# An Introduction to Git

Shelley Knuth
Research Computing

# Outline

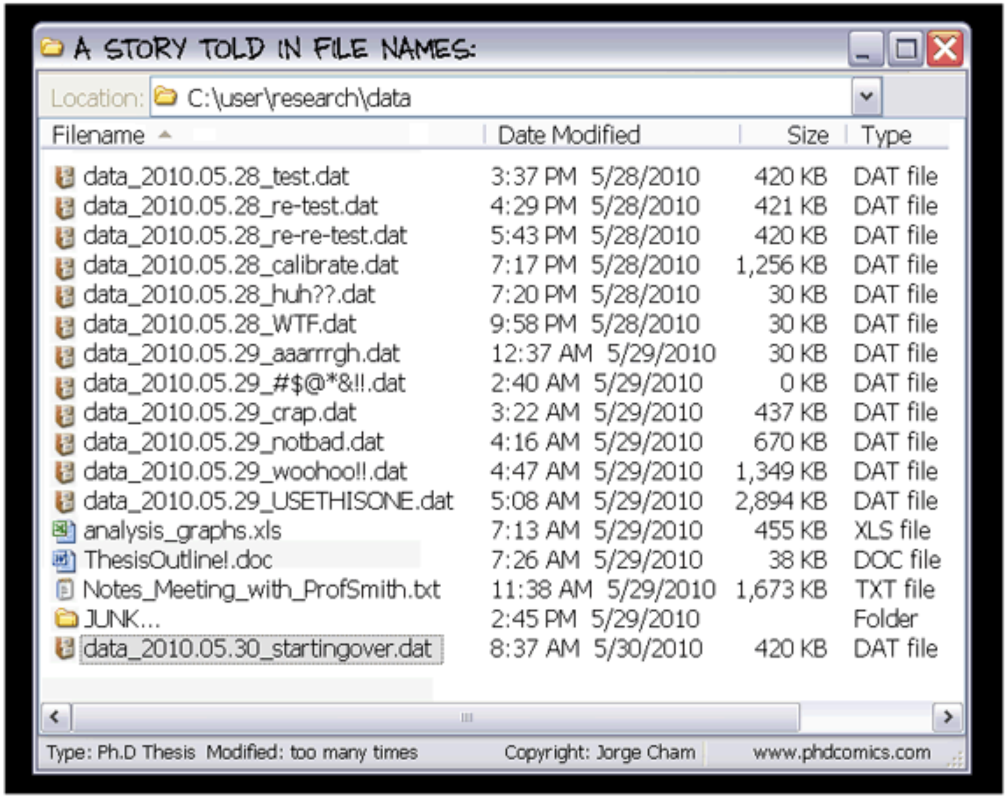
- What is version control?
- What is git?
- Let’s manage some code!
- Differences between git and other version control systems?

# What is Version Control?

- Version control records changes to a file or to code
- Keeps track of:
  - What changes are made
  - Who makes them
- By keeping track of changes, you can recall previous versions later

# Version Control and You

- Probably already using version control
  - Rename files
  - Copy into another directory
  - Easy to overwrite
  - Which version is newest?



A STORY TOLD IN FILE NAMES:

