An Introduction to Git and GitHub

Zheng Tian

1 What are Git and GitHub?

1.1 What is Git?

- Git is a version control system (VCS) that records changes to a file or set of files over time so that you can recall specific versions later.

- Using Git to manage versions means that if you screw things up or lose files, you can easily recover.

1.2 How does Git work?

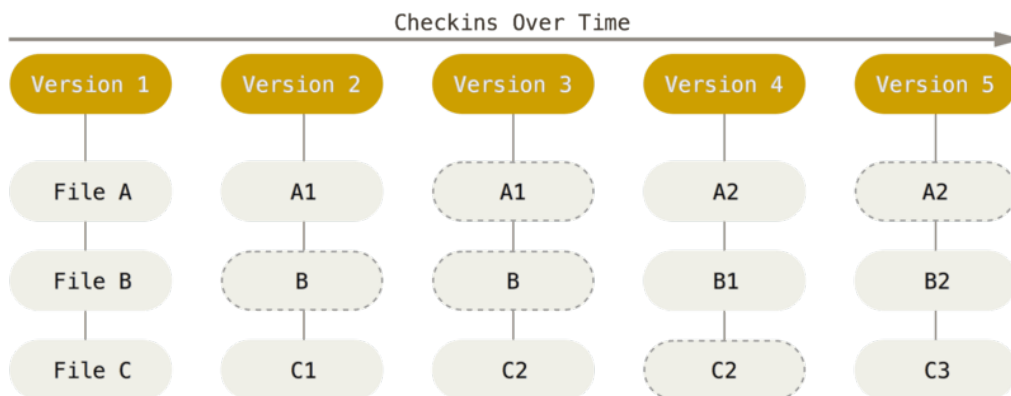- Git takes a snapshot of a whole project whenever you make a commit to the changes.

Figure 1:   Storing data as snapshots of the project over time

1.3 The basic workflow of Git goes like this

1. You modify files in your working tree.

2. You stage the files, adding snapshots of them to your staging area.

3. You do a commit, which takes the files as they are in the staging area and stores that snapshot permanently to your Git directory.
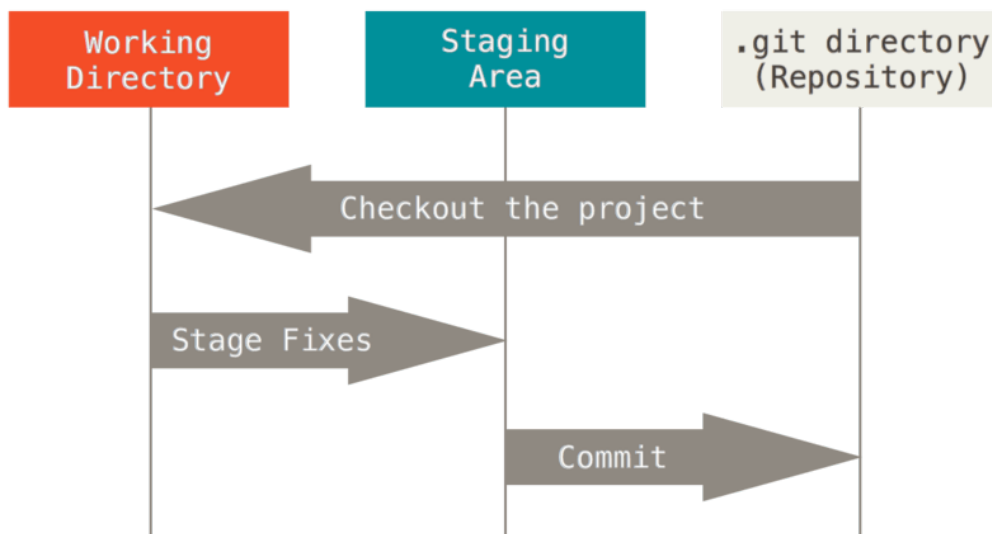

Figure 2: Working tree, staging area, and Git directory

1.4 What is GitHub?

- GitHub is the single largest host for Git repositories, and is the central point of collaboration for millions of developers and projects.

- Visit `https://github.com/` and sign up a free account right now!

2 Get Started

2.1 Install Git and GitHub Desktop

Download and install Git

You can download the installers of Git for Windows, MacOS, and Linux at `https://git-scm.com/downloads`.

Download and install GitHub Desktop

- GitHub Desktop is a GUI for Git and integrate your repositories on GitHub with local counterparts.

- Download the installer of GitHub Desktop at `https://desktop.github.com/`.

