

Programación de servicios y procesos
Curso 2020/2021

ACTIVIDAD 3. SINCRONIZACIÓN DE HILOS

Objetivos

El objetivo de esta actividad individual es la de repasar, asentar y adquirir un mayor conocimiento de lo impartido en clase mediante la realización de ejercicios teórico/prácticos.

Esta actividad puntuará como calificación para el apartado de "Actividades" de la Evaluación del 1er trimestre.

La entrega de esta actividad se realizará mediante el Aula Virtual. Si alguna persona no pudiera acceder al Campus Virtual, se le dará la opción de entregar esta actividad por correo electrónico al profesor.

Todo alumno que no entregue esta actividad o suba dicha actividad individual posterior a la fecha límite de subida se calificará con un 0 para esta actividad.

La entrega de esa actividad se encuentra ya habilitada en el Aula Virtual. Además, esta actividad deberá ser subida en formato ZIP EXCLUSIVAMENTE.

Formato

EL NOMBRE DEL FICHERO A ENTREGAR EN FORMATO ZIP DEBERÁ TENER LA SIGUIENTE ESTRUCTURA:

Apellido1_Apellido2_Nombre
EJEMPLO: Lopez_Fernandez_Aurelio.ZIP

Aquella persona que, entregado en plazo, incumpla con este formato y forma se le dará 48 horas como máximo para corregirlo y se le penalizará un punto de la calificación de esta actividad. Por el contrario, si no se entrega la actividad en dicho plazo, la calificación de esta actividad será de 0.

Plagio

Se considerará falta grave cuando una persona muestre síntomas de plagio en la resolución de problemas y/o código fuente de los programas, ya sea por código proporcionado por otro alumno o por código copiado de internet. Además, la calificación en esta actividad será de 0.

Ejercicios

EJ1. En mitad de un restaurante de buffet libre hay una mesa capaz de almacenar como máximo cinco platos (buffer). Quien va elaborando e incorporando los platos en dicha mesa es el único cocinero que está contratado por el restaurante. Por otro lado, los comensales del restaurante van cogiendo de forma simultánea esos platos de la mesa para ir comiendo. Por ejemplo, existen dos platos en la mesa y 5 comensales de forma simultánea van a la mesa para conseguir un plato. Por lo tanto, tan sólo dos comensales serán capaces de conseguir un plato y el resto deberá esperar a que el cocinero reponga nuevos platos en la mesa.

Además, se debe utilizar la metodología del algoritmo FIFO para aquellos comensales que se encuentren esperando un nuevo plato. En definitiva, se pretende crear una cola de comensales en el que el primer comensal que se encuentra esperando, será el primero que reciba un plato nuevo.

Se pretende desarrollar un software en Java capaz de solventar este problema real que se encuentra en un restaurante de buffet libre.

EJ2. El primer día post-Covid de la historia, cinco estudiantes filósofos quedan para “filosofear” en el parque de San Bernardo. Estos cinco estudiantes de filosofía dedican su tiempo en pensar y comer, alternativamente. Cuando llegan al parque, éstos se sientan en el césped formando un círculo. En el centro del círculo ponen un plato repleto de sushis y entre cada uno de los estudiantes ponen un palillo chino. Para comer un sushi, un estudiante necesita usar los dos palillos que tiene a cada lado. Si otro estudiante ha cogido uno de los palillos, éste deberá esperar a que dicho palillo se encuentre en la mesa.

El tiempo que tarda un estudiante en comerse un sushi y dejar los palillos en la mesa es de 2 segundos. El tiempo que un estudiante dedica a pensar después de comerse un sushi es de 6 segundos.

Elaborar un software en Java capaz de solventar este problema real.

EJ3. Para calcular un número combinatorio:

$$\binom{n}{k} = \frac{n * (n-1) * (n-2) * \dots * (n-k+1)}{k!}$$

creamos un programa con 2 threads. El thread P1, calcula el numerador y deposita el resultado en una variable compartida, denominada x, mientras que P2 calcula el factorial y deposita el valor en la variable y. Se trata de sincronizar los threads P1 y P2, utilizando semáforos, para que el thread P2 realice correctamente la división x/y. Analice e implemente este problema en lenguaje C. Puede seguir el siguiente pseudocódigo:

Declaración-inicialización de variables globales y semáforos n, k, x=1, i=1 : enteros : semáforos	
P1	P2
For i=n-k+1 to n do Begin x := x*i; End	y:=1 For j=2 to k do Begin y:=y*j; End Escribir(x/y);

EJ4. Una clínica dental está compuesta por:

- Una sala de espera con un número determinado de sillas (N).
- Habitación del dentista con la silla correspondiente para atender a un paciente.

