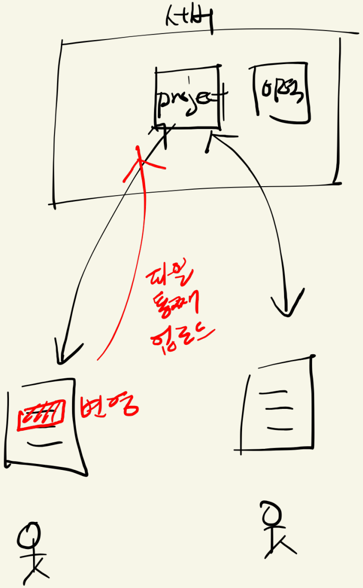
* Git - Software Configuration Management (SCM)

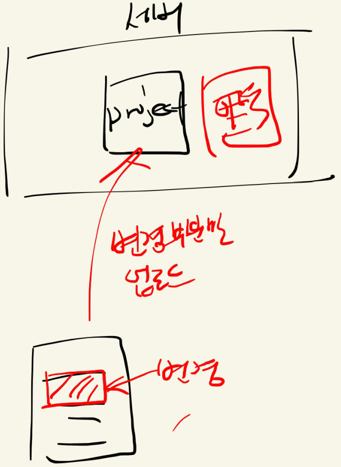
↳ 소스코드의 변경 관리

① 1세대



예) CSV

② 2세대



문제점

- 서버 의존

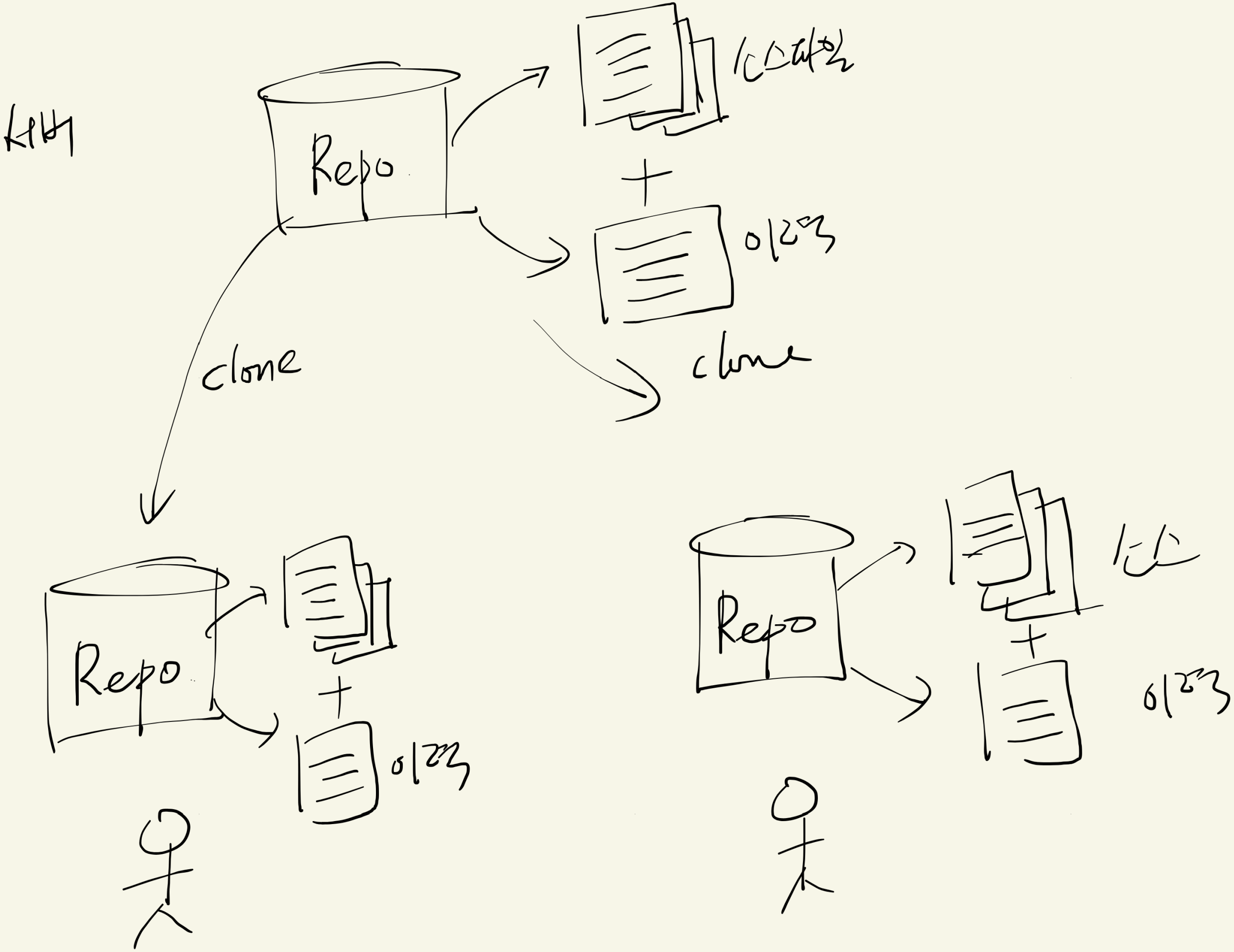
↓

"변경내역 손실"

