Version control

2019-05-06 22:55
1a0f14a

Contents

Git resources

- Pro Git[1] book by Scott Chacon. Available online (gratis) or paper. I suggest working through at least chapters 1–3. We'll also pick up some things from chapters 7–8 later in the course.
- Understanding Git Data Model[2] article by Zvonimir Spajic. Great intro to the three types of objects: blobs, trees, and commits. Part of a series.
- Oh Shit, Git![3] is a fun overview and printable cheat sheet/booklet available for $10. (There's also a version without explicit language, if you prefer.)
- Beginner's Guide to `git bisect`[4] by Tony Rost. Incidentally, here's the invocation that I used to automate the search:

  ```
  git bisect run sh -c "! grep --count car test.txt"
  ```

  It would be run *after* doing `start` and marking the initial `good` and `bad` commits.

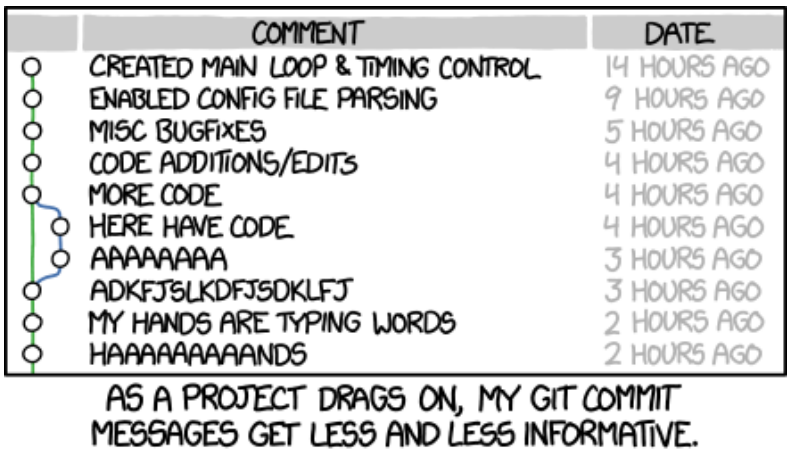

Figure 1: xkcd on git commit messages

Purpose

- We have files that represent code, configuration, documentation.
- Need tools to manage modifications
- Team environment means that multiple devs may edit the same file(s). Need to be careful about integrating changes.

[1] git-scm.com/book/en/v2

[2] hackernoon.com/https-medium-com-zspajich-understanding-git-data-model-95eb16cc99f5

[3] gumroad.com/l/oh-shit-git

[4] www.metaltoad.com/blog/beginners-guide-git-bisect-process-elimination

- However, VC even useful for a lone developer: to see previous versions, undo changes, redo, manage different configurations, etc.

History

Although git's model of snapshot-based, concurrent, and distributed version management is now dominant, it can be useful to understand some of the other design points that were used in the past.

Snapshots vs deltas

- One way that VC tools differ: do they manage **snapshots** of your files, or do they manage changes (aka **deltas** or **diffs**) to files?
  - Either store original and forward deltas,
  - Or store most recent and reverse deltas.
- Git (and friends) instead store snapshots – every version of every file. Faster to find old versions compared to applying deltas.
- Does take up more space than delta-based versions. Can use compression to reduce space.

Centralized vs distributed

- Centralized means there's some designated server that keeps all the history.
- Centralized also means browsing the history or adding to it requires network access to the server.
- Distributed means each developer has their own copy of the entire history.
- Distributed also means I can work while disconnected and then later push/pull.
- A distributed VC can also be a much-improved centralized VC.
- GitHub/GitLab are central servers for a distributed tool.

Winner: snapshots and distributed

- git (free/OSS) – GitHub (commercial)
- mercurial
- bzr "bazaar"
- BitKeeper