2.2   Configure Git

Next, follow the steps of configuration at `https://help.github.com/articles/set-up-git/`.

1. Open a **terminal**.

   - In MacOS, search `terminal` in Spotlight Search.

   - In Windows, open `Git Shell` that is installed with GitHub Desktop.

2. Configure user name and user email. It is wise to use the same user name and email when you sign up with GitHub.

   ```
   $ git config --global user.name "YOUR NAME"
   $ git config --global user.email "EMAIL ADDRESS"
   ```

3. Setup a default editor. GitHub recommend using Atom.

   - Download Atom at `https://atom.io/`.

   - Configure Atom to be the default editor of Git, as instructed at `https://help.github.com/articles/associating-text-editors-with-git/` In MacOS,

     ```
     $ git config --global core.editor "atom --wait"
     ```

     In Windows

     ```
     $ git config --global core.editor "C:/Users/USERNAME/AppData/Local/atom/app-VERSI
     ```

4. Authenticating Git with GitHub

   - In MacOS. Install `osxkeychain helper` in by entering the following command.

     ```
     $ git credential-osxkeychain
     ```

     If `osxkeychain helper` has already been installed, nothing happens. If not, your computer will prompt you to download it as a part of the Xcode Command Line Tools.

   - In Windows. Open GitHub Desktop. If it is your first time opening it, a window will pop up asking for your GitHub account.

2.3   Book: *Pro Git*

- Git is fully documented in this book, `https://git-scm.com/book/en/v2`.

3 TODO Git Basics [1/2]

This section introduces the very basic commands of Git.

3.1 DONE Create a Git repository

A Git repository is simply a folder containing the files pertaining to a project that you want to have versions controlled. Within the folder, there is a sub-folder, usually invisible, called. `.git`, where all data regarding each different version are stored.

Create a Git repository in a computer

Let us first create a git repository in our computer. We do the following things

- Create a new folder, called `git_test`

- Go to this folder

- Make it a git repository.

```
$ mkdir git_test
$ cd git_test
$ git init
$ git status
```

We use `git status` to check the status of the repository.

Create a Git repository in GitHub

At `https:github.com`, click **New Repository** and follow the instruction to create a new repository, names `github_test`. This repository is now at GitHub not in your computer. To make it in your computer, the quickest way is to press the button of **Set up in Desktop**, which will call GitHub Desktop and set up the repository in your computer.

Another way is the `HTTPS` of the repository and type the following command, assuming that you are under a folder in which you want this new repository to be downloaded.

```
$ git clone https://github.com/zngtian/github_test.git
$ git status
```

3.2 A life-cycle of a file in a git repository

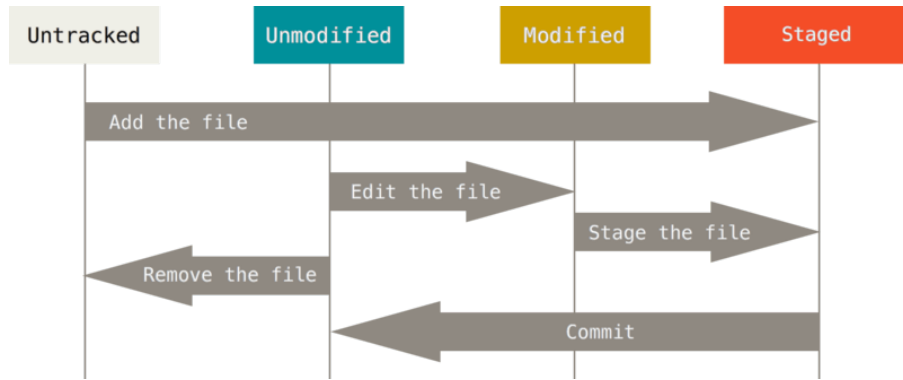Figure 3 displays a life cycle of a file in a git repository.

Figure 3: The lifecycle of the status of your files

- When a file is first created, its status in git is **untracked**. For example, we create a new file `README.md`.

- We want git to know that we have added a new file. With git, we need to **stage** the file.

- After we have made all changes regarding the new file, we need to tell git that all changes are done. With git, we need to **commit** the file.

- After committing, if we make any change, we need to repeat staging and committing.

All these things are implemented with the following command.

```
$ touch README.md
$ git status
$ git add README.md
$ git status
$ git commit -m "add a README.md file"
$ git status
```

If there are more than one files that we have made some changes, we can use `git add -A` to stage all files.

Removing and renaming files also need to be staged so that git can know and records such changes.

3.3 TODO Upload your work to GitHub

4 TODO A Workflow of Group Working