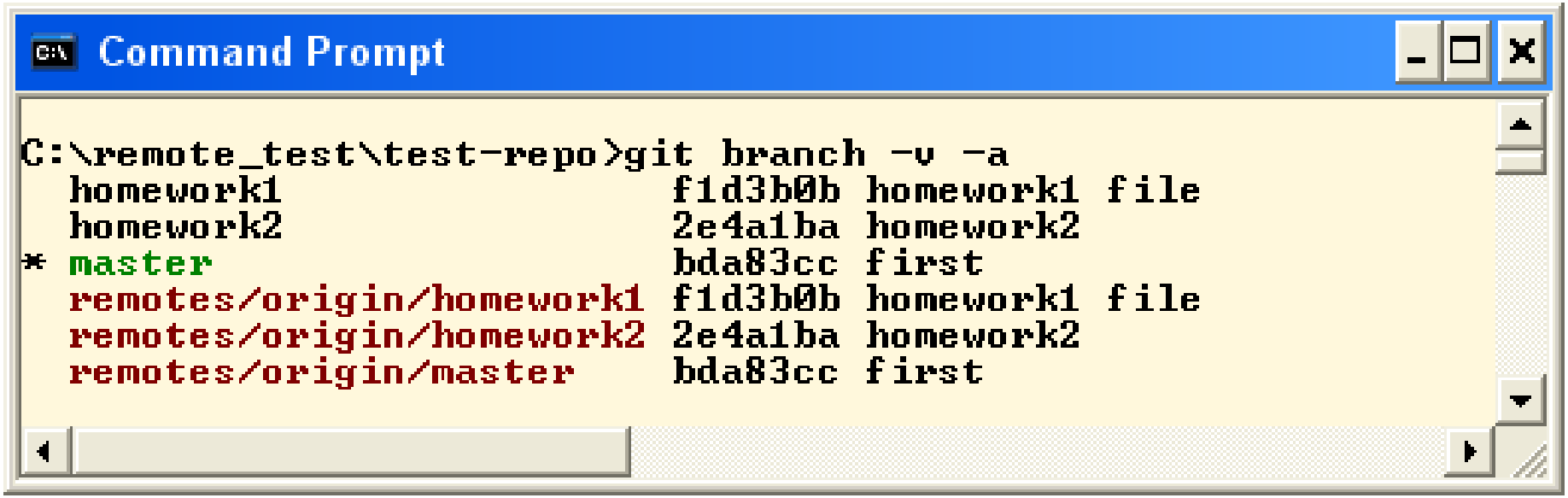
Git Branches and Merges

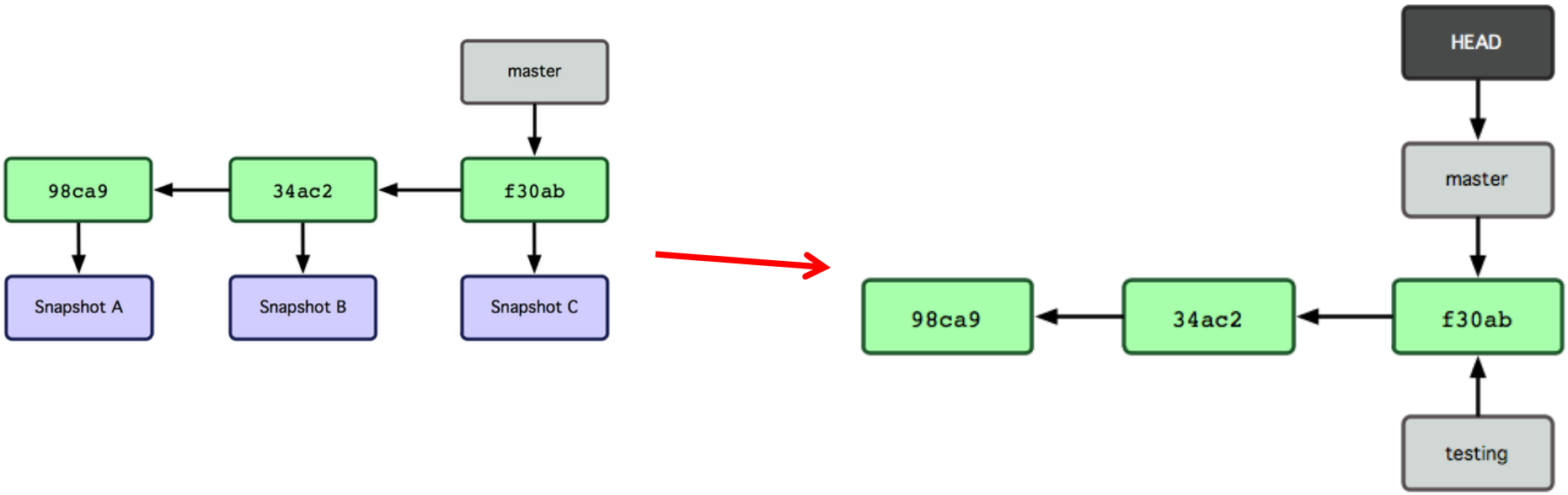
git branch

- `$git branch –v -a`
  - Displays all branches in the repo
  - Asterisk next to the branch you are currently working on



