

Git: Distributed Version Control

An Introduction

Peter E. DeWitt, Ph.D.¹
peter.dewitt@cuanschutz.edu

SOM — Department of Pediatrics — Informatics and Data Science

5 December 2019



SCHOOL OF MEDICINE

Department of Pediatrics

UNIVERSITY OF COLORADO ANSCHUTZ MEDICAL CAMPUS

Topics

- 1 Overview
- 2 Acquiring git
- 3 Basic Use
- 4 Remotes
- 5 Other Tools and Resources

"FINAL".doc



FINAL.doc!



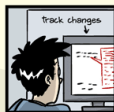
FINAL_rev.2.doc



FINAL_rev.6.COMMENTS.doc



FINAL_rev.8.comments5.
CORRECTIONS.doc



FINAL_rev.18.comments7.
corrections9.MORE.30.doc



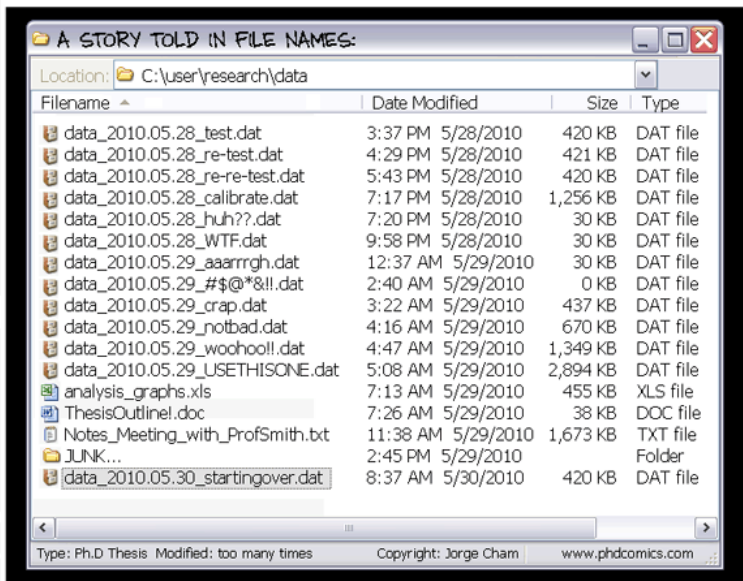
FINAL_rev.22.comments49.
corrections.10.#@%\$%WHYDID
ICOMETOGRADSCHOOL?????.doc

JORGE CHAN © 2012

WWW.PHDCOMICS.COM

1

¹<http://phdcomics.com/comics/archive.php?comid=1531>



²<http://phdcomics.com/comics/archive.php?comid=1323>

Tracking Changes

- The history of a project can be viewed as a series of changes:
 - ▶ A unique identifier
 - ▶ What changed?
 - ▶ When did it change?
 - ▶ Who changed it?
 - ▶ Why did it change?
- Difficult to manually track multiple files

Git: A Version Control System

A snapshot of the working directory is taken and *committed* to the git data base.

