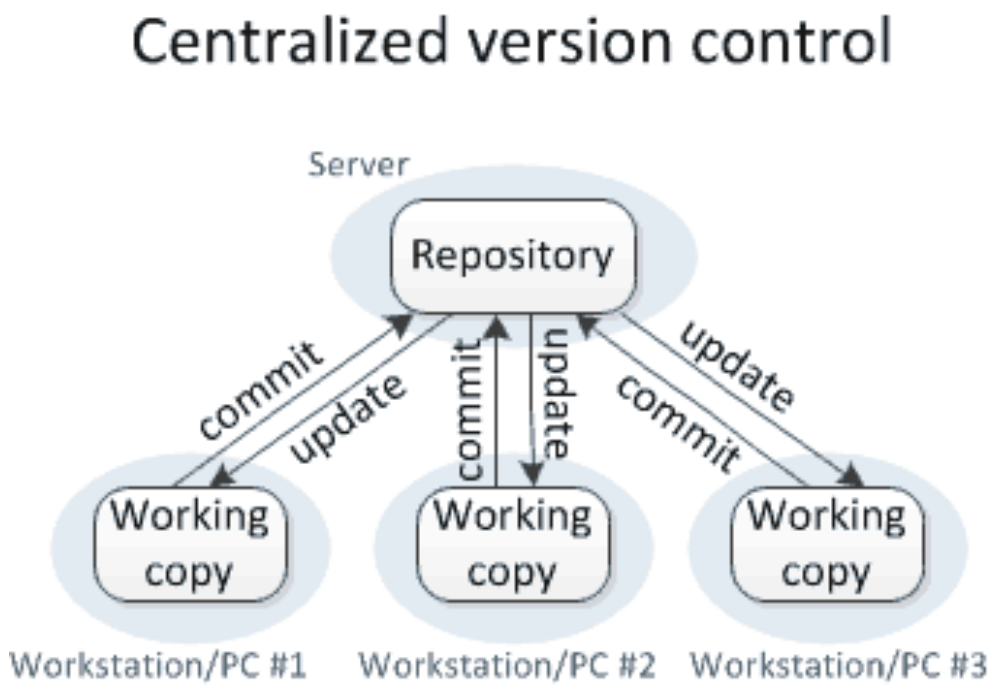
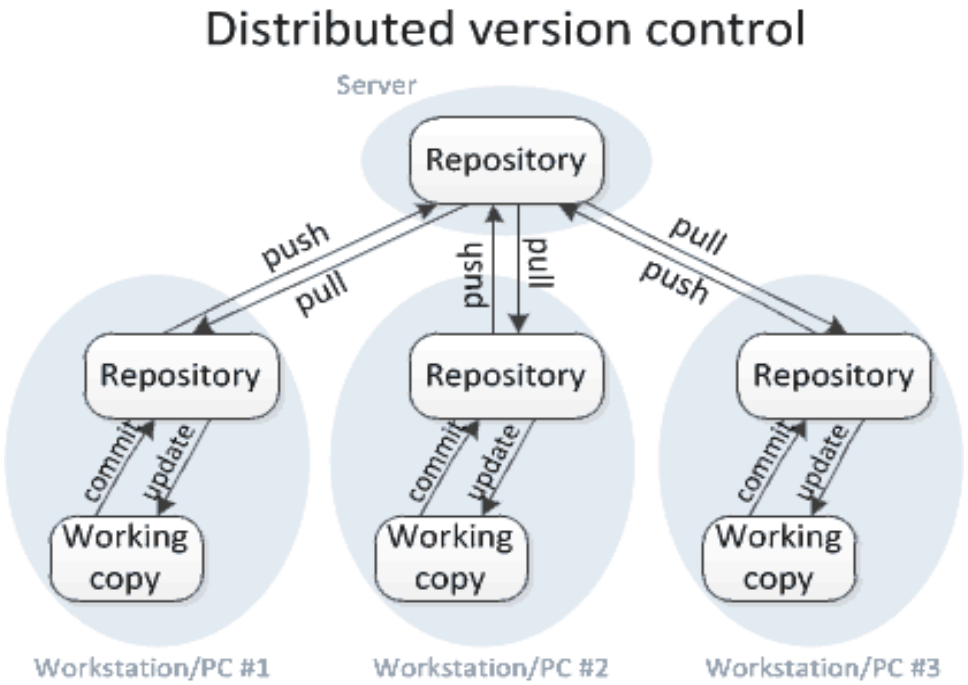
