Git

Amir Masoumzadeh

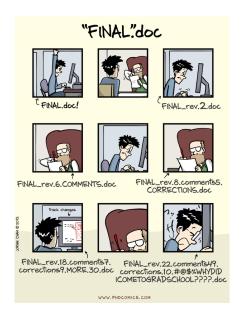
CSI 402 - Systems Programming

January 25, 2018

Administrivia

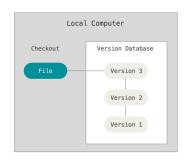
- GitHub ID
 - Have you submit it?
- 1st homework will be released by end of today
 - due next Tuesday (end of the day)
- TA₂ assigned (course page updated)
 - office hours Friday 2pm-4pm
- team names on Socrative

No version control



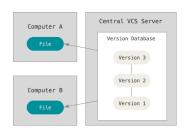
Local version control

- RCS
- simple database keeps all changes to files
- or, keeping patch sets



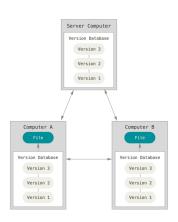
Centralized version control

- CVS, Subversion, Perforce, . . .
- central server repository (repo) holds "official copy"
 - also sole version history of the repo
- you make "checkouts" of it to your local copy
 - you make local modifications
 - your changes are not versioned
- when done, you "check in" back to the server
 - your checkin increments repo's version



Distributed version control

- Git, mercurial, . . .
- you "clone" from a repo and "pull" changes from it
- your local repo is a complete copy of everything on the remote server
 - yours is "just as good" as theirs
- many operations are local:
 - check in/out from local repo
 - commit changes to local repo
 - local repo keeps version history
- when you're ready, you can "push" changes back to server



Git snapshots

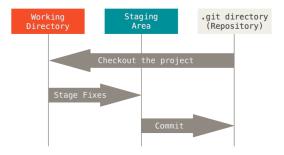
- centralized VCSs track version data on each individual file
- Git keeps "snapshots" of the entire state of the project
 - each checkin version of overall code has a copy of each file in it
 - some files change on a given checkin, some do not
 - more redundancy, but faster



Local Git areas

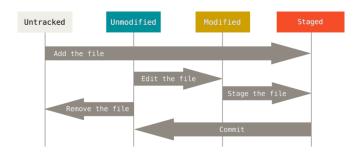
On your local machine files can be:

- in your local repo (committed)
- checked out and modified, but not yet committed (working copy)
- in-between, in a "staging" area
 - Staged files are ready to be committed
 - A commit saves a snapshot of all staged state



Basic Git life cycle

- modify files in your working tree
- stage files (selectively), adding snapshots of them to your staging area
 - also track files in your working directory
- commit, which takes staged files and stores that snapshot permanently to your Git directory



Using Git for lecture slides

- [optional] create a folder for this course and clone/create repos within that folder
- clone our lecture repo: https: //github.com/ualbany-csi402-s18/lectures
 - just pull it each time to retrieve the new lectures!

Using Git for homework assignments

- on release date of homework
 - we send you a link
 - you follow link and push a button so that a private repo will be created for you for that specific homework
 - e.g., homework01-azerrima
- for doing your homework, you
 - clone the repo
 - create/edit assignment files
 - add changes
 - commit
 - push [as often as you want]
- at the deadline, we
 - clone your repo to our machine
 - grade it (as soon as possible)
 - push a version that contains feedback/grade

Exercise

- Retrieve repo from: https: //github.com/ualbany-csi402-s18/inclass125
- In which snapshot random.txt contains random number 15139?
- Record the first 6 letter of SHA-1 hash code for the commit
- Submit it as your team answer on Socrative