

i, you, and **Git**

Whoami

Muslih / @muslih / <http://muslih.me>

Front end at Alfanous

Who knows?

- Github
- Bitbucket
- Gitlab

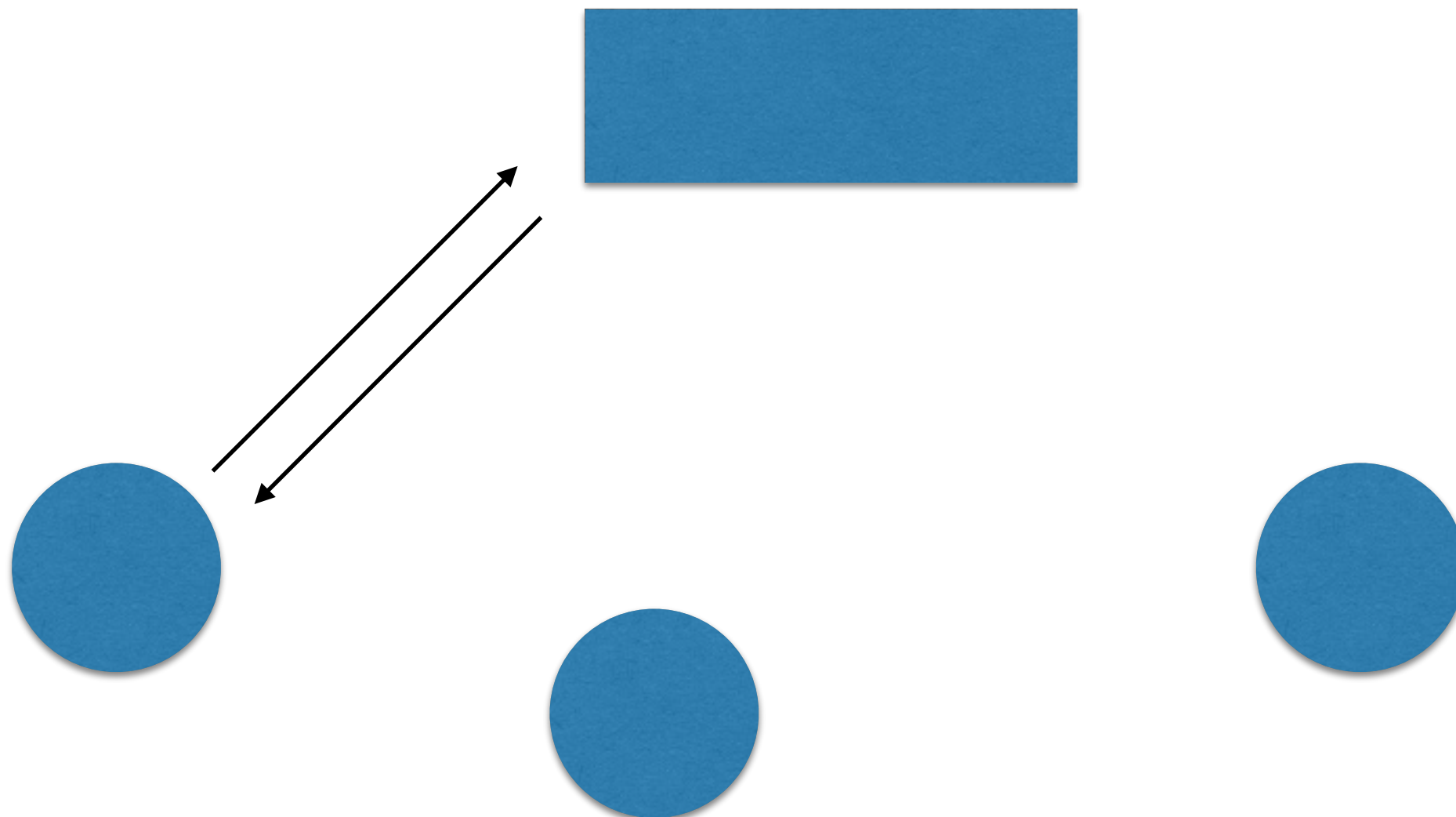
1 Developer

100/100 Developers

It will be complicated!

Working with team

- Productivity
- Social Challenge
- Ship Software faster and smarter



GIT

Make it not complicated anymore!

Git Just A tool!

Open source distributed
version control system
designed for speed & efficiency

version control?

Version control?

Record data change system

Team development possibility

Know who changed and when changed

Possibility to restore back data

Distributed version
control?

Data Stored locally?

- Backup
- Work fastly
- Work Offline (Without internet connection)

What can we do without internet?

- Performing diff
- View file history
- Commit change
- Merging branches
- Obtaining other revisions of a file
- Switching branches

designed for speed &
efficiency?

data mostly not
deleted, **snapshots**



Storing data as snapshots of the project over time

be happier developer
with GIT

working directory

a working copy
of your project

index

repository

working directory

index

object database

repository

working directory

index

“staging”

repository

Index

- git config
- git init
- git add
- git status
- git commit
- git log



Install

- MAC => `brew install git`
- Windows => git for Windows

Settings

```
$ git config --global user.name "Your Name"  
$ git config --global user.email "your@email.com"
```


Start

```
$ mkdir ~/git-learn && cd ~/git-learn
```

git init

