

Process Synchronization

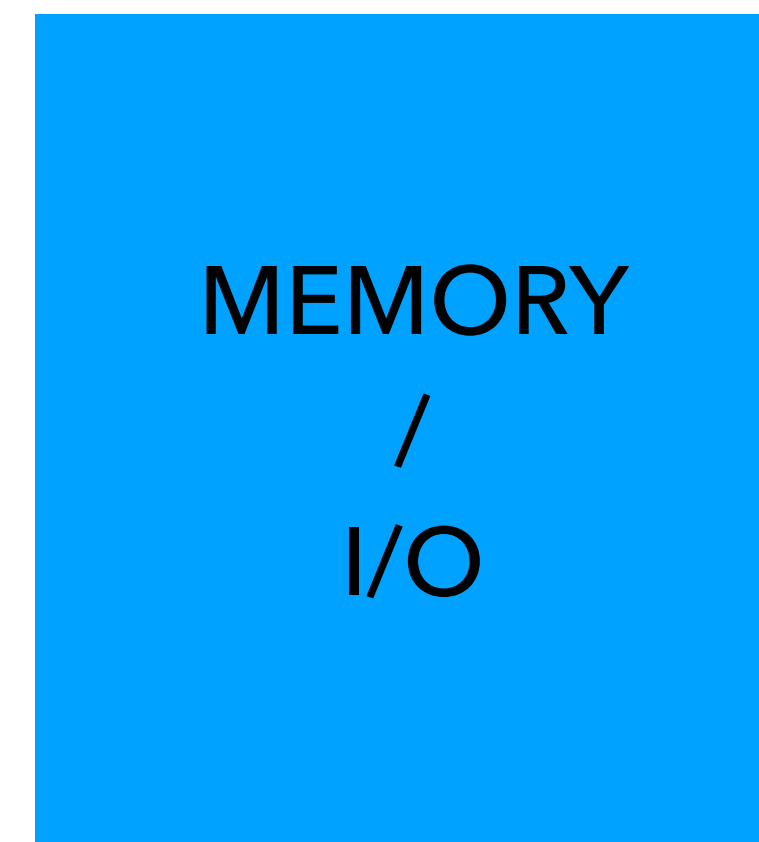
동기화의 문제와 원인

데이터의 접근

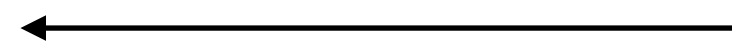
Execution - Box



Storage - Box



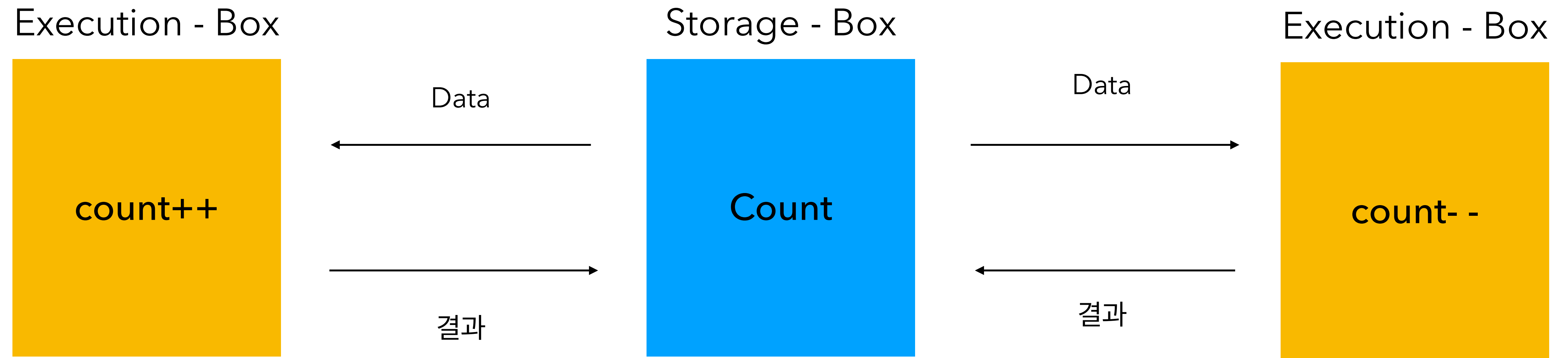
Data



결과



Race Condition



1을 더하고(count++) 1을 빼면(count-) 원래 값이 저장되어야 한다. 하지만..?

