

# Condvars

Técnicas de Programación Concurrente

# Condvars

- *Condition variables*
- Bloquean un thread hasta que se cumpla una condición
- La condición puede ser arbitrariamente compleja y se verifica siempre dentro de un mutex
- Se puede despertar a uno a **todos** los threads que estén esperando en la condvar

# Condvars

```
let pair = Arc::new((Mutex::new(true), Condvar::new()));
```

## Thread actualizando la condvar

```
thread::spawn(move || {  
    let (lock, cvar) = &*pair_clone;  
    // "Expensive computation"  
    thread::sleep(Duration::from_millis(1000));  
    let mut pending = lock.lock().unwrap();  
    *pending = false;  
    cvar.notify_all();  
});
```

## Thread esperando en la condvar

```
let (lock, cvar) = &*pair;  
  
// As long as the value inside the  
// `Mutex<bool>` is `true`, we wait.  
  
let _guard =  
    cvar.wait_while(lock.lock().unwrap(),  
        |pending| { *pending }).unwrap();
```

# Un ejemplo de condvars en la vida real

Servo un rendering engine de browser implementado en Rust.

`fetch()` es una API web candidata a ser el nuevo standard para realizar requests HTTP desde JS

La implementación de cache de `fetch` de Servo sufría de una race condition cuando concurrentemente se realizaban dos requests al mismo recurso.

Lo solucionaron con una condvar

<https://github.com/servo/servo/pull/24318/files#diff-5b7cef56895a34b09812a744ac8376802d4c38b76115148d8f9e3fc3458ca42eR84>



# Condvars

Ejemplos:

- Hello world
- Spurious Wakeup
- Wait on multiple
- Multiple Waiters

# Problema del Lector-Escritor

Un estado se comparte entre varios procesos.

Algunos procesos necesitan actualizar dicho estado, mientras que otros solo necesitan leerlo.

Mientras que un proceso está leyendo el estado, otros pueden leerlo, pero ninguno modificarlo

Mientras que un proceso está modificando el estado, ningun otro puede leerlo ni modificarlo.



Pierre Jacques Courtois

# Problema del Lector-Escritor

Process 1	Process 2	Allowed / Not Allowed
Writing	Writing	Not Allowed
Reading	Writing	Not Allowed
Writing	Reading	Not Allowed
Reading	Reading	Allowed

# Problema del Lector-Escritor

- Starvation
- Writer Preference
- Fair



# Problema de los fumadores

Implementemos el problema de los fumadores usando condvars

Recordamos: Consideremos que para fumar un cigarrillo se necesitan 3 ingredientes: tabaco, papel y fósforos.

Hay 3 fumadores alrededor de una mesa. Cada uno tiene una cantidad infinita de uno solo de los ingredientes y necesita los otros dos para fumar. Cada fumador posee un ingrediente distinto y le faltan los otros dos.

Existe además un agente que pone aleatoriamente dos ingredientes en la mesa. El fumador que los necesita los tomará para hacer su cigarrillo y fumará un rato. Cuando el fumador termina, el agente pone otros dos ingredientes.

# Referencias

Concurrent control with "readers" and "writers" - <https://dl.acm.org/doi/10.1145/362759.362813>