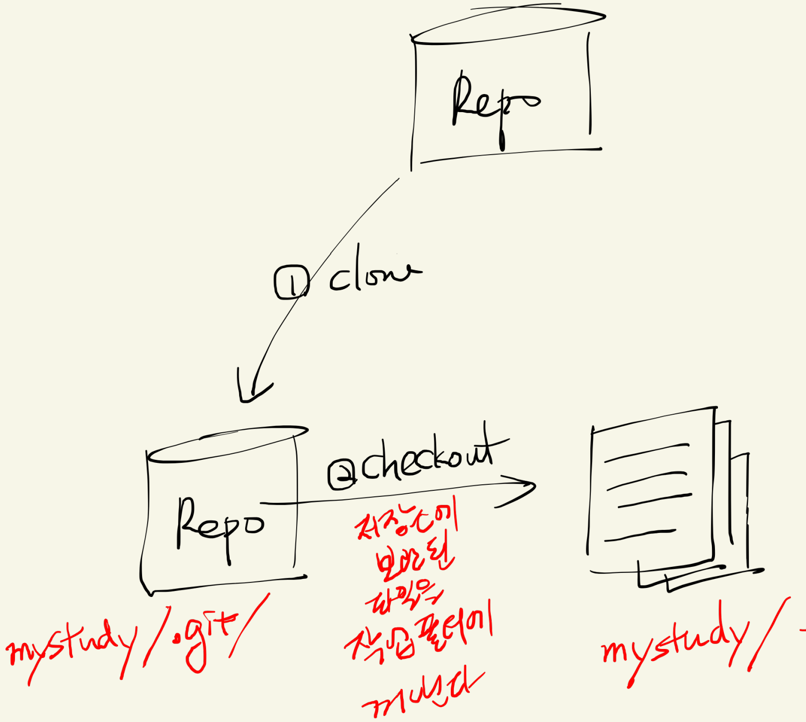
↓

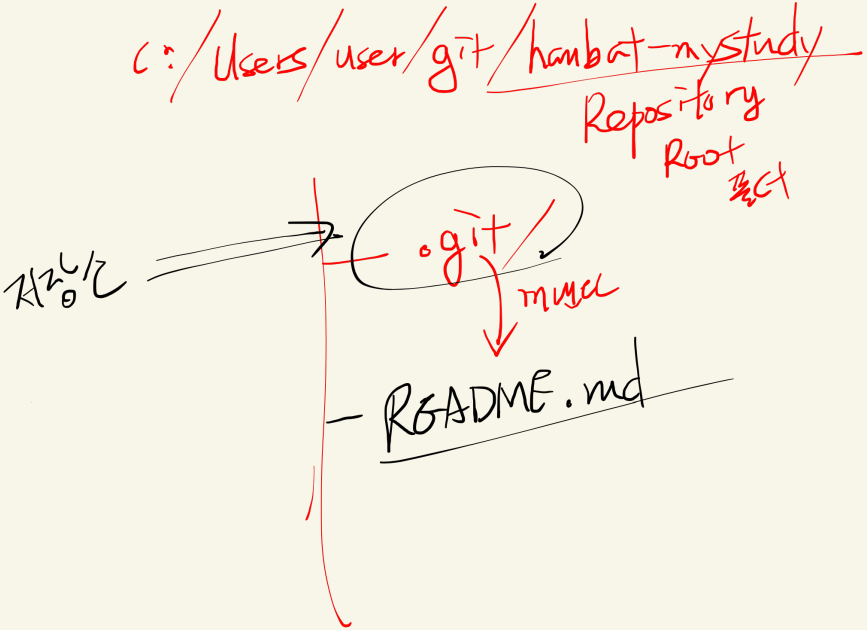
분산 저장

"Git"

