CSCI-UA-4-005

Intro to Web Design + Computer Principles

Version Control

Professor Emily Zhao

M/W 12:30PM – 1:45PM

Agenda

- Lecture: Version Control
- Github Demo
  - Creating a repository
  - Adding to a repository
  - Changing a repository
  - Markdown syntax
  - Forking a repository

Version Control

A system that records changes to a file or set of files over time so that you can recall specific versions later

Commonly used for software source code but any type of file can be placed under version control

A Version Control System (VCS) allows you to:

- Revert files back to a previous state
- Review changes made over time
- Collaborate more efficiently
- Maintain project backups

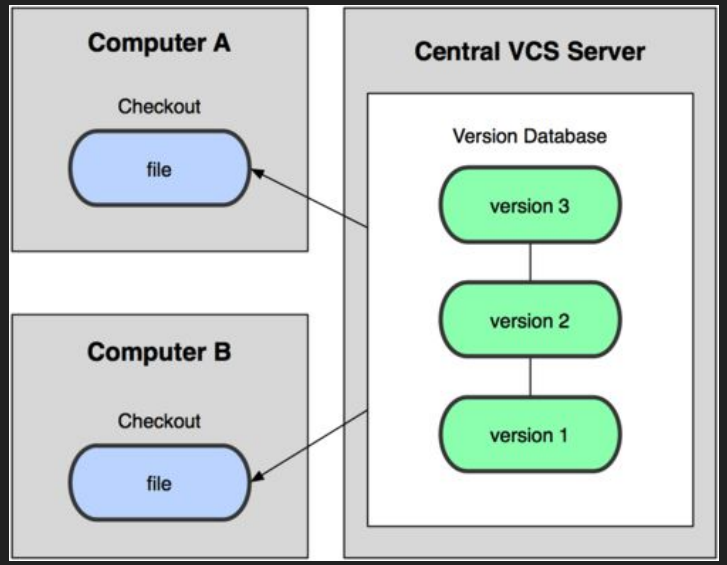
my-project-1.txt

my-project-2.txt

my-project-3.txt

Centralized Version Control

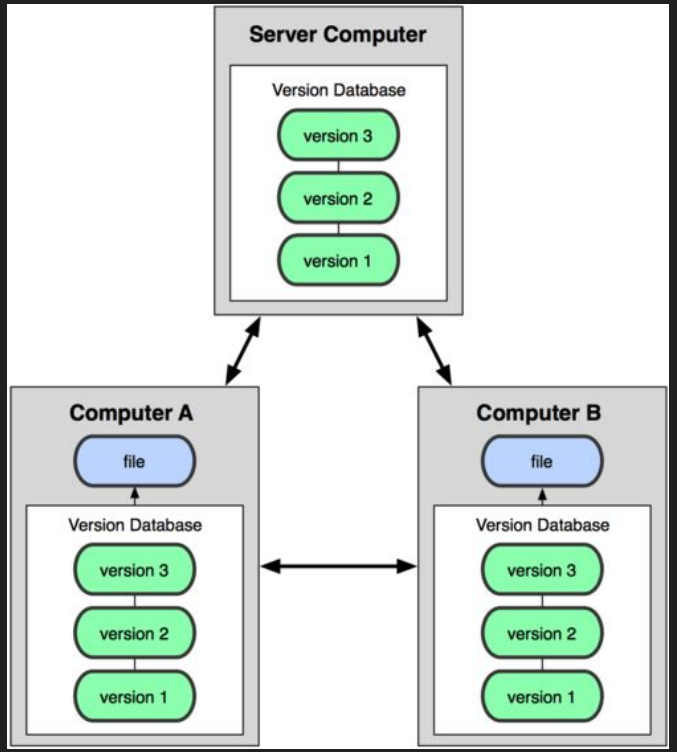
- Centralized Version Control Systems were developed to allow collaboration with developers on other systems.
- With a CVCS, a single server contains all the versioned files and clients “check out” files from that central place.
- For many years, this has been the standard for version control.
- The downside of centralized version control is the vulnerability of having the entire history of a project in one place.