- Unique identifier
 - ▶ SHA-1 (determined by the files, the author, date, description of change, and the prior history)
- What changed
 - ▶ git diff
- Who changed it
 - ▶ git blame
- Why did it change
 - ▶ git log
- for example:

```
* 169a9d7 - Wed, 4 Dec 2019 21:39:52 -0700 (HEAD -> master)
|\
| * e68bab7 - Sun, 30 May 2010 09:14:00 -0600 (startover-again)
| |
| | Starting over, again - Peter DeWitt
* | ac99acc - Sun, 30 May 2010 08:37:00 -0600
|/
| Starting over - Peter DeWitt
* 00668c0 - Sat, 29 May 2010 11:38:00 -0600
|
| Add notes from meeting with Prof Smith - Peter DeWitt
* fadfdd4 - Sat, 29 May 2010 04:47:00 -0600 (tag: v0.1.0)
|
| USETHISONE - Peter DeWitt
* 2d09fc1 - Sat, 29 May 2010 04:47:00 -0600
|
| woohoo!! - Peter DeWitt
* efc90c6 - Sat, 29 May 2010 04:16:00 -0600
|
| notbad - Peter DeWitt
* 98f98e1 - Sat, 29 May 2010 03:22:00 -0600
|
| crap - Peter DeWitt
* e733f09 - Sat, 29 May 2010 02:40:00 -0600
|
| #$$*%!! - Peter DeWitt
* add83fb - Sat, 29 May 2010 00:37:00 -0600
|
| aaarrgh - Peter DeWitt
* 4eb4c09 - Fri, 28 May 2010 21:58:00 -0600
|
| WTF - Peter DeWitt
* 9b56981 - Fri, 28 May 2010 19:20:00 -0600
|
| huh? - Peter DeWitt
* e9288ae - Fri, 28 May 2010 19:17:00 -0600
|
| calibrate - Peter DeWitt
* d7f2fc7 - Fri, 28 May 2010 17:43:00 -0600
|
| re-re-test - Peter DeWitt
* 7250e57 - Fri, 28 May 2010 16:29:00 -0600
|
| re-test - Peter DeWitt
* 0244886 - Fri, 28 May 2010 15:37:00 -0600
|
| my test data and analysis - Peter DeWitt
```

What changed?

```
diff --git a/data.dat b/data.dat
```

```
index 88c8fc0..55fa6d2 100644
```

```
--- a/data.dat
```

```
+++ b/data.dat
```

```
@@ -1,2 +1,3 @@
```

```
1,2,3
```

```
5,6,7
```

```
+1,1,1,2,2,3,4,5,7,9,16,...
```

```
diff --git a/notes_metting_with_prof_smith.txt b/notes_metting_with
```

```
new file mode 100644
```

```
index 0000000..e69de29
```

- Diffs are easier to see in several GUI thanks to color coding. More on this later.

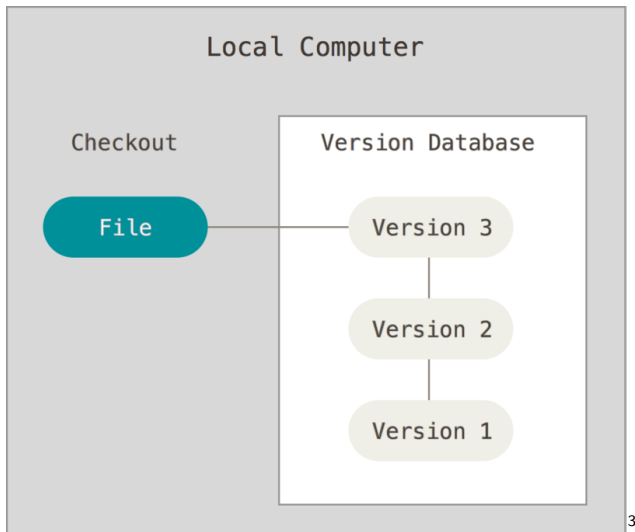
Skipping over errors

```
* 169a9d7 - Wed, 4 Dec 2019 21:39:52 -0700 (HEAD -> master)
|\      Merge branch 'startover-again' - Peter DeWitt
| * e68bab7 - Sun, 30 May 2010 09:14:00 -0600 (startover-again)
| |      Starting over, again - Peter DeWitt
* | ac99acc - Sun, 30 May 2010 08:37:00 -0600
|/      Starting over - Peter DeWitt
* 00668c0 - Sat, 29 May 2010 11:38:00 -0600
|      Add notes from meeting with Prof Smith - Peter DeWitt
* fadfdd4 - Sat, 29 May 2010 04:47:00 -0600 (tag: v0.1.0)
|      USETHISONE - Peter DeWitt
```

Conceptual Git and Version Control

- Local Version Control
- Centralized Version Control System
- Distributed Version Control System
- Short History and Design of Git

