

버전 관리 시스템과 Git

학습목표

- 버전 관리 시스템을 이해할 수 있다.
- Git을 이해하고 사용할 수 있다.

개요

그런 기사를 본 적이 있을 것이다. 프로그램이 업데이트를 한 후 치명적인 버그 혹은 운영의 문제가 생겨서 프로젝트를 롤백하는 경우 말이다. 이것이 어떻게 가능할까? 그 뒤에는 버전 관리 시스템이 있기 때문이다. 모든 회사에서는 프로젝트의 히스토리를 추적하기 위해 버전 관리 시스템을 사용한다. 이번 강의에서는 버전 관리 시스템이란 무엇인지, 그리고 그중 가장 폭 넓게 사용되고 있는 Git에 대해서 알아본다.

버전 관리 시스템

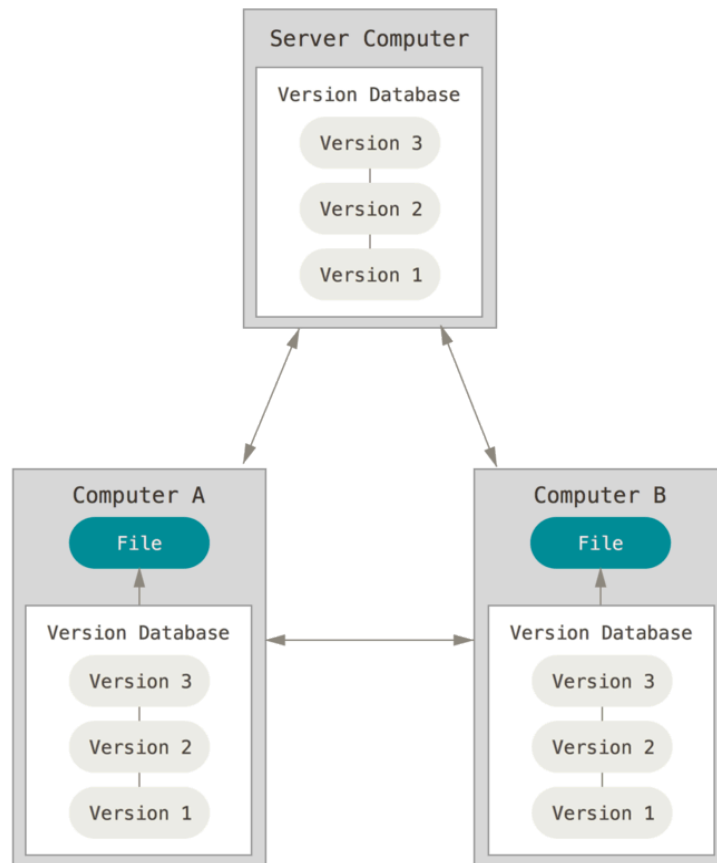
버전 관리 시스템(VCS; Version Control System)이란 **파일의 변화를 기록하여 추후 특정 버전으로 불러올 수 있는 시스템**을 말한다. 이미 여러분들은 일상 속에서 경험을 해본 적 있다.



위와 같은 것도 일종의(?) 버전 관리라고 할 수 있다.

위와 같은 그림은 우리가 수동적으로 버전 관리를 한 것인데, 버전 관리 시스템은 이를 프로그램이 대신 해준다.

현재의 버전 관리 시스템은 크게 **로컬 / 중앙집중형 / 분산형**으로 나눌 수 있는데, 이는 버전 관리가 어디서 이뤄지는지에 따라 구분된 것이다. 로컬은 본인 컴퓨터에서만, 중앙집중형은 어떤 특정 컴퓨터에서만, 분산형은 팀원 모두가 관리하는 것을 의미한다. 이 중 우리가 알아볼 Git은 분산형에 속하며 모든 변화의 히스토리를 각 개인이 들고 있게 된다.



