

동기와 비동기

블로킹과 논-블로킹

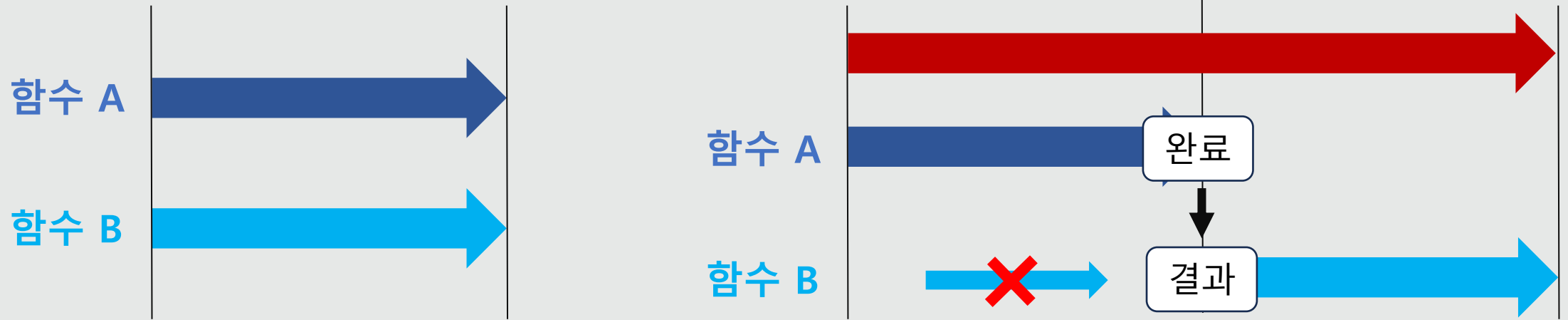
동기 (*Synchronous*) VS 비동기 (*Asynchronous*)

Syn + chrono
함께 시간

데이터베이스에서 같은 시간에
두 개의 다른 데이터베이스 저장소에서 같은 데이터를 가지고 있도록 맞춰주는 것

대상들의 시간이 맞춰지는가?

동기 (Synchronous)

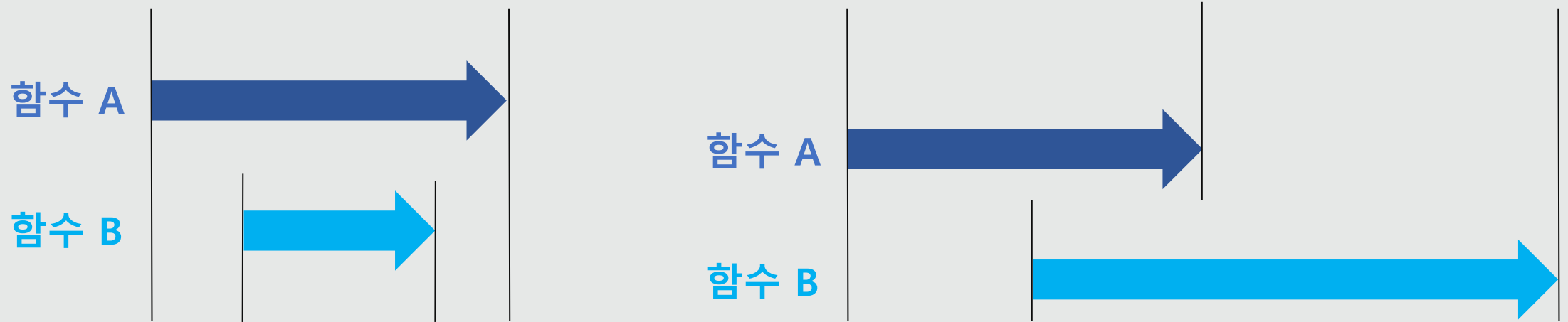


작업들이 같이 시작해서 같이 끝났다.
요청한 작업에 대해 완료 여부를 따져 순차대로 바로 처리한다.



