

Programación Concurrente y de Tiempo Real^{*}

Grado en Ingeniería Informática

Asignación de Prácticas Número 8

En esta asignación aplicará control de exclusión mutua y sincronización, utilizando para ello el API estándar de Java, a diferentes situaciones que se le plantean. Documente todo su código con etiquetas (será sometido a análisis con `javadoc`). Si lo desea, puede también agrupar su código en un paquete de clases, aunque no es obligatorio.

1. Enunciados

1. Una base de datos de la DGT puede abstraerse de forma elegante mediante un array de objetos de clase `Conductor.java`. Desarrolle una clase `Conductores.java`. Como toda base de datos, sus elementos deben estar protegidos frente a transacciones concurrentes sobre los mismos. Para lograr esto, determine un API de la clase anterior cuyos métodos estén sincronizados y sus datos sean privados. Escriba ahora un diseño de hilos que haga uso de la base de dato anterior en `usaConductores.java`.

2. La tripulación de un *drakkar* vikingo comparte una marmita con un almuerzo a base de m anguilas cocinadas al eneldo. Cuando un vikingo quiere comer, se sirve una anguila. Si ya no quedan, avisa al vikingo cocinero para que proceda a llenar la marmita de nuevo, utilizando las inagotables provisiones de anguilas disponibles. Desarrolle un programa en java que modele esta curiosa situación, y provea la sincronización necesaria utilizando métodos `synchronized` y notificación `wait()-notifyAll()`. Llame a su programa `drakkarVikingo.java`.

3. El problema de los lectores-escritores es otro problema clásico de la concurrencia. En él, hilos lectores y escritores tratan de acceder a un recurso común bajo los siguientes criterios de control:

- Los lectores pueden acceder al recurso siempre que no haya un escritor escribiendo en él.
- Los escritores deben esperar a que no haya ni otro escritor, ni lectores, accediendo al recurso. La siguiente es una solución que utiliza un monitor teórico *Hoare* para resolver el problema. Se le pide que programe una solución equivalente en java utilizando el API estándar de control de la

^{*}©Antonio Tomeu

conurrencia. Guárdela en `lectorEscritor.java`. Escriba también un diseño de hilos lectores y escritores `usalectorEscritor.java` que emplee la solución anterior.

```
monitor LE;
var
  n: integer; (*numero de lectores leyendo concurrentemente*)
  lector, escritor: condition;
  escribiendo: boolean (*indica si hay un escritor activo*)

procedure inicia_lectura;
begin
  if(escribiendo) then wait(lector);
  n++;
  send(lector)
end;

procedure fin_lectura;
begin
  n--;
  if(n=0) then send(escritor)
end;

procedure inicia_escritura;
begin
  if(n!=0) or (escribiendo) then wait(escritor);
  escribiendo=true;
end;

procedure fin_escritura;
begin
  escribiendo=false;
  if(empty(lector)) then send(escritor);
  else send(lector);
end;

begin (*principal*)
  n=0;
  escribiendo=false;
end.
```

2. Procedimiento y Plazo de Entrega

Se ha habilitado una tarea de subida en *Moodle* que le permite subir cada fichero que forma parte de los productos de la práctica de forma individual en el formato original. Para ello, suba el primer fichero de la forma habitual, y luego siga la secuencia de etapas que el propio *Moodle* le irá marcando. Recuerde además que:

- No debe hacer intentos de subida de borradores, versiones de prueba o esquemas de las soluciones. *Moodle* únicamente le permitirá la subida de los ficheros por **una sola vez**. utilizará paralelismo de grano grueso, particionando un problema y definiendo el número de hilos necesarios para su resolución en función de ecuaciones de balanceado de carga.
- La detección de plagio (copia) en los ficheros de las prácticas, o la subida de ficheros vacíos de contenido o cuyo contenido no responda a lo pedido con una extensión mínima razonable, invalidará plenamente la asignación, sin perjuicio de otras acciones disciplinarias que pudieran corresponder.
- El plazo de entrega de la práctica se encuentra fijado en la tarea de subida del Campus Virtual.
- Entregas fuera de este plazo adicional no serán admitidas, salvo causa de fuerza mayor debidamente justificadas mediante documento escrito.
- Se recuerda que la entrega de todas las asignaciones de prácticas es recomendable, tanto un para un correcto seguimiento de la asignatura, como para la evaluación final de prácticas, donde puede ayudar a superar esta según lo establecido en la ficha de la asignatura.