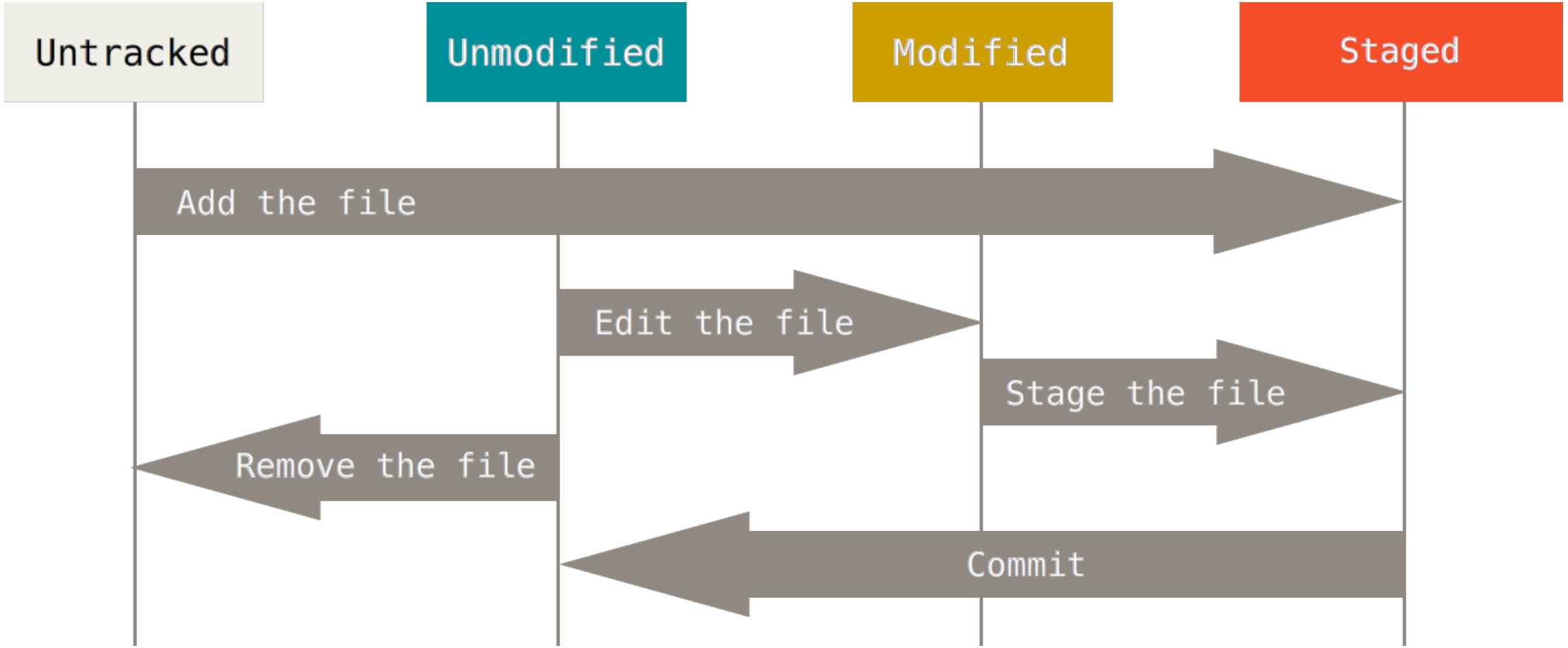