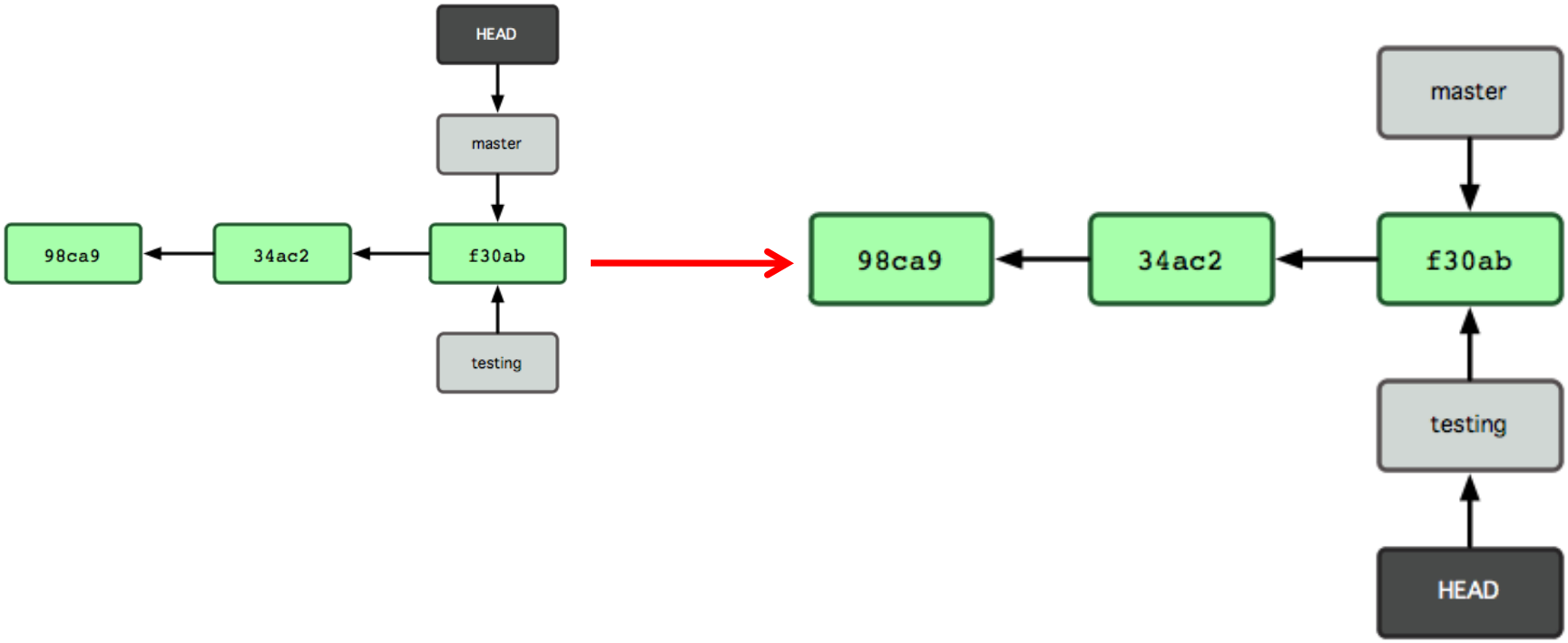
Branching - Example

- `$git branch testing`



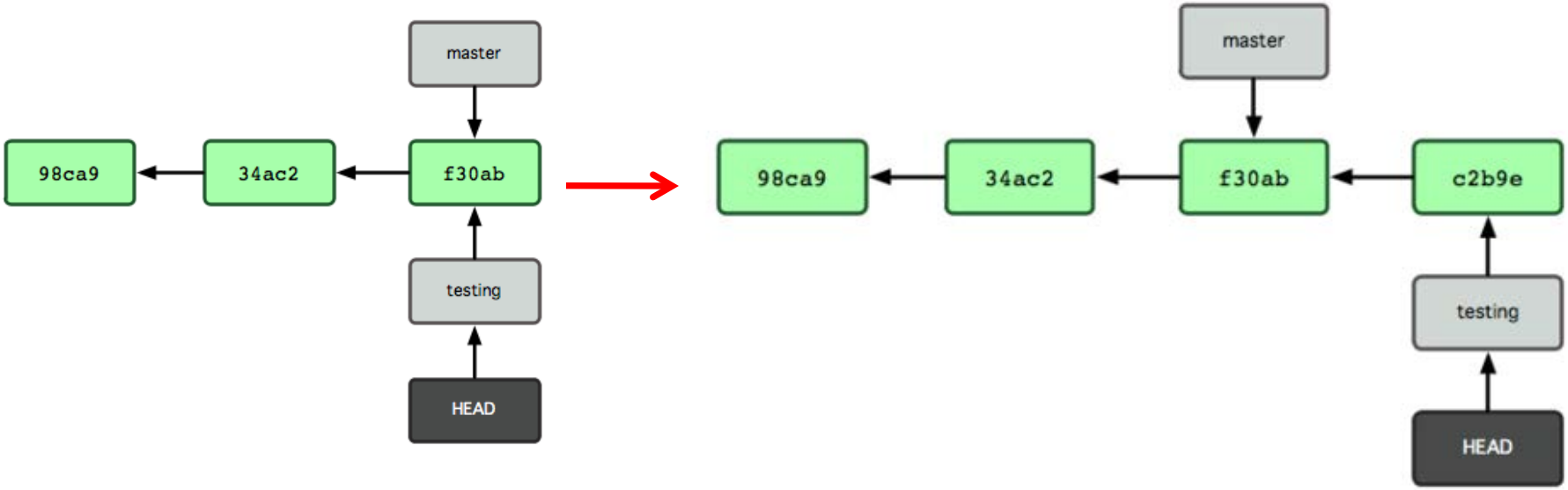
