

Tema 1. Práctica 5: Ejercicios de hilos avanzados.

1. Implementa el esquema productor/consumidor con una lista de elementos. Es decir, habrá un thread que se dedicará a crear datos y otro a consumirlos. Es un esquema bastante habitual de concurrencia, con uno o varios threads creando datos y con uno o varios consumiendo dichos datos. Es parecido al ejercicio que hemos hecho Vamos a crear 3 productores y 2 consumidores. Además, por simplicidad, los datos a producir/consumir serán enteros, empezando por el 1, luego el 2, el 3 y así sucesivamente hasta el 30. Cada thread esperará un tiempo aleatorio entre datos (random), entre 0 y 2 segundos, para simular el tiempo que tarda en producir/consumir cada dato. Utilice un buffer circular, con capacidad para 4 datos, para pasar los datos entre los dos threads. Asegúrese de sincronizarlos correctamente para que el productor no meta datos en el buffer cuando este esté lleno ni que el consumidor coja datos cuando este está vacío. Haga que el consumidor imprima los datos una vez que los haya consumido (tras la espera) para ver que no se pierde ninguno. Imprima también un mensaje cada vez que un thread tenga que esperar porque el buffer esté lleno o vacío.
2. En una mesa hay procesos que simulan el comportamiento de unos filósofos que intentan comer de un plato. Cada filósofo tiene un cubierto a su izquierda y uno a su derecha y para poder comer tiene que conseguir los dos. Si lo consigue, mostrará un mensaje en pantalla que indique «Filósofo 2 comiendo». Después de comer, soltará los cubiertos y esperará al azar un tiempo entre 1000 y 5000 milisegundos, indicando por pantalla «El filósofo 2 está pensando».

En general todos los objetos de la clase Filósofo está en un bucle infinito dedicándose a comer y a pensar. Simular este problema en un programa Java que muestre el progreso de todos sin caer en problemas de sincronización ni de inanición.

3. Acaba de salir a la venta la PlayStation 5 pero sólo hay 20 unidades. Hay 200 clientes en la entrada de la tienda sin ningún tipo de orden (no están en fila). Por la puerta de la tienda sólo cabe un cliente, por lo que tienen que entrar de uno en uno. Cada cliente intentará entrar 10 veces por la puerta. Si después de intentar 10 veces nunca ha encontrado la puerta libre desistirá y se irá sin comprar. Entre intento e intento de entrar cada cliente espera un tiempo aleatorio.

Si un cliente consigue entrar por la puerta hay dos opciones. El cliente cogerá una PlayStation 5 y se marchará. No quedan PlayStation 5 y se marchará.

Realizar la simulación en Java de dicha situación imprimiendo por pantalla información para que se entienda el resultado (quién consigue la PlayStation 5, quién entra y no la puede comprar y quién pierde la paciencia y no llega a entrar).