

< 동기 방식 vs 비동기 방식 >

* 동기 방식 (Synchronous)

직렬적으로 작업을 처리. ~~요청을 하고~~ 요청을 하고 기다렸다가 응답을 받음
요청을 하는 주체와 응답을 받은 주체가 같음. ~~결과~~ 결과의 순서가 보장

* 비동기 방식 (Asynchronous)

요청 후 기다리지 않고 바로 다음 작업을 처리. 요청을 하고 응답을 받은 주체가 다름.

* ~~Processor~~ Processor의 수가 같을 때 동기 vs 비동기

작업을 처리하는데 2초. 프로세서의 수가 5개. 20개의 job.



~~Requester~~ Requester가 여러개의 차이.

* ~~시간~~ 시간당 처리량이나 처리속도의 차이는 크지 않지만

처리중 다른 작업을 할 수 있느냐 없느냐, 혼란성의 차이.

Snapshot의 경우 동기식으로 처리하면 다음 요청을 받을 수 없다 (처리중기름).

Snapshot의 수행시간이 길지 않음. 대용량인 것도 있기 때문에

요청자 입장에서는 느리다고 생각될 위험도가 있음.

* OSI 7 Layer.

1. 물리계층 - 데이터를 비트로 바꾸어 전송.

2. 데이터링크 ~~계층~~ - MAC주소

3. ~~네트워크~~ 네트워크

4. 전송계층 레이어

5. 세션

6. 표현 ~~계층~~ (presentation)

7. Application.

preorder → 전위 순회: root부터 방문.

inorder → 중위 순회: 왼쪽부터 방문.

post order → 후위 순회: 오른쪽부터 방문.

< Stateless vs Stateful >

* Stateful: 이전 동작에 따른 상태나 데이터를 저장하고 다음 수행시 해당 데이터를 사용.

* Stateless: 이전 동작의 상태 및 데이터를 저장하지 않고 다음 수행시에도 처음 수행한 것과 마찬가지로 동작.

* Monitoring agent를 stateless 하게 설계한 이유.

1. 설계의 단순화 → stateful 한 설계를 위해서는 추가적인 요소가 필요하여 이에 대한 부하를 고려하여 간단한 구조로 설계.

2. 비즈니스적인 요인으로 인한 모니터링 비용 변경에 따른 작업량 최소화.

3. 운영자 관리 포인트 ↓ → 크래시는 PM시 항상 백업 복구.

상주 process로 stateful 하게 수행하면

추가 F/U 필요.

* 네트워크 OSI 7계층, 서브넷, 파이버와 PPP 차이

DI, IoC, Spring Framework bean scope

* ~~패턴~~ 패턴 IOC를 이용한 클래스 호출 방식.

