Version Control - Git

A Basic CS Skill, ABC Winter School

박원

Objective

- What is Git?
- Git
  - make & clone repository
  - add, commit
  - push, pull
  - rm, mv
- Branch
  - branch, checkout, merge

What is Git? - Version Control

> 버전 관리 시스템(Version Control System)은 파일의 변화를 시간에 따라 기록했다가, 나중에 특정 시점의 버전을 다시 불러올 수 있는 시스템이다

- 소프트웨어 개발의 규모가 점점 커지고, 이에따라 협력해서 개발을 하게 된다.
- 이 과정 속에서 하나의 소프트웨어 프로젝트라도 다양한 버전이 생기고, 이를 통합적으로 관리해야한다.

Snapshot

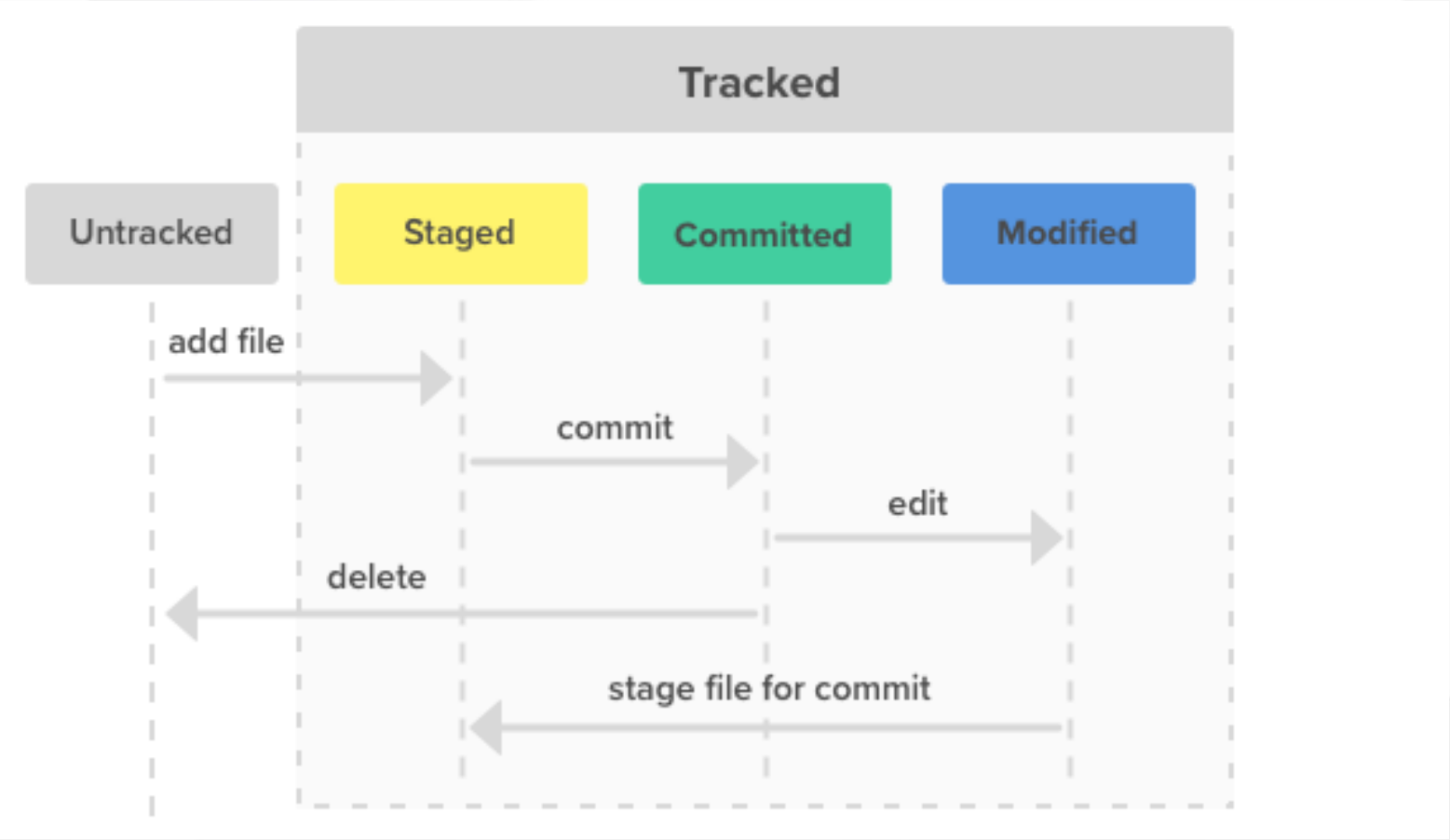
- 각각의 모든 버전은 snapshot 형태로 저장이 된다.
- 파일시스템은 하나의 트리구조로 되어있고, snapshot은 이러한 트리를 저장하고 있다.

```
<root> (tree)
|
+- foo(tree)
||
|+ bar.txt(blob, contents="abc")
|
+-baz.txt(blob, contents="it is git!")
```

