

Laboratorio de Sistemas Operativos  
Semestre A-2018  
Práctica Cinco  
Laboratorio

Prof. Rodolfo Sumoza  
Prep. Alvaro Araujo  
Prep. Luis Sanchez

## 1. Sincronización.

Un proceso cooperativo es aquel que puede afectar o verse afectado por otros procesos que se ejecutan en el sistema. Los procesos de cooperación pueden compartir directamente un espacio de direcciones lógicas (es decir, tanto códigos como datos) o se les permite compartir datos solo a través de archivos o mensajes. Un ejemplo de lo expuesto anteriormente, es el acceso concurrente a datos compartidos, esta tarea puede provocar inconsistencias en los datos.<sup>1</sup>

## 2. Mutex.

La función del bloqueo mutex es proteger las regiones críticas y así evitar las condiciones de carrera. Es decir, un proceso debe adquirir el bloqueo antes de ingresar a una sección crítica y luego libera el bloqueo cuando sale de la sección crítica. Un bloqueo mutex posee una variable booleana cuyo valor indica si el bloqueo está disponible o no. Un proceso que intenta adquirir un bloqueo no disponible se bloquea hasta que se libera el bloqueo.<sup>1</sup>.

---

<sup>1</sup>Silberschatz, A., (Ed. 9)(2013)*Operating System Concepts*, United States of America: Wiley.

### 3. Semáforo.

El semáforo es una herramienta más robusta que puede comportarse de manera similar a un bloqueo mutex, pero también puede proporcionar formas más sofisticadas para que los procesos sincronicen sus actividades.

Los semáforos se pueden usar para controlar el acceso a un recurso, dado que consiste en un número finito de instancias. El semáforo se inicializa en una cantidad de recursos disponibles. Cada proceso que desea usar un recurso realiza una operación en el semáforo que disminuye en uno las instancias disponibles. Cuando un proceso libera un recurso, realiza una operación que incrementa en uno las instancias disponibles. Cuando el conteo del semáforo se encuentra en 0, todos los recursos están siendo utilizados. De ocurrir esto último, los procesos que deseen usar un recurso, serán bloqueados hasta que el recuento de instancias disponibles sea mayor a 0.<sup>1</sup>

### 4. Monitor.

Un tipo de monitor es un TAD que incluye un conjunto de operaciones definidas por el programador que se proporcionan con exclusión mutua dentro del monitor. El tipo monitor también declara las variables cuyos valores definen el estado de una instancia de ese tipo, junto con los cuerpos de funciones que operan en esas variables. La representación de un tipo monitor no puede ser utilizada directamente por los diversos procesos. Por lo tanto, una función definida dentro de un monitor puede acceder solo a aquellas variables declaradas localmente dentro del monitor y sus parámetros formales. De forma similar, las variables locales de un monitor solo se pueden acceder mediante las funciones locales.<sup>1</sup>