Location: C:\user\research\data

| Filename | Date Modified | Size | Type |
|---|---|---|---|
| data_2010.05.28_test.dat | 3:37 PM 5/28/2010 | 420 KB | DAT file |
| data_2010.05.28_re-test.dat | 4:29 PM 5/28/2010 | 421 KB | DAT file |
| data_2010.05.28_re-re-test.dat | 5:43 PM 5/28/2010 | 420 KB | DAT file |
| data_2010.05.28_calibrate.dat | 7:17 PM 5/28/2010 | 1,256 KB | DAT file |
| data_2010.05.28_huh??.dat | 7:20 PM 5/28/2010 | 30 KB | DAT file |
| data_2010.05.28_WTF.dat | 9:58 PM 5/28/2010 | 30 KB | DAT file |
| data_2010.05.29_aaarrgh.dat | 12:37 AM 5/29/2010 | 30 KB | DAT file |
| data_2010.05.29_#$@*&!!.dat | 2:40 AM 5/29/2010 | 0 KB | DAT file |
| data_2010.05.29_crap.dat | 3:22 AM 5/29/2010 | 437 KB | DAT file |
| data_2010.05.29_notbad.dat | 4:16 AM 5/29/2010 | 670 KB | DAT file |
| data_2010.05.29_woohoo!!.dat | 4:47 AM 5/29/2010 | 1,349 KB | DAT file |
| data_2010.05.29_USETHISONE.dat | 5:08 AM 5/29/2010 | 2,894 KB | DAT file |
| analysis_graphs.xls | 7:13 AM 5/29/2010 | 455 KB | XLS file |
| ThesisOutline!.doc | 7:26 AM 5/29/2010 | 38 KB | DOC file |
| Notes_Meeting_with_ProfSmith.txt | 11:38 AM 5/29/2010 | 1,673 KB | TXT file |
| JUNK... | 2:45 PM 5/29/2010 | | Folder |
| data_2010.05.30_startingover.dat | 8:37 AM 5/30/2010 | 420 KB | DAT file |

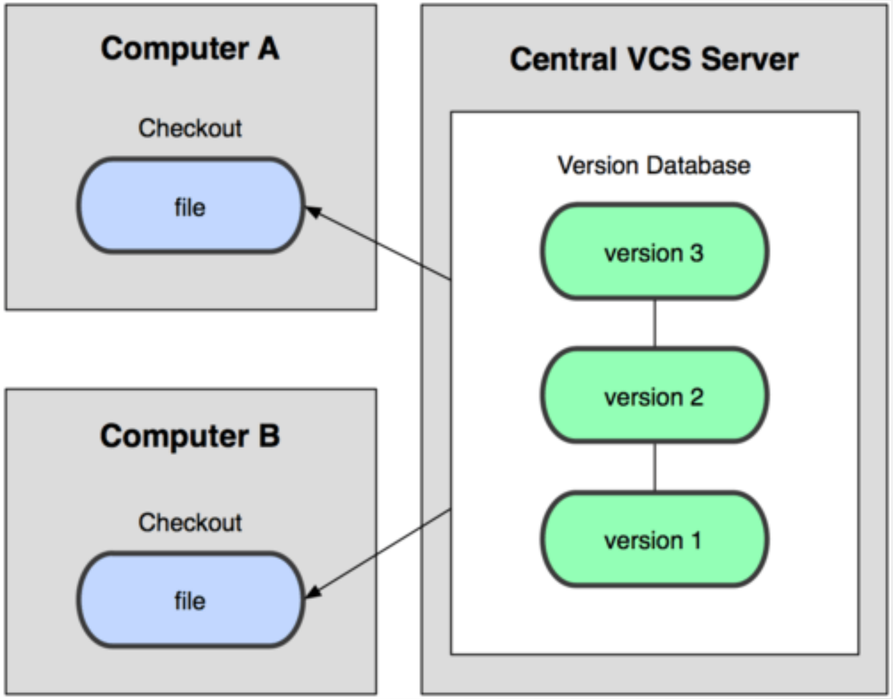
Type: Ph.D Thesis Modified: too many times Copyright: Jorge Cham www.phdcomics.com

# Version Control Advantages

- Revert files back to a previous state
- Easy recovery
- See who might have changed a file that is causing issues
  - History of changes by who and when
  - More efficient tracking of documents, code
- Local or centralized

# Centralized Version Control

- Collaborations
- Single server
- Holds all versions of files
- Different people can check out files from a central location