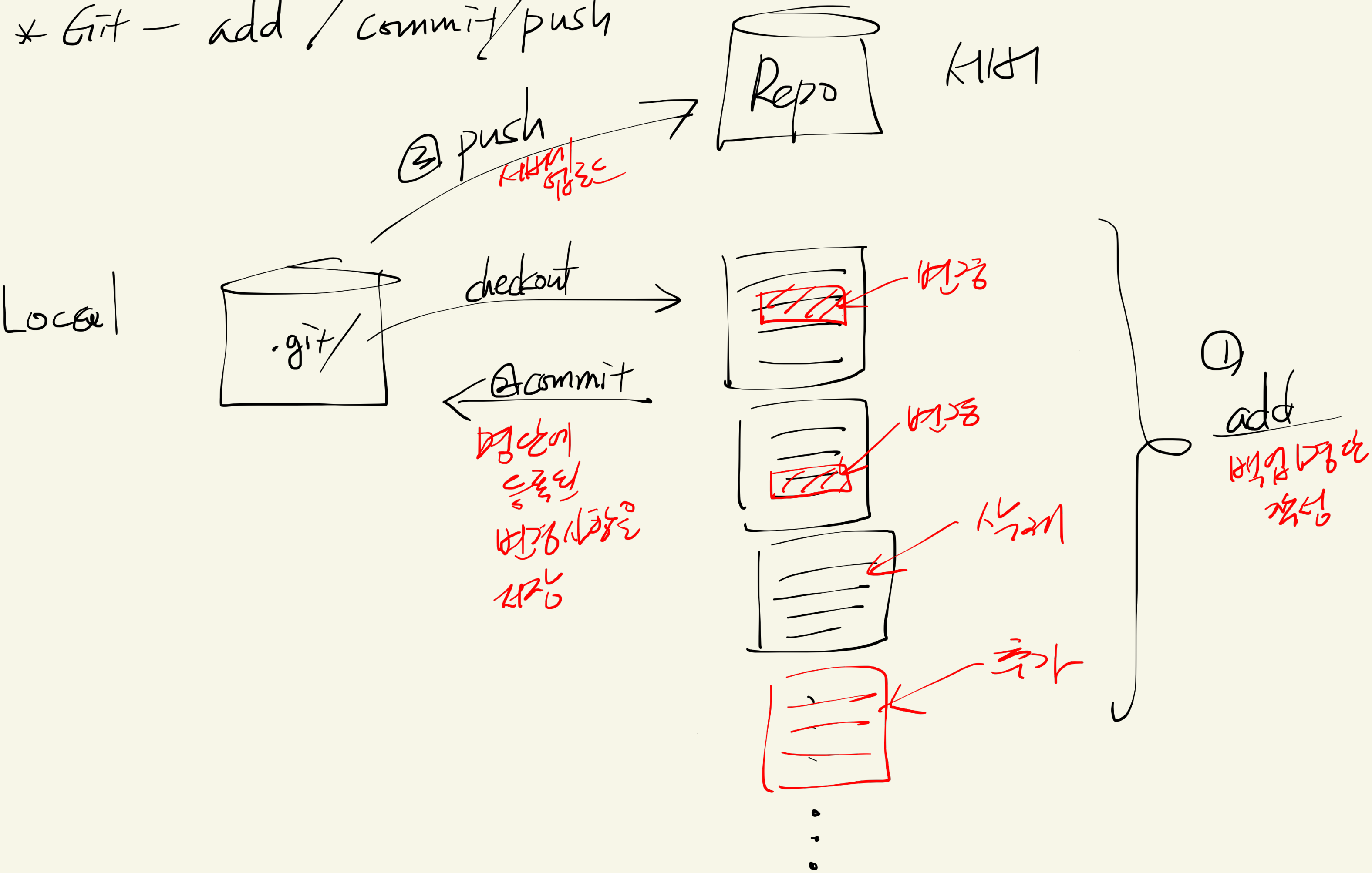

서버
Repo
소스파일
+
이력
clone
clone
Repo
+
이력
Repo
소스
+
이력

* GIT 사용법 - checkout

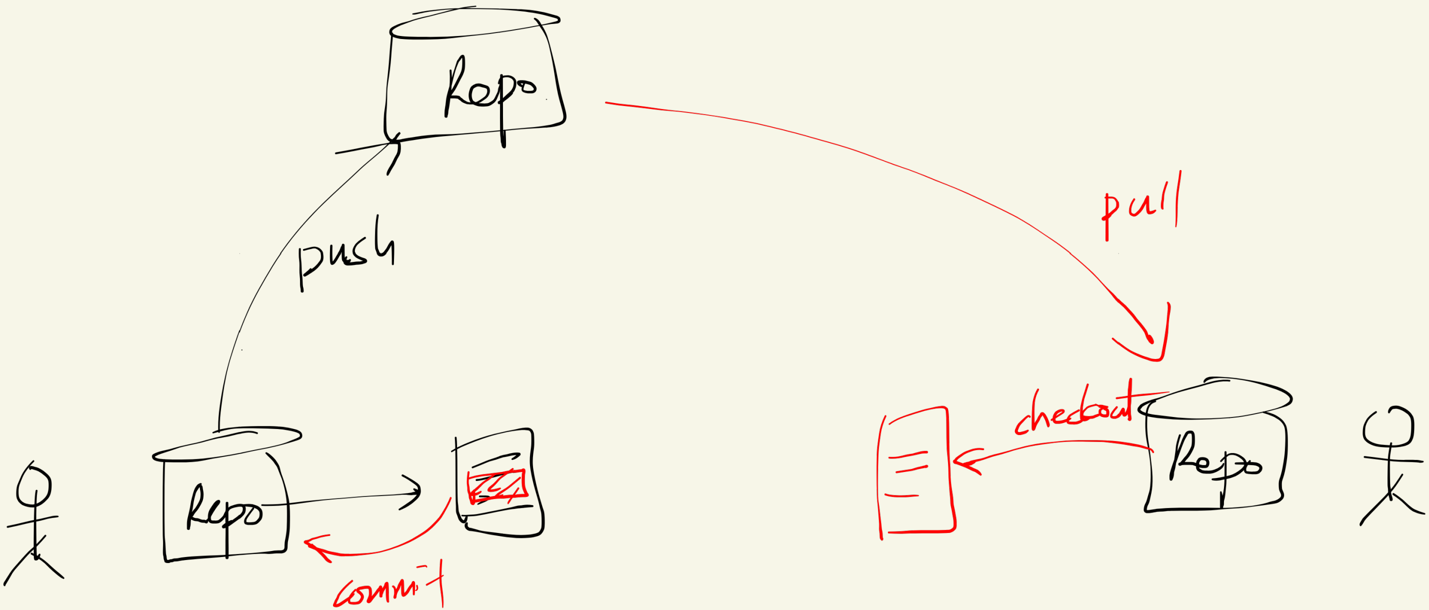