둘 이상의 작업이 동시에 시작해, 동시에 끝나게 한다.
작업 중에 다른 작업이 끼어들지 못하게 한다.

비동기 (*Asynchronous*)



작업들의 시작과 끝이 다르다.
작업이 완료될 때까지 기다리지 않고 다른 작업을 실행한다.



작업 중에 끼어들어도 괜찮다.
작업들의 시작과 종료 시기를 신경 쓰지 않는다.
작업을 순차적으로 처리 하지 않아도 되고 바로 처리 하지 않아도 된다.

블로킹 (*Blocking*) VS 논-블로킹 (*Non-Blocking*)

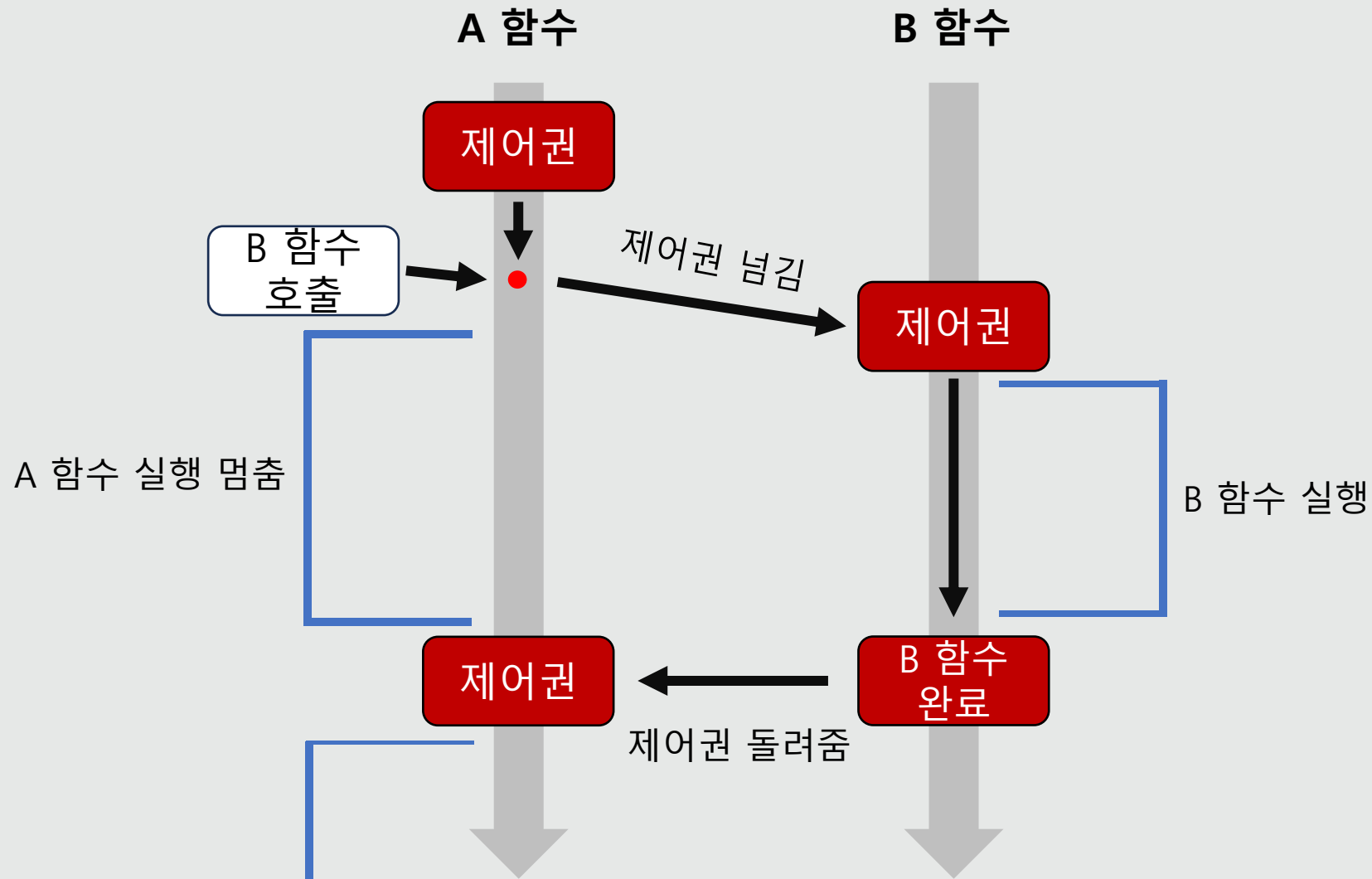
제어권

Blocking과 Non-Blocking은 호출된 함수가 호출한 함수에게 제어권을 바로 주느냐 안 주느냐로 구분된다.

누가 제어권을 가지고 있고 제어권을 어떻게 처리하느냐,
처리하고 있는 작업이 전체 흐름을 막느냐 안 막느냐에 따라 달라진다.

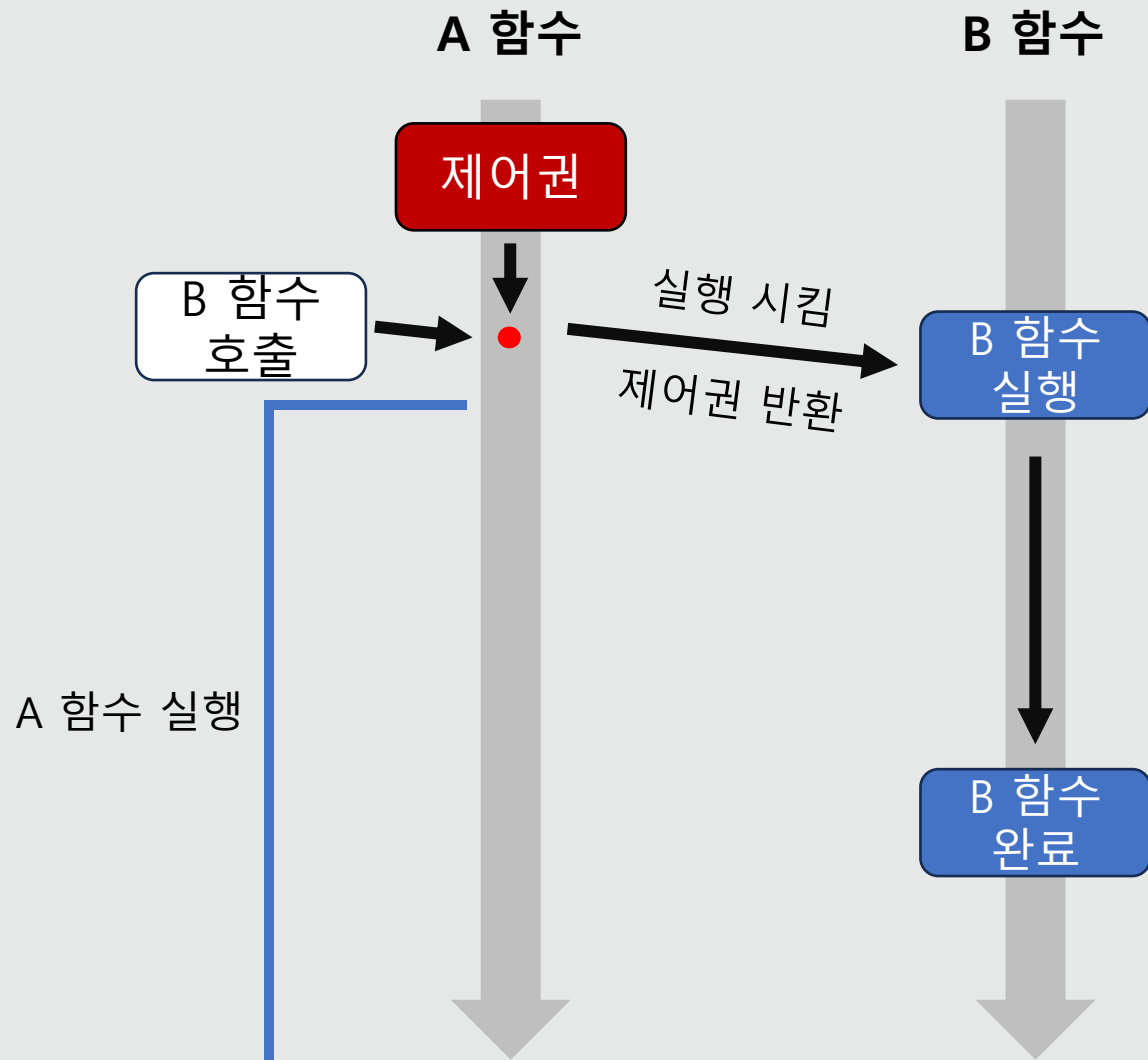
블로킹 (*Blocking*)