git

Why is git using distributed model?





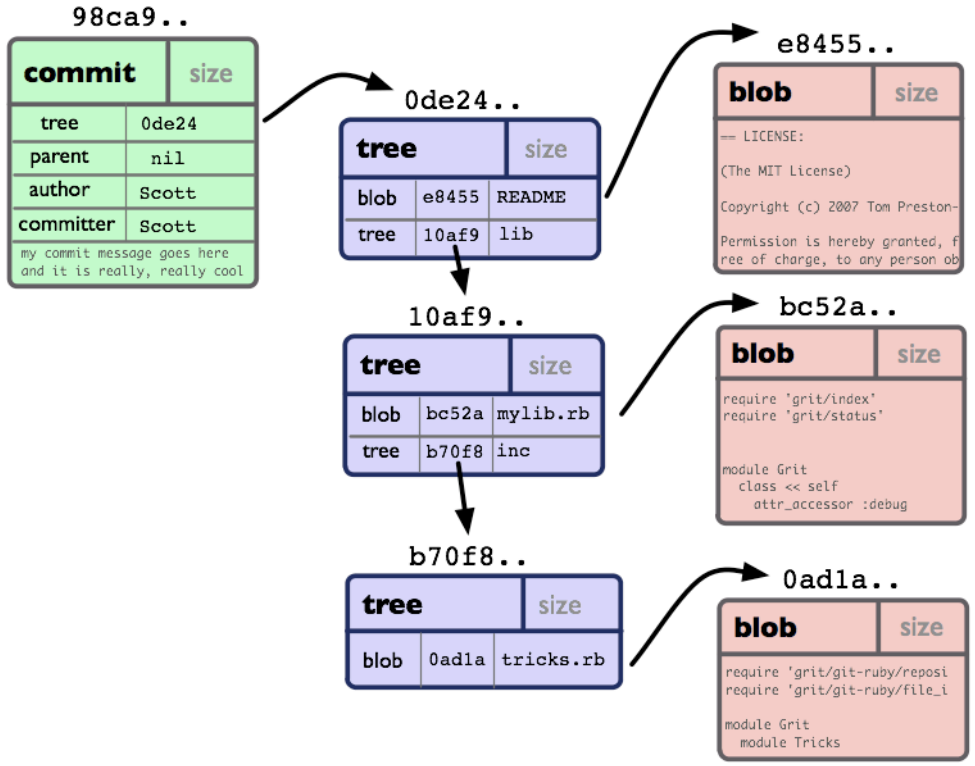
High-level View



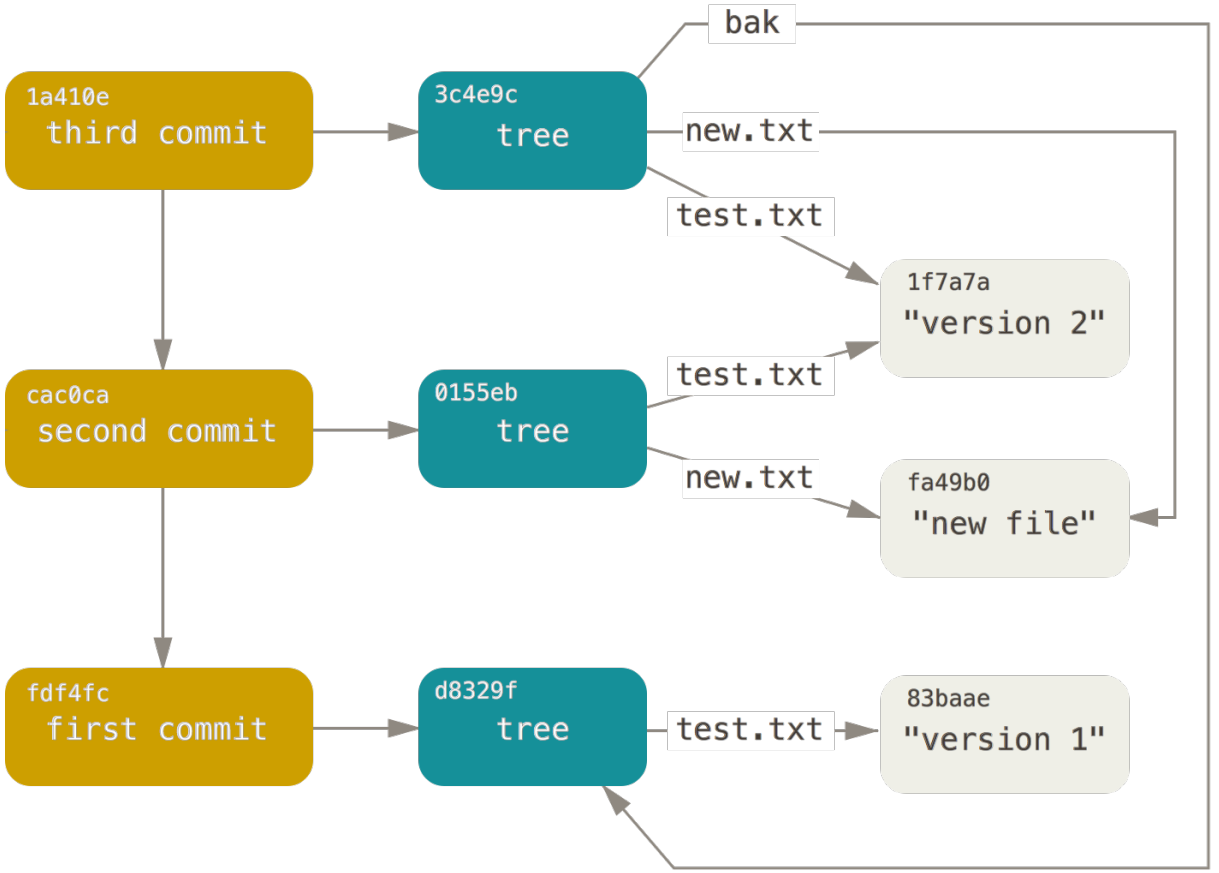
Git Internals

```
$ ls -F1
HEAD
config*
description
hooks/
info/
objects/
refs/
```