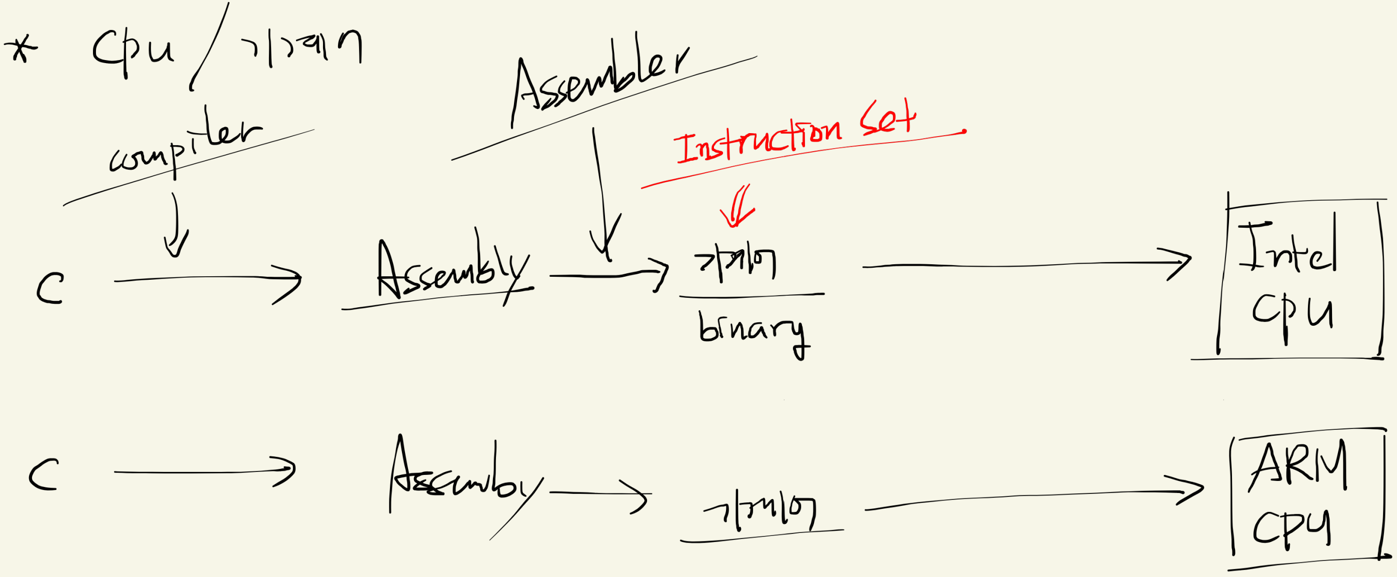
* Git – add / commit / push

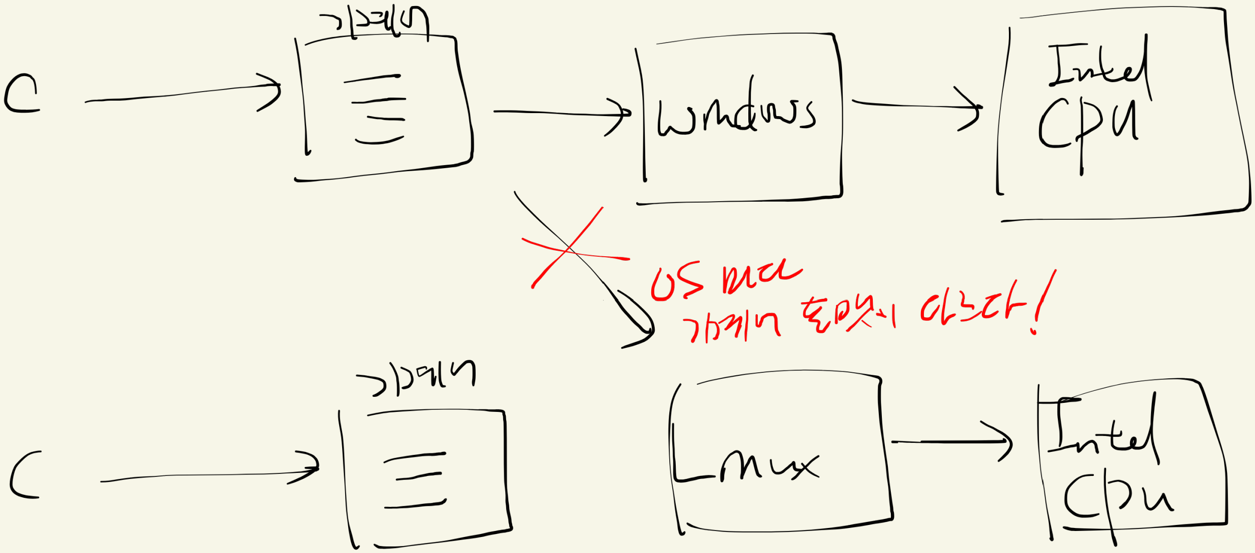


* Git — pull