Race Condition

여러 프로세스들이 **동시**에 공유 데이터를 **접근**하는 상황
데이터의 최종 연산 결과는 마지막에 그 데이터를 다룬 프로세스에 따라 달라짐

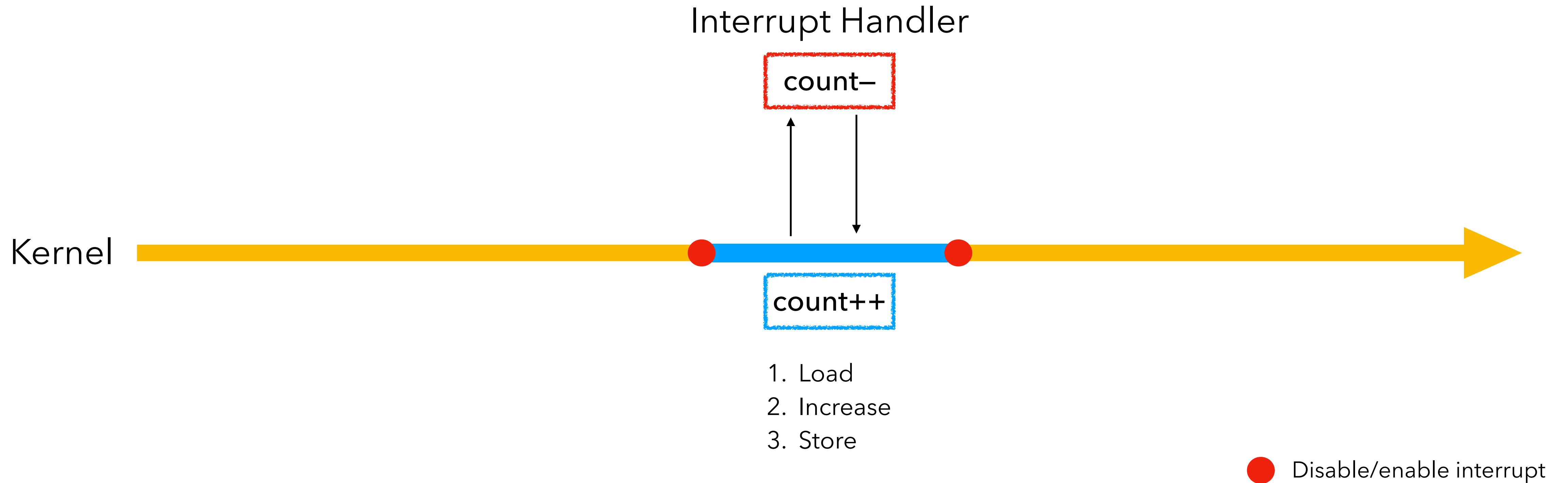
문제는 운영체제가 끼어드는 경우 발생할 수 있는데..
(System Call을 할 경우)

