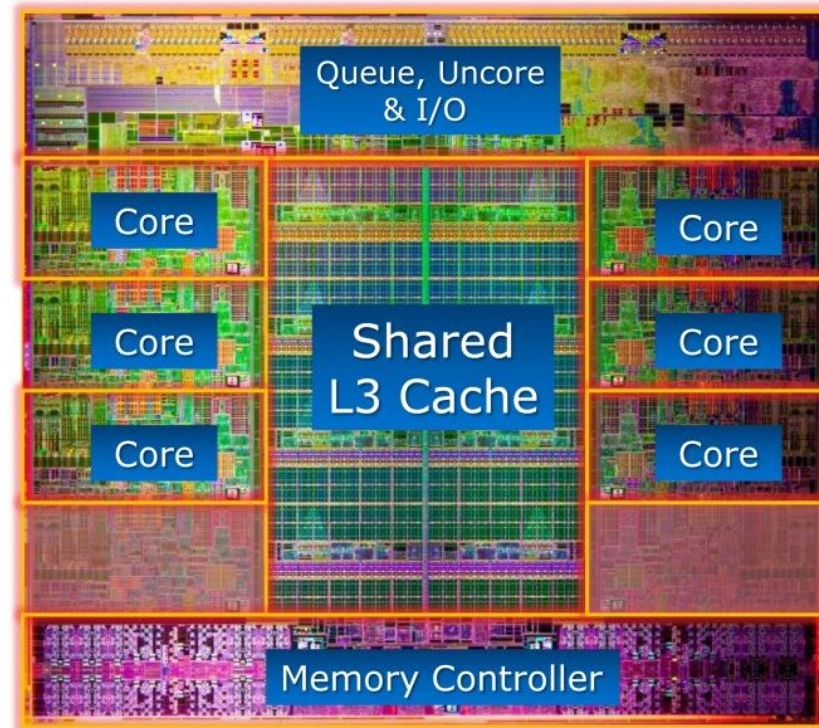


# Concurrencia

Dr. Francisco E. Cabrera

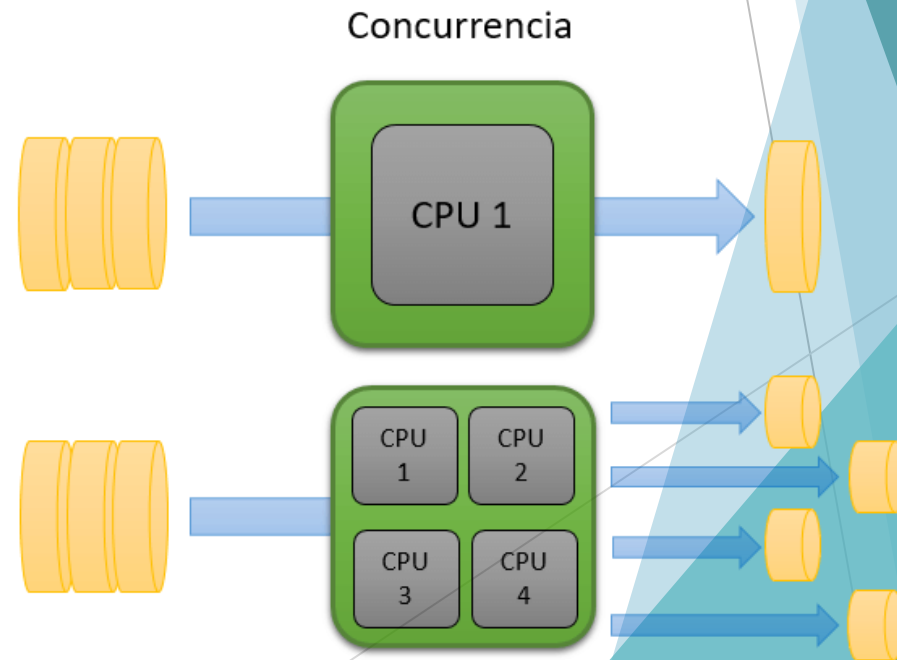
# Concurrencia



# Concurrencia

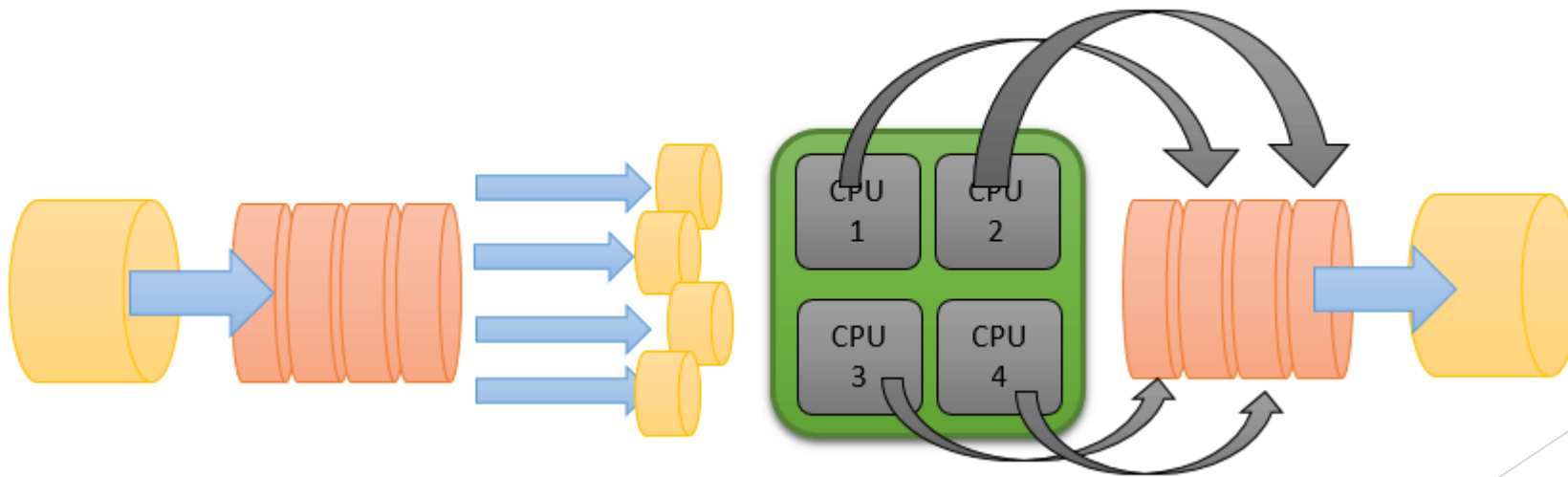
Es la capacidad del procesador para ejecutar mas de un proceso al mismo tiempo.

- ▶ El Task Scheduler del SO se encarga de gestionarlo.
  - ▶ Decide qué proceso se va a ejecutar y cuándo.
  - ▶ Gestiona los recursos.
  - ▶ Prioriza procesos.
  - ▶ Interrumpe procesos.
  - ▶ Realiza cambios de contexto.



# Paralelismo

Consiste en aprovechar la concurrencia para realizar una tarea más rápido.



# Paralelismo

- ▶ Dividimos el problema inicial en tareas más pequeñas.
  - ▶ Que podamos realizar al mismo tiempo.
- ▶ Ejecutamos de manera concurrente estas tareas.
- ▶ Una vez estén todas completadas, unimos los resultados.



