Hooks and Bisect

by Jonathan Miedel and Alvin Wang

Last time on Git

- Git internals
  - Refspec
  - Recovering Files
- HW 5 was released, due Today 11:59PM
  - Questions?

Hooks

- provide a method to run scripts before certain commands are run
- stored in .git/hooks
- client-side
- server-side

Use Cases

- Prevent rebasing of published commits
- Enforcing commit policy
- Continuous integration
  - verifying tests across program before allowing commit

Implementation

- Place a script with the appropriate name in the .git/hooks folder
  - Has examples
  - pre-commit, pre-push, pre-rebase
- Script must be executable

Client Side Hooks

- Local to the repository not cloned with git clone
- Pre-commit
- Pre-merge
- Pre-rebase
- Post-checkout
- Pre-push

Server Side Hooks

- Useful for project administration
- Can return errors to clients
- Pre-receive
- Update
  - once per branch
- Post-receive

Pre-commit hook framework by Yelp

"At Yelp we rely heavily on pre-commit hooks to find and fix common issues before changes are submitted for code review. We run our hooks before every commit to automatically point out issues like missing semicolons, whitespace problems, and testing statements in code. Automatically fixing these issues before posting code reviews allows our code reviewer to pay attention to the architecture of a change and not worry about trivial errors." --Yelp

Continuous Integration

- Integral in maintaining code quality
- Implies large number of merges per day
- Important in industry

Atlassian talk on CI with Git

https://www.youtube.com/watch?v=TWM1YNzSwB8

- It starts off briefly on branches in Git
- Talks about good workflows when working with CI

2:31 - 11:18

How to integrate CI

From a high level perspective:
- Set up your CI Server
  - Jenkins - very popular
  - CodeShip - web CI service (free plan available)
- Set up webhooks on development and master branches on the "post-receive" and "pre-commit" hook
- In the script your CI would return a non-zero if the tests fail
- Developer must have passing test before pushing to master

Git Bisect

- Git bisect helps you track down when bugs were introduced to your code
- Git bisect is a very useful command when working with many developers
- When working with a small team it is still fairly easy to figure out where these bugs were introduced, but on bigger projects with more collaborators, this is much harder.

git bisect

git bisect start <bad revision> <good revision>
label commits as bad or good after testing
- git bisect good
- git bisect bad

Bisect Activity

1. clone repo
2. run bisect to find the commit in which the bug was introduced
3. fix the error in every commit up to the recent one (something that you would normally do only on local branches if you plan to rebase into master) Hint: use rebase!
4. create a pull request (counts as participation)

Next Week in Git Stuco

- filter branch