Git

History and Alternatives

Shannon Lee
Jonathan Miedel
Alvin Wang

Last time on Git Stuco (or perhaps earlier today)

- Working directory/index(stage)/repository
- add files to the staging area
- viewing changes in your working directory
- viewing git history
- writing good commit messages

Version Control Systems

- Three generations
- By time period
  - First generation 1972
  - Second generation 1990
  - Third generation 2005

First Generation

- RCS, SCCS (claimed to be the first VCS)
- operations completed one file at a time
- concurrency controlled through locks
- Drawbacks
  - only one developer per file at any given moment
  - difficult to manage large amounts of interdependent files
  - very little merge support

Second Generation

- CVS, SourceSafe, Subversion (very popular especially Apache subversion SVN), Team Foundation Server
- Multifile operations common
- Merge must be completed before committing

Third Generation

- BitKeeper, Bazaar, Mercurial, and Git
- Merge and commit separate allowing for faster development

Advantages of Distributed

- Workspace is more private
- Much faster
- Offline is possible
- Workflow more flexible (commits are not as big of a deal; branching is easier)
- Implicit Backups

Advantages of Centralized

- Locks are easier to implement
- Repository size
- Deletion
- Access Control
- Ease of use

Snapshotting vs Changeset

- Snapshotting
  - Stores compressed versions of files (ie. .pak)
  - git, SVN
- Changesets or patch sets
  - stores changes in files
  - mercurial

History of Git

- Linus Torvalds - father of Linux and Git
  - developed for the Linux Kernel project
- Bitkeeper revoked free-use access
- Linus set out to make his own
- Git was born in 2005

Goals of Git

- Speed - fast/scales pretty well
- Distributed - everyone has their own copy of the repository
- Secure - SHA1 hashing of files to maintain data integrity
- Non-linear development - easy branching

Next Week

- Viewing git log in more detail
- Modifying history
- Reverting files and branches

Homework 2 est (3-7 minutes)

- Download the .zip from the website, unzip it
- Add a text (.txt) file with your andrew id
  - ex. alvinw.txt
- Commit the changes with an appropriate message
- Set your remotes
- Push the changes
- Due before the **start** of next class

git push error

During this HW, you will find that if you try to push you may get an error similar to this:

```
To https://github.com/Dogfalo/dogfalo.github.io.git
 ! [rejected]        master -> master (fetch first)
error: failed to push some refs to 'https://github.com/Dogfalo/dogfalo.github.io.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

This is expected and means that someone has updated the repo and you have to grab these new changes before you can `git push`

To fix: `git pull`

Use `:wq` save and quit out of vim if it appears

Next Time on Git Stuco

- viewing history
- rewriting history

Questions


LOST
CONFUSED
UNSURE
UNCLEAR
PERPLEXED
DISORIENTED
BEWILDERED

Basic Command Line

- Directory Navigation and Management
  - cd <directory> (changes directory)
    - cd .. (goes up folder) cd - (goes to previous)
  - ls (lists files in current directory)
    - -a (list all)
  - mkdir <directory> (makes directory)
- File Management
  - rm <file> (removes file
    - rm -r (deletes directories) rm -f (forces delete)
  - mv <file-old> <file-new> (moves file)
  - cp <file> <directory> (copy file)
  - touch (creates file)

Continued

- Miscellaneous commands
  - cat <file> (outputs file contents)
  - less <file> (outputs file contents and allows scrolling)
  - clear (clears command window)

Basic Vim Commands

- :q (quit)
- :w (write)
- :! (force)
- a (append)
- u (undo)
- /<text> (search)