

02-3 - git

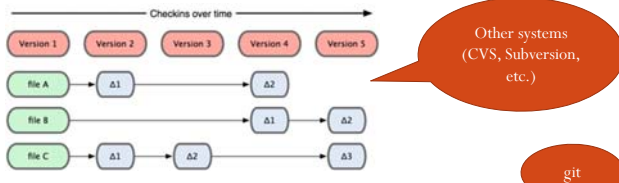
Much of this is taken from...
Pro Git
 professional version control
<http://progit.org/book>



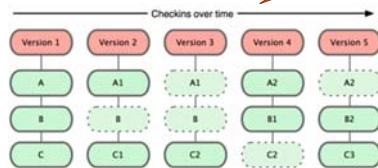
What is git?

- A distributed revision control system with an emphasis on being fast
- Initially designed and developed by Linus Torvalds for Linux kernel development
- Every Git working directory is
 - a full-fledged repository
 - with complete history and
 - full revision tracking capabilities,
 - not dependent on network access or a central server

Keeps Snapshots, not Differences



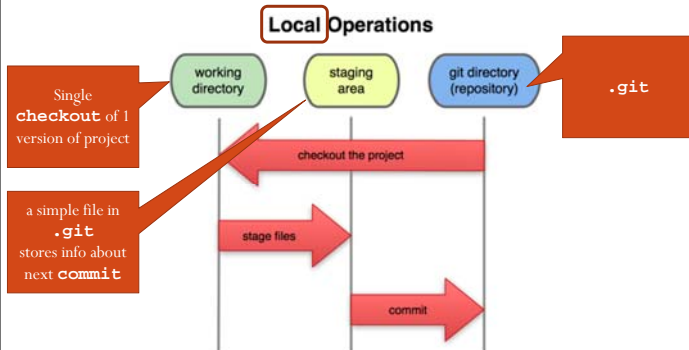
- Stores a picture of all files at every commit
- Uses links for unchanged files



Directory Details

- The Git directory (**.git**) is where Git stores the metadata and object database for your project
- This is the most important part of Git
- It is what is copied when you **clone** a repository from another computer

Three sections of a Git project



Git workflow

- Modify files in your working directory
- Stage the files, adding snapshots of them to staging area
- Commit, takes the files as they are in the staging area and stores that snapshot permanently to your Git directory



Terms

- **Committed:** Version of file in `.git` directory
- **Staged:** modified and *added* to the staging area
- **Modified:** changed since it was checked out but has not been staged



Configuration Files

- `/etc/gitconfig` : values for every user on the system
- `~/.gitconfig` file: Specific to you.
- `.git/config` : config file for current repository
 - Specific to that single repository
 - Each level overrides values in the previous level

```
$ cat ~/.gitconfig
[user]
    name = Mark A. Yoder
    email = Mark.A.Yoder@Rose-Hulman.edu
[github]
    user = MarkAYoder
    token = a8836c841ce558a8f52af0a7bd1dbe79
```

`.git/config`

```
$ cat .git/config
[core]
    repositoryformatversion = 0
    filemode = true
    bare = false
    logallrefupdates = true
[remote "origin"]
    fetch = +refs/heads/*:refs/remotes/origin/*
    url = git@github.com:MarkAYoder/gitLearn.git
[branch "master"]
    remote = origin
    merge = refs/heads/master
```

Things to configure

```
$ git config --global user.name "Mark A. Yoder"
$ git config --global user.email Mark.A.Yoder@Rose-Hulman.edu
$ git config --global core.editor gedit
```

```
$ git config user.name
Mark A. Yoder
```

```
$ git help
$ git help config
```

The *git lab* leads you to **github** which will lead you through these commands.

Your Diff Tool and Settings

- Configure the default diff tool to use to resolve merge conflicts.
 - Say you want to use **vimdiff**:
- ```
$ git config --global merge.tool vimdiff
```
- git accepts `kdiff3`, `tkdiff`, `meld`, `xxdiff`, `emerge`, `vimdiff`, `gvimdiff`, `ecmerge`, and `opendiff` as valid merge tools.
- ```
$ git config --list
```
- To see your settings

Here's what you did

```
$ git clone git://git.angstrom-distribution.org/setup-scripts oe
    • Creates a directory named "oe",
    • initializes a .git directory inside it,
    • pulls down all the data for that repository, and
    • checks out a working copy of the latest version

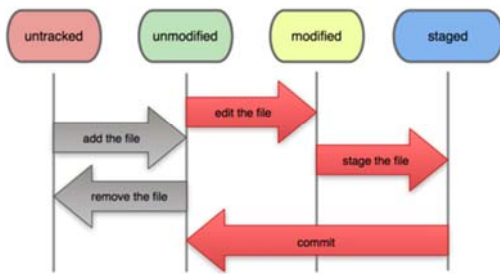
$ ls -F oe/.git
branches/  description  hooks/    info/     objects/  refs/
config    HEAD        index     logs/    packed-refs
```

