**

* Dado [./arcport] como es esperado el numero de aviones por defecto a despegar es 4 y se muestra 8 mensajes indicados por el TRACKBOSS con una secuencia correcta de el id de los aviones
* Dado [./arcport -1 2 2 1 6] el mensaje “ERROR los argumentos no pueden ser negativos es retornado como esperado”
* Dado [./arcport 0 2 2 1 6] como esperado el TRACKBOSS no realiza ninguna acción
* Dado [./arcport 20 2 2 1 6] como esperado el TRACKBOSS procesa los aviones a despegar 20 veces y retorna los dos mensajes correspondientes a cada uno

## 2.2 Radar aéreo

Dado el ejemplo anterior se esperan que 2 aviones aterricen. Por lo tanto, se espera la siguiente secuencia dos veces:

[RADAR] Plane with id <id> detected!

[RADAR] Plane with id <id> ready to land

En la línea 10 se observa como el RADAR detecta el primer avión a aterrizar. A este le asigna un id correspondiente con la secuencia existente en cola y añade este avión a la cola. Luego en la línea 17 podemos observar como el RADAR notifica que existe un avión listo para aterrizar. Podemos observar el mensaje y también podemos observar como el CONTROL eventualmente recibe la señal y aterriza este avión en la línea 32.

El RADAR repite este ciclo por cada uno de los 2 aviones supuestos a aterrizar. Se puede observar una cantidad correcta de aviones recibidos y alistados para aterrizar por el RADAR en líneas 10 y 17, 18 y 23. Se observan dos mensajes por cada uno de los 2 aviones como es esperado.

De forma similar al TRACKBOSS, es importante notar la secuencia numérica en el id de los aviones procesados es la esperada, y encaja con el procesamiento concurrente de los aviones que intentan despegar.

**Otros casos importantes y casos extremos:**

* Dado [./arcport] como es esperado el número de aviones por defecto a aterrizar es 3 y se muestra 6 mensajes indicados por el RADAR con una secuencia correcta del id de los aviones
* Dado [./arcport 4 -1 2 1 6] el mensaje “ERROR los argumentos no pueden ser negativos es retornado como esperado”
* Dado [./arcport 4 0 2 1 6] como esperado el RADAR no realiza ninguna acción
* Dado [./arcport 4 20 2 1 6] como esperado el RADAR procesa los aviones a aterrizar 20 veces y retorna los dos mensajes correspondientes a cada uno

## 2.3 Torre de control

Mientras no haya aviones en la cola, el CONTROL debe retornar el mensaje:

[CONTROL] Waiting for planes in empty queue

Lo cual podemos apreciar en la línea 4 ya que en este punto ni el TRACKBOSS o el RADAR han insertado aviones en la cola.

Adicionalmente, el CONTROL debe mostrar un mensaje indicando que un avión ha sido extraído de la cola circular para ejecutar un despegue. De ser así el siguiente mensaje debe ser retornado:

[CONTROL] Putting plane with id <id> in track

En el ejemplo anterior dado que 4 vuelos están supuestos a despegar, esperamos encontrar este mensaje 4 veces por cada vez que el CONTROL reciba señal del TRACKBOSS de hay un avión listo para despegar. En el ejemplo observamos el mensaje en las líneas 12, 28, 34 y 40.

De forma similar el siguiente mensaje es esperado por cada avión extraído de la cola para aterrizar:

[CONTROL] Track is free for plane with id <id>

Al haber 2 aviones para aterrizar, esperamos que el CONTROL retorne el mensaje dos veces lo cual podemos comprobar en las líneas 31, 37

Consecuentemente, se debe retornar un mensaje indicando

[CONTROL] After plane with id <id> the airport will be closed

Para indicar que el ultimo avión va a ser ejecutado. Una vez este ha sido ejecutado se espera el mensaje

Airport Closed!

Los dos mensajes anteriores los podemos encontrar en líneas 41 y 43 respectivamente.

Finalmente, se crea o se trunca un archivo resume.air que en este ejemplo obtenemos como esperado:

Total number of planes processed: 6

Number of planes landed: 2

Number of planes taken off: 4

**Otros casos importantes y casos extremos:**

* Dado [./arcport] como es esperado el tiempo por defecto para los aviones a despegar es 4 y el tiempo para aterrizar es 3
* Dado [./arcport 4 2 -2 1 6] el mensaje “ERROR los argumentos no pueden ser negativos es retornado como esperado”
* Dado [./arcport 4 2 2 -1 6] el mensaje “ERROR los argumentos no pueden ser negativos es retornado como esperado”
* Dado [./arcport 4 2 2 1 -6] el mensaje “ERROR los argumentos no pueden ser negativos es retornado como esperado”
* Dado [./arcport 4 10 2 10 6] como es esperado las operaciones de aterrizaje y despegue tardan el tiempo especificado para cada una, lo que se pueda apreciar ya que en esta caso el programa tarda mucho mas en terminar de lo que terminaba antes
* Dado [./arcport 4 0 2 0 6] es fácil apreciar que el programa termina más rápido asumiendo que las operaciones de despeje y aterrizaje son instantáneas
* Dado [./arcport 4 2 2 1 0] se obtiene el mensaje “ERROR el tamano del buffer debe ser mayor que 0.”
* Dado [./arcport 4 2 2 1 1] el programa funciona correctamente sin ningún problema
* Después de correr [./arcport 4 2 2 1 0] y luego correr [./arcport] se observa que el contenido de resume.air has sido sobrescrito.

## 2.4 Cola sobre un buffer circular

El funcionamiento correcto de todos los componentes en la prueba usando [./arcport 4 2 2 1 6] es en sí prueba del funcionamiento correcto de la cola sobre el buffer circular. Siendo mas especifico, se espera obtener el siguiente mensaje siempre que queue\_put() se embocase:

[QUEUE] Storing plane with id <id>

Lo cual podemos apreciar en líneas 7, 13, 16, 19, 22, y 24 con un total de las 6 ocurrencias esperadas, 4 para despegar y 2 para aterrizar.

Adicionalmente se muestra el mensaje:

[QUEUE] Getting plane with id <id>

Cada vez que se invoque la función queue\_get() lo cual podemos comprobar funcionando exitosamente ya que encontramos las 6 ocurrencias esperadas en las líneas 11, 27, 30, 33, 36, 39

## 2.5 Modos de ejecución

Ya hemos comprobado la ejecución sin parámetros y la ejecución con los 5 parámetros esperados funciona exitosamente.

Adicionalmente es importante notar los siguientes casos:

* Dado [./arcport 4 2 2 1 6 7] es decir dado más de 5 argumentos se retorna el siguiente error:

usage 1: ./arcport

usage 2: ./arcport <n\_planes\_takeoff> <time\_to\_takeoff> <n\_planes\_to\_arrive> <time\_to\_arrive> <size\_of\_buffer>

* Dado [./arcport 4 2 2] es decir dado menos de 5 argumentos se retorna el siguiente error:

usage 1: ./arcport

usage 2: ./arcport <n\_planes\_takeoff> <time\_to\_takeoff> <n\_planes\_to\_arrive> <time\_to\_arrive> <size\_of\_buffer>

* Dado [./arcport 4 2 c 2 6] se toma el valor de c como 0 y se obtiene el valor esperado usando 0
* Dado [./arcport 3 1 2 1 2] es decir un buffer menor que el numero de aviones a procesar, el programa debe funcionar exitosamente y procesar todo los aviones

## 2.6 Integración y ejemplos de ejecución

El ejemplo al principio de la batería de pruebas nos sirve de muestra del comportamiento exitoso de la integración de los componentes del programa.