

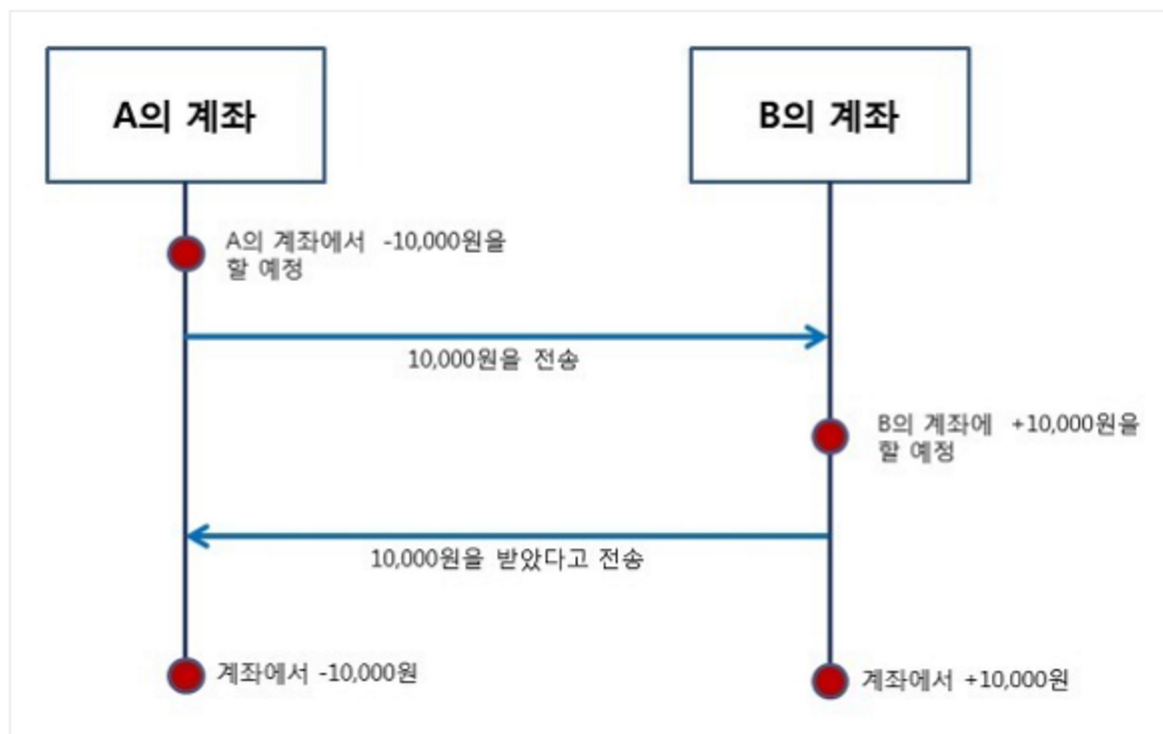
동기(Synchronous)

와 비동기(Asynchronous)