Branching - Example (Continued)

- `$git checkout testing`



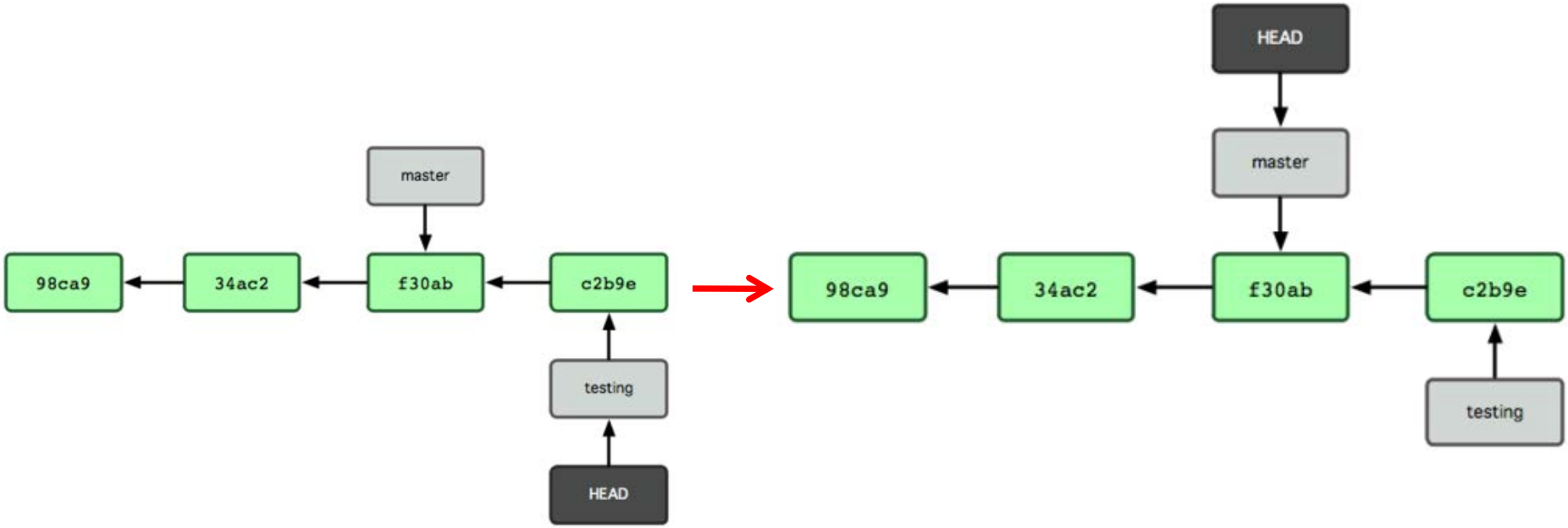
Branching - Example (Continued)