File Status Lifecycle

See *git lab* for more details

- See: <http://progit.org/book/ch2-2.html>

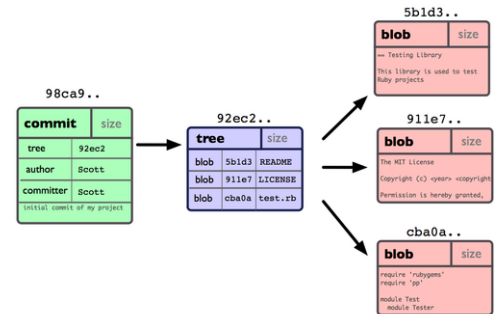
File Status Lifecycle



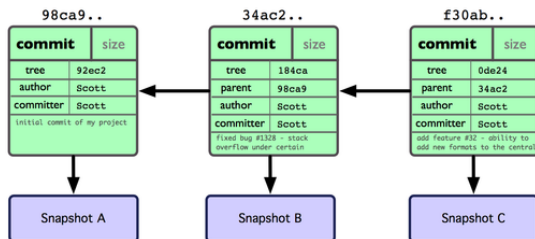
Branching

```
$ git add README test.rb LICENSE
```

```
$ git commit -m "initial commit of my project"
```

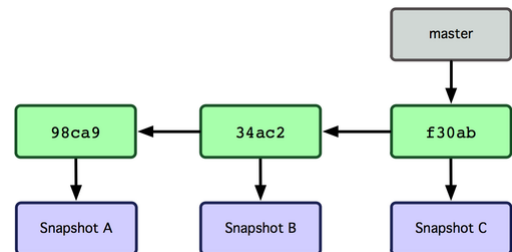


After 2 more commits



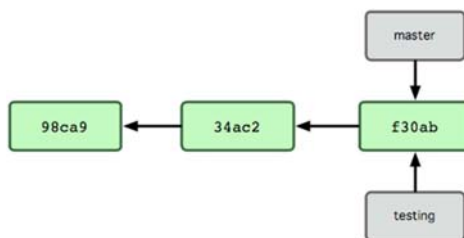
master

- A branch is a lightweight movable pointer to a commit
- Default: **master**



New branch

```
$ git branch testing
```

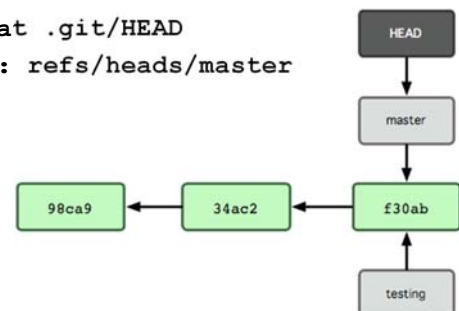


What's the current branch?

- HEAD**

```
$ cat .git/HEAD
```

```
ref: refs/heads/master
```

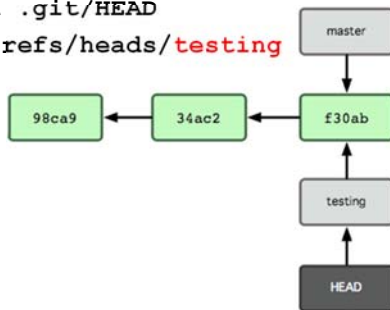


Switch branches

```
$ git checkout testing
```

```
$ cat .git/HEAD
```

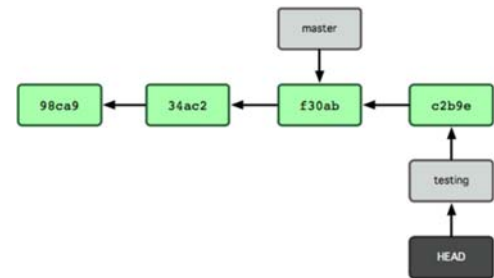
```
ref: refs/heads/testing
```



Another commit

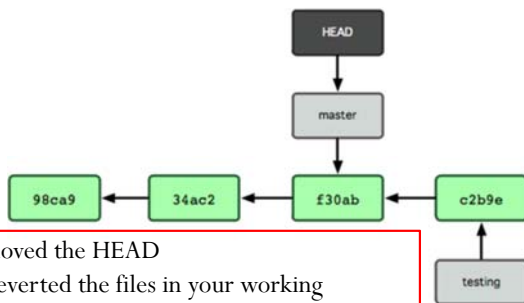
```
$ gedit test.rb
```

```
$ git commit -a -m 'made a change'
```



What does this do?

```
$ git checkout master
```

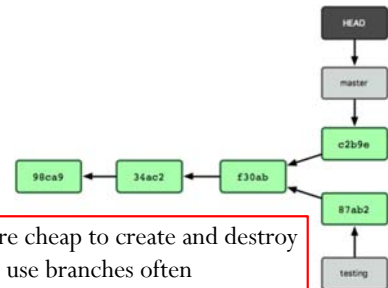


- Moved the HEAD
- Reverted the files in your working directory back to the snapshot that master points to.

More changes

```
$ gedit test.rb
```

```
$ git commit -a -m 'made other changes'
```



- Branches are cheap to create and destroy
- Create and use branches often

Merge

- See: <http://progit.org/book/ch3-2.html> for a merge example

You'll do this in the *git lab*