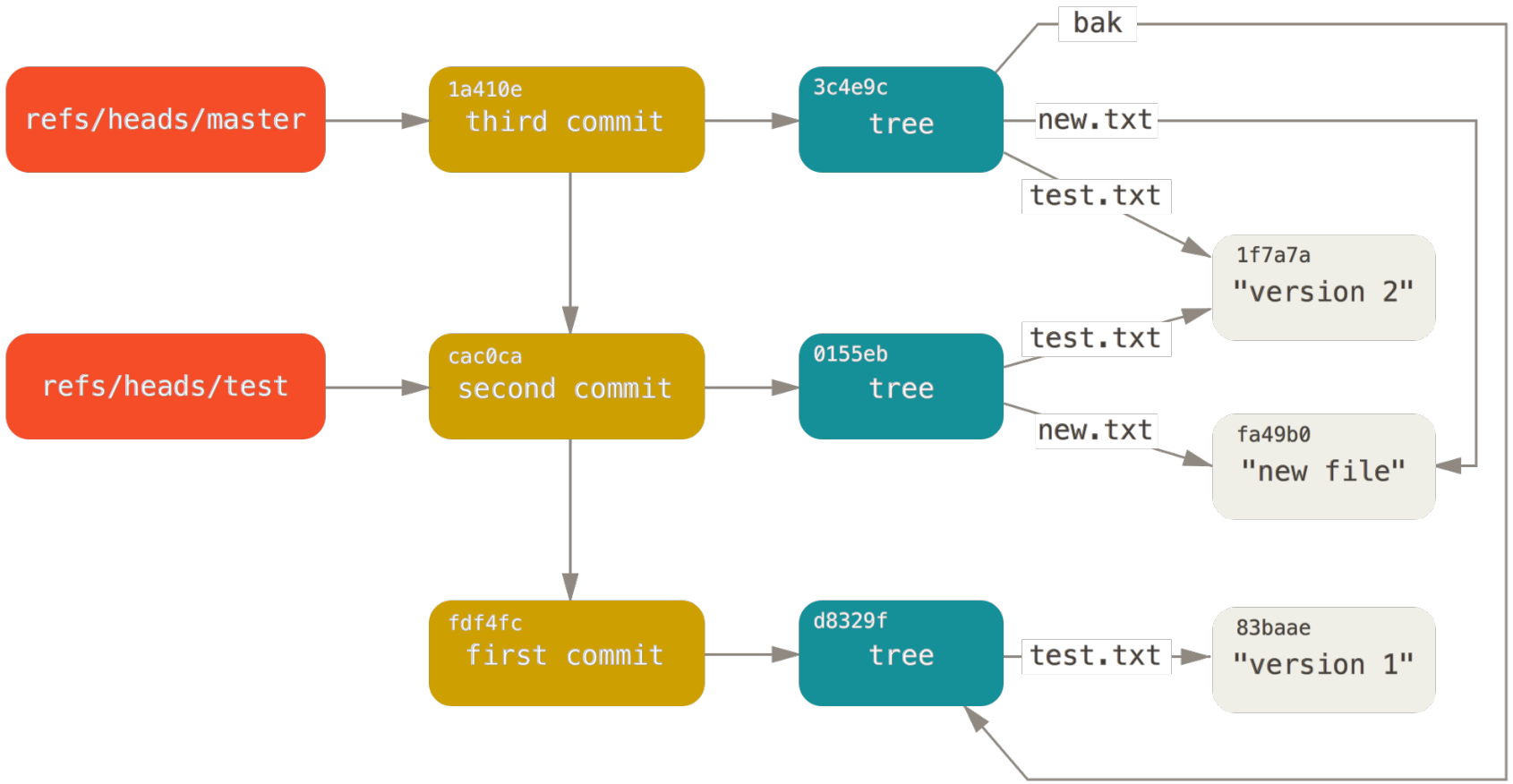
Git Object Model



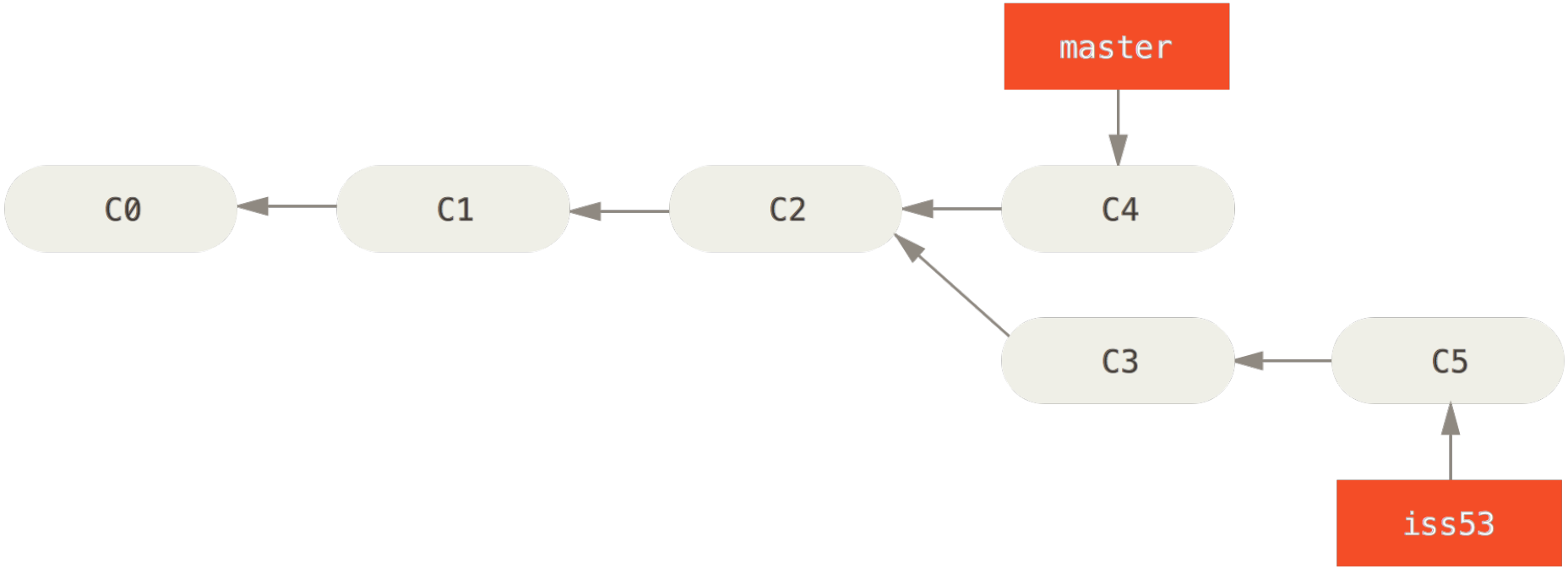
Git Object Model



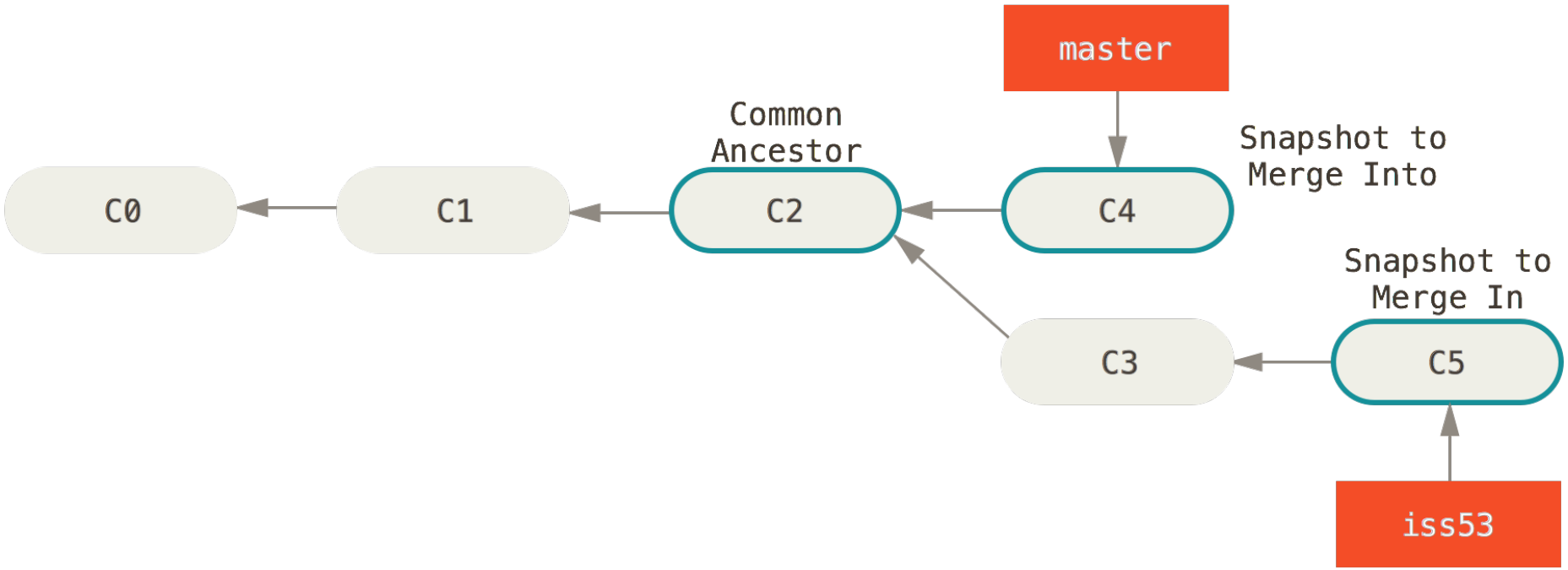
Refs


refs/heads/master
1a410e
third commit
3c4e9c
tree
bak
new.txt
test.txt
1f7a7a
"version 2"
refs/heads/test
cac0ca
second commit
0155eb
tree
test.txt
new.txt
fa49b0
"new file"
fdf4fc
first commit
d8329f
tree
test.txt
83baae
"version 1"

