Version Control Systems

Git, Subversion, and all that

By Neil Roeth

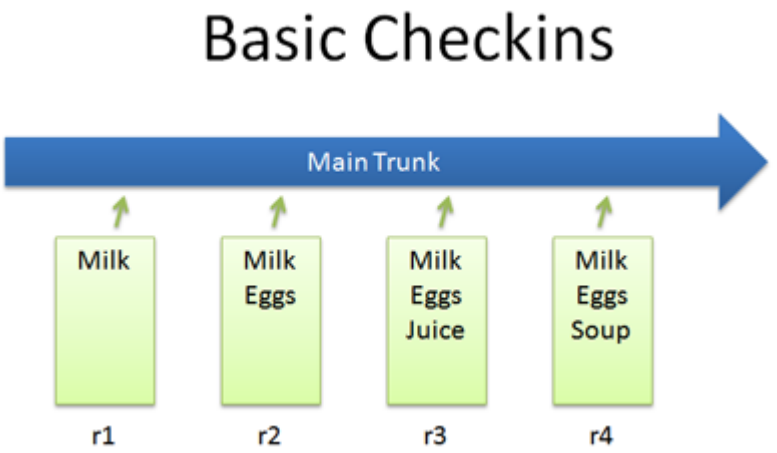
Why?

- Desire: You want to keep a history of a file or files (documents, images, source code…)
- Chaos: Name them file.txt, file.txt.old, file.txt.savethisone, file.txt.2017-05-18
- Better: Have file.txt and some mechanism to keep track of each version – version control
- Broader: Keep track of many related files (e.g., all of the source files for a program)

One File

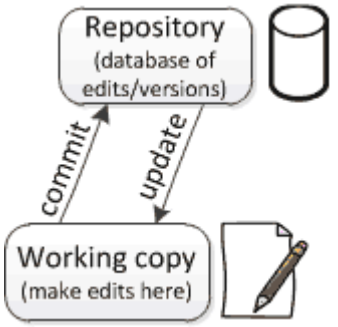
- Keep a record of each version – what changed and when
- Usually want the latest version, so most version control systems make it easy to get that
- You can get an exact copy of any version at any time
- Technical detail – only diffs are stored to save space

Version control for a grocery list



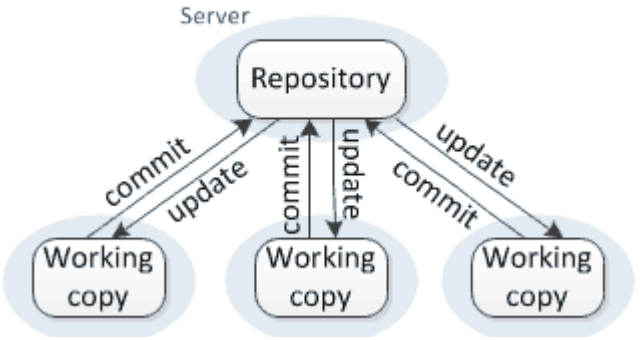
- “Check in” or “Commit” - put the current working copy in the repository – you choose when
- “Diff” - show differences between versions

Version Control Repository



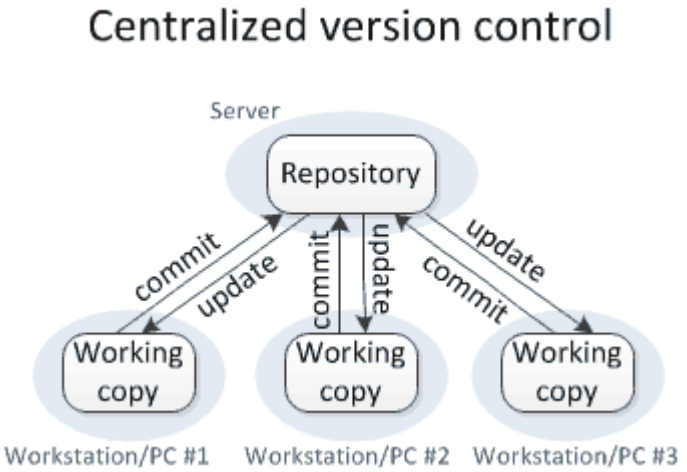
- Working copy is the version you are working on
- It does not get saved to the repository until you explicitly decide to save the current version

