CI-0117 Laboratorio 3 : Métodos de Sincronización y Problemas Asociados / Paralela y Concurrente
Marco Antonio Ferraro Rodriguez | B82957

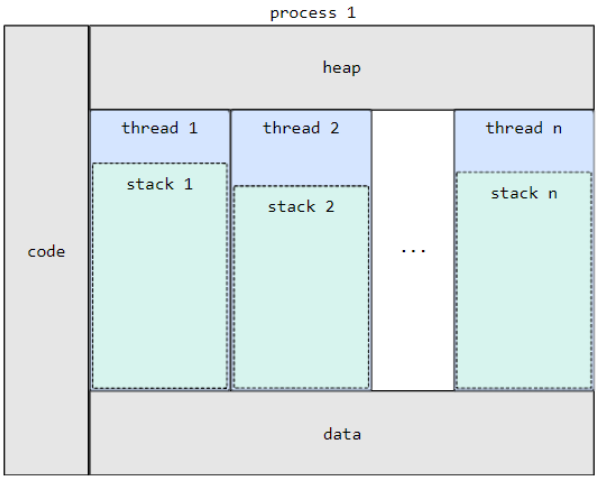
---

1. ¿Cuáles son los problemas que se encuentran al utilizar hilos?

Como se vio en clases, los hilos de ejecución sobre un mismo proceso compartes los mismos recursos de la computadora. Si no se diseña un buen programa que utilice las herramientas de manera correcta, se pueden generar una serie de errores. En otras palabras, el codigo no es **thread safe**

1.1. Condición de Carrera

Recordemos que 2 o más hilos comparten los mismos recursos que la computadora le asigna a un proceso.



Dicho esto, un problema que ocurre al manejar hilos **sin algún método de sincronización** y estos hilos acceden a la misma variable o dirección de memoria es denominado **condición de carrera**. Se llama así porque los hilos están compitiendo en una çarrera"por quien accede o quien modifica el valor en una dirección de memoria. El problema que esto genera es que la salida va a depender del **orden de ejecución de los hilos**
Si 2 o mas hilos modifican una variable compartida, sea **x**, esta variable puede tener diferentes valores con la misma cantidad de iteraciones sobre cada hilo.

```
x = 2 + x;
```

Cabe también notar, que cabe otra tipo de condición de carrera llamado **data condición** en inglés. Esta condición ocurre cuando varios hilos modifican una variable por medio de operaciones **write**, entonces lo que ocurre es que el valor de la variable puede ser corrompida. Expresada en un ejemplo cotidiano seria como si un estudiante estuviera resolviendo un problema matemático en la pizarra, pero sin que el estudiante termine el problema, otro estudiante llegue y siga resolviendo el siguiente paso del problema matemático. Hay modificación sin control sobre los datos. **Se pierde el determinismo de datos.**

1.2. Spinning

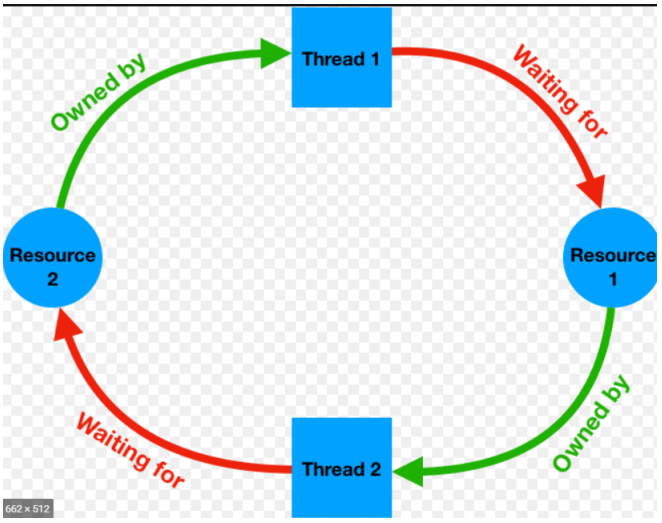
Spinning es una condición que ocurre cuando un hilo se queda consumiendo recursos computacionales del CPU esperando que una condición se cumpla para avanzar.
Usualmente lo que establece esta condición, es otro hilo de ejecución, no el que esta haciendo spinning.
Esta técnica es considerada poco útil ya que un hilo de ejecución se queda básicamente ejecutando nada pero esta consumiendo recursos computacionales del CPU que se podría usar de una manera mas eficiente.

1.3. Deadlock

En el contexto de programación con hilos, un deadlock ocurre cuando un programa se bloquea, ya que los hilos están esperando condiciones que nunca se van a cumplir. Esto ocurre cuando el hilo 1 espera que la condición del hilo 2 sea cierta, pero el hilo 2 espera que la condición del hilo 1 sea cierta.

1.4. Starvation

Stravation es un error de **sincronizacion**. Se origina cuando un hilo entra a un estado de **sleep** pero nadie lo vuelve a activar.



Es importante destacar que deadlock también es un problema de sincronización.

2. ¿Qué métodos de sincronización se pueden usar para mitigar estos problemas?

2.1. Mutex

El mutex es una herramienta de sincronización que permite que solo **1 hilo** ejecute una sección de código al mismo tiempo. Tiende a usarse solo para el **write de una variable, para evitar que 2 o mas hilos modifiquen una variable compartida al mismo momento**

2.2. Semáforo

El semáforo es una herramienta que se utiliza sobren los hilos para ponerlos a "dormir". Es una variable que le indica a un hilo si puede çorrer.º estar no.

2.3. Barrier

Una barrera o barrier es una herramienta que lo que ocasiona es que un hilo va a estar esperando en la sección de la barrera, hasta que todos los hilos lleguen a la barrera. Básicamente es una señal para que todos los hilos lleguen al mismo punto de ejecución.

3. ¿Qué analogías del mundo real utilizaría para explicar estos métodos de sincronización? Puede usar como referencia/inspiración la analogía utilizada en Locks, Mutexes, and Semaphores

La analogia que se me ocurre para explicar los métodos de sincronización seria un oficial de transito. El oficial controla el flujo de los carros (hilos en este caso) para controlar el buen flujo vehicular. El oficial puede detener un vehículo para que otros fluyan como el semáforo, o puede detener a todos en un momento dado como una barrera. También puede pasar un carro al mismo tiempo en una calle peligrosa como un mutex en una sección critica.