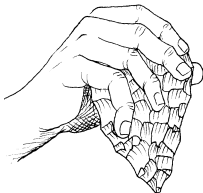


An introduction to git

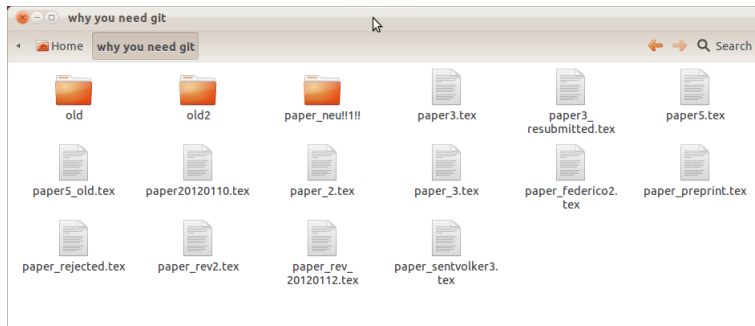
André Gaul Federico Poloni

TU Berlin

Tools Seminar
26 Jan 2012



Why you need git



Hmm, which one was the latest version?

Sounds familiar?

What git does

- Saves lots of “previous versions” of your work
- Manages them smartly (and lets you do extra tricks)
- Lets you share a “current working version” with your collaborators
- Different people can work on the same file seamlessly

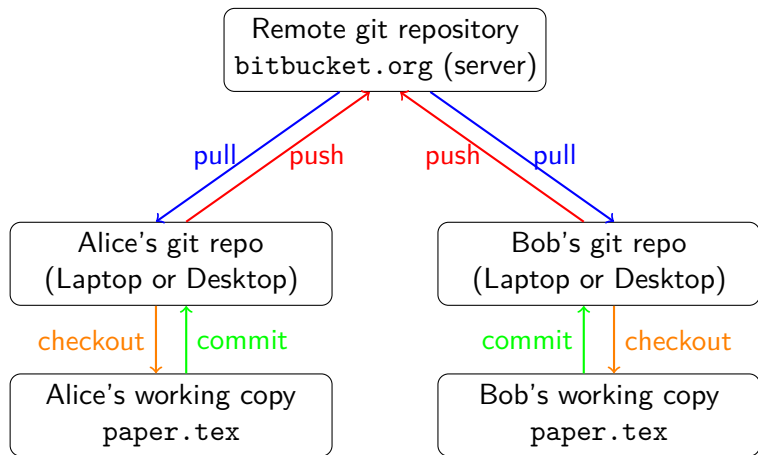
No need to bounce “here’s the new version” e-mail messages and combine files manually (with the risk of losing your work)

Git on a local machine

- One-time configuration:
 - ▶ `git config --global user.name "Donald E. Knuth"`
 - ▶ `git config --global user.email "knuth@microsoft.com"`
- `git init`
- `git add {filename}`
- `git commit [-a]`
- `git checkout`
- `git log`
- `git status`
- `gitk`

Easier to see it directly on the machine...

Local and remote repositories



Setting up a remote git repository

- One-time configuration:
 - ▶ ssh-keygen
 - ▶ Upload public key to Bitbucket account
- Once per project:
 - ▶ Create repo on Bitbucket, then
`git clone git@bitbucket.org:user/repo.git`
 - ▶ Or: `git remote add origin git@bitbucket.org:user/repo.git`,
then `git push -u origin master`
- `git pull`
- ...hack hack...!!!11!
- `git commit`
- `git push`

What can go wrong

- Two people edit different files → ok
- Two people edit different parts of the same file → ok
- Two people make the exact same modification → ok
- Two people make different modifications in the same point → **conflict**

Further topics

- Git is line-oriented, happier if your editor doesn't “reflow text”
- Best working practice: 1 “logical” change == 1 commit
Change = {rewritten introduction, changed a notation, fixed a typo}
- Lots of other useful commands: `git tag`, `git diff`, `git branch/merge`, `git bisect`, ...
- Works great with code
- Useful for you even if your collaborators don't use it

Further topics

- Git is line-oriented, happier if your editor doesn't “reflow text”
- Best working practice: 1 “logical” change == 1 commit
Change = {rewritten introduction, changed a notation, fixed a typo}
- Lots of other useful commands: `git tag`, `git diff`, `git branch/merge`, `git bisect`, ...
- Works great with code
- Useful for you even if your collaborators don't use it
- Thanks!

Further topics

- Git is line-oriented, happier if your editor doesn't “reflow text”
- Best working practice: 1 “logical” change == 1 commit
Change = {rewritten introduction, changed a notation, fixed a typo}
- Lots of other useful commands: `git tag`, `git diff`, `git branch/merge`, `git bisect`, ...
- Works great with code
- Useful for you even if your collaborators don't use it

- Thanks!
- Q & A ?

git cheatsheet

One-time config:

set name	<code>git config --global user.name "Donald E. Knuth"</code>
set email	<code>git config --global user.email "knuth@microsoft.com"</code>

Once per project:

clone remote repository	<code>git clone git@bitbucket.org:user/repo.git</code>
create empty repository	<code>git init</code>
link to bitbucket repo	<code>git remote add origin git@bitbucket.org:user/repo.git</code>
first push	<code>git push -u origin master</code>

Every day commands:

pull changes from remote repo	<code>git pull</code>
show changes in all files	<code>git diff</code>
show changes in specified files	<code>git diff paper.tex gmres.m</code>
overview of changes	<code>git status</code>
add specified files for commit	<code>git add paper.tex gmres.m</code>
commit to local repo	<code>git commit</code>
commit all changes	<code>git commit -a</code>
push commits to remote repo	<code>git push</code>
show last commit messages	<code>git log</code>
revert to last committed state	<code>git checkout paper.tex</code>

HELP!

show help for command	<code>git CMD --help</code>	$CMD \in \{add, commit, \dots\}$
-----------------------	-----------------------------	----------------------------------

Want to become a pro?

list all branches	<code>git branch</code>
create branch 'andre_hack'	<code>git branch andre_hack</code>
create version tag 'release2.3'	<code>git tag release2.3</code>