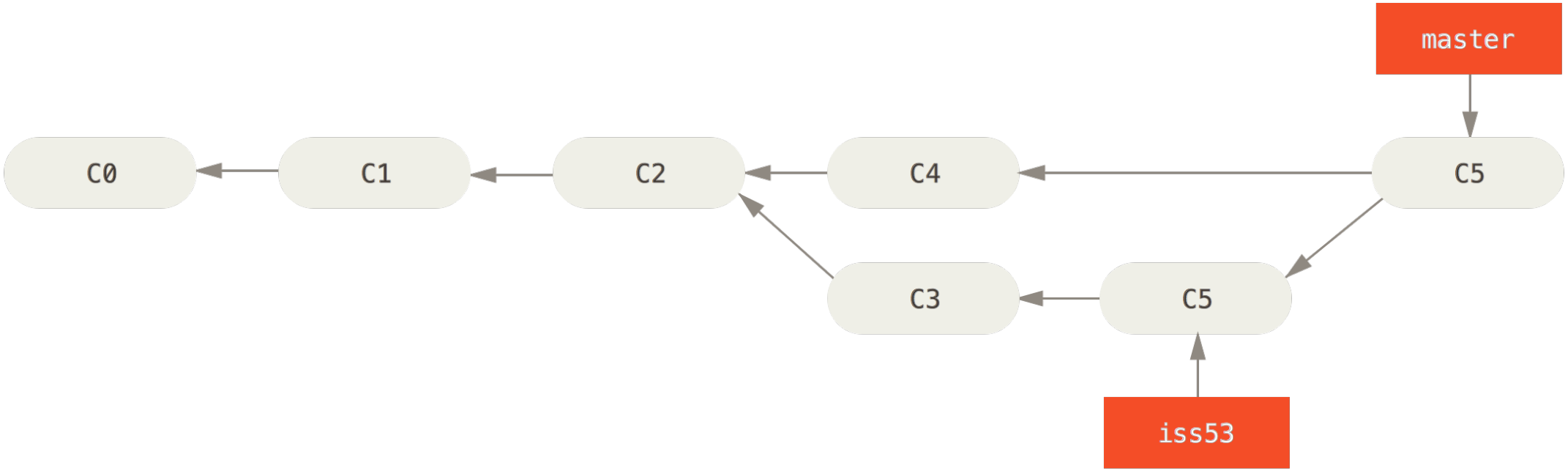
Branch



Branch


master
Common
Ancestor
Snapshot to
Merge Into
C0
C1
C2
C4
Snapshot to
Merge In
C3
C5
iss53

Branch



Working with remote repository

`git remote -v` # shows references to remote repository URLs

`git clone https://github.com/sukwon0709/CSC309.git`

- Clones remote repository and creates a local repository under directory named CSC309

`git remote add mashiyat https://github.com/mashiyat/CSC309.git`

- Creates a reference called mashiyat which points to Mashiyat's CSC309 repository.

Working with remote repository

`git fetch mashiyat`

- Downloads changes from Mashiyat’s repository to my local repository

`git pull mashiyat`

- Downloads changes and merges them to my local repository

`git push origin master`

- Uploads a local commit to my remote repository’s master branch

`git push mashiyat master`

- Uploads a local commit to Mashiyat’s remote repository’s master branch

How to commit your changes

0. make changes to file.txt
- file.txt will be in “modified” state.

1. `git add file.txt`
- Stage a file named “file.txt” before commit.

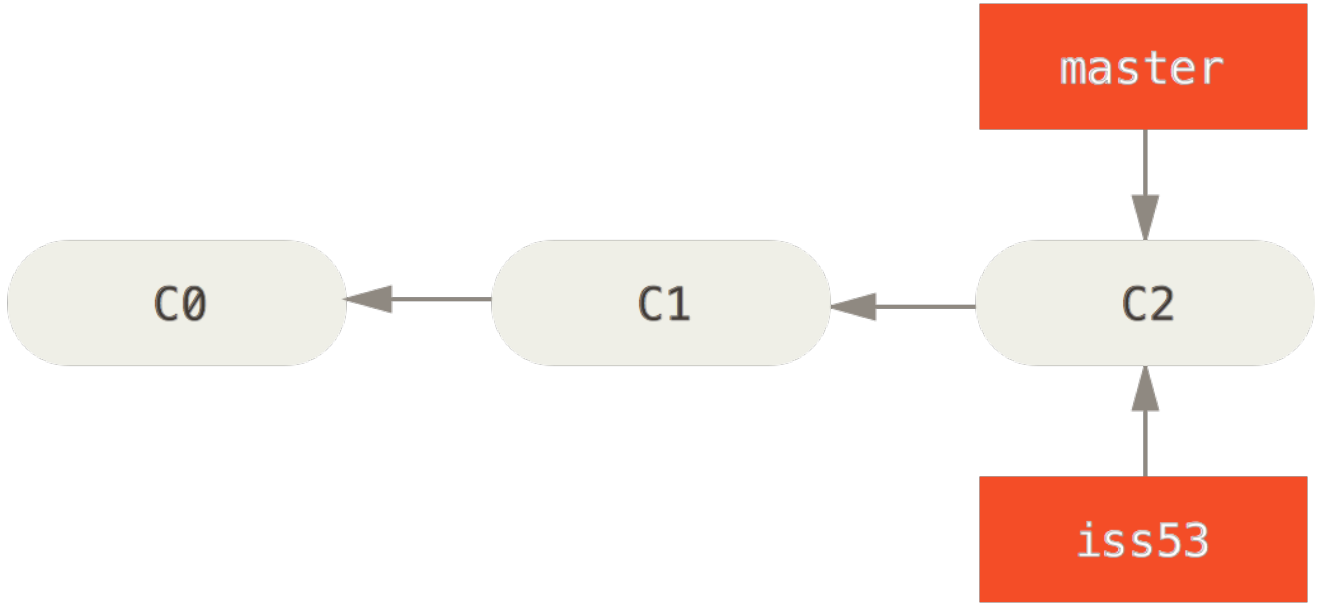
2. `git commit –m “this is a commit!”`
- Commit all staged changes.