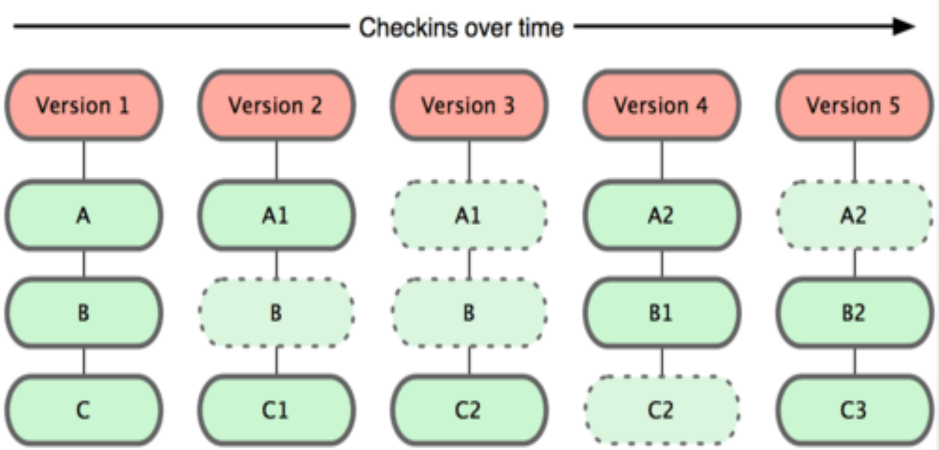
git-scm.com

# Version Control Disadvantages

- Server goes down, loss of productivity
  - Centralized system
- How good is everyone at keeping backups?
  - Centralized and local

# Distributed Version Control Systems

- Git is a distributed version control system
- Instead of checking out the latest version of the file, checks out a snapshot of all files at that time
- Easier to recover from loss
- Only needs local resources to operate, generally



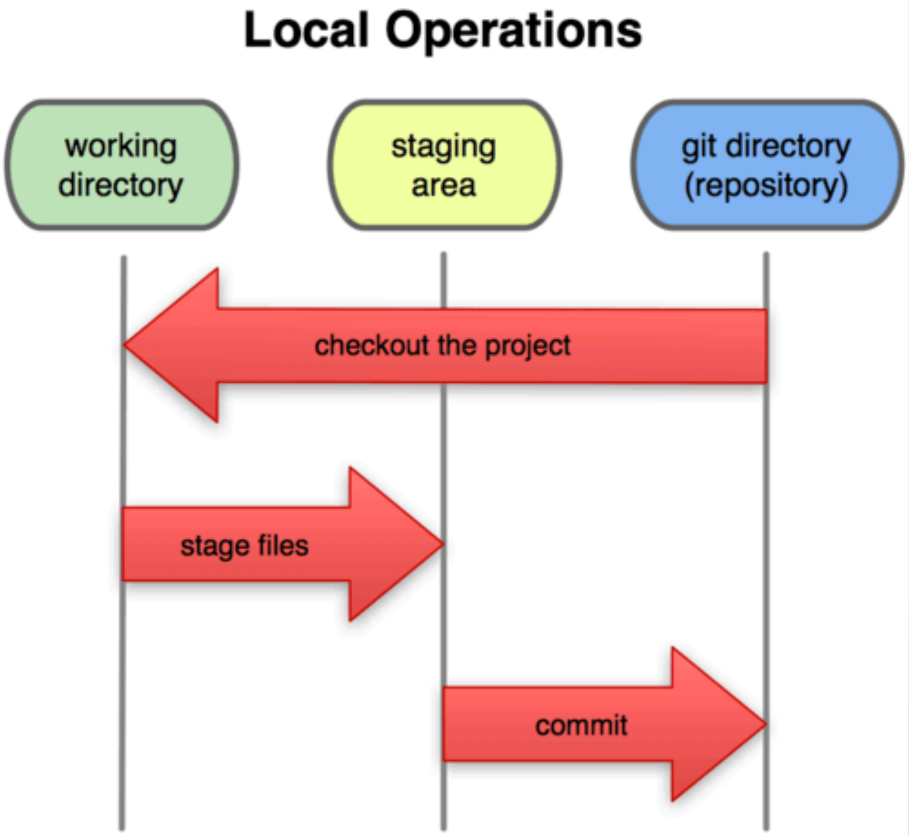
git-scm.com

# Local Control

- Most of what you need to do on git can be done locally
- Don’t need to continually connect to a server
- Useful if are out of internet contact
- Can commit all your changes and upload to a server when you have access

