

Intermediate Git

Day 1: Understanding Git's Worldview

Raman A. Shah

Copyright (c) 2015 by Raman A. Shah.

Creative Commons BY-NC-SA 3.0 Unported.

https://github.com/ramanshah/intermediate_git

Some initial configuration

```
git config --list
```

If your user name and email are not set:

```
git config --global user.name \  
  "Raman A. Shah"  
git config --global user.email \  
  "raman@uchicago.edu"
```

If you don't like vim firing up in the middle of doing Git stuff:

```
git config --global core.editor "nano"
```

Git is...

... a distributed
version control
system.

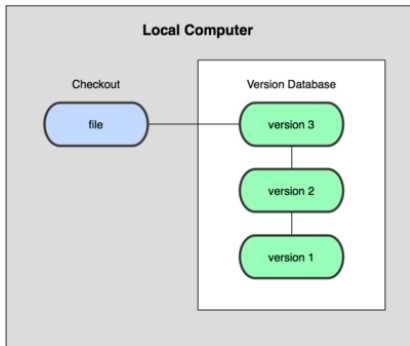
Git is...

... a distributed
version control
system.

Git is...

... a **distributed**
version control
system.

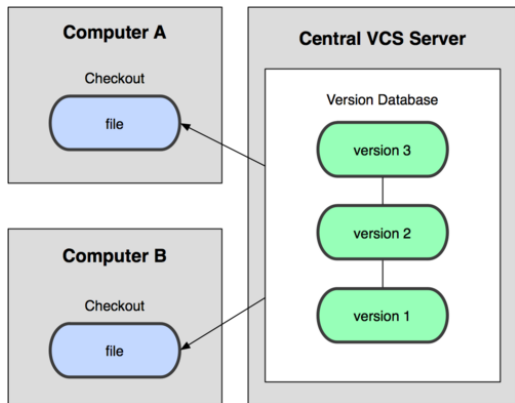
Git is...



Local version control (e.g., rcs).

Scott Chacon, *Pro Git*, Fig. 1-1. CC-BY-NC-SA.
<https://progit.org/>

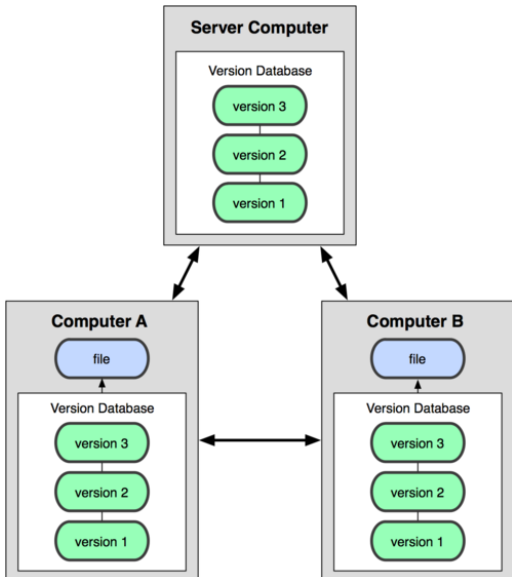
Git is...



Centralized version control (e.g., CVS, Subversion (SVN), Perforce).

Scott Chacon, *Pro Git*, Fig. 1-2. CC-BY-NC-SA.
<https://progit.org/>

Git is...



Distributed version control (e.g., rcs).

Scott Chacon, *Pro Git*, Fig. 1-3. CC-BY-NC-SA.
<https://progit.org/>

Git is...

... a great way to collaborate on projects consisting of many code or text files.

Git is...

... meant for perfecting
(software) *products*.

Git is...

... a content
addressable
filesystem.

Exploring repository internals

From a place where you wouldn't mind a new subdirectory:

```
git clone [URL]
```

```
cd [repo name]
```

```
git status
```

Exploring repository internals

Explore the contents of `.git` and `.gitignore`. To list a directory's contents including hidden “dotfiles”:

```
ls -al
```

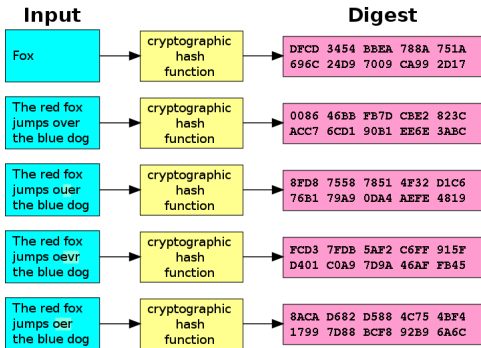
To write out the contents of a file to the terminal:

```
cat [filename]
```

Git is...

... safe because it
tracks every single
bit in your files and
commits with hash
functions.