* cpu / 기계어

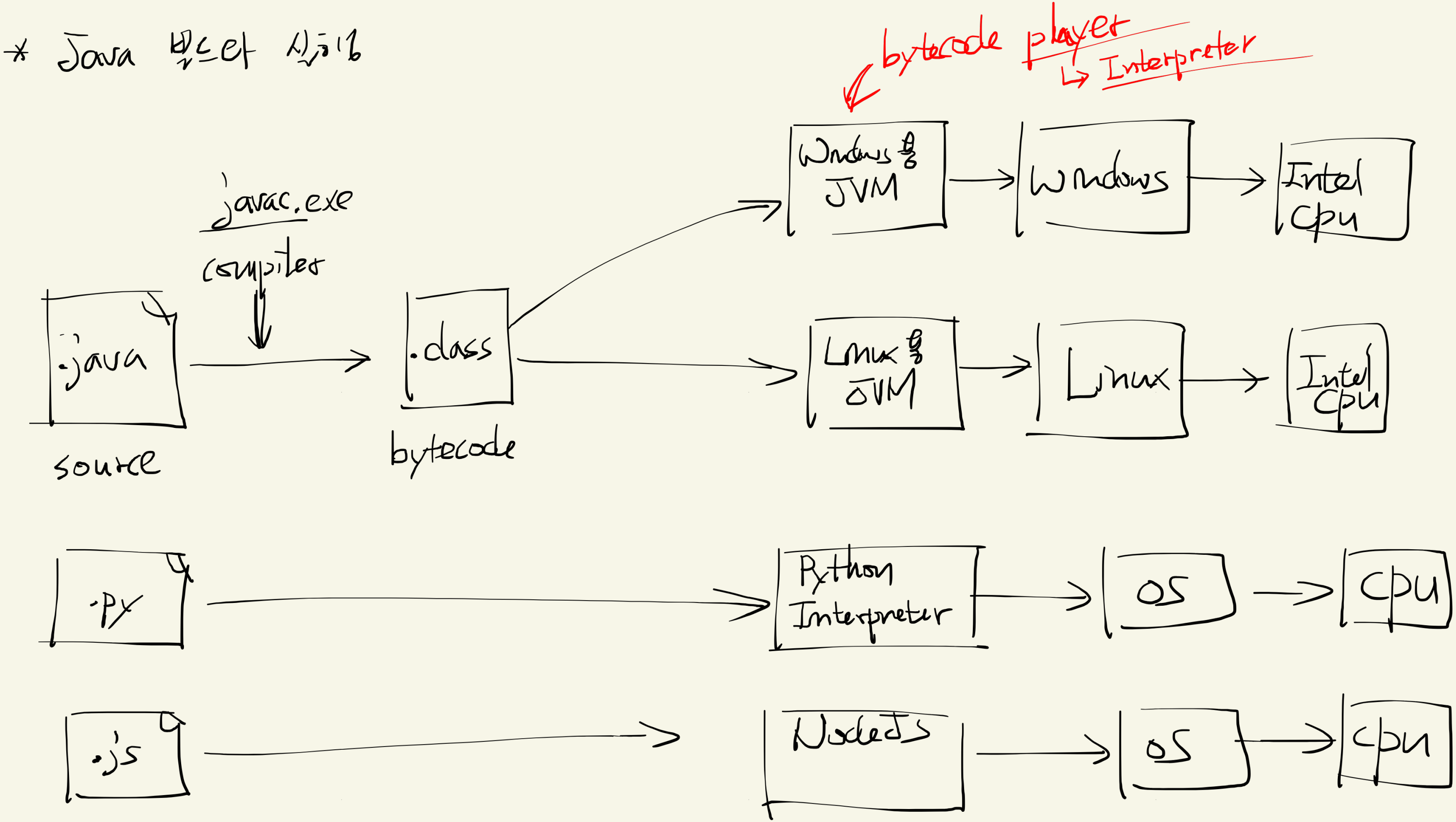

compiler
Assembler
Instruction set
C
Assembly
기계어
binary
Intel cpu
C
Assembly
기계어
ARM cpu