Pro Git

Distributed Version Control

- With Distributed Version Control Systems, clients don’t just check out the latest snapshot of files, they fully mirror the entire history of the project.
- If a server dies, anyone with a copy of all the versioned files can restore it to the server.
- Every checkout is really a full backup of all the data.
- You can also collaborate with different groups of people in different ways simultaneously within the same project.



Pro Git

Git History

Git was was created in 2005 by Linus Torvalds and the Linux development community for Linux kernel maintenance

Linux is an open source operating system project of fairly large scope

Its goal was to be a fully distributed VCS with a simple design, support for non-linear development, and the ability to handle large projects efficiently

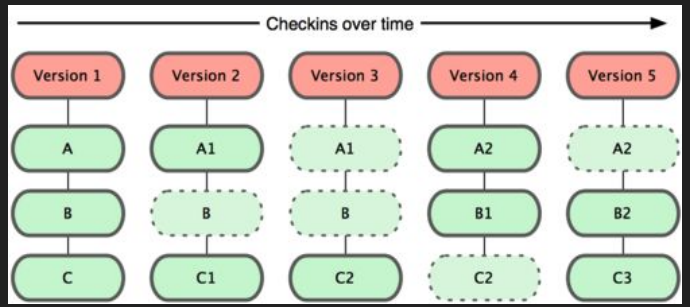
Git Basics

Git thinks of its data like a set of snapshots of a mini file system.

Every time you save the state of your project, it basically takes a picture of what all your files look like then and stores a reference to that snapshot.

To be efficient, if files have not changed, Git doesn't store the file again—just a link to the previous identical file it has already stored.

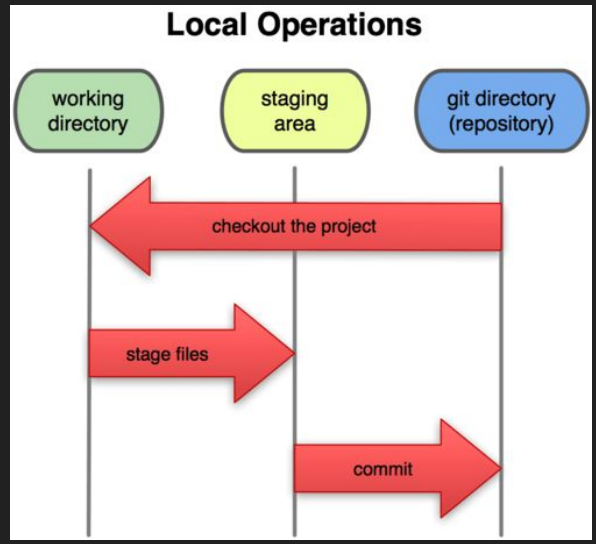
This makes Git more like a mini file system with some powerful tools built on top of it.



Git States

Git has three main states that your files can reside in: modified, staged, and committed.

- Modified means that you have changed the file but have not committed it to your database yet.
- Staged means that you have marked a modified file in its current version to go into your next commit snapshot.
- Committed means that the data is safely stored in your local database.



Git Workflow

1. Modify files in your working directory
2. Stage the files, adding snapshots of them to your staging area.
3. Commit changes, which takes the files as they are in the staging area and stores that snapshot permanently to your Git directory.

Github

GitHub is a web-based hosting service that uses the Git VCS.

- The site also provides social networking functionality such as feeds, followers, wikis, and statistics.
- The company was founded in 2008 and is located in San Francisco.
- In addition to computer programmers, architects, musicians, municipal governments, and academics are among its users.

[Demo] Web page with HTML audio, video, and inline frame

[Demo] Github

Homework

— Be ready to discuss and show initial work on your final website!