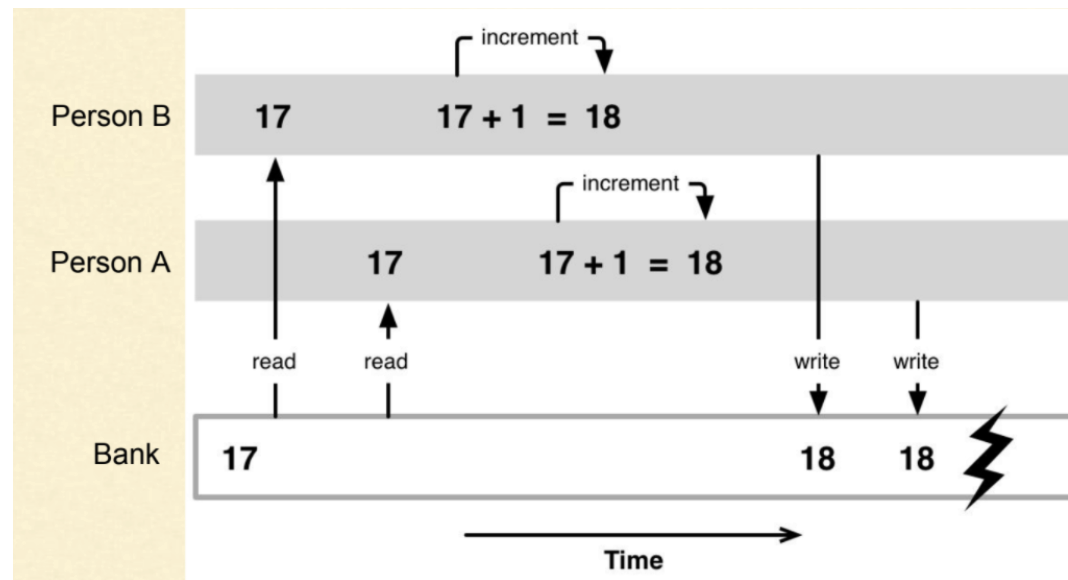
# Problemas comunes en concurrencia

Al ejecutar varios procesos al mismo podemos tener los siguientes problemas:

- ▶ Condiciones de carrera (Race conditions).
- ▶ Interbloqueos (Deadlocks).
- ▶ Sincronización.

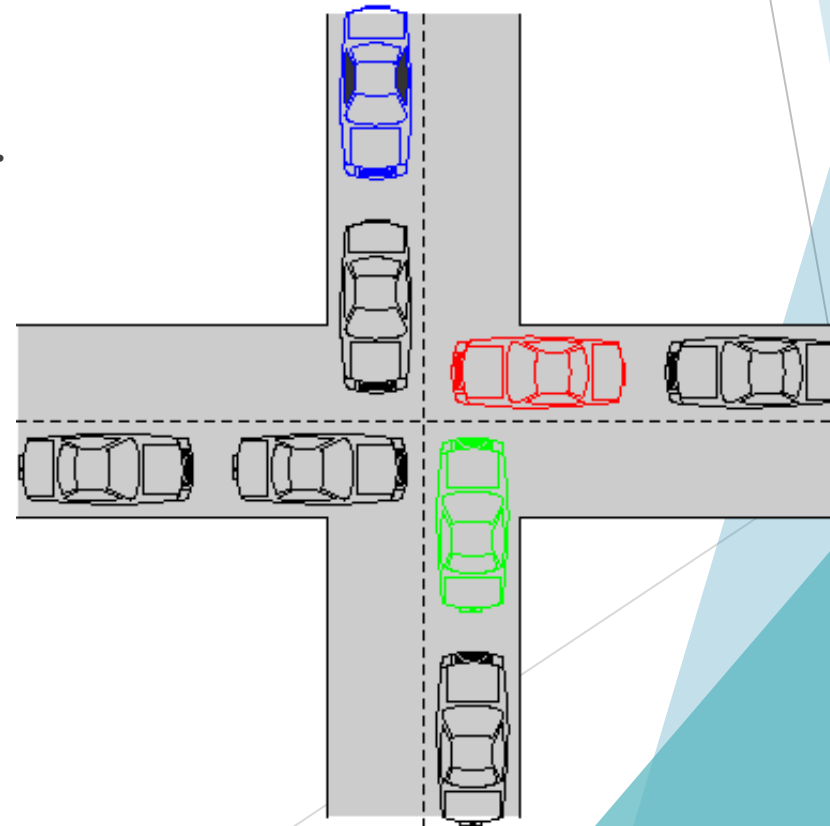
# Race Conditions

- ▶ Se producen cuando varios procesos acceden y modifican un mismo recurso.
- ▶ Sin sincronización puede llevar a resultados impredecibles.



