

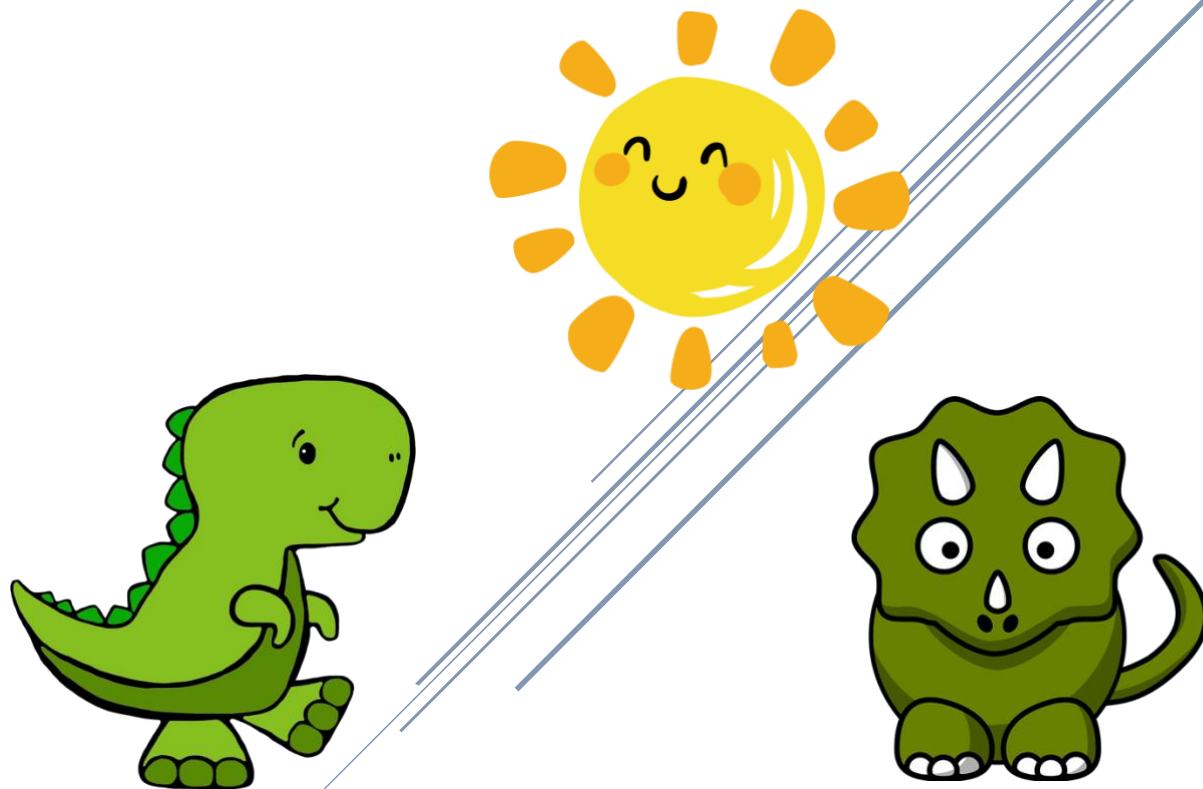
# PRACTICA 2

## Implementación de Algoritmo de Peterson y Filtro modificado

**Profesor:** Salvador Gonzalez Arellano.  
email: salvador\_gonzalez\_a@ciencias.unam.mx

**Ayudante de teoría:** Rogelio Alcantar Arenas.  
email: rogelio-aa@ciencias.unam.mx

**Ayudante de laboratorio:** Luis Angel Leyva Castillo  
email: luis\_angel\_howke@ciencias.unam.mx



FACULTAD DE CIENCIAS, UNAM  
Computación Concurrente

# INTRODUCCIÓN

El matemático holandés, T- Dekker. combinó la idea de los turnos y la idea de las variables de cerradura y las variables de advertencia con lo que fue el primero en diseñar una solución en software a problema de exclusión mutua sin el requisito de la de alternancia estricta Véase Dijkstra (1965.) para un análisis del algoritmo de Dekker.

En 1981. G. L. Peterson descubrió una forma más sencilla para conseguir la exclusión mutua, lo cual hizo obsoleta la solución de Dekker.

El algoritmo de Peterson se utiliza para la exclusión mutua y permite que dos procesos compartan un recurso de un solo uso sin conflictos. Utiliza solo memoria compartida para la comunicación. La fórmula de Peterson originalmente funcionó solo con dos procesos, pero desde entonces se ha generalizado durante más de dos.

El algoritmo de filtro generaliza el algoritmo de Peterson a  $N > 2$  procesos. Requiere una variable entera por proceso, almacenada en un registro atómico de un solo escritor y múltiples lectores, y  $N - 1$  variables adicionales en registros similares.

## EJERCICIOS

### 1.- Implementación del Algoritmo de Peterson

Implementar el algoritmo de Peterson visto en clase, esto con el fin de generar un **CANDADO**, con el fin de solucionar el problema de Exclusión Mutua, para 2 hilos.

Debera venir implementado con los métodos lock() y unlock(). (Se proporcionara un archivo que contiene dichos métodos).

### 2.- Implementación del Algoritmo del Filtro Modificado

Utilizando la idea del Algoritmo del Filtro visto en la clase, tienen que crear una implementación que permita pasar a lo más L hilos a la sección Crítica con  $L < N$ , es decir, que varios hilos puedan entrar a la sección crítica.

Deben venir los metodos acquire() y release()

# TEORÍA

1. Proponer 4 problemas donde se pueda utilizar el algoritmo de Peterson para su solución
2. Proponer 2 problemas donde se pueda utilizar el Algoritmo del Filtro

Responde las siguientes preguntas, justificando tu respuesta:

- ¿Los algoritmos cumplen con No Deadlock?
- ¿El Algoritmo de Peterson cumple con la propiedad de Justicia?
- ¿Cuál de estos algoritmos cumple con la propiedad Libre de Hambruna?
- ¿Cumplen con Exclusión Mutua?

## Extra (+1.5)

Implementa el Algoritmo del Filtro Clásico.

# ENTREGABLE

**Para la parte práctica, que es la programación, cada algoritmo deben de ir en un archivo distinto, todo debe ir debidamente documentado.**

**Para la parte teórica, es necesario que se haga a computadora, en el editor de su elección. Deben de poner las referencias bibliográficas en donde consultaron la información, esta debe de ir en formato APA.**

**Se les dará 0.5 extra si la realizan en LaTeX.**

**Todo debe de ir comprimido en un archivo ZIP, Taro cualquiera de tu preferencia, este debe de llevar el siguiente formato:**

**[NOMBRE DEL EQUIPO].[EXTENSION]**

**EJEMPLO:**

**LASZERATUNAS.zip**

**NOTA:**

**Esta practica se vera para la siguiente practica, por lo que si alguno de tus métodos no funciona, se vera complicada la siguiente.**

**Implementenlo con un lenguaje orientado a objetos, pues la siguiente practica se utilizara esta función, para resolver diversos problemas concurrentes.**