



Threading en Python

Alvarado Morán Óscar Anuar
Computación Concurrente
IIMAS, 2020-1
Universidad Nacional Autónoma de México

SINCRONIZACIÓN CON THREADING EN PYTHON

1. lock

Según entendí, el módulo 'lock' funciona tal y como lo usamos en 'multiprocessing', sólo que aquí la acción de bloqueo no le pertenece a un hilo en particular, sino que un hilo puede invocar el bloqueo que por default aparecerá en modo -desbloqueado- y se activará o desactivará mediante los métodos 'acquire()' y 'release()' no necesariamente por el mismo hilo.

2. rlock

Aquí entiendo que este método se usa para activar o desactivar (igual con 'acquire()' y 'release()') varias veces un bloqueo que fue invocado por el mismo hilo con el que se está activando o desactivando el mismo.

3. event

Es el método más básico de comunicación, simplemente un hilo espera una señal y otro la envía. Aquí se utilizan los métodos 'set()', 'clear()' y 'wait()', donde los primeros dos se

usan para configurar la señal de comunicación (una bandera) como verdadero o falso, respectivamente, y el último método no hace nada más que bloquear lo que hace un hilo hasta que la señal emitida por el otro hilo esté configurada en verdadero.

4. condition

Por lo que entiendo aquí, las variables de 'condition' siempre se relacionan con un bloqueo. En la documentación lo explican como un tipo de comunicación productor - consumidor, de modo que mediante el método 'wait()' está esperando a la llamada de algo ajeno para despertar y seguir haciendo su trabajo (productor) y aquí es donde entra el método 'notify()' que le notifica al método 'wait()' que ya haga sus cosas, sin embargo, esto depende aún de que haya bloqueos en el ambiente.

5. semaphore

Como ya lo vimos en clase, un semáforo es una variable compartida que cambia mediante invocaciones de los métodos 'acquire()' y 'release()', no puede ser negativa, por lo que al llegar a cero se bloquea y espera a la llamada del método 'release()' de algún hilo, esto es, dicho método aumenta el semáforo y el otro lo disminuye.