Multiple files



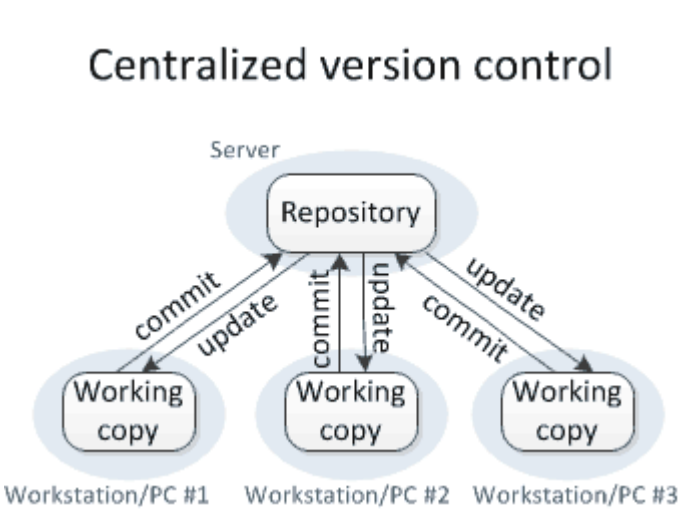
- Version control systems keep track of many related files
- Again, usually want latest of each
- Conceptually, “keep track of all but show me the latest of each”

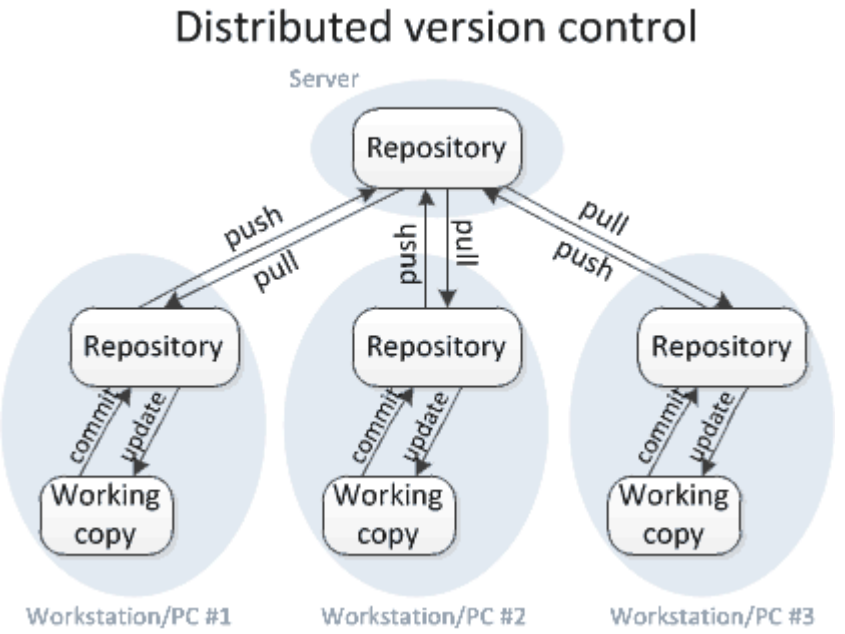
Multiple People



- Not just multiple files, but multiple people can work with version control
- Using a single repository, each person can commit and save their changes to the overall system

Centralized vs Distributed





- Can have a central repository where everyone checks in files (Subversion, CVS)
- Can have local repository that is periodically synchronized with central repository (Git, Mercurial)
- Can have no central repository at all (Git, Mercurial)

Groups of Files

- Can keep track of individual files
- People often want to track a group of related files – e.g., all source for a program, all files for a single Raspberry Pi Night
- Latest of all files is the current state of the whole system
- Can get all files as of a given time, or can tag them to retrieve as a group
- Branches for parallel tracks

Uses of Version Control

- In general, history – each commit has a timestamp and a note
- Program source code
- Configuration (/etc)
- Personal document history
- Drafts to final versions
- Make files available across multiple computers
- Share files with other people

Git

- A popular version control tool
- http://github.com/Hackerspace-Charlotte/RaspberryPiNight
- Distributed – usually a central repository (“repo”)
- “git clone” - copy central repo to local repo
- “git commit” - save changes to your local repo
- “git pull” - merge others’ changes to your local repo
- “git push” - merge your local repo with central repo
- “git status” - see files changed, working vs. local repo
- “git add” - add a changed or new file to next commit

Try it

- Clone repo (URL to be given) – git clone ….
- Create a file with a unique name, add it – git add neilsfile.txt
- Commit the file – git commit neilsfile.txt
- Make it part of the central repo – git push
- Get others' changes – git pull
- Check status – git status
- Show history – git log