- `$git commit –m “some change”`



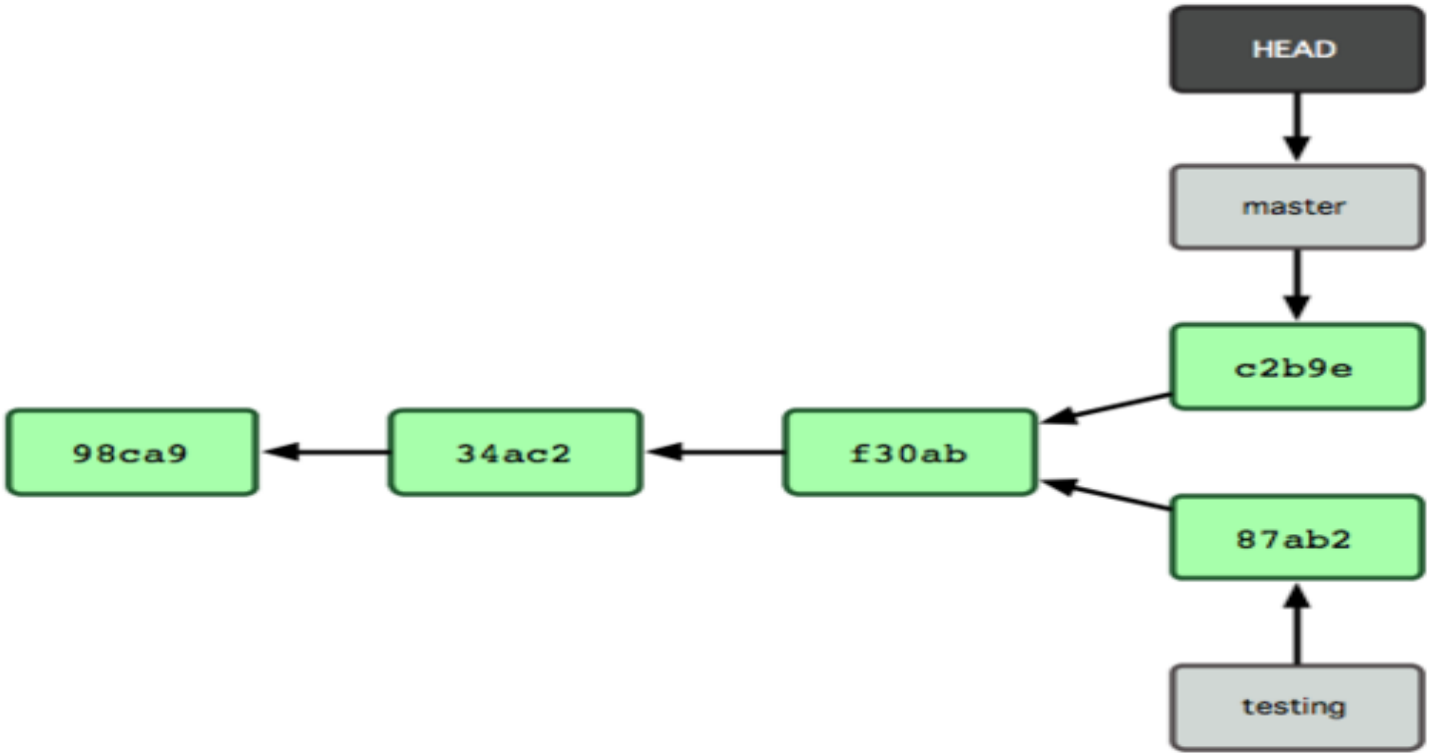
Branching - Example

- `$git checkout master`



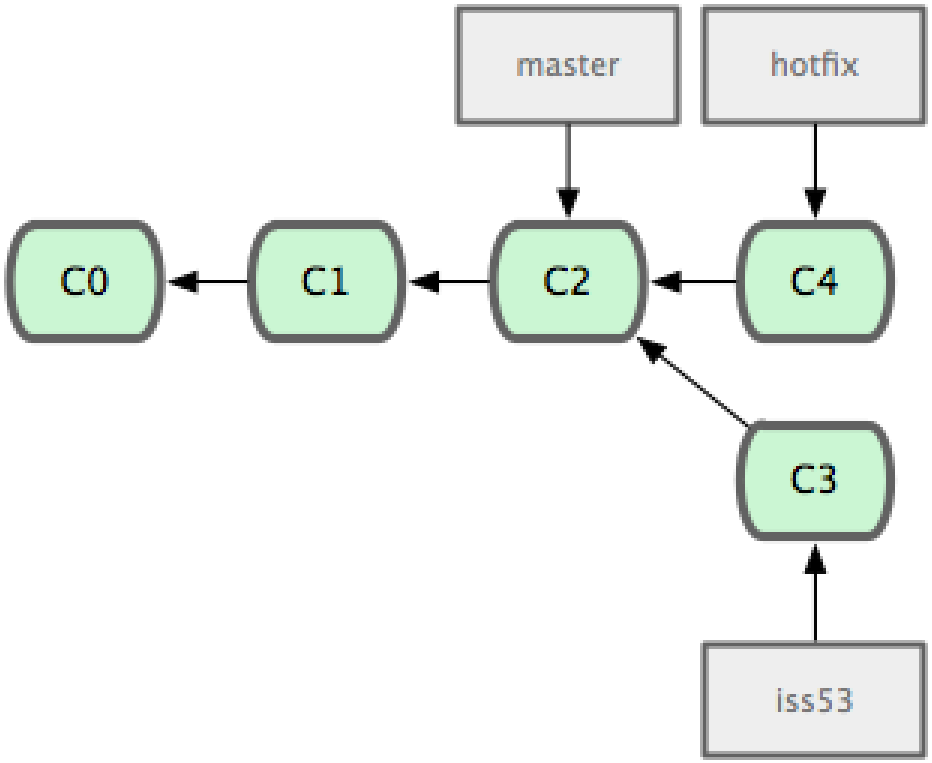
Branching - Example

- `$git commit –m “another change”`



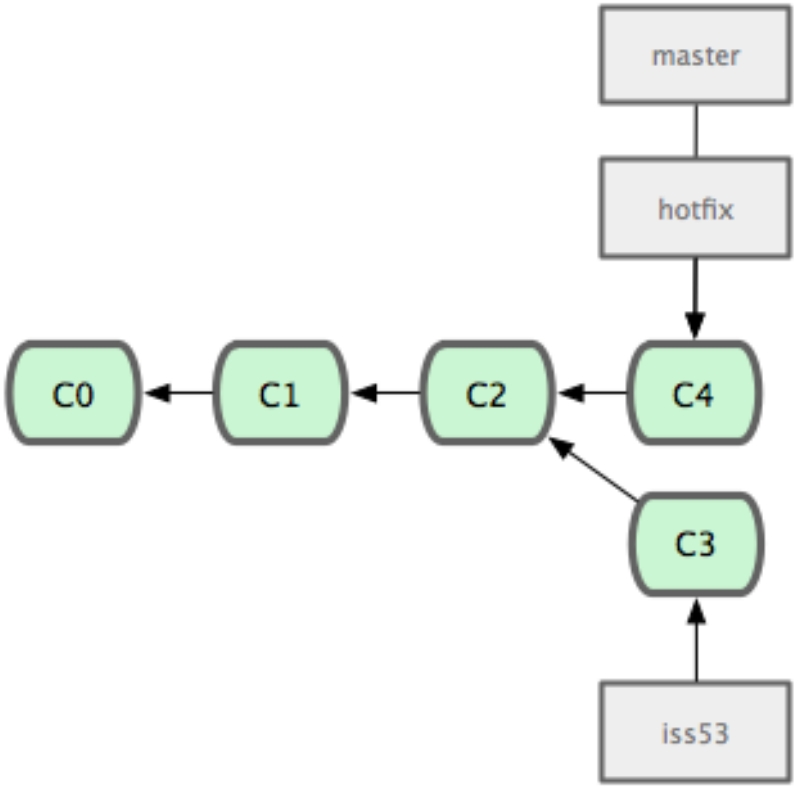
Simple Merge - Fast-forward merge

```
$ git checkout -b 'hotfix'
Switched to a new branch "hotfix"
$ vim index.html
$ git commit -a -m 'fixed the broken email address'
[hotfix]: created 3a0874c: "fixed the broken email address"
 1 files changed, 0 insertions(+), 1 deletions(-)
```