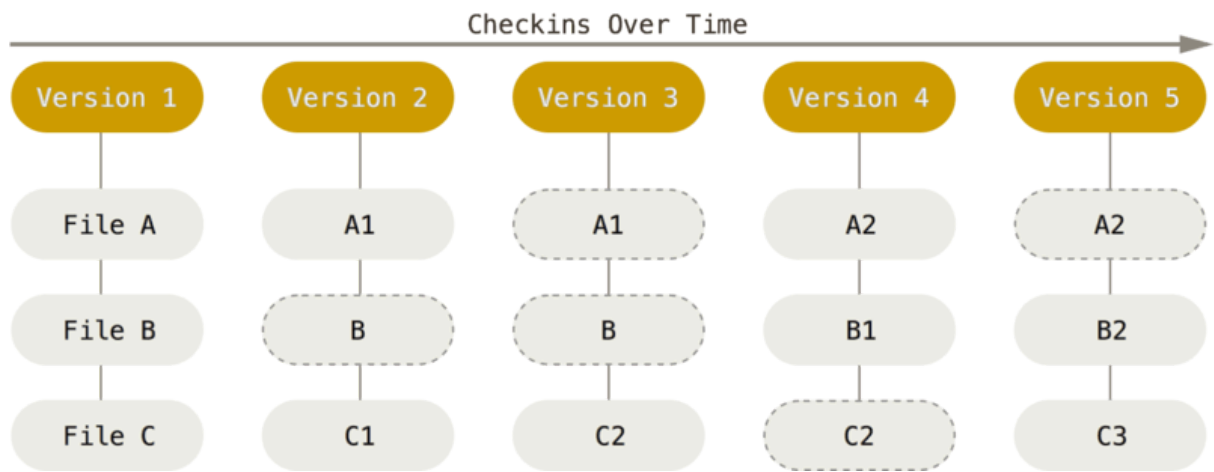
분산형 버전 관리 시스템의 구조

Git

버전을 저장하는 방법

Git은 어떻게 다양한 버전을 기록해둘 수 있을까? Git은 여기에 **스냅샷(Snapshot)***이라는 기술을 사용하며, **상태를 저장할 때마다 파일이 존재하는 그 순간을 중요하게 여긴다**. 아래의 그림은 Git이 버전을 기록하는 걸 그림으로 나타낸 것이다.

*스냅샷 : 특정 시점에 스토리지(Storage)의 파일 시스템을 포착해 보관하는 기술



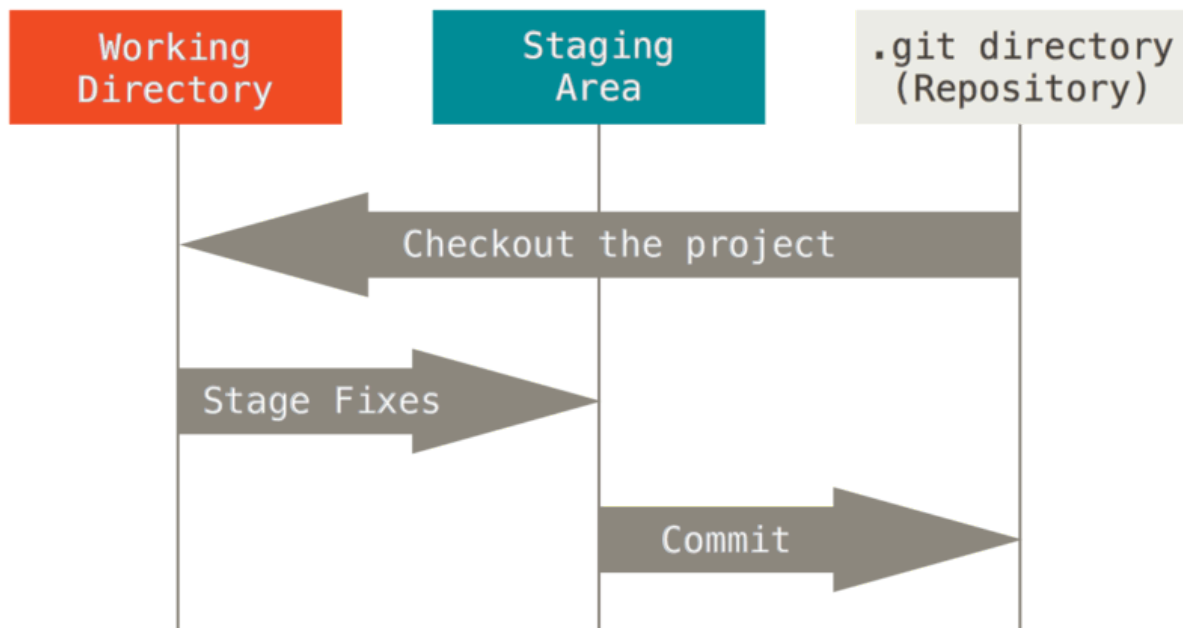
Git은 특정 시점에 파일이 존재하는지, 해당 파일의 어떤 버전인지를 기록한다.

