Programación de Sistemas y Concurrencia

Examen 2ª convocatoria ordinaria
Curso 2020-2021

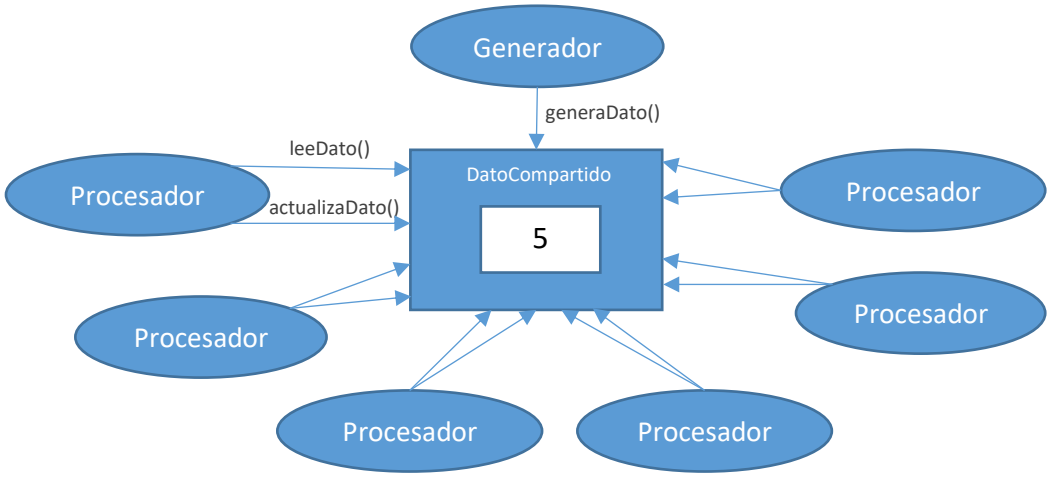
APELLIDOS_____ NOMBRE_____

DNI _____ ORDENADOR _____ GRUPO/TITULACIÓN_____

Bloque 2 – Concurrencia

Descripción del sistema

Queremos diseñar un sistema de procesamiento de datos concurrente formado por **una** hebra *generador de datos* y **N** hebras *procesadores de datos*. La comunicación entre las distintas hebras se realiza mediante un recurso compartido, donde se almacena el dato que se está procesando en cada momento. Para simplificar el ejercicio, el dato a procesar es un valor de tipo entero y el procesamiento realizado por cada hebra procesador consiste simplemente en incrementar en 1 el dato que ha leído.



Cada entidad mostrada en la figura anterior se comportará de la siguiente forma:

- Clase Generador: Hebra que genera un dato y lo almacena en el recurso compartido DatoCompartido, invocando el método generaDato(). Una vez generado un dato, el generador no podrá generar un nuevo dato hasta que todos los procesadores lo hayan procesado. El resultado del procesamiento se devolverá como valor de retorno del método generaDato() (esta clase se da implementada)
- Clase Procesador: Hebra que lee un dato utilizando el método leeDato() y lo procesa. Una vez procesado el dato, esta hebra vuelve a almacenar el valor resultante en el recurso compartido utilizando para ello el método actualizaDato() (esta clase se da implementada)
- Clase DatoCompartido: Recurso compartido mediante el cual se sincronizan el generador de datos con los procesadores de datos, a través de los siguientes métodos que deben ser implementados:

public DatoCompartido(**int** nProcesadores): El constructor de esta clase recibe como parámetro el número de procesadores que tienen que manipular cada dato generado. Debe ser un número mayor que 0.

public int generaDato (**int** dato): El Generador utiliza este método para almacenar un nuevo dato a procesar. Una vez almacenado el dato se debe avisar a los procesadores de que se ha almacenado un nuevo dato. Por último, el Generador tendrá que esperar en este método a que todos los procesadores terminen de procesar el dato. Una vez

que todos terminen, se devolverá el resultado del procesamiento, permitiendo al Generador la generación de un nuevo dato.

public int leeDato(**int** id): El Procesador con identificador id utiliza este método para leer el dato que debe procesar (el dato se devuelve como valor de retorno del método). Deberá esperarse si no hay datos nuevos para procesar o si otro procesador está manipulando el dato.

public int actualizaDato(**int** id, **int** datoActualizado): El Procesador con identificador id almacena en el recurso compartido el resultado de haber procesado el dato. Una vez hecho esto actuará de una de las dos formas siguientes: (1) Si aún hay procesadores esperando para procesar el dato los avisará, ó (2) Si él era el último procesador avisará al Generador de que han terminado.

Además, la clase Driver incluye el método principal (ya implementado) que crea el generador de datos y los N procesadores.

