Procesador de paquetes.

Objetivos.

- Aprender a implementar el TAD Lista usando una lista de nodos simplemente enlazados.
- Aprender a implementar el TAD Pila usando una lista.
- Aprender a implementar y utilizar el TAD Cola usando dos pilas.

Descripción.

Desarrollar un programa que simule un procesador de paquetes. Para ello se implementará un buffer con una capacidad máxima de paquetes que será gestionado con una cola (sin prioridad). Para implementar la cola se usarán dos pilas.

Cuando un paquete llega al procesador pueden ocurrir las siguientes circunstancias:

- Que esté vacío el buffer, por lo que el procesador pasa a procesar el paquete de forma inmediata.
- Que no esté vacío el buffer, pero haya aún espacio en él. En ese caso el paquete se encola para ser procesado en orden de llegada.
- Que el buffer esté lleno, en ese caso el paquete será desechado (“drop”).

Nótese que pueden llegar varios paquetes en el mismo instante de tiempo.

El programa mostrará para cada paquete, el instante de tiempo en el que comenzó a ser procesado o -1 indicando que el paquete fué desechado.

En el ficheros de tests, la primera fila indica el tamaño del buffer y el número de paquetes que se simulan. A continuación cada paquete (uno por línea) está modelado por una tupla de valores “a p” donde ‘a’ es el tiempo medido en milisegundos de llegada y ‘p’ es el tiempo en milisegundos necesario para procesar el paquete.

Por ejemplo si el fichero test de entrada es:

2 4
1 2
2 2
3 2
3 1

La primera fila indica usar un <tamaño buffer>=2 y se modelan <número de paquetes>=4. Las siguientes filas indican los 4 paquetes (uno por fila) con el formato <tiempo de llegada> <tiempo de procesado>, esto es, el primer paquete llega en tiempo 1 ms y tiene un tiempo de proceso de 2 ms, por lo que terminará de ser procesado en t=1+2= 3ms. Como es el primer paquete (la cola estará vacía) comienza a procesarse directamente.

Después en t=2ms llega el siguiente paquete con un tiempo de proceso de 2 ms. Este paquete empezará a procesarse en t=3ms que es el tiempo en que termina de procesarse el anterior paquete, y a su vez terminará de procesarse en t=3+2= 5 ms.

Ahora llega un tercer paquete en t=3ms, con tiempo de proceso 2 ms. El primer paquete ya ha sido procesado por lo que es desencolado. Como queda espacio libre para

un paquete, se encola para ser procesado en t=5ms y terminando de ser procesado en t=5+2=7ms.

Por último llega un cuarto paquete en t=3ms. La cola está llena con dos paquetes que aún no han sido procesados por lo que este paquete se desecha (drop).

La salida debería del programa debería ser:
1
3
5
-1

Evaluación

Esta práctica se evaluará de la forma siguiente:

Superar todos los tests de	**Puntos**
test_slist tests_slist_simple	2
test_slist tests_slist_cursor (opcional)	2
test_slist tests_slist_fold_unfold (opcional)	1
test_stack tests_stack	1
test_queue tests_queue	2
test_packet_processor tests_packet_processor	2