다른 작업의 실행이 현재 작업의 실행을 막는다.



논-블로킹 (*Non-Blocking*)

다른 작업의 실행이 현재 작업을 막지 않는다.



동기 = 블로킹? 비동기 = 논-블로킹?

동기 (Synchronous)
비동기 (Asynchronous)

시간, 타이밍
대상들의 시간을
일치시키는가?

블로킹 (Blocking)
논-블로킹 (Non-Blocking)

제어권
제어할 수 없는 대상을
어떻게 처리하는가?

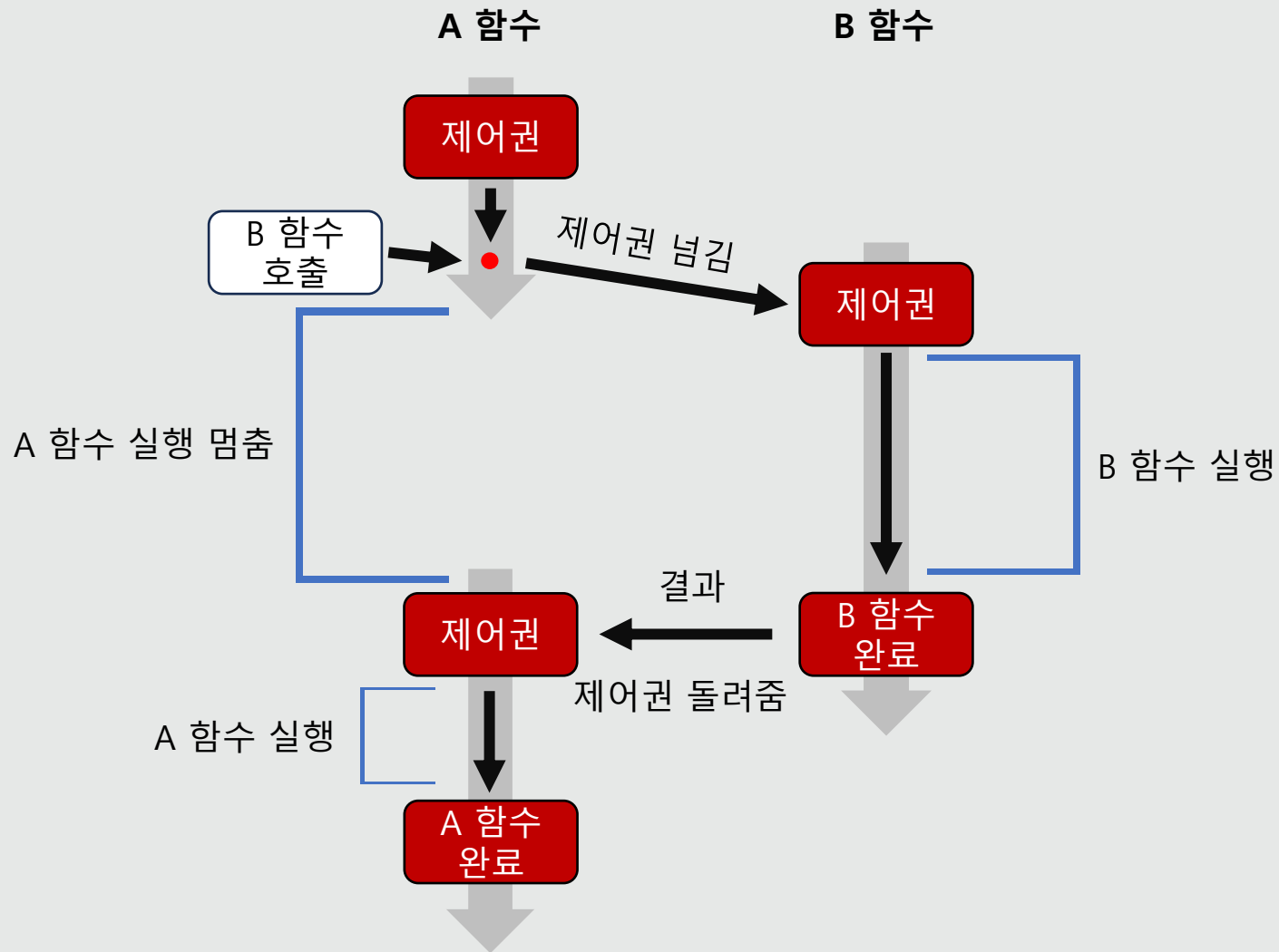
4가지 조합

1. 동기와 블로킹
2. 동기와 논-블로킹
3. 비동기와 블로킹
4. 비동기와 논-블로킹

4가지 조합

1. 동기화 블로킹

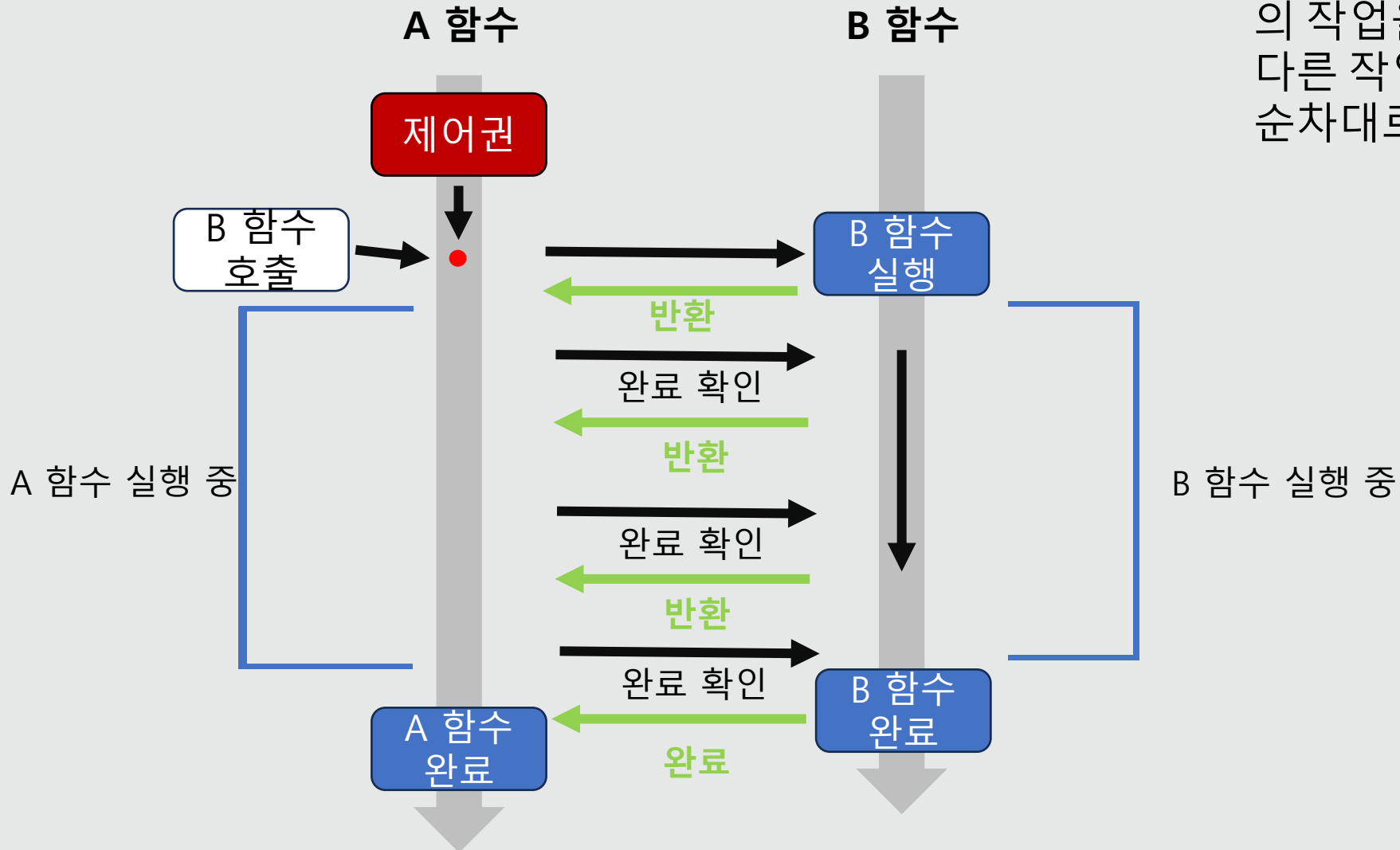
다른 작업이 진행되는 동안 자신의 작업을 처리하고 앎고(Blocking), 다른 작업의 완료 여부를 바로 받아 순차적으로 처리하는 (Sync) 방식