Si no hay pacientes, el dentista se duerme. Si un paciente entre en la sala de espera y todas las sillas están ocupadas, entonces éste abandona la clínica enfadado. Si el dentista está ocupado y hay sillas disponibles, entonces el paciente se sienta a esperar en una de las sillas. Si el dentista está dormido, el paciente entra a la habitación dejando una silla libre en la sala de espera, le despierta al dentista y le dice: ¡Oye tú, atiéndeme!

Escribe un programa que sea capaz de gestionar debidamente la cola de los pacientes que se encuentran en la sala de espera y la atención que el dentista ofrece a los pacientes en su habitación. El dentista puede tardar entre uno y diez segundos en atender a un paciente (número aleatorio).

EJ5. La hermandad de los Negritos de Sevilla ha comprado un nuevo edificio con 15 plantas y cuatro ascensores (identificados con valores entre 0 y 3) para su nueva casa hermandad. En cada planta hay un pulsador para solicitar un ascensor. Cuando una persona llama a un ascensor desde una planta en la que se encuentra, el ascensor libre más cercano será el que deba acudir a dicha planta para que se monte. Si todos los ascensores están ocupados, la petición queda bloqueada hasta que un ascensor quede libre.

No es necesario tratar las peticiones de las personas en el orden en el que llegan. Además, suponga que nunca habrá más de una petición desde la misma planta.

Se pide desarrollar un software que gestione el funcionamiento de forma inteligente de los ascensores de la nueva casa hermandad.

EJ6. DAMAIR es una compañía que construye aviones de pasajeros. Sin embargo, estos aviones son de bajo coste y al despegar generan una gran cantidad de turbulencias, por lo que entre dos despegues consecutivos tiene que transcurrir un intervalo de tiempo mínimo.

En el aeropuerto Aurelinex únicamente despegan aviones de esta compañía y éstos son clasificados como aviones normales y aviones Premium. Los aviones "Premium" tienen preferencia para despegar, sin embargo, no pueden despegar dos aviones Premium de forma consecutiva siempre y cuando haya aviones normales esperando para despegar.

El intervalo de despegue entre dos aviones es de 15 segundos y el tiempo que tarda un avión en llegar a la pista es de 5 segundos.

Se pretende desarrollar un software capaz de gestionar un sistema para controlar los múltiples despegues productivos en el aeropuerto.

Problemas

P1. Se pretende desarrollar un software que gestione una montaña rusa de un parque de atracciones. En esta atracción se contempla que hay m visitantes (pasajeros) y n coches de tan sólo una plaza. Los visitantes del parque de atracciones suelen tardar un rato en llegar hasta la atracción y luego hacen cola para montarse en la atracción.

Un visitante se montará en un coche de la atracción cuando se encuentre alguno libre y el coche realizará el recorrido programado hasta llegar nuevamente al punto de recogida donde el visitante debe bajar de la atracción. Si todos los coches de la atracción están ocupados con un visitante a bordo, entonces el visitante que quiere montarse en la atracción se espera; si algún coche está disponible pero no hay visitantes esperando, entonces el coche es el que se espera.

- Utilice semáforos para sincronizar los m hilos (threads) de los visitantes con los n hilos (threads) de los coches de la atracción.
- La cantidad de tiempo que un visitante tarda en llegar a la atracción es aleatoria. Para generar estos tiempos se puede utilizar la función `rand()`, y los tiempos deberán estar comprendidos en el intervalo $[1,5]$.
- No es necesario que el programa termine cuando todos los visitantes del parque han realizado el recorrido en coche.

Un ejemplo de ejecución del sistema donde hay 10 pasajeros y 2 coches es el siguiente:

```
Visitante 2 llegando a la atracción en 3 segundos.
Visitante 1 llegando a la atracción en 1 segundos.
Visitante 3 llegando a la atracción en 3 segundos.
Visitante 0 llegando a la atracción en 1 segundos.
Visitante 4 llegando a la atracción en 2 segundos.
Visitante 5 llegando a la atracción en 1 segundos.
Visitante 6 llegando a la atracción en 1 segundos.
Visitante 8 llegando a la atracción en 3 segundos.
Visitante 9 llegando a la atracción en 1 segundos.
Visitante 7 llegando a la atracción en 1 segundos.
Visitante 0 se ha montado en un coche de la atracción.
Coche 0 comienza a realizar el recorrido.
Visitante 1 se ha montado en un coche de la atracción.
Coche 1 comienza a realizar el recorrido.
Coche 0 ha finalizado el recorrido.
Visitante 0 sale de la atracción.
Visitante 6 se ha montado en un coche de la atracción.
Coche 1 ha finalizado el recorrido.
Coche 0 comienza a realizar el recorrido.
Visitante 1 sale de la atracción.
Visitante 5 se ha montado en un coche de la atracción.
Coche 1 comienza a realizar el recorrido.
Coche 0 ha finalizado el recorrido.
Visitante 6 sale de la atracción.
.....
```

