

COMP6700/2140 Version Control with *Git*

Steve Blackburn and Josh Milthorpe

Research School of Computer Science, ANU

9 March 2017

Version Control

- Indispensable software engineering tool
- Solitary work
 - Personal audit trail and time machine
 - Establish when bug was introduced
 - Fearlessly explore new ideas (roll back if no good)
- Teamwork
 - Concurrently develop
 - Share work coherently

Distributed version control system

- hg, git, others (conceptually very similar)

Contrast with centralized version control

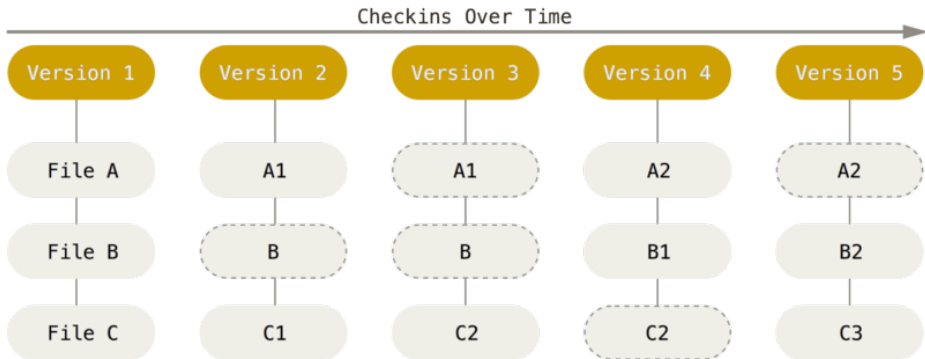
- cvs, svn, others

We will focus on distributed version control systems and not discuss centralized version control any further.

Git - Filesystem Snapshots

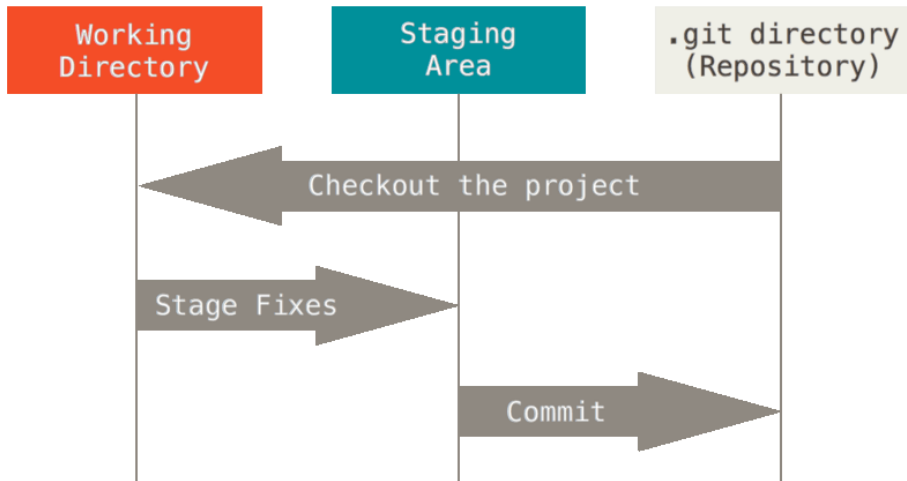
Git stores a series of snapshots of a filesystem over time.

Every time you *commit*, Git stores a snapshot.



Git - Project Areas

Files in Git can be modified, staged or committed. Any file that has never been staged or committed is *untracked*.



Git Concepts

- Changeset
- Commit (atomically commit changes to your local repo)
- Push (push outstanding local changes to a remote repo)
- Pull (pull new changes from a remote repo)
- Update (update your working version)
- Merge
- Reset and Revert

Git Changesets

A Git commit captures a set of changes (including modifications, additions and deletions) that may span multiple files.

- Globally unique commit ID (large hexadecimal number)
- Parent – child relationship (based on changeset ID)
 - Single parent, single child is simple case
 - Multiple children indicates a *branch*
- Multiple parents indicates a *merge*
- A *push* sends commits, a *pull* gets commits
- Commits are usually never deleted

Git Update

By default, *update* will take you to the “HEAD” (the most recent known commit).

You can, however, “update” to any particular revision, moving yourself back and forward in time. To do this, you need to specify the revision.

In IntelliJ you can do this by double-clicking on the revision (VCS -> Git -> Show History, then select the revision)

IntelliJ Git Integration

Create a new repository:

- VCS -> Import into Version Control -> Create Git Repository...

Clone an existing repository:

- VCS -> Checkout from Version Control -> Git...

Other operations:

- VCS
- VCS -> Git
- right mouse click -> Git

Further Reading

- Chacon & Straub *Pro Git*
<https://git-scm.com/book>
(The images used in this lecture were taken from the Pro Git Book.)
- Code School *Try Git* <https://try.github.io/>