Basics of version control

Data plumbers' corner, session 7

Arianna Masciolini
Språkbanken Text, University of Gothenburg

23 January 2025

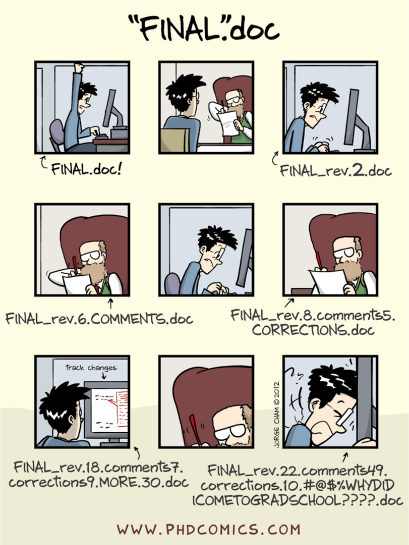
Today’s session

1. intro to version control and Git (again)
2. hands-on Git(Hub) tutorial

What you will need

- a Unix-like shell
- the basic Git CLI program
- a GitHub account

Manual versioning



"FINAL".doc
FINAL.doc!
FINAL_rev.2.doc
FINAL_rev.6.COMMENTS.doc
FINAL_rev.8.comments5.CORRECTIONS.doc
track changes
FINAL_rev.18.comments7.corrections9.MORE.30.doc
FINAL_rev.22.comments49.corrections.10.#@$%WHYDID ICOMETOGRADSCHOOL????.doc
JORGE CHAM © 2012
WWW.PHDCOMICS.COM

Version Control Systems

1. need to save disk space: **patch-based VCS** (e.g. RCS)
2. need for collaborative development: **centralized VCSs** (e.g. SVN)
3. need to work offline and prevent the server from being the “single point of failure”: **distributed VCSs** (e.g. Git, the current *de facto* standard)

Behind Git

Linus Torvalds, the creator of both Linux and Git

recommended: youtube.com/watch?v=4XpnKHJAok8

Git: basic concepts

	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
○	AAAAAAAA	3 HOURS AGO
○	ADKFJ5LKDFJ5DKLFJ	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAAAAAAAANDS	2 HOURS AGO

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

image from xkcd.com

- (named) filesystem snapshots rather than patches
- version history as a directed graph
- distributed, supports easy branching
- several interfaces and “hubs”

Hands-on Git tutorial

1. **local usage** (simple versioning)
2. **setting up a remote GitHub repository** (versioning + backup)
3. collaborative usage with GitHub