

Introduction to Git and GitHub

Mark Anderson

anderson.mark@queensu.ca



May 7, 2020

What is Git?

- Software for version control
- Specifically, a *distributed version control system*

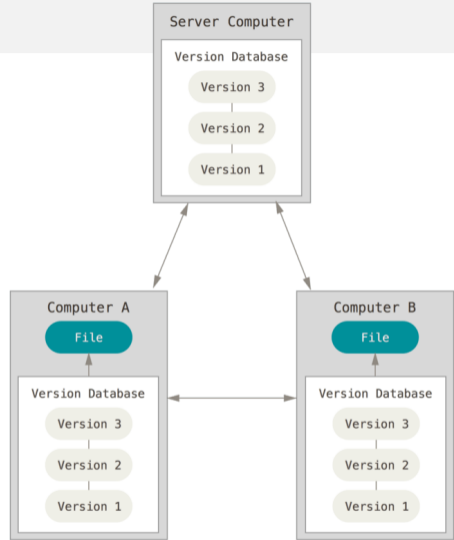
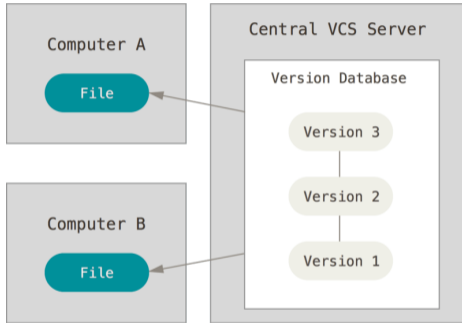
Version Control System

- Tracks and stores changes to information
- Tracks the history of changes over time

Distributed

- A full copy of the code and history is available at every repository and to every developer
- No need to connect to the Internet unless pushing or pulling changes from remote repository

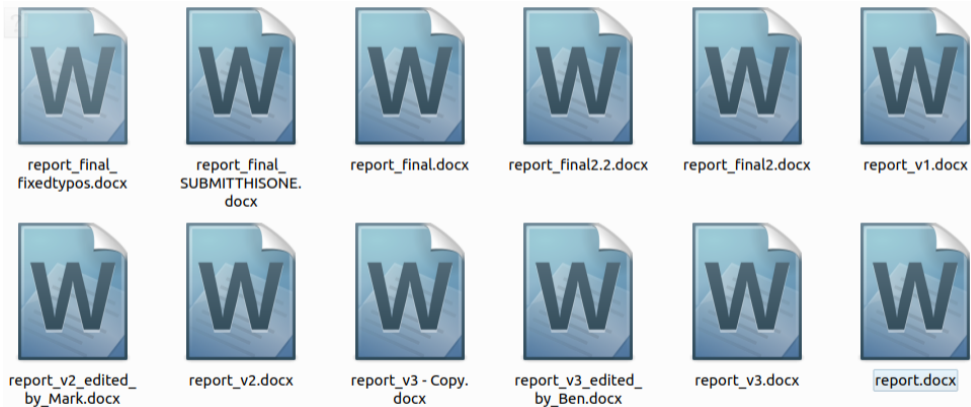
Central vs. Distributed



<https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control>

Why is Git useful?

The “undergraduate version control system” in action



Why is Git useful?

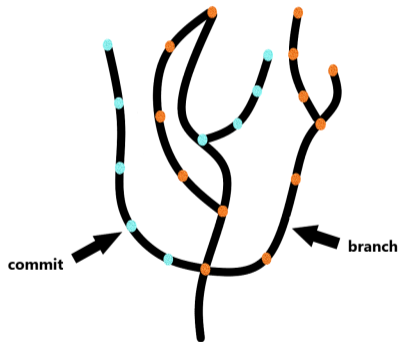
- Multiple developers can work on the same project
 - Prevents confusion when multiple people are editing the same file
 - Each developer can work on different parts of the project simultaneously
 - Independent parts can easily be merged in
 - If two developers modify the same portion of code, this will become obvious
 - Conflict can be resolved
- Can look back through the history of the project
 - Figure out *who* did that weird hack six years ago
 - Figure out *why* someone did that weird hack six years ago
 - You can blame your coworkers when things go wrong!
 - Maybe `git blame` should have been called
`git praise` or `git objectively-determine-contributer`

Why is Git useful?

- Undoing accidental changes to code
- Imagine you are working up until the deadline for a project and not using git
 - 11:34pm – You find a fairly minor bug that requires a few changes
 - 11:46pm – You believe you have fixed the bug
 - 11:48pm – You go to compile or run the code and find that it no longer works!
 - In your rush to fix the code, you accidentally changed something else
 - You no longer remember exactly what used to be where
 - 11:49pm – You say “no worries, I still have 10 minutes to fix it!”
 - 11:55pm – You still haven’t fixed it and you’re starting to panic
 - 11:58pm – You have to decide whether to submit non-working code or to submit it late.

Git

- Git “sits on top of” your file system to track changes and manage files
- Git is tree-based
 - Each node in the tree is a *commit*
 - Can have multiple *branches* from a given node
- Git stores files and tracks changes in a *repository*
 - A repository is basically a set of commits



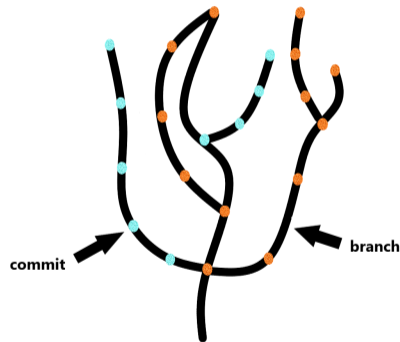
Git

Commits

- A commit is like a snapshot of a project at a given time
 - While Git *tracks* differences, it does not *store* the data as a set of differences
- The most important parts of a commit object are:
 - A pointer to a snapshot of the project
 - A message (hopefully detailed!) and author information
 - A hash to uniquely identify the commit

Branches

- A branch is a pointer to a commit
 - Moves forward automatically as you make more commits
 - A special pointer HEAD keeps track current branch



	COMMENT	DATE
○	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
○	ENABLED CONFIG FILE PARSING	9 HOURS AGO
○	MISC BUGFIXES	5 HOURS AGO
○	CODE ADDITIONS/EDITS	4 HOURS AGO
○	MORE CODE	4 HOURS AGO
○	HERE HAVE CODE	4 HOURS AGO
○	AAAAAAA	3 HOURS AGO
○	ADKFJSLKDFJSDKLFJ	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAAAAAAAAAANDS	2 HOURS AGO

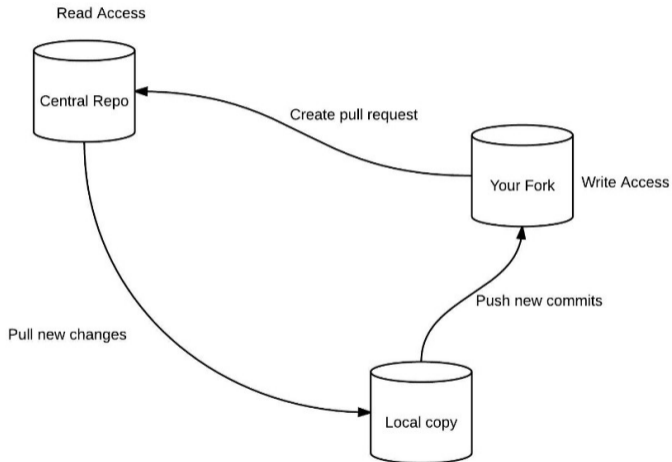
AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.

<https://xkcd.com/1296/>

What is GitHub?

- GitHub is a *service* for hosting Git repositories
 - GitHub is **not** Git (Git is a *tool*)
- Provides numerous useful features in addition to Git functionality
 - Review and compare code
 - Make pull requests
 - Submit issues and request features
 - Add wikis
- Also acts as a backup for important files
- GitHub is great, but you don't actually *need* GitHub to use Git
 - GitHub exists because Git does, not the other way around
- Other services like GitHub exist (although GitHub is the most popular and well-known)
 - BitBucket
 - GitLab
 - SourceForge

Git workflow (with GitHub)



<https://dab1nmslvntp.cloudfront.net/wp-content/uploads/2016/02/14550049531.jpg>

Git workflow (with GitHub)

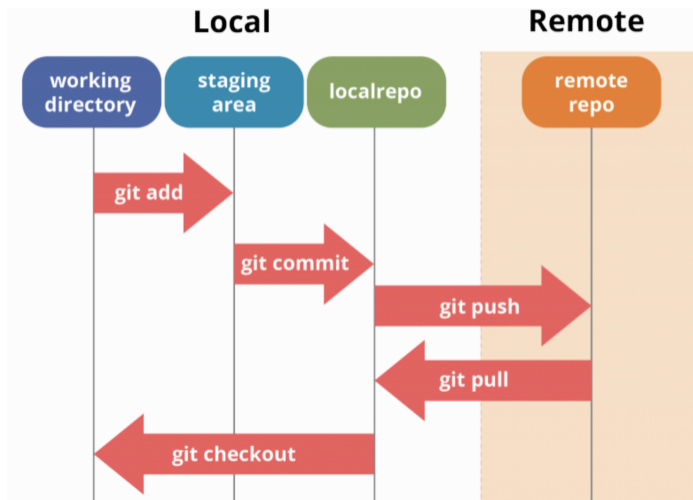
Cloned repository (local copy of the repository)

- Make changes to files
- Add them the “staging area”
- Make the change “permanent” by committing it
- Push changes to remote repository

Remote repository (your fork of the main repository)

- Compare changes to the main repository (can also do this from the local copy)
- Create a pull request

Git workflow (with GitHub)



<https://dev.to/mollynem/git-github--workflow-fundamentals-5496>

Git commands

git clone <GIT_REPOSITORY.git>	clone an existing repository
git diff <FILENAME.EXTENSION>	display changes to file
git add <FILENAME.EXTENSION>	add a file to the staging area
git checkout <BRANCH>	switch to different branch
git checkout <FILENAME.EXTENSION>	discard uncommitted changes to file
git log	displays commit logs of branch
git status	displays modified, added, and untracked files
git commit (-m "MY MESSAGE")	commit files in staging area (with message)
git push	push changes to remote
git pull	pull changes from remote

More Git commands (less frequently used, but still important)

git init	initialize repository (can do on GitHub)
git help <COMMAND>	get details on a git command
git remote -v <COMMAND>	view remote
git remote add <NAME> <GIT_REPOSITORY.git>	add remote repository to track
git config --list	list settings
git config --global user.name <FIRSTNAME LASTNAME>	change name
git config --global user.name <EMAIL@DOMAIN>	change email
git config --global credential.helper "cache --timeout=7200"	cache password in memory



Time for a demonstration

Further resources

- Once again, lots of material is not covered here
 - Git has numerous features that are more complicated, but very useful
 - Commit squashing and rebasing
 - Submodules
 - ...
 - Similarly, GitHub has numerous useful features that we didn't cover
 - Personal websites hosted via GitHub for free
 - File finder and searching
 - Tasks, assignees, mentions, milestones
 - Security vulnerability checks and alerts
- A lot of the material taken from the *Pro Git* book
 - Free and online: <https://git-scm.com/book/en/v2>