3. `git push origin master`
- If you want to share your commit with others, upload your commit to the remote branch like this.

Branching

```
git checkout -b iss53
```

- Creates a local branch named “iss53” and switches to that branch



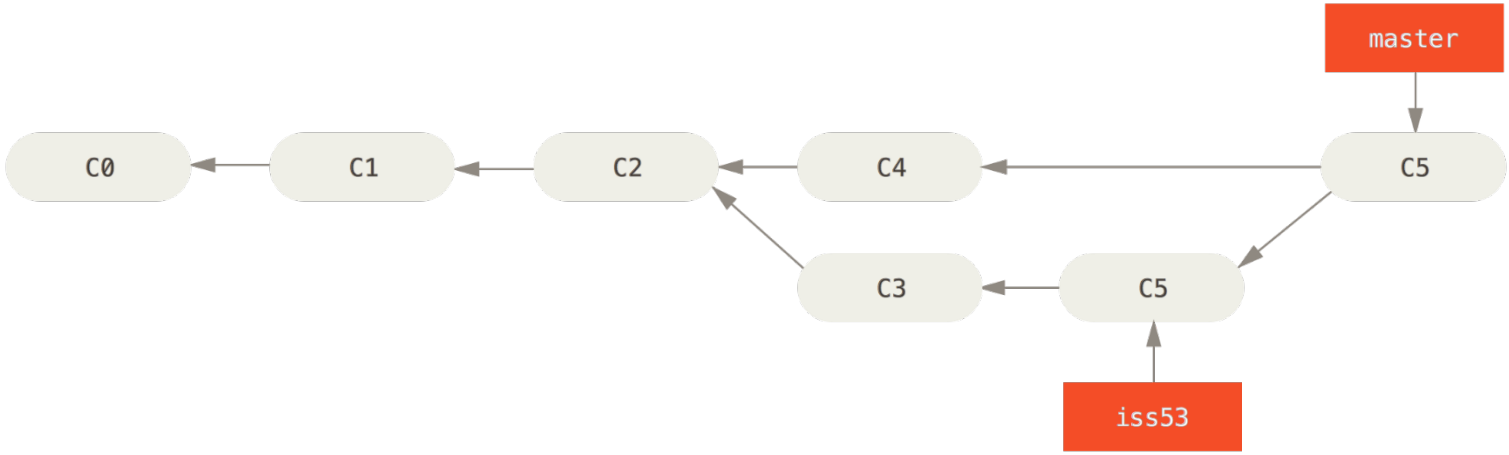
Branching

`git checkout master`

- Switches back to local master branch

`git merge iss53`

- Merges changes made in iss53 branch to current branch



Removing Tracked Files

- You must be careful when you are trying to remove a “tracked” file.
- Using “`rm <filename>`” will remove the file from your filesystem but Git will not consider this file to be removed from the repository.
- The correct way to remove/move a “tracked” file is to prepend “git” in front.