C
기계어
Windows
Intel CPU
OS 따라
기계어 호환이 다르다!
C
기계어
Linux
Intel CPU

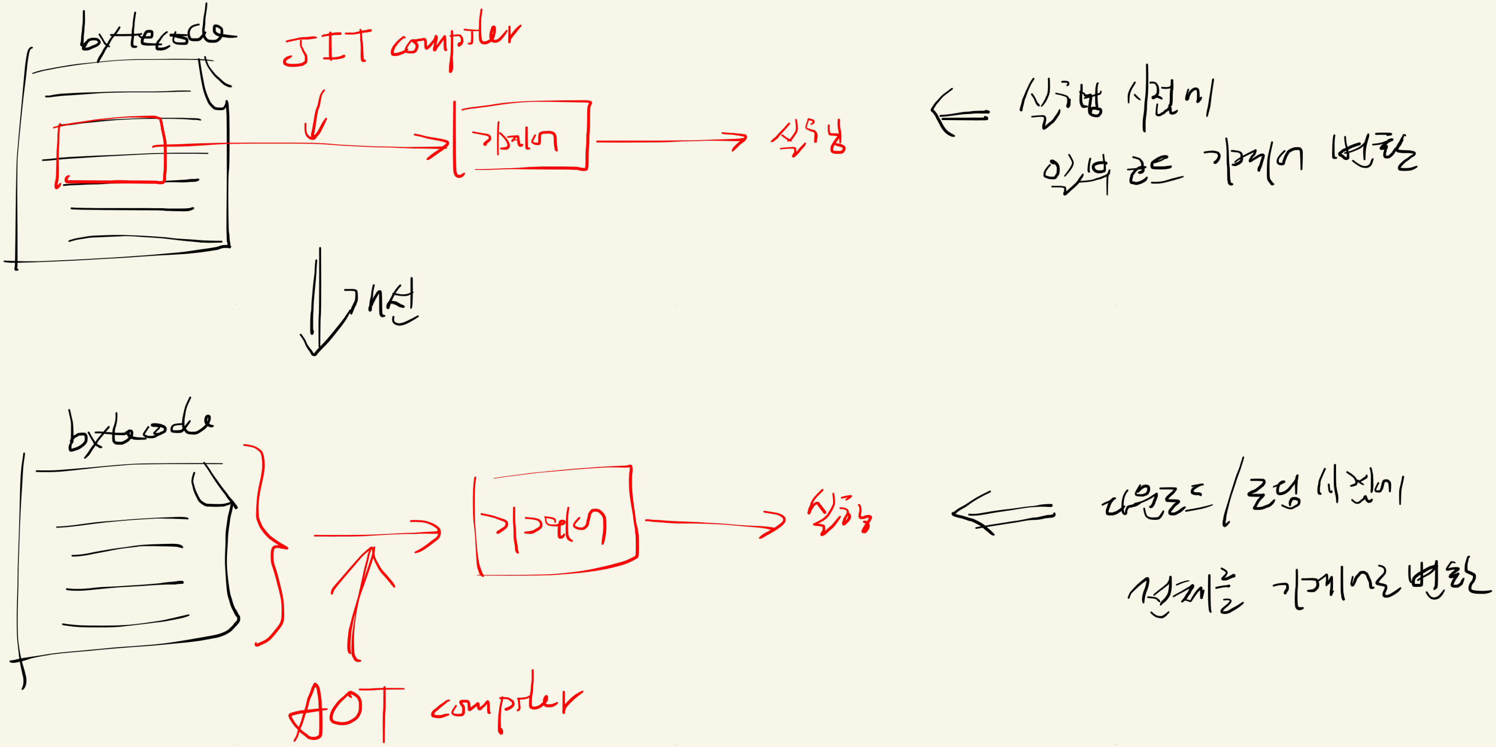
티디CD 210 866 3081

인출 111222

* Java 빌드와 실행

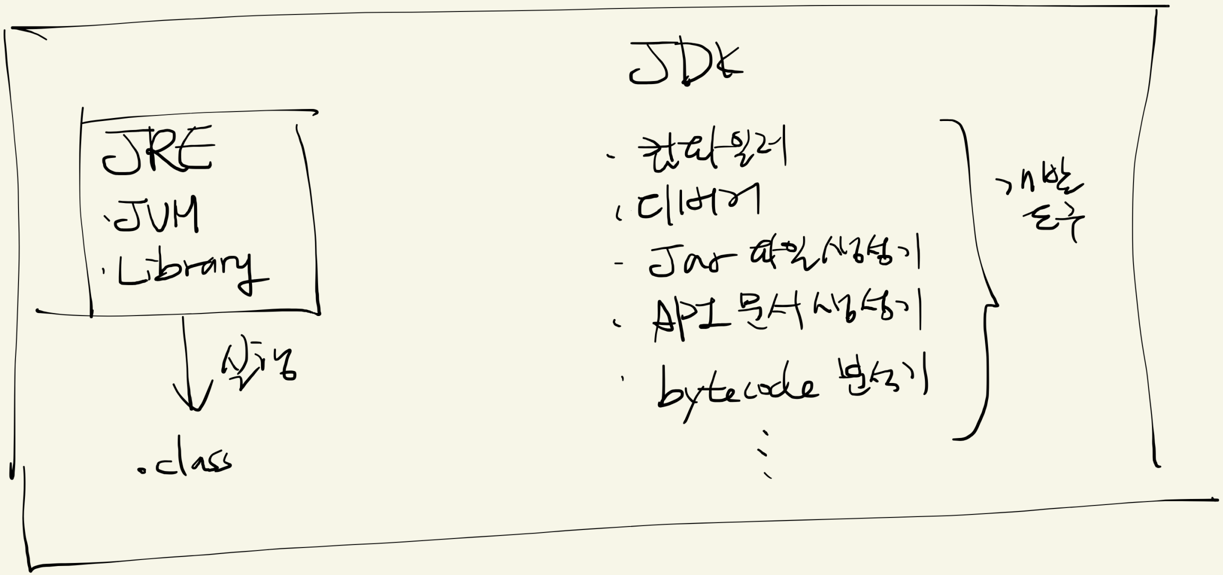

bytecode player
↳ Interpreter
javac.exe
compiler
.java
source
.class
bytecode
Windows 용
JVM
Windows
Intel
cpu
Linux 용
JVM
Linux
Intel
cpu
.py
Python
Interpreter
OS
cpu
.js
NodeJS
OS
cpu

