



# Sistemas Operativos 2021-2022

---

## Lectores-Escritores Problema de concurrencia



# Lectores-Escritores

## Descripción del problema

---

- Varios hilos comparten un **objeto**
  - Unos consultan su contenido: *lectores*
  - Otros lo modifican: *escritores*
  - Los *lectores* pueden leer simultáneamente
  - El acceso de un *escritor* es incompatible con cualquier otro hilo

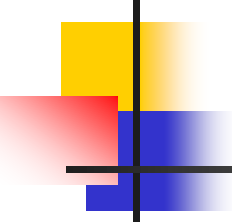


# Lectores-Escritores

## Detalles de implementación

---

- Solución
  - Semáforo *escr* binario
    - Garantiza la exclusión mutua al **objeto** (tal y como se nos indica en la teoría)
    - Es decir, excluye *escritores* cuando hay algún *lector* y cualquier hilo (sea *lector* o *escritor*), cuando hay un *escritor*
  - Necesitaremos además una variable *lect* que almacenará el numero de *lectores* que están accediendo, en un momento dado, al **objeto** (están en la SC)
  - Solo el primer *lector* necesita pedir permiso para entrar y solo el último *lector* puede liberar la SC



# Lectores-Escritores

## Implementación inicial

### **Función *lector***

```
// Si soy el primer lector
// que quiere entrar, pido
// permiso (el resto no)
lect++;
if (lect==1) P(escr);
LEER OBJETO;
lect--;
// Si soy el último lector
// libero la SC
if (lect==0) V(escr);
```

### **Función *escritor***

```
// Siempre debe pedir
// permiso para entrar
P(escr);
MODIFICAR OBJETO;
V(escr);
```

# Lectores-Escritores

## Implementación inicial

### Función *lector*

```
// Si soy el primer lector
// que quiere entrar, pido
// permiso (el resto no)
lect++;
if (lect==1) P(escr);
LEER OBJETO;
lect--;
// Si soy el último lector
// libero la SC
if (lect==0) V(escr);
```

### Función *escritor*

```
// Siempre debe pedir
// permiso para entrar
P(escr);
MODIFICAR OBJETO;
V(escr);
```

**Secciones críticas de *OBJETO***

# Lectores-Escritores

## Implementación inicial

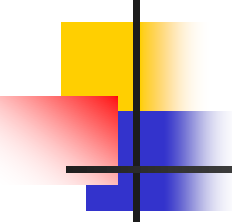
### Función *lector*

```
// Si soy el primer lector
// que quiere entrar, pido
// permiso (el resto no)
lect++;
if (lect==1) P(escr);
LEER OBJETO;
lect--;
// Si soy el último lector
// libero la SC
if (lect==0) V(escr);
```

### Función *escritor*

```
// Siempre debe pedir
// permiso para entrar
P(escr);
MODIFICAR OBJETO;
V(escr);
```

**Rodear de operaciones P y V  
las secciones críticas**



# Lectores-Escritores

## Implementación inicial

### Función *lector*

```
// Si soy el primer lector
// que quiere entrar, pido
// permiso (el resto no)
lect++;
if (lect==1) P(escr) ;
LEER OBJETO;
lect--;
// Si soy el último lector
// libero la SC
if (lect==0) V(escr) ;
```

**¡Falta garantizar la exclusión mutua de la variable *lect*!!**

### Función *escritor*

```
// Siempre debe pedir
// permiso para entrar
P(escr) ;
MODIFICAR OBJETO;
V(escr) ;
```

#### Nota:

- Valor inicial del *contador* del semáforo *escr* es 1
- Valor inicial de variable entera *lect* es 0



# Lectores-Escritores

## Exclusión mutua de la variable global *lect*

---

- Añadir a la anterior solución:
  - Semáforo *mutex* binario
    - Garantiza exclusión mutua al actualizar n<sup>o</sup> lectores
      - Variable *lect*: n<sup>o</sup> de lectores



# Lectores-Escritores

## Implementación final

### Función *lector*

P(mutex);

lect++;  
if (lect==1) P(escr);

V(mutex);

LEER OBJETO;

P(mutex);

lect--;  
if (lect==0) V(escr);

V(mutex);

### Función *escritor*

P(escr);

MODIFICAR OBJETO;

V(escr);

**Secciones críticas de *lect***

# Lectores-Escritores

## Implementación final

### Función *lector*

**P(mutex) ;**

```
lect++;  
if (lect==1) P(escr);
```

**V(mutex) ;**

LEER OBJETO;

**P(mutex) ;**

```
lect--;  
if (lect==0) V(escr);
```

**V(mutex) ;**

### Función *escritor*

P(escr) ;

MODIFICAR OBJETO;

V(escr) ;

**Rodear de operaciones P y V  
las secciones críticas**



# Lectores-Escritores

## Implementación final

### Función *lector*

**P(mutex) ;**

```
lect++;  
if (lect==1) P(escr) ;
```

**V(mutex) ;**

LEER OBJETO;

**P(mutex) ;**

```
lect--;  
if (lect==0) V(escr) ;
```

**V(mutex) ;**

### Función *escritor*

**P(escr) ;**

MODIFICAR OBJETO;

**V(escr) ;**

### Nota:

- Valores iniciales del contador de *mutex* y *escr* es 1
- Valor inicial de *lect* es 0



# Lectores-Escritores

## Casos

---

- Se indican, a continuación, distintas situaciones que se pueden dar con este problema y como evolucionarían las variables y los semáforos

# Lectores-Escritores

## Casos

- **Caso 1:** 1<sup>er</sup> lector y ningún escritor
  - $P(mutex) \rightarrow mutex=0$
  - $lect++ \rightarrow lect=1$
  - $P(escr) \rightarrow escr=0$  (bloquear a los escritores)
  - $V(mutex) \rightarrow mutex=1$
  - *LEER OBJETO*
- **Caso 2:** caso 1 y llegan escritores
  - $P(escr) \rightarrow$  escritor bloqueado
- **Caso 3:** último lector termina
  - $P(mutex) \rightarrow mutex=0$
  - $lect-- \rightarrow lect=0$
  - $V(escr) \rightarrow escr=?$  (desbloquear a un escritor o bien  $escr=1$ )
  - $V(mutex) \rightarrow mutex=1$
- **Caso 4:** llegan 2 primeros lectores y ningún escritor
  - Uno de los lectores hace antes  $P(mutex) \rightarrow mutex=0$
  - El otro lector se bloquea en  $P(mutex)$
- **Caso 5:** 1<sup>er</sup> escritor (E) y luego 2 lectores (L1 y L2)
  - (E)  $P(escr) \rightarrow escr=0$
  - (E) MODIFICAR OBJETO
  - (L1)  $P(mutex) \rightarrow mutex=0$
  - (L1)  $lect++ \rightarrow lect=1$
  - (L1)  $P(escr) \rightarrow$  lector bloqueado
  - (L2)  $P(mutex) \rightarrow$  lector bloqueado