위 그림을 보면 점선으로 표시된 그림이 있는데, 이는 이전 상태의 파일에 대한 바로가기이며*, 특별한 변화가 없다는 것을 의미한다.

* 이를 링크(Link)라 한다.

Git을 이용한 작업 순서

아래 그림과 함께 작업 순서를 살펴보자.



Git 디렉토리, 작업 디렉토리, 스테이지 영역

1. 저장소를 생성한다.

Git 디렉토리 혹은 **저장소(Repository)**는 **Git이 동작하기 위한 데이터와 우리의 프로젝트의 버전이 기록되어 있는 영역**이다. 저장소는 다른 컴퓨터에 있는 저장소를 복제(clone) 하거나 새로운 프로젝트를 만들 때(init) 생기게 된다.

2. 저장소에서 특정 버전을 불러온다.

저장소에 저장된 여러 버전 중 작업하고 싶은 버전을 불러온다(checkout). 불러온 후에 파일을 생성 / 삭제 / 수정 등의 작업을 한다.

3. 수정된 파일은 스테이지 영역으로 올린다.

버전을 기록할 파일에 대한 정보를 저장하며, 스테이지 영역을 **인덱스(Index)**라고도 한다. **스테이지 영역에는 내가 선택한 파일에 대해서만 기록한다.** 이때, 스냅샷을 찍는다.

4. 저장소에 버전을 기록한다.

저장소에 버전을 기록하는 것(스냅샷을 저장하는 것)을 커밋(Commit)이라고 한다. 이 단계까지 와야지만 비로소 버전이 기록되는 것이다.

게임으로 명령어 배우기

Git에는 여러 가지 명령어가 있으며, [여기](#)에서 게임처럼 실습해볼 수 있다. 실습 하면서 각 명령어가 어떻게 동작하는지를 이해해보자. 그 외의 명령어는 뒤에서 실습한다.

컴퓨터에서 실습해보기

실습은 강의 시간에 이뤄진다. Git을 사용하기 위해 Git 프로그램을 설치해야 한다. [여기](#)서 설치하자. 추가적으로 배울 명령어가 있는데, 아래의 목록을 참고하라.

- init : 저장소 생성
- remote : 리모트 설정
- config : Git 프로그램 설정
- log : 히스토리 확인

- .gitignore

프로그램을 만들다보면 임시 파일이 만들어지게 되는데, 이러한 파일까지 관리할 필요는 없을 것이다. .gitignore 파일은 추적하지 않을 파일이나 폴더를 기록한 파일이다. 대부분 프로젝트에 대해 패턴이 만들어져 있으며, [여기](#)서 복사해서 사용하면 된다.

- Git GUI

명령어에 익숙해졌다면 GUI 프로그램으로 좀 더 편하게 사용할 수 있다. 목록은 [여기](#)를 참고하자.

더 나아가기

1. [이 실습](#)에서 아래의 단계를 수행해봅시다.
 - a. git 기본 - 1, 2, 3, 4
 - b. 다음 단계로 - 1, 4
 - c. 고급 문제 - 1
 - d. Push & Pull - 1, 2, 3, 4, 5, 6, 7, 8
2. 버전 관리 시스템이란 무엇인가요?
3. Git을 이용한 작업 순서는 어떻게 되나요?
4. git merge와 rebase에는 어떤 차이가 있는지 서술해주세요.
5. 버전을 롤백하고 싶을 때 사용하는 명령어와 차이점을 서술해주세요.
6. .gitignore 파일을 무엇이며 왜 사용하나요?

참고자료

- [Git - Book](#)
- [git - 간편 안내서 - 어렵지 않아요!](#)
- [A Visual Git Reference](#)
- [누구나 쉽게 이해할 수 있는 Git 입문~버전 관리를 완벽하게 이용해보자~](#)