Stage

- Git은 파일을 세 가지 Stage를 통해 관리한다.
- commited: 데이터가 로컬 데이터베이스에 안전하게 저장된 상태이다
- modified: 수정한 파일을 아직 커밋하지 않은 상태를 의미한다
- staged: 현재 수정한 파일을 곧 커밋할 것 이라 표시한 상태이다.

Stage


Tracked
Untracked
Staged
Committed
Modified
add file
commit
edit
delete
stage file for commit

Git - Build Repository

- git을 사용하기에 앞서, 우리가 git을 사용하고 싶은 directory를 지정하고, git이 관리하는 repo로 설정해야한다.
- git init를 통하여 설정할 수 있고, 실행하면, `.git` 이라는 git에 대한 정보가 담긴 숨김 디렉토리가 생성된다.

```
$ git init
```

Git - Clone Repository

- 이미 존재하는 repository를 가져올 수 있다.

```
$ git clone <url>
$ git clone https://github.com/circle-oo/ABC-unist-2021-winter.git
```

Git - Add & Commit

```
$ git status
On branch master
No commits yet
nothing to commit (create/copy files and use "git add" to track)
```

Git - Add & Commit

```
$ echo "wow" > README
$ git status
On branch master
No commits yet
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        README
nothing added to commit but untracked files present (use "git add" to track)
```

Git - Add & Commit

```
$ git add README
$ git status
On branch master
No commits yet
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   README
```

Git - Add & Commit

```
$ echo "hello" >> README
$ git status
On branch master
No commits yet
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   README
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README
```

Git - Add & Commit

- git add로 추가된 파일을 commit하여 추가한다
- staged에 속하는 스냅샷을 커밋

```
$ git commit -m "wow"
[master (root-commit) 76319a1] wow
 1 file changed, 2 insertions(+)
 create mode 100644 README
```

Git ignore

- staging할 필요가 없는 파일들을 자동으로 걸러주게 하는 파일에 대한 정보를 담은 파일이 `.gitignore` 이다.
- 아무것도 없는 줄이나 `#` 으로 시작하는 줄은 무시된다.
  - 주석
- 정규표현식을 사용한다
- `/`로 시작하면 하위 디렉토리에 적용되지 않는다.
- 디렉토리는 `/`를 끝에 사용하여 표현한다.
- `!`로 시작하는 패턴은 무시하지 않는다.

Git ignore

```
# 확장자가 .a인 파일 무시
*.a
# 위의 예외로 lib.a는 제외
!lib.a
# 현재 디렉토리의 TODO만 무시
/TODO
# build 디렉토리에 있는 모든 파일 무시
build/
# doc/a.txt는 무시, doc/a/b.txt는 무시 안함
doc/*.txt
# doc 디렉토리 내의 모든 .pdf 파일 무시
doc/**/*.pdf
```

rm & mv

- `rm`으로 tracked된 파일을 삭제할 수 있다.
- `mv`로 tracked된 파일의 이름을 바꾸거나 옮긴다

```
$ git rm <file>
$ git mv <from> <to>
```

GitHub

- GitHub: github.com

Branch

- `git branch`: 새로운 브랜치를 생성
- `git checkout`: 브랜치 변경
- `git merge`: 현재 브랜치에 다른 브랜치를 머지

```
$ git checkout <branch>
$ git branch <branch>
$ git merge <branch>
```

Exercise

- Learn Git Branching

Remote Repository 생성

Questions?

Next

Version Control - Git