2022년 5월 17일 화요일 오전 9:53

동기

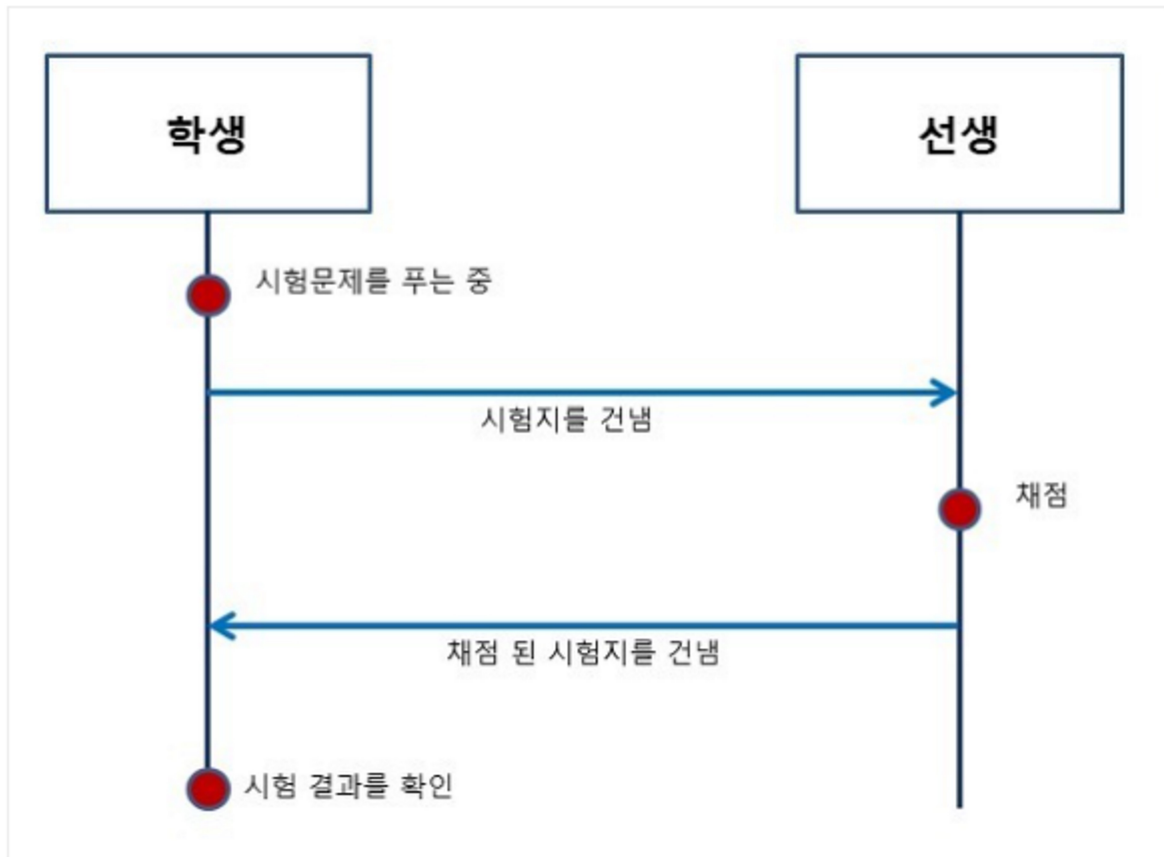
1. 요청과 그 결과가 한 자리에서 동시에 일어남
2. A노드와 B노드 사이의 작업 처리 단위를 동시에 맞춤
3. 설계가 간단하고 직관적이지만, 결과가 주어질 때까지 아무것도 못하고 대기해야 하는 단점



- A. A의 계좌는 10,000원을 뺄 생각을 하고 있다.
- B. A의 계좌가 B의 계좌에 10,000원을 송금한다.
- C. B의 계좌는 10,000원을 받았다는 걸 인지하고, A의 계좌에 10,000원을 받았다고 전송한다.
- D. A, B 계좌 각각 차감과 증가가 동시에 발생하였다.

비동기

1. 요청 결과가 동시에 일어나지 않음
2. 노드 사이의 작업 처리 단위를 동시에 맞추지 않음
3. 동기보다 설계가 복잡하지만, 결과가 주어지는데 시간이 걸리더라도 그 시간 동안 다른 작업을 활용함으로써 자원을 효율적으로 사용 가능



- A. 학생은 시험 문제를 푼다.
- B. 시험 문제를 모두 푼 학생은 선생님에게 전송한다.
- C. 선생은 학생의 시험지를 채점한다.
- D. 채점이 다 된 시험지를 학생에게 전송한다.
- E. 학생은 선생이 전송한 시험지를 받아 결과를 확인한다.
- F. 서로의 행위(목적)가 다르기때문에 둘의 작업 처리 시간은 일치하지 않고, 일치하지 않아도 된다.

블록과 논블록

1. 학생이 시험지를 받기만 기다린다면 블록 상태
2. 학생이 시험지를 받을 때 까지 공부, 게임 등 다른 일을 한다면 논블록 상태