Hashes (checksums)

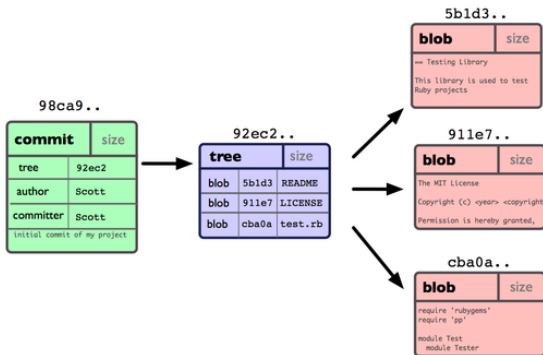


SHA-1 maps a file or text to a 160-bit value in a scramble way.

```
echo 'a' | sha1sum
```

```
sha1sum standup_snitch.py
```

Content addressability



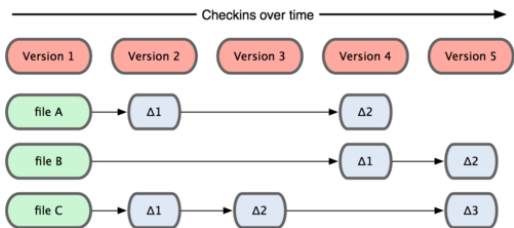
Content is snapshotted at the blob, tree, and commit levels.

Scott Chacon, *Pro Git*, Fig. 3-1. CC-BY-NC-SA.
<https://progit.org/>

Git is...

... fast because it stores
a (compressed) copy
of every version of
every file locally.

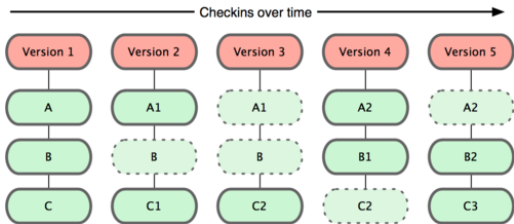
Git is...



Other version control systems require calculating versions of a file with diffs.

Scott Chacon, *Pro Git*, Fig. 1-4. CC-BY-NC-SA.
<https://progit.org/>

Git is...



Git just stores all (unique) versions.

Scott Chacon, *Pro Git*, Fig. 1-5. CC-BY-NC-SA.
<https://progit.org/>

Git is...

... hard because
efficiently managing
version control and
collaboration is
hard.*

Playing with the Past

`git log`

`git diff`

`git blame`

`git show`

`git checkout`

Reviewing history: git log

Default log; type q to quit:

```
git log
```

Limit the output to just the two most recent commits, and show some extra statistics:

```
git log --stat -2
```

A single line of output per commit:

```
git log --oneline
```

And much, much more.

```
git help log
```

Finding changes: git diff

HEAD is a “You Are Here” pointer. Tilde notation lets us walk back in history.

```
git diff HEAD~
```

Equivalently:

```
git diff HEAD~1
```

From three commits ago to one commit ago:

```
git diff HEAD~3 HEAD~1
```

You can specify with hashes, and single out specific files:

```
git diff [older hash] [newer hash] \  
[path]
```

Finding authors: git blame

```
git blame [path]
```

Good for:

- Blaming people for mistakes (as advertised)
- Figuring out whom to ask for guidance or code review

Seeing old versions: git show

To see the contents of an old version of a single file on the screen:

```
git show [commit]:[path]
```

You can redirect it to a file outside of the repo to recover an old version.

Time travel: git checkout

Rewrite the contents of the directory to reflect the repository one commit ago:

```
git checkout HEAD~1
```

Rewrite them back:

```
git checkout master
```

Git is not...

... GitHub.

Git is not...

... a great system for
documenting
(experimental)
projects.

Git is not...

... ideal for storing
bulky data.*

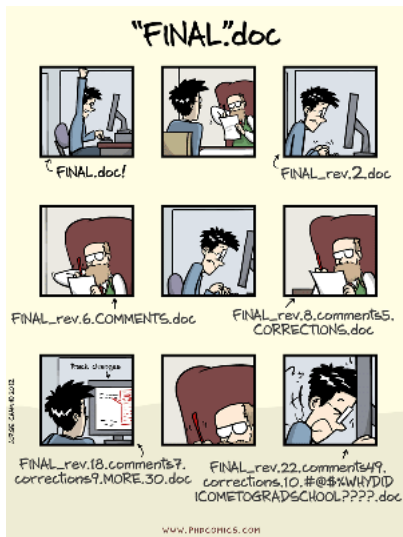
Git is not...

... quite as helpful for
binary files as for
text files.

Git is not...

... a silver bullet for
collaborating on
written works.

Git is...



... better than many alternatives!

"Piled Higher and Deeper" by Jorge Cham
www.phdcomics.com