INSTITUTO TECNOLÓGICO SUPERIOR DE JEREZ

INGENIERÍA EN SISTEMAS COMPUTACIONALES

4to Semestre

20/03/2020

Actividad 1: Mapa Conceptual.

Tema 3: Programación Concurrente (Multihilos)

Materia: Tópicos Avanzados de Programación.

Nombre del Alumno: Marín Ramírez Mario.

Número de Control: S18070186

Correo electrónico: mariomarin502t@gmail.com

Profesor: I.S.C Acevedo Sandoval Salvador

Exercises

1. Fill in the blanks in each of the following statements:

a) A thread enters the *terminated* state when it successfully completes its task or otherwise terminates.

b) To pause for a designated number of milliseconds and resume execution, a thread should call method sleep of class Thread

c) Method signal of class Condition moves a single thread in an object's *waiting* state to the *runnable* state.

d) Method signal all of class Condition moves every thread in an object's *waiting* state to the *runnable* state.

e) A(n) runnable thread enters the terminated state when it completes its task or otherwise terminates.

f) A *runnable* thread can enter the timed waiting's state for a specified interval of time.

g) At the operating-system level, the *runnable* state actually encompasses two separate states, ready and running.

h) Runnables are executed using a class that implements the Executror interface.

i) ExecutorService method shutdown ends each thread in an ExecutorService as soon as it finishes executing its current Runnable, if any.

j) A thread can call method await on a Condition object to release the associated Lock and place that thread in the waiting state.

k) In a(n) consumer/producer relationship, the producer generates data and stores it in a shared object, and the consumer reads data from the shared object.

l) Class ArrayBlockngQueue implements the BlockingQueue interface using an array.

m) Keyword synchronized indicates that only one thread at a time should execute on an object.

2. State whether each of the following is *true* or *false*. If *false*, explain why.

a) A thread is not *runnable* if it has terminated. ***TRUE***

b) Some operating systems use timeslicing with threads. Therefore, they can enable threads to preempt threads of the same priority. ***FALSE. DEADLOCKS CAN OCCUR IF THE LOCK ON AN OBJECT IS NEVER RELEASED.***

c) When the thread's quantum expires, the thread returns to the *running* state as the operating system assigns it to a processor. ***TRUE***

d) On a single-processor system without timeslicing, each thread in a set of equal-priority threads (with no other threads present) runs to completion before other threads of equal priority get a chance to execute. ***TRUE***

3. *(True or False)* State whether each of the following is *true* or *false*. If *false*, explain why.

a) Method sleep does not consume processor time while a thread sleeps. ***TRUE***

b) Declaring a method synchronized guarantees that deadlock cannot occur. ***FALSE. DEADLOCKS CAN OCCUR IF THE LOCK ON AN OBJECT IS NEVER RELEASED.***

c) Once a ReentrantLock has been obtained by a thread, the ReentrantLock object will not allow another thread to obtain the lock until the first thread releases it. ***TRUE***

d) Swing components are thread safe. ***FALSE. SWING COMPONENTS ARE NOT THREAD SAFE.***

4. *(Multithreading Terms)* **Define each of the following terms.**

a) thread

An individual execution context of a program

b) multithreading

The ability of more than one thread to execute concurrently.

c) *runnable* state

A state in which the thread is capable of running

d) *timed waiting* state

A state in which the thread cannot use the processor because it is

waiting for a time interval to expire or a notification from another thread.

e) preemptive scheduling

A thread of higher priority enters a running state and is assigned

to the processor.

f) Runnable interface

only declare a specific member function execute, which have been

defined by the classes that implement this interface

g) notifyAll method

Transitions all threads waiting on an object's monitor to the runnable

state.

h) producer/consumer relationship

A relationship in which a producer and a consumer share

common data.

i) quantum

A small amount of processor time, also called a time slice

5. ***(Multithreading Terms)*** Define each of the following terms in the context of Java’s threading mechanisms:

a) synchronized

it is the concept of monitor, which controls access to an object

b) producer

A thread that writes data to a shared memory resource

c) consumer

A thread that reads data from a shared memory resource

d) wait

Places a thread in the waiting state until another thread call notify or notifyAll