1. 커널 수행 중 **인터럽트** 발생할 때

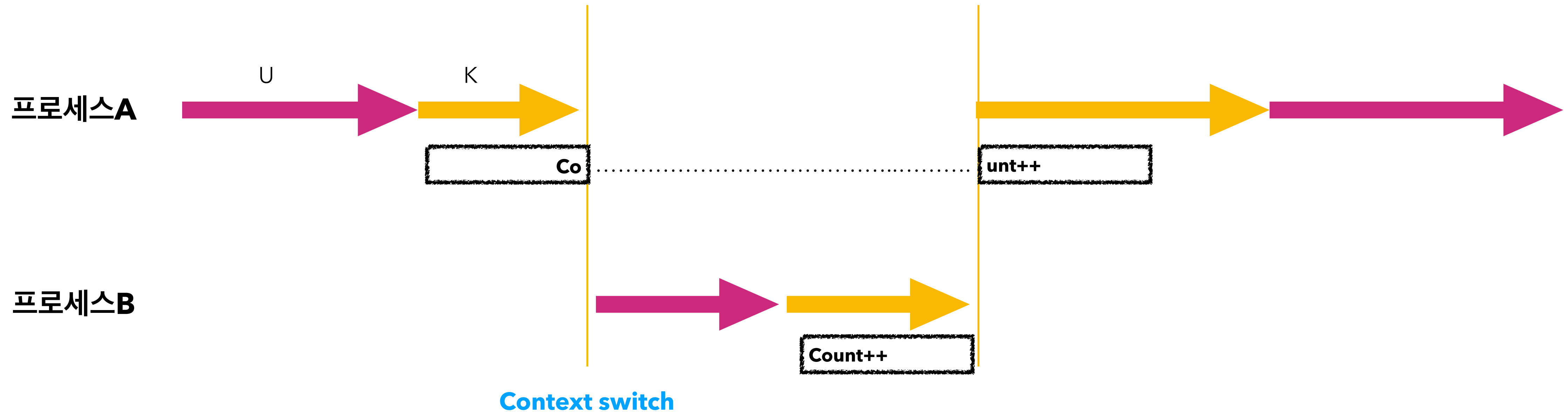
2. 프로세스가 **System call** 하여 커널 모드로 수행 중인데 **Context switch**가 일어날 때

3. 멀티 프로세서에서 **shared memory** 내의 커널 데이터 공유할 때

1. 커널 수행 중 인터럽트 발생할 때

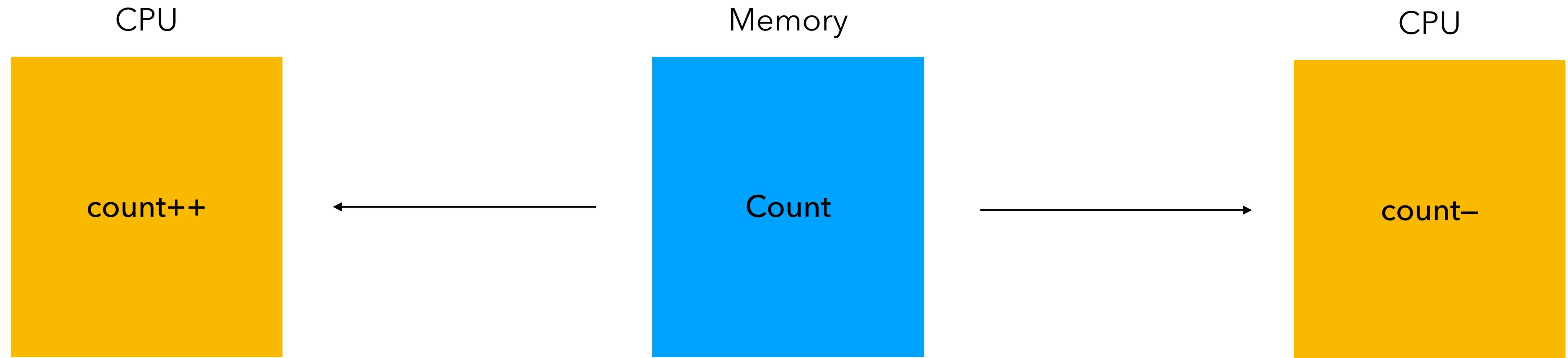


2. 프로세스가 System call 하여 커널 모드로 수행중인데 Context switch가 일어날 때



어떻게 해결해야 할까?