P2. Una tribu de N estudiantes de 2º DAM nada más que comen serranitos recién elaborados del Mercadona. Estos serranitos se encuentran en un armario que tiene una capacidad máxima para almacenar M serranitos. Cuando un estudiante quiere comer, coge un serranito, a menos que el armario esté vacío. Si el armario está vacío, el estudiante le avisa al cocinero del Mercadona y espera a que el cocinero reponga el armario al completo. Cuando lo ha hecho, y antes de dejar que otro

estudiante le quite un serranito, lo coge él primero. Por parte del cocinero del Mercadona, en el momento que haya repuesto el armario se duerme y espera a ser avisado por cualquier estudiante para elaborar nuevos serranitos.

Sin embargo, los estudiantes de 2º DAM tienen un hambre tan voraz que después de darse una vuelta por el aparcamiento que dura 10 segundos, vuelven a entrar al Mercadona para volver a coger otro serranito.

Desarrollar un programa capaz de gestionar el proceso de reponer serranitos y que los estudiantes cojan serranitos del armario.

P3. Un grupo de N estudiantes sufren casi una depresión con este boletín de ejercicios y desean salir a la calle para fumar. Cada estudiante debe seguir dos procesos continuamente: liarse un cigarrillo y después fumárselo. Para conseguir que un estudiante pueda fumarse un cigarrillo necesita tres ingredientes: tabaco, papel y cerillas.

La empresa Marlboro dispone una cantidad infinita de estos tres materiales. Por este motivo, esta empresa ha puesto una mesa en la puerta de MEDAC y deja dos de los ingredientes en dicha mesa. El estudiante que tiene el ingrediente que le falta lía y se fuma un cigarrillo. Cuando termine de fumarse el cigarro, le avisa al empleado de Marlboro que está en la mesa que quiere más ingredientes para fumarse más cigarros. Entonces, este empleado pone otros dos de los tres ingredientes en la mesa. Este proceso se repite hasta que los estudiantes se hayan fumado 10 cigarros.

Escribir un programa en Java que gestione este problema real.

P4. Por las circunstancias del Covid, en la puerta de MEDAC han puesto un carril por sentido para que los estudiantes puedan entrar y salir al edificio. Sin embargo, las escaleras para subir al edificio son tan estrechas que únicamente pueden pasar estudiantes en un solo sentido al mismo tiempo. Si un estudiante del sentido contrario pretende pasar por esa escalera deberá esperar hasta que 10 estudiantes del sentido de la marcha terminen de subir las escaleras. Cuando los 10 estudiantes terminen de subir, entonces se habilitará para el otro sentido.

Tenga en cuenta que si no hay estudiantes en uno de los sentidos para bajar o subir las escaleras, seguirán subiendo estudiantes del otro sentido.

Elaborar un programa en Java que gestione las escaleras para acceder al edificio de MEDAC.

P5. La asignatura de Programación de Servicios y Procesos dispone de dos vías de comunicación para ayudar a los estudiantes fuera del horario lectivo a través de Discord y de Google Meet. Sin embargo, Aurelio es el único profesor para atender a todos los estudiantes. Cuando un estudiante desea ser atendido, espera en una de las dos aplicaciones, hasta que el profesor le atiende.

Una vez que Aurelio atiende al estudiante descansa durante 10 segundos. Si no hay estudiantes esperando, éste sigue descansando hasta que llegue algún estudiante con alguna duda por cualquiera de las dos aplicaciones. Por otro lado, Aurelio siempre atenderá a aquel estudiante en la aplicación que haya mas estudiantes esperando con dudas. Si el número de estudiantes es igual para las dos aplicaciones, se elige de forma aleatoria un estudiante de cualquiera de las dos aplicaciones.

No es necesario garantizar el orden en que se atiende a los estudiantes. Desarrollar un software capaz de gestionar esta problemática real.