4가지 조합

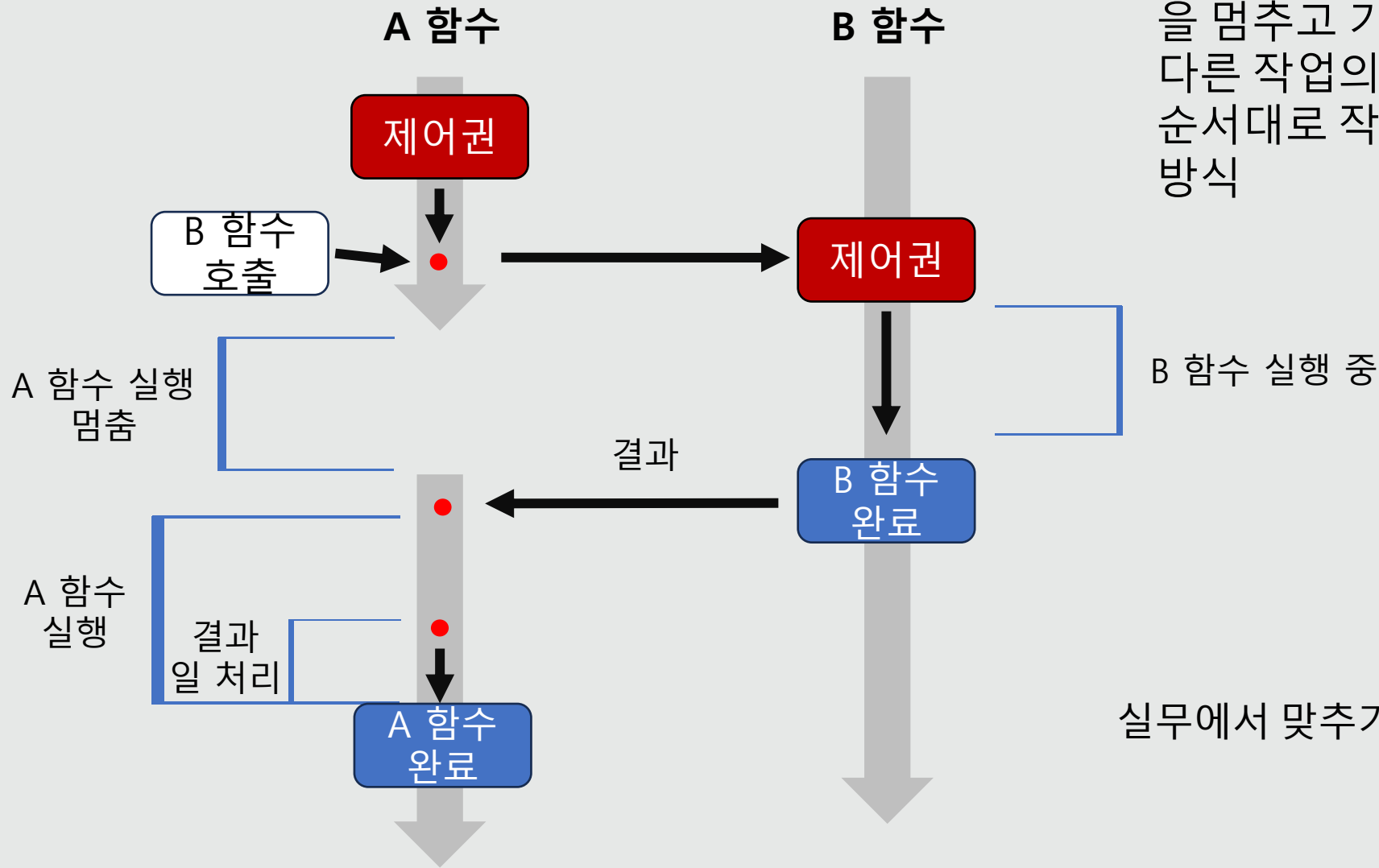
2. 동기화 및 논-블로킹

다른 작업이 진행되는 동안에도 자신의 작업을 처리하고 (Non Blocking), 다른 작업의 결과를 처리하여 작업을 순차대로 수행하는 (Sync) 방식