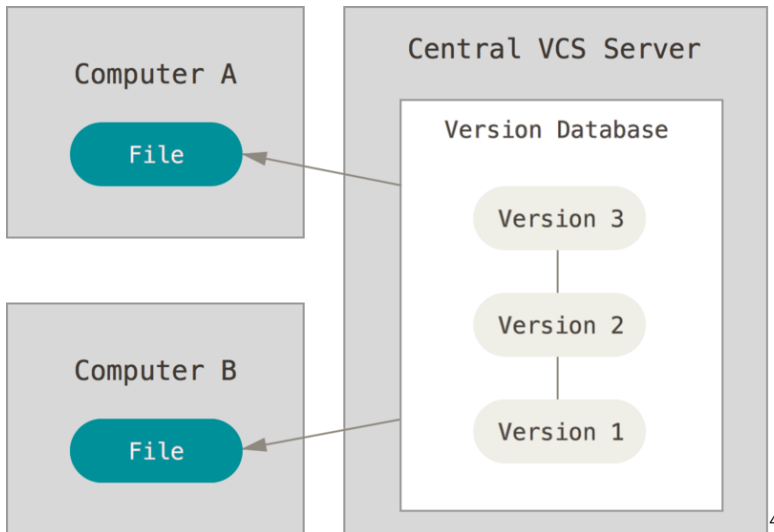
Local Version Control System



3

³<https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control>

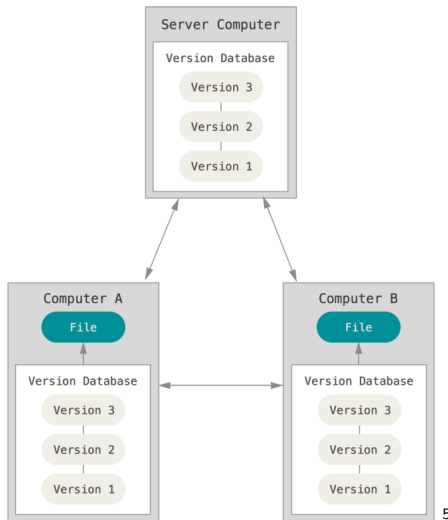
Centralized Version Control System



4

⁴<https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control>

Distributed Version Control System

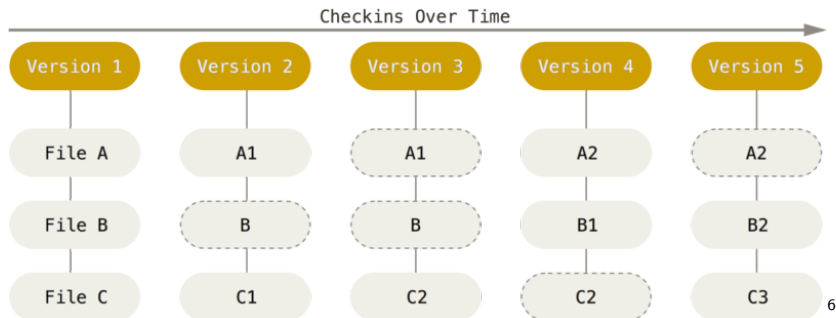


⁵<https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control>

Short History of Git

- Original Author: Linus Torvalds
- Version 0.99 released July 2005
- Current version 2.24
- Linux vs BitKeeper
- Linux dev community sets out to develop their own DVCS with the goals of:
 - ▶ speed
 - ▶ simple design
 - ▶ strong support for non-linear development (thousands of parallel branches)
 - ▶ fully distributed
 - ▶ able to handle large projects, like the Linux kernel, efficiently

Snapshots; Not Differences



⁶<https://git-scm.com/book/en/v2/Getting-Started-What-is-Git%3F>

Nearly Every Operation is Local

- Most operations are local
 - ▶ No need to talk to other computers on a network
 - ▶ The entire project history is on your local machine
- You can work off vpn
- You can work offline