# Deadlock

- ▶ Ocurre cuando dos o mas procesos se bloquean mutuamente.
  - ▶ Proceso A espera a proceso B para continuar.
  - ▶ Proceso B espera a proceso A para continuar.
- ▶ En esta situación ningún proceso puede continuar.





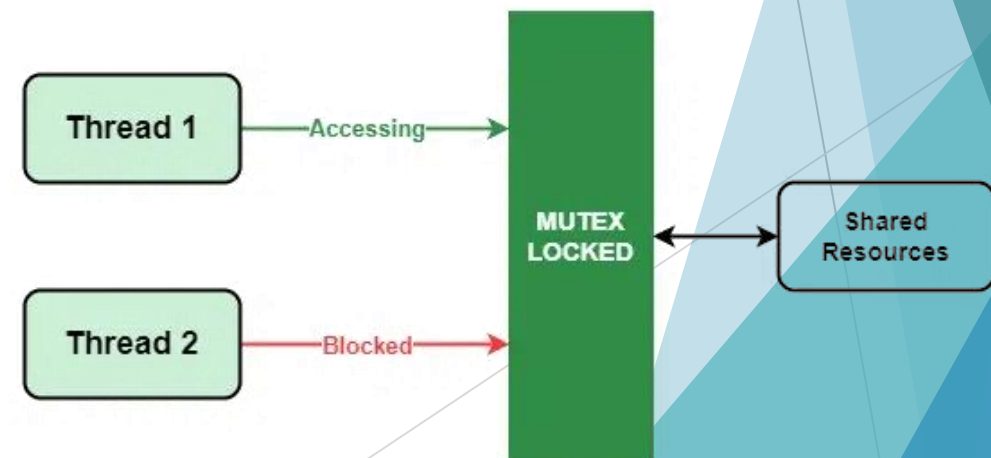
# Sincronización

- ▶ Esencial cuando se emplean recursos compartidos.
  - ▶ Técnicas y mecanismos que aseguran que las operaciones se realicen de manera ordenada y segura.
- ▶ Herramientas de sincronización:
  - ▶ Bloqueos (Locks y Mutex).
  - ▶ Semáforos.
  - ▶ Colas (Queues).

# Sincronización

## Cerraduras (Locks o Mutex)

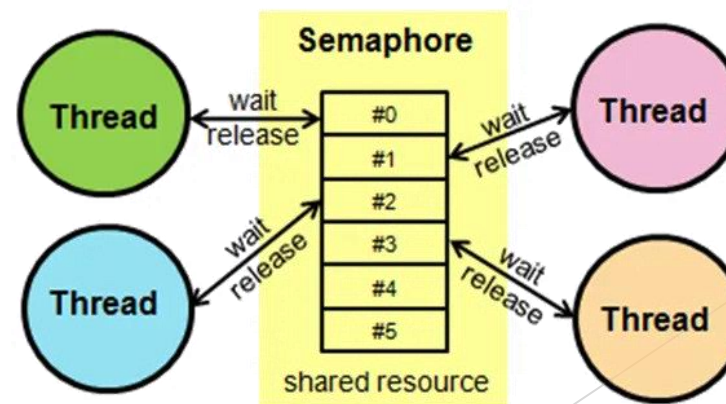
- ▶ Son mecanismos que permiten que solo un hilo acceda a un mecanismo en un momento dado.
  - ▶ Antes de acceder al recurso, el hilo adquiere el lock.
  - ▶ Hasta que el hilo no libere el lock, ningún otro proceso puede acceder al recurso.
    - ▶ Cuando necesiten el recurso, deberán esperar.
    - ▶ Cuando deja de utilizarlo, el hilo libera el lock.
- ▶ Se usa para prevenir las Race Conditions.