3. 멀티프로세서에서 Shared memory의 커널 데이터 공유할 때



어떤 CPU가 마지막으로 count를 수행하고 저장 했는가에 따라서 Race Condition 발생 가능



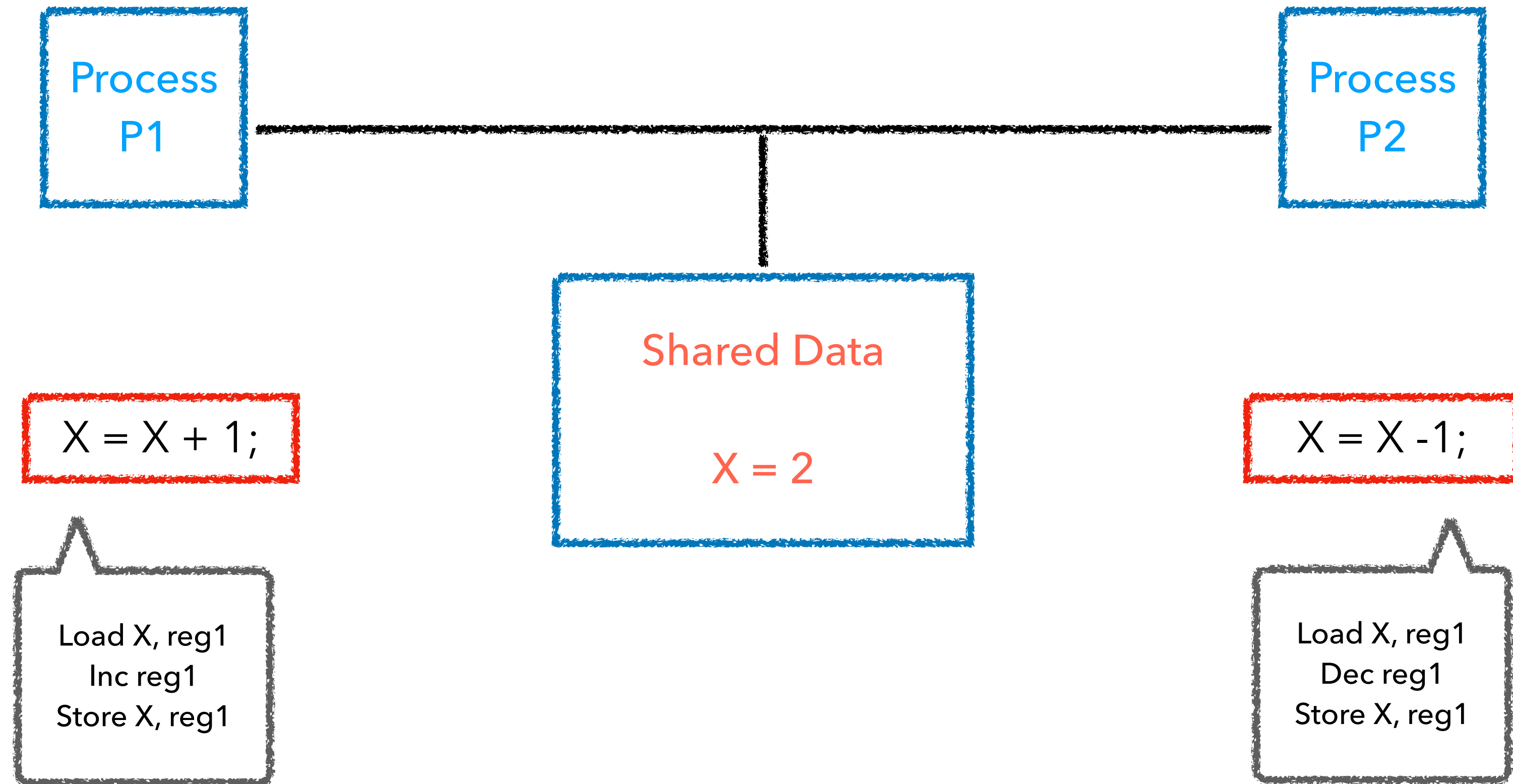
- 1.한번에 하나의 CPU만 커널에 들어갈 수 있게 하자.
- 2.커널 내부에 있는 각 공유 데이터에 접근 할 때마다 데이터에 대한 lock/unlock 수행하자.

Process Synchronization 문제

일관성 유지를 위해

협력 프로세스 간의 실행순서를 정해주는 메커니즘이 필요

Race Condition의 예 & Critical Section



감사합니다