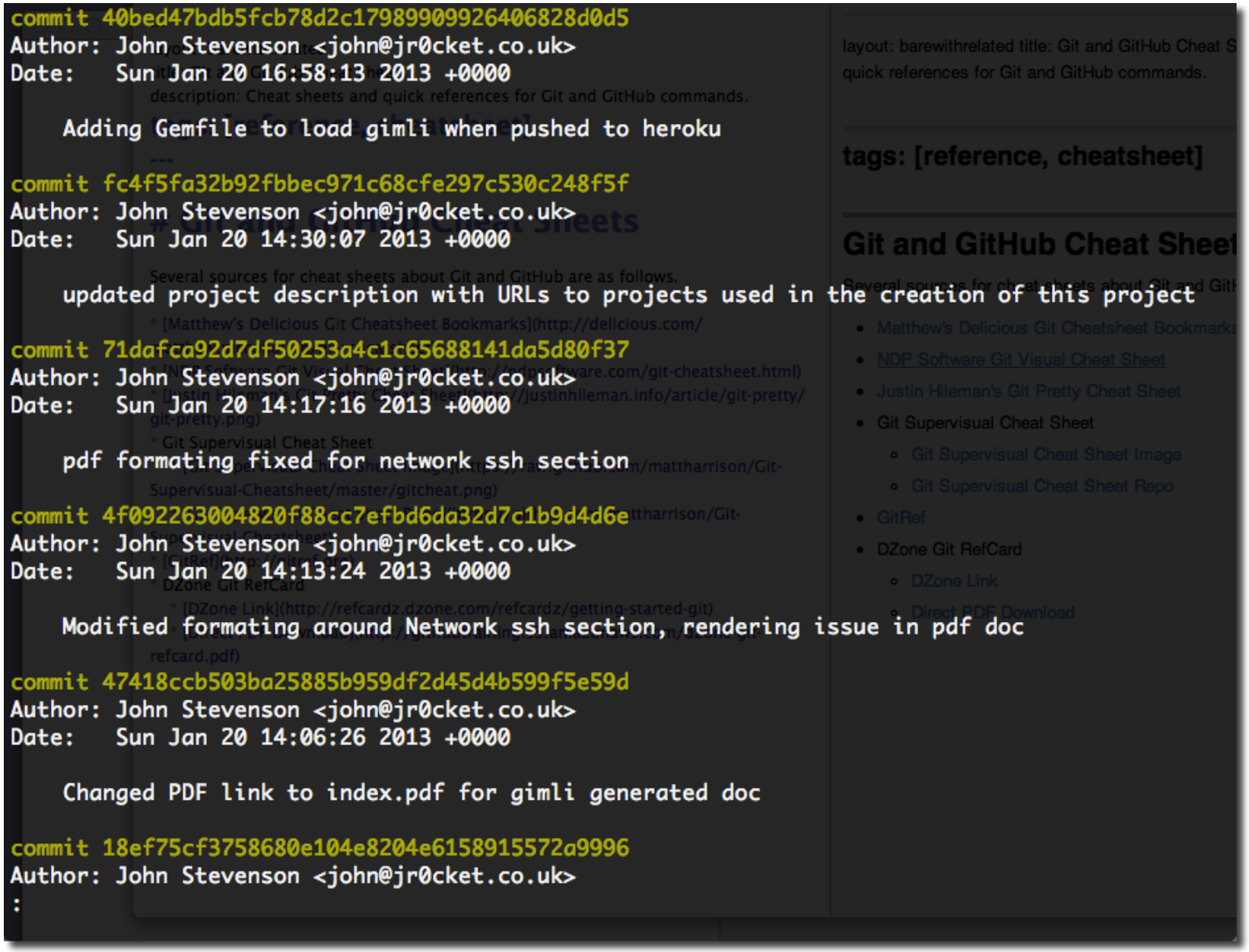
e.g. `git rm <filename>`, `git mv <filename> <new filename>`

- git rm will **remove** the file from both git repository and your filesystem!!
- Use `git rm --cached` if you only want to unstage the file but keep it in your filesystem.

Looking at history

`git log`

- Displays all of previous commits in descending time order.


commit 40bed47bdb5fcb78d2c17989909926406828d0d5
Author: John Stevenson <john@jr0cket.co.uk>
Date: Sun Jan 20 16:58:13 2013 +0000
Adding Gemfile to load gimli when pushed to heroku
commit fc4f5fa32b92fbbec971c68cfe297c530c248f5f
Author: John Stevenson <john@jr0cket.co.uk>
Date: Sun Jan 20 14:30:07 2013 +0000
updated project description with URLs to projects used in the creation of this project
commit 71dafca92d7df50253a4c1c65688141da5d80f37
Author: John Stevenson <john@jr0cket.co.uk>
Date: Sun Jan 20 14:17:16 2013 +0000
pdf formating fixed for network ssh section
commit 4f092263004820f88cc7efbd6da32d7c1b9d4d6e
Author: John Stevenson <john@jr0cket.co.uk>
Date: Sun Jan 20 14:13:24 2013 +0000
Modified formating around Network ssh section, rendering issue in pdf doc
commit 47418ccb503ba25885b959df2d45d4b599f5e59d
Author: John Stevenson <john@jr0cket.co.uk>
Date: Sun Jan 20 14:06:26 2013 +0000
Changed PDF link to index.pdf for gimli generated doc
commit 18ef75cf3758680e104e8204e6158915572a9996
Author: John Stevenson <john@jr0cket.co.uk>
:

Tracking changes with diff

`git diff <commit1 hash> <commit2 hash>`

- Shows changes made between commit1 and commit2

- Use `git diff --color <commit1> <commit2>` if you want nice colored output.

Tagging

- You can create a tag like version numbers to your commits.

```
git tag –a v1.0 –m “my version 1.0”
```

- Creates a tag named v1.0 to the latest commit.