

# Sistemas Operativos

Departamento de Computación  
Facultad de Ciencias Exactas y Naturales  
Universidad de Buenos Aires

## Trabajo Práctico 2 - Scrable

Integrante	LU	Correo electrónico
Ortiz de Zarate, Juan Manuel	403/10	jmanuoz@gmail.com
Kujawski, Kevin	459/10	kevinkuja@gmail.com
Carreiro, Martin	45/10	carreiromartin@gmail.com

Reservado para la cátedra

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		

# Índice

1. Ejercicio 1	3
1.1. Librería RWLock sin inanición . . . . .	3
1.2. Pseudocódigo . . . . .	3

# 1. Ejercicio 1

## 1.1. Librería RWLock sin inanición

Para implementar una librería Read Write Lock libre de inanición, utilizamos semáforos POSIX mediante la librería de C semaphore.h. La librería creada por nosotros consta de los mismo métodos que la clase RWLock brindada por la cátedra. Disponemos de 4 atributos privados: readersQuantity (contador de lectores, inicializado en 0), mutexReaders (para implementar la sección crítica a la hora de tocar el atributo readersQuantity), writers (Semáforo para saber si hay un escritor esperando, inicializado en 1) y freeResource (para saber si el recurso esta disponible, inicializado en 1).

A continuación el pseudocódigo de la nueva libreria RWSemaphoreLock:

## 1.2. Pseudocódigo

```
WLOCK /**espero que no haya ningún escritor**/  
hayEscritores.wait()  
/**espero que se libere el recurso y lo tomo**/  
recurso.wait()
```

---

WUNLOCK

```
freeResource.signal() // Libero el recurso  
writers.signal() // Libero la exclusividad de los escritores
```

---

RLOCK

```
writers.wait() // Espero que no haya escritores, para no çolarmez evitar la inanición.  
writers.post() // Libero el semaforo para que los que vengan puedan seguir.  
mutexReaders.wait() // Tomo el mutex  
readersQuantity++ // Incremento la cantidad de lectores  
if(readersQuantity == 1) // Si soy el primero...  
freeResource.wait() // .. Tomo el recurso para los lectores  
mutexReaders.signal() // Libero el mutex
```

---

RUNLOCK

```
//espero el mutex para asegurarme el uso exclusivo sobre lectores  
mutexLectores.wai().  
//disminuyo la cantidad de lectores  
lectores--  
//si soy el ultimo lector que quedaba libero el recurso así puede tomarlo un escritor (en el caso que hubiese uno esperando)  
if(lectores == 0)  
recurso.signal()  
//libero el mutex  
mutexLectores.signal()
```