4가지 조합

3. 비동기와 블로킹



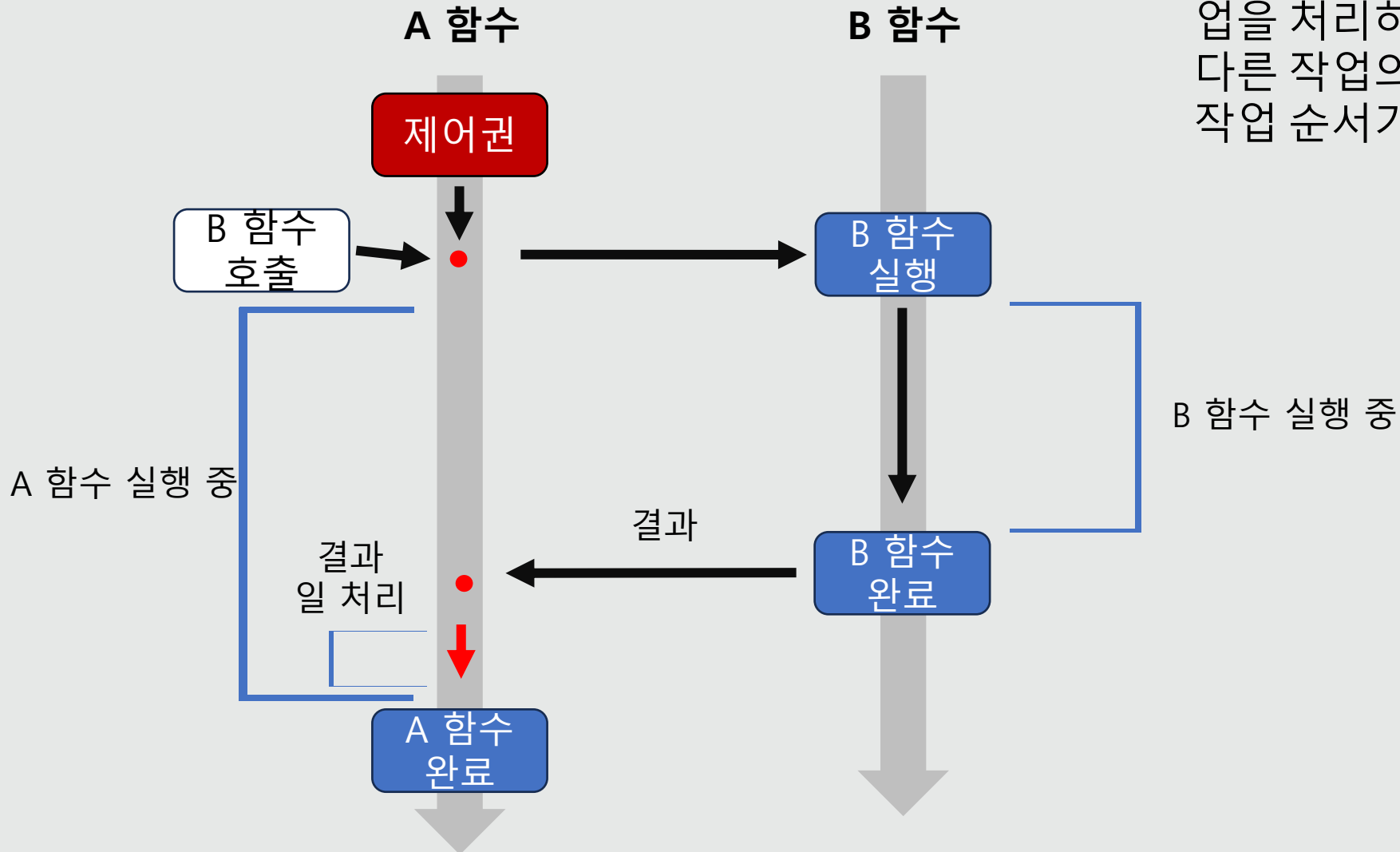
다른 작업이 진행되는 동안 자신의 작업을 멈추고 기다리는 (Blocking),
다른 작업의 결과를 바로 처리하지 않아
순서대로 작업을 수행하지 않는 (Async)
방식

실무에서 맞추기가 쉽지 않아 다룰 일이 거의 없다.

4가지 조합

4. 비동기와 논-블로킹

다른 작업이 진행되는 동안에 자신의 작업을 처리하고(Non Blocking), 다른 작업의 결과를 바로 처리하지 않아 작업 순서가 지켜지지 않는 (Async) 방식



Reference

<https://inpa.tistory.com/entry/%EB%B3%B9-%EB%A6%B4%EB%A6%B4-%EB%B3%B9%EB%B3%B9-%EA%B3%B9-%EC%A0%B9>

<https://velog.io/@nittre/%EB%B3%B9-Vs.-%EB%B3%B9-%EB%B3%B9-Vs.-%EB%B3%B9>

<https://velog.io/@ddclub12/%EB%B3%B9%EB%A6%B4%EB%A6%B4-VS-%EB%B3%B9%EB%B3%B9>

유튜브

<https://www.youtube.com/watch?v=oEloqGd-Sns>

<https://www.youtube.com/watch?v=ldpkfygWIMk>

<https://www.youtube.com/watch?v=ohz7uHnza6A>

감사

합니다