Simple Merge - Fast-forward merge

```
$ git checkout master
$ git merge hotfix
Updating f42c576..3a0874c
Fast forward
 README |    1 -
 1 files changed, 0 insertions(+), 1 deletions(-)
```



Branches

Summary:

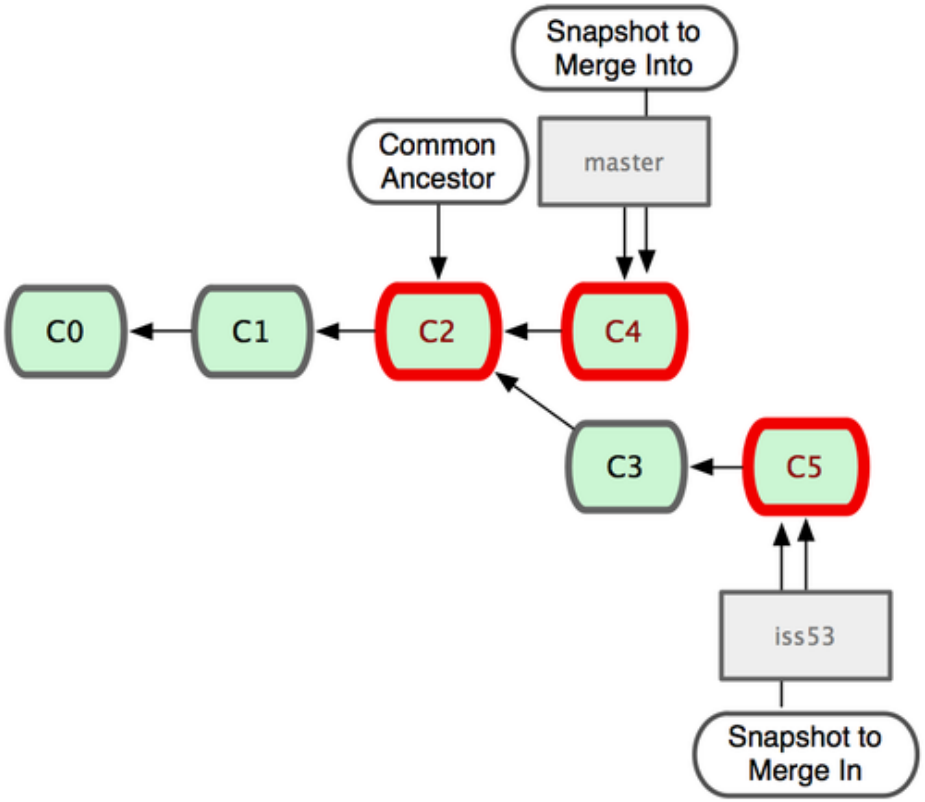
- `$git branch <branch name>`
  - Create a new branch
- `$git checkout <branch name>`
  - Switch to a branch
- `$git checkout –b <branch name>`
  - Create a new branch and switch to it (shortcut instead of the previous 2 commands)