Observa que en este ejercicio hay tres condiciones de sincronización:

CS-Generador: Una vez generado un dato, el Generador espera a que todos los procesadores terminen antes de generar el siguiente dato.

CS1-Procesador: El Procesador espera si no hay un nuevo dato que procesar. Esto puede ocurrir porque el generador aún no haya almacenado ningún dato o porque el Procesador ya haya procesado el dato almacenado en ese momento en el recurso compartido.

CS2-Procesador: Espera a que el dato esté disponible para poder procesarlo (es decir, no hay otro Procesador procesando al dato).

Se pide realizar dos implementaciones de la clase DatoCompartido utilizando una de tipo 1 y otra de tipo 2.
Tipo 1: Semáforos binarios (5 puntos)
Tipo 2: Monitores (5 puntos) o Locks (5 puntos)

La ejecución del programa para N=10 procesadores y M=3 datos que procesar, puede producir, entre otras, la siguiente ejecución:

```
Dato a procesar: 29
Numero de procesadores pendientes: 10
        Procesador 0 ha procesado el dato. Nuevo dato: 30
Numero de procesadores pendientes: 9
        Procesador 1 ha procesado el dato. Nuevo dato: 31
Numero de procesadores pendientes: 8
        Procesador 4 ha procesado el dato. Nuevo dato: 32
Numero de procesadores pendientes: 7
        Procesador 2 ha procesado el dato. Nuevo dato: 33
Numero de procesadores pendientes: 6
        Procesador 3 ha procesado el dato. Nuevo dato: 34
Numero de procesadores pendientes: 5
        Procesador 7 ha procesado el dato. Nuevo dato: 35
Numero de procesadores pendientes: 4
        Procesador 8 ha procesado el dato. Nuevo dato: 36
Numero de procesadores pendientes: 3
        Procesador 5 ha procesado el dato. Nuevo dato: 37
Numero de procesadores pendientes: 2
        Procesador 6 ha procesado el dato. Nuevo dato: 38
Numero de procesadores pendientes: 1
        Procesador 9 ha procesado el dato. Nuevo dato: 39
Numero de procesadores pendientes: 0
Resultado procesamiento = 39
------------------------------
Dato a procesar: 23
Numero de procesadores pendientes: 10
```

```
        Procesador 7 ha procesado el dato. Nuevo dato: 24
Numero de procesadores pendientes: 9
        Procesador 0 ha procesado el dato. Nuevo dato: 25
Numero de procesadores pendientes: 8
        Procesador 1 ha procesado el dato. Nuevo dato: 26
Numero de procesadores pendientes: 7
        Procesador 2 ha procesado el dato. Nuevo dato: 27
Numero de procesadores pendientes: 6
        Procesador 5 ha procesado el dato. Nuevo dato: 28
Numero de procesadores pendientes: 5
        Procesador 3 ha procesado el dato. Nuevo dato: 29
Numero de procesadores pendientes: 4
        Procesador 4 ha procesado el dato. Nuevo dato: 30
Numero de procesadores pendientes: 3
        Procesador 9 ha procesado el dato. Nuevo dato: 31
Numero de procesadores pendientes: 2
        Procesador 6 ha procesado el dato. Nuevo dato: 32
Numero de procesadores pendientes: 1
        Procesador 8 ha procesado el dato. Nuevo dato: 33
Numero de procesadores pendientes: 0
Resultado procesamiento = 33
------------------------------
Dato a procesar: 79
Numero de procesadores pendientes: 10
        Procesador 5 ha procesado el dato. Nuevo dato: 80
Numero de procesadores pendientes: 9
        Procesador 0 ha procesado el dato. Nuevo dato: 81
Numero de procesadores pendientes: 8
        Procesador 4 ha procesado el dato. Nuevo dato: 82
Numero de procesadores pendientes: 7
        Procesador 1 ha procesado el dato. Nuevo dato: 83
Numero de procesadores pendientes: 6
        Procesador 3 ha procesado el dato. Nuevo dato: 84
Numero de procesadores pendientes: 5
        Procesador 9 ha procesado el dato. Nuevo dato: 85
Numero de procesadores pendientes: 4
        Procesador 7 ha procesado el dato. Nuevo dato: 86
Numero de procesadores pendientes: 3
        Procesador 2 ha procesado el dato. Nuevo dato: 87
Numero de procesadores pendientes: 2
        Procesador 6 ha procesado el dato. Nuevo dato: 88
Numero de procesadores pendientes: 1
        Procesador 8 ha procesado el dato. Nuevo dato: 89
Numero de procesadores pendientes: 0
Resultado procesamiento = 89
------------------------------
```