* 실행 속도 개선

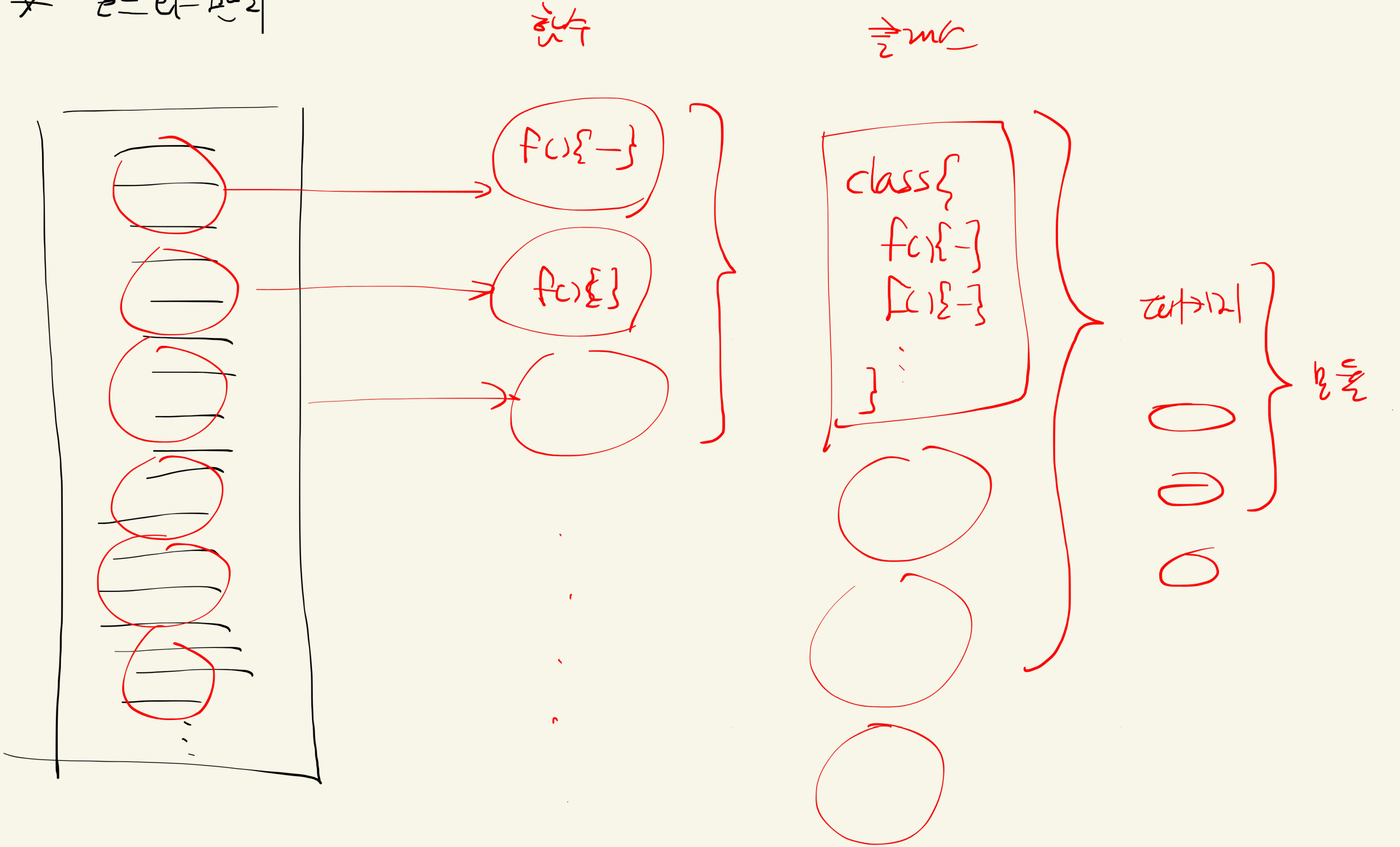

bytecode
JIT compiler
기계어
실행
실행 시점에
일부 코드 기계어 변환
개선
bytecode
기계어
실행
다운로드/로딩 시점에
전체를 기계어로 변환
AOT compiler

* JRE/JDK

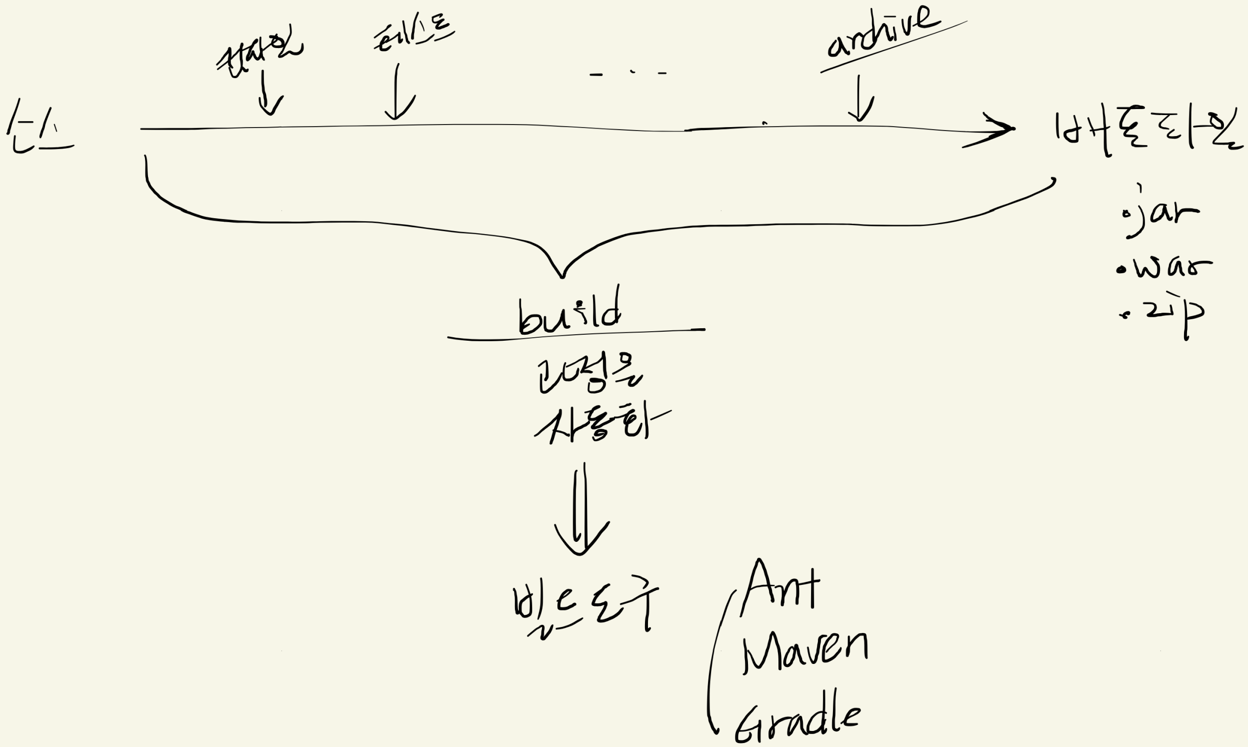
"Java SE" (제품명)



