

Concurrencia y Paralelismo

Grado en Informática 2017

Práctica #2 (Productores y Consumidores)

Ejercicio 1 (Implementar iterations)

Añadir la opción de que en total se produzcan y consuman `iterations` elementos. Los productores deberían terminar cuando se hayan producido `iterations` elementos, y los consumidores cuando se hayan consumido `iterations` elementos. Asegurense que se producen/consumen exactamente `iterations` elementos.

Ejercicio 2 (Imprimir número de iteraciones)

En el `thread` principal, después de que hayan terminado todos los `productores` y `consumidores`, imprimir la lista de elementos producidos por cada productor, ordenada por número de productor y a continuación el número de elementos consumidos por cada consumidor, también ordenada por número de consumidor.

Ejercicio 3 (Implementar control de flujo)

Controlar cuantas veces un productor se encuentra lleno el buffer. Si el buffer esta lleno más de 10 veces en un segundo, hay que crear un consumidor nuevo. Usar un contador compartido que se incrementa en los productores, y un `thread` que duerma un segundo, compruebe el contador y lance un nuevo consumidor si es necesario.

Ejercicio 4 (Implementar control de flujo II)

Controlar cuantas veces un consumidor se encuentra vacío el buffer. Si el buffer esta vacío más de 10 veces en un segundo, hay que matar uno de los consumidores. Esto debe hacerse de forma “civilizada”, es decir, sin utilizar `pthread_kill`.