Git Has Integrity

- Everything is check-summed (remember the sha?)
- You cannot lose information in transit nor get a file corruption without git being able to detect it
- git stores everything in its database not by file name but by the hash value of its contents
- git has been designed for multiple parallel development, i.e., multiple programmers contributing to one project non-linearly in time

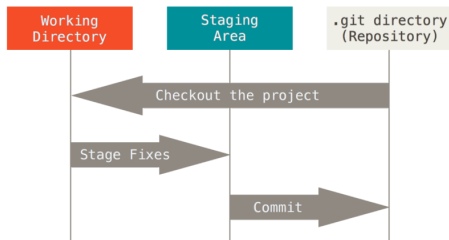
Git Generally Only Adds Dat

- Nearly all actions in git add data to the git database
- It is difficult to do anything that is not undoable
- You can lose/corrupt un-committed changes
- It is very difficult to lose anything after a commit, especially with frequent pushes to other repositories

The Three Stages

- Git has three main states that files reside in
 - ▶ Committed: data is safely stored in the local database
 - ▶ Modified: changed the file(s) but have not committed to the data base yet
 - ▶ Staged: marked modified files(s) in current version to go into the next commit snapshot
- This leads to three main sections of a git project
 - ▶ the .git directory
 - ▶ the working directory
 - ▶ the staging area

The Three Stages



Basic Git workflow

- Modify: make changes in the working directory
- Stage: adding snapshots to the staging area
- Commit: move the snapshot in the staging area to the .git database

Git is Free!

- Download from: <https://git-scm.com/download>
- Other options:
 - ▶ Linux: install via your favorite package manager
 - ▶ apt-get install git
 - ▶ yum install git
 - ▶ Mac: Xcode command line tools
 - ▶ Install from source: <https://github.com/git/git>

Terminal or GUI

- git was built for the terminal
- Graphical User Interface (GUI) for git exists
 - ▶ Gitkraken
 - ▶ RStudio
 - ▶ A list of GUI clients is available at <https://git-scm.com/downloads/guis/>

Set Up

- You only need to do this once per machine:

```
git config --global user.name "Firstname Lastname"
```

```
git config --global user.email "first.last@institution.xxx"
```

- Use the terminal (git bash shell even on Windows) or some GUIs will support this

Basic Use

- Since this is an R in Data Science Class, let's walk through the use of git in RStudio.
- Please remember that you can interact with git in many different ways:
 - ▶ terminal (my preferred method)
 - ▶ Gitkraken
 - ▶ RStudio
 - ▶ ...
- A list of GUI clients is available at <https://git-scm.com/downloads/guis/>

Working with Remotes

- So far we have only looked at working with git locally
- git is *distributed* version control
- remotes are copies of the repository on
 - ▶ another directory on your computer
 - ▶ a network drive
 - ▶ another computer
 - ▶ a repository hosting server
 - ▶ gitlab.com
 - ▶ github.com
 - ▶ bitbucket.org
 - ▶ your own institutional server

Working with remotes

Pros and Cons

- Pros

- ▶ Collaboration!
- ▶ Every copy of the repo is a whole copy of the history - every copy is a back up for every other copy
- ▶ Additional tools can be used, depending on the remote/server
- ▶ Controls for who can read/write to the project

- Cons

- ▶ Be very careful about publicly hosted repository hosts. *DO NOT COMMIT SENSITIVE DATA TO A REPO*

Setting Up a Remote

- Let's create a github repository for the example project
- Introduce more git verbs
 - ▶ push
 - ▶ fetch
 - ▶ merge
 - ▶ pull

Resources

- git documentation
<https://git-scm.com/doc>
- GitKraken <https://www.gitkraken.com/>
- Peter DeWitt:
peter.dewitt@cuanschutz.edu



^a<https://xkcd.com/1597/>