* 코드 다 관리



* 빌드 도구



소스

컴파일

테스트

- - -

archive

배포파일

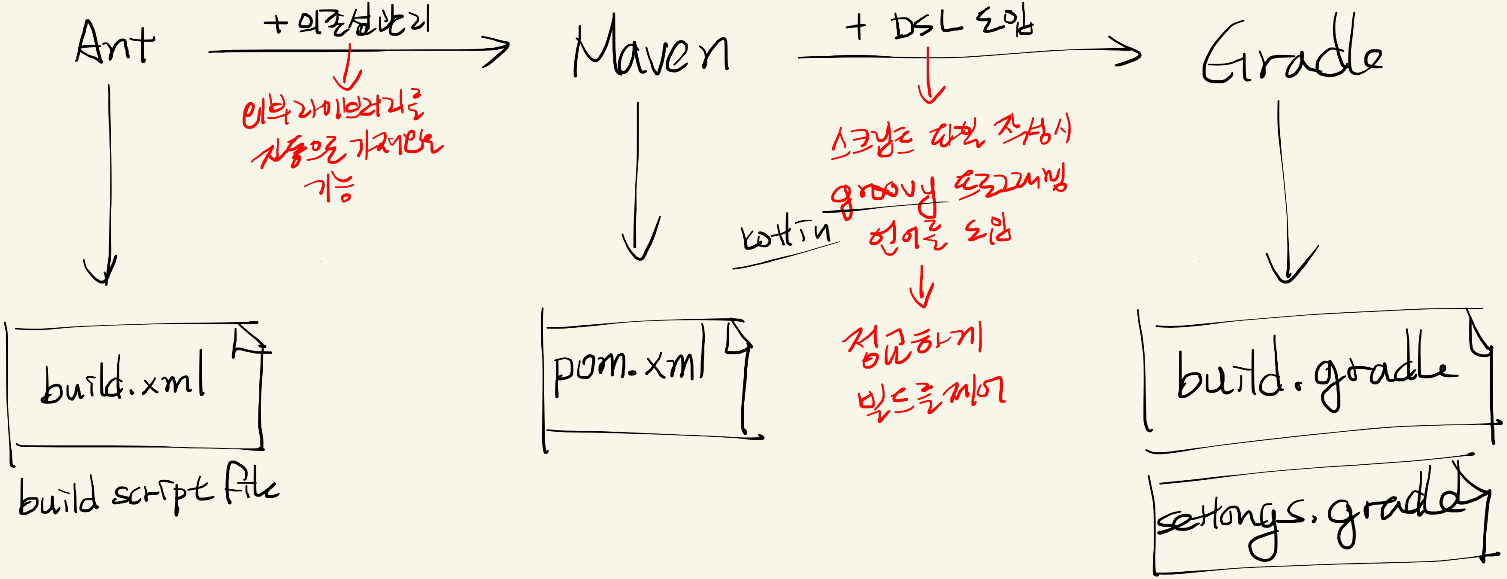
.jar
.war
.zip

build
과정을
자동화

빌드 도구

Ant
Maven
Gradle

* Ant vs Maven vs Gradle



Ant

+ 의존성관리

외부 라이브러리를 자동으로 가져오는 기능

Maven

+ DSL 도입

스크립트 파일 작성시 groovy 프로그래밍 언어를 도입

kotlin

정교하게 빌드를 제어

Gradle

build.xml

build script file

pom.xml

build.gradle

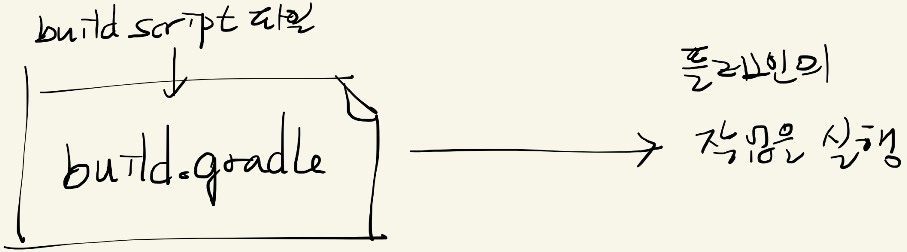
settings.gradle

* Gradle 사용법

gradle task

Gradle에 장착된
plug-in에 따라
사용할 수 있는
작업이 결정된다

* Gradle 작업



build script 파일

build.gradle

플러그인의 작업을 실행

· plug-in

:

① 기본 작업

* 클래스와 객체 / 메서드

```
class A {

    static void m1(){-}

    void m2(){}

}
```

데이터 없이 실행

데이터가 있어야만 실행가능

parameter

```
int plus(int a, int b){—}
```

```
plus(100, 200);
```

argument

* Static Method

입력

Math.abs(-100);

클래스

기능

작업할때 필요한 부가 데이터

str.substring(2, 6)

작업에 사용될 필수 데이터

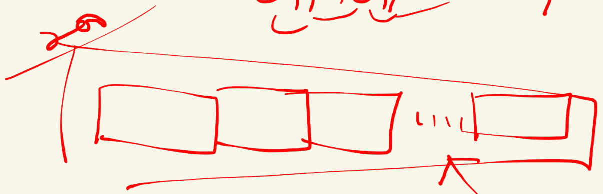
↓

객체

↓

인스턴스

인스턴스 = 객체



클래스에 선언된 변수 생성

String str = new String();

레퍼런스