Removing a branch

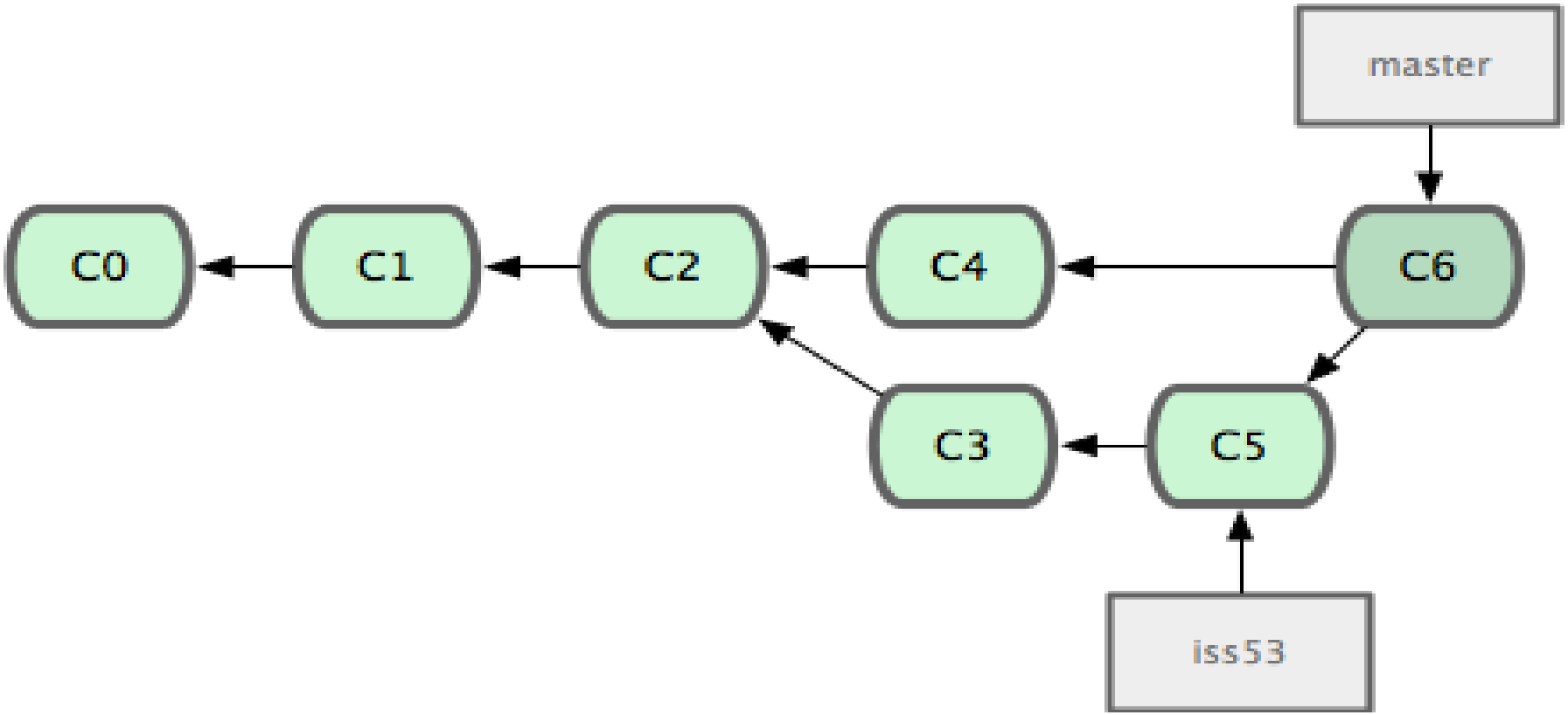
- At this point, we can delete hotfix branch
  - `$git branch –d hotfix`
- `-d` will stop you if the branch hasn't been merged into master yet
  - If you still want to delete after that – use `–D`
  - `$git branch –D somebranch`

Complicated (non-fast-forward) merge

```
$ git checkout master
$ git merge iss53
Merge made by recursive.
 README |    1 +
 1 files changed, 1 insertions(+), 0 deletions(-)
```



Complicated merge - End Result



Merge conflicts

- Conflicts happen

```
$ git merge iss53
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the result.
```

```
[master*]$ git status
index.html: needs merge
# On branch master
# Changed but not updated:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       unmerged:   index.html
#
```

Merge conflicts - resolution

- Resolve manually or use a merge tool
- You'll need to play with mergetools to see what you like. Consider the following:
  - RubyMine (http://vimeo.com/6907909)
  - TortoiseGit (http://code.google.com/p/tortoisegit/)
  - http://code.google.com/p/gitextensions/

Merge conflicts - manual resolution

```
<<<<<<< HEAD:index.html
<div id="footer">contact : email.support@github.com</div>
=======
<div id="footer">
  please contact us at support@github.com
</div>
>>>>>>> iss53:index.html
```

```
<div id="footer">
please contact us at email.support@github.com
</div>
```

Merge conflicts - manual resolution

- After resolving conflicts, run `git add` on the file(s)
- Staging (adding) the file marks it as resolved
- Commit the changes