# Sincronización

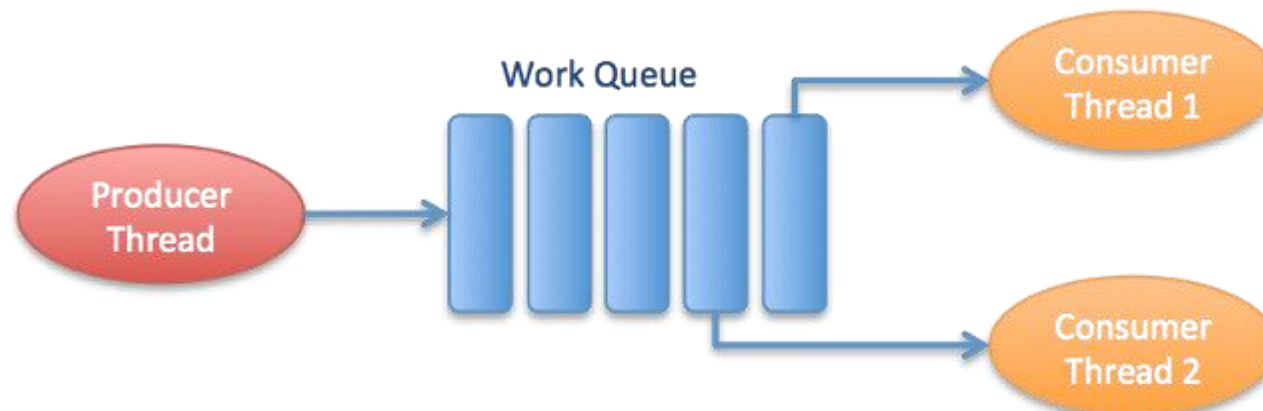
## Semáforo

- ▶ Utiliza un contador para controlar el acceso a recursos compartidos.
- ▶ El contador indica el número de accesos permitidos simultáneamente.
- ▶ Permite que un número específico de hilos acceda de forma simultanea.
  - ▶ Controla el uso del recurso compartido.



# Colas

- ▶ Permiten la comunicación entre hilos y procesos.
- ▶ Gestionan los elementos en orden FIFO (First-in, First-out) e implementan mecanismos de bloqueo interno.
- ▶ Se usan para modelos de productor-consumidor.
  - ▶ Unos hilos producen datos y otros los consumen.
- ▶ Facilitan la coordinación sin tener que gestionar manualmente los locks.



# Ejemplo

- ▶ Queremos aplicar la función  $f(x)$  a cada elemento de una lista.

# Ejemplo

- ▶ Enfoque iterativo

```
res = []  
for x in datos:  
    resultados.append(f(x))
```

- ▶ Fácil implementación.
- ▶ Sin problemas de sincronización.
- ▶ No aprovecha múltiples núcleos.
- ▶ Tiempo (t) = Suma de los tiempos individuales.

- ▶ Enfoque concurrente

Dividir datos en k partes

Para cada parte en paralelo:

procesar aplicando f(x) a cada elemento.

Combinar los resultados de todas las partes.

- ▶ Aprovecha la capacidad de múltiples núcleos.
- ▶ Posible reducción del tiempo (idealmente  $t/\text{núcleos}$ ).
- ▶ Introduce overhead
  - ▶ Tiempo de dividir, sincronizar y combinar el trabajo.

# Consideraciones

¿Qué enfoque debería elegir?

## Proceso Iterativo

- ▶ Tareas pequeñas.
- ▶ Coordinar las subtarear es computacionalmente costoso.
- ▶ El overhead es mayor que el beneficio.
- ▶ No tengo una ventaja apreciable en un enfoque multinucleo.

## Concurrencia

- ▶ Cada subtarea tiene una carga de trabajo significativa.
  - ▶ Por ejemplo, una función difícil de calcular.
- ▶ Tengo un problema grande y computacionalmente costoso.

# Ejemplo práctico

Contar números primos.

Dado dos números, contar cuántos números primos hay entre ellos.

- ▶ Enfoque Iterativo:
  - ▶ Recorrer el rango completo.
- ▶ Enfoque Concurrente:
  - ▶ Dividir el rango en N partes.
  - ▶ Lanzar N procesos en el que cada uno cuente los primos en su rango.
  - ▶ Sumarlo todo al final.