e) notify

Wake a thread currently waiting on the given object

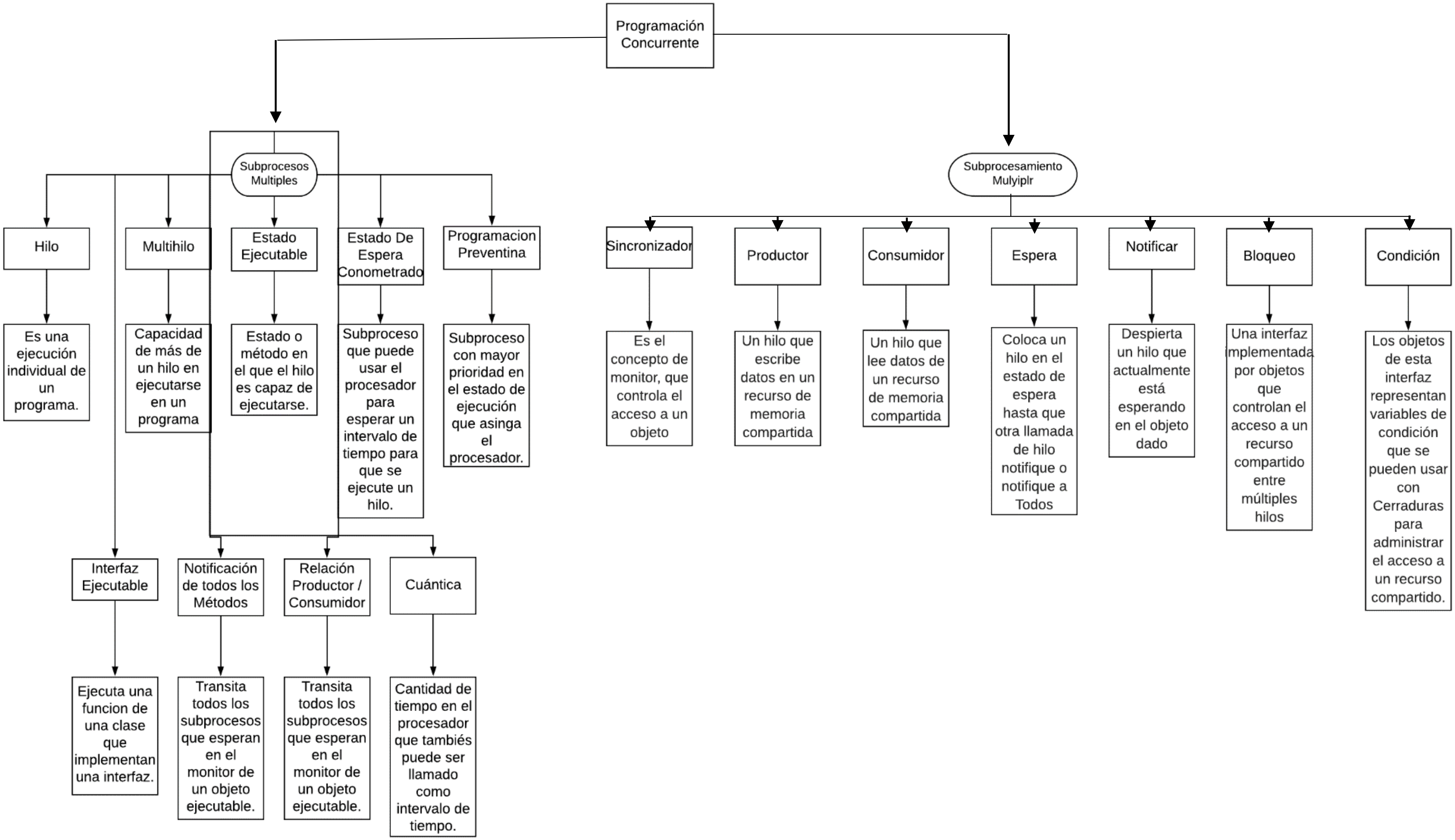
f) Lock

An interface implemented by objects that control access to a resource shared among multiple threads

g) Condition

Objects of this interface represent condition variables that can be used with

Locks to manage access to a shared resource.



Programación Concurrente

Subprocesos Multiples

Hilo

Es una ejecución individual de un programa.

Multihilo

Capacidad de más de un hilo en ejecutarse en un programa

Estado Ejecutable

Estado o método en el que el hilo es capaz de ejecutarse.

Estado De Espera Conometrado

Subproceso que puede usar el procesador para esperar un intervalo de tiempo para que se ejecute un hilo.

Programacion Preventina

Subproceso con mayor prioridad en el estado de ejecución que asinga el procesador.

Interfaz Ejecutable

Ejecuta una funcion de una clase que implementan una interfaz.

Notificación de todos los Métodos

Transita todos los subprocesos que esperan en el monitor de un objeto ejecutable.

Relación Productor / Consumidor

Transita todos los subprocesos que esperan en el monitor de un objeto ejecutable.

Cuántica

Cantidad de tiempo en el procesador que tambiés puede ser llamado como intervalo de tiempo.

Subprocesamiento Mulyiplr

Sincronizador

Es el concepto de monitor, que controla el acceso a un objeto

Productor

Un hilo que escribe datos en un recurso de memoria compartida

Consumidor

Un hilo que lee datos de un recurso de memoria compartida

Espera

Coloca un hilo en el estado de espera hasta que otra llamada de hilo notifique o notifique a Todos

Notificar

Despierta un hilo que actualmente está esperando en el objeto dado

Bloqueo

Una interfaz implementada por objetos que controlan el acceso a un recurso compartido entre múltiples hilos

Condición

Los objetos de esta interfaz representan variables de condición que se pueden usar con Cerraduras para administrar el acceso a un recurso compartido.

REFERENCIAS:

- Programación Concurrente. 2020, de EcuRed Sitio web: https://www.ecured.cu/Programaci%C3%B3n_concurrente
- ¿Qué es la programación concurrente?. 2020, de CodigoFacilito.com Sitio web: https://codigofacilito.com/articulos/programacion-concurrente
- La clase derivada de Thread. 2020, de www.sc.ehu.es Sitio web: http://www.sc.ehu.es/sbweb/fisica/cursoJava/applets/threads/subprocesos.htm
- Qué es la programación multihilo y qué ventajas tiene. 2020, de Ceac.es Sitio web: https://www.ceac.es/blog/que-es-la-programacion-multihilo-y-que-ventajas-tiene