# How Does Git Operate? - States

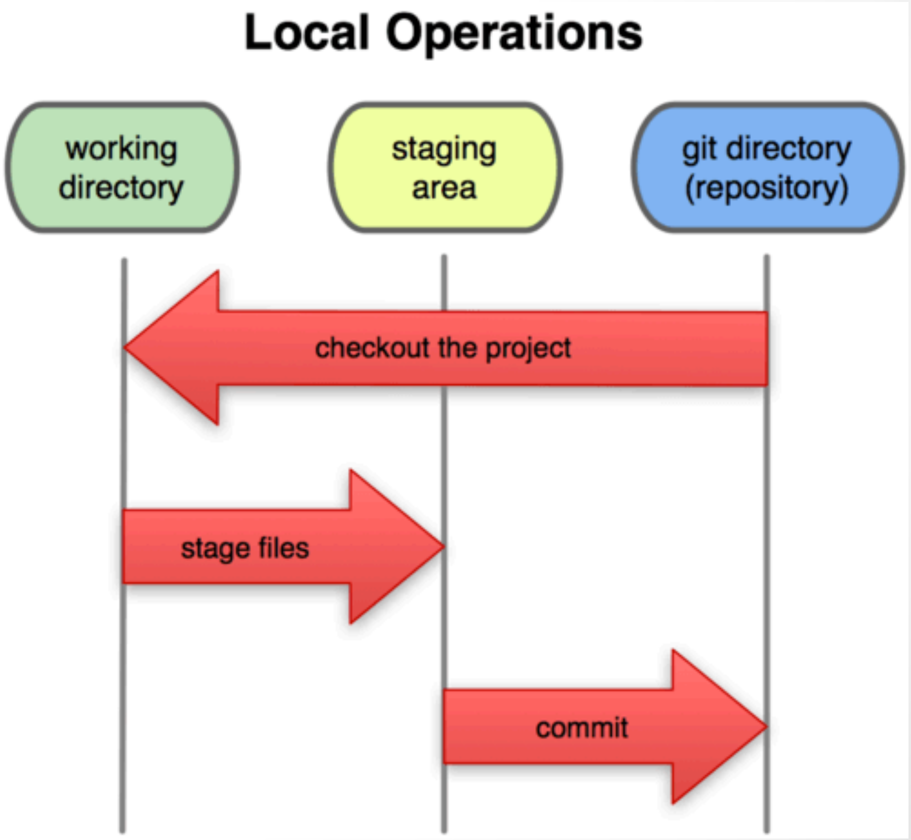
- Commit state
  - Data is stored in database
- Modified state
  - The file has been changed
  - Not yet committed
- Staged state
  - Marked a modified file to go into the next commit



git-scm.com

# How Does Git Operate? – Sections of a Project

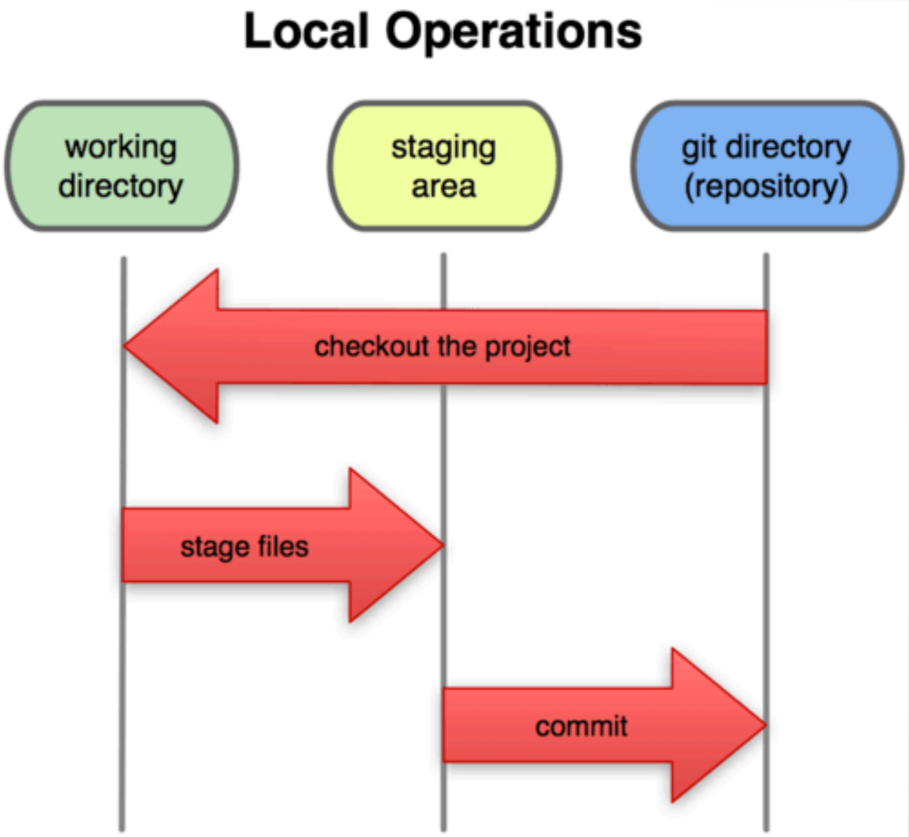
- Git directory
  - Where the metadata is located
  - Very important
- Working directory
  - Contains a checkout of a version of your project
- Staging area
  - Contains a file that has information about what goes into your next commit



git-scm.com

# Life Cycle of a Project

- Files in working directory are modified
- Files get staged
- You commit the files
  - Files in staging area have a snapshot that gets permanently stored to the git directory



git-scm.com

# Getting a Git Repository

- Get a repository from either copying one that exists or importing an existing project to git
- You import the existing project based on what we've already done
- To copy an existing git repository, type

```
git clone https://github.com/mlunacek/meetup_d3_2014.git
```

- Receive a copy of all the data the server has
- Not a checkout of the latest version of code

# Getting a Git Repository

- You never write directly to a remote repository
- You need to create a corresponding local branch which tracks the remote branch
- You can push or pull changes to or from the remote branch

**git pull**

**git push**

- Usually you have a public repository where everyone pulls changes from
- You push your changes there

# Let’s Work on Some Code

- Git_tutorial_doc.txt

# A Few Key Items About Git

- Git operates so that each working copy carries its own repository with it (found in .git)
  - Can have local and remote branches
  - Svn has a single repository for each project at a remote central location
- Git identifies revisions with long hexadecimal numbers
  - Svn defines revisions with growing decimals

# Git Vs. SVN

| Task | Git | SVN |
|---|---|---|
| **Tracking someone else's project** | Git clone *url*<br>Git pull | Svn checkout *url*<br>Svn update |
| **Committing** | Git commit | svn commit |
| **Check status** | Git status | Svn status |
| **Check changes in files** | Git diff | Svn diff |
| **Restore a file from the last version** | Git checkout *path* | Svn revert *path* |

# Questions?

- git-scm.com (Lots of great information in this talk was found here!)
- try.github.io/levels/1/challenges/1 (Learn git in 15 minutes!)
- Git.or.cz/course/svn.html

- Shelley.Knuth@colorado.edu