

Version Control

- An industry standard tool for managing software projects
 - Allows multiple developers to work on the same code
 - Manages changes to code over time
- Traditional version control
 - RCS, CVS, SVN
 - Still very common in big organizations
- Distributed version control
 - Git, Hg, Bazaar
 - Better for team work

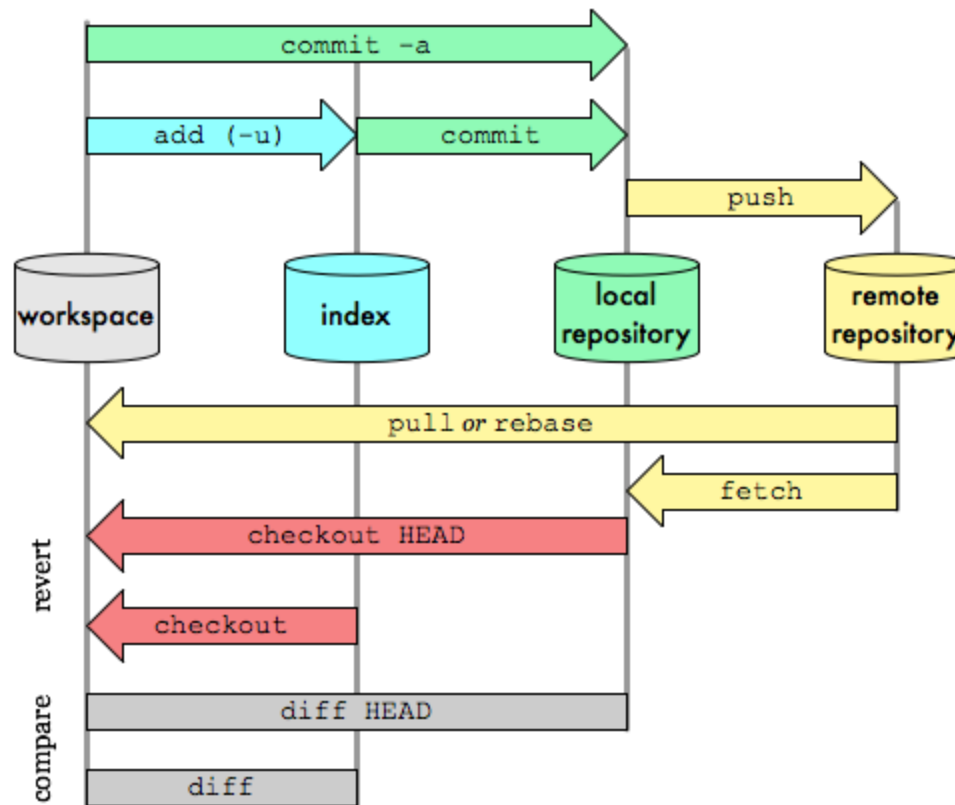
Version Control

- A *repository* is a store of all the files in a project
 - To work on the project, a developer *checks out* or *clones* a *working copy* of the project
 - Changes are made to that local copy, then saved back to the repository
 - All changes are kept, so the working copy can reflect the project at any point in time
 - This allows the project to roll back to earlier versions very easily
 - Changes are linked to a specific developer as well
 - The repository can be backed up so that no work is ever lost

How Git Works

Git Data Transport Commands

<http://osteele.com>



Setting up Git

- Linux
 - Command line interface, just install through the package manager
- Windows
 - Install Git for Windows
 - Then install TortoiseGit as a better graphical interface
- Mac
 - Install SourceTree
- GitHub (<https://github.com>)
 - Create a free account (just need an email, don't use your University email)

Git Classroom

- A free educational service that allows me to create assignments and send them out to you as GitHub repositories
 - You can then modify the code, commit and push to turn in
 - Commit all the time, always!
 - Push whenever you can without breaking the codebase for your team members
 - When you're working alone, that's all the time!

Git Classroom

- For each lab/assignment, there will be a GitHub URL
- Follow it and it will automatically set up your copy of the assignment to finish
 - Log into GitHub (if not already)
 - Authorize GitHub Classroom to automatically create repos for you (first time only)
 - Select your username from the class roster to link your GitHub account to your class identity (first time only)
 - Accept the assignment to create your repo copy

Submitting Your Work

- On your local computer, *clone* the assignment repo
 - This creates a local repo and local working copy
- Edit or add files to complete the assignment
- *Commit* to the local repository
- *Push* your changes to GitHub
 - All your work will be saved and timestamped
 - Push every time you work on the project, to save your work and show progress
 - Whatever is there at the deadline is considered your submission!

Submitting Your Work

- In the TortoiseGit context menu
 - *Check for Modifications* gives a quick way to see what changes you've made to the working directory
 - *Show Log* lets you see your past commits in detail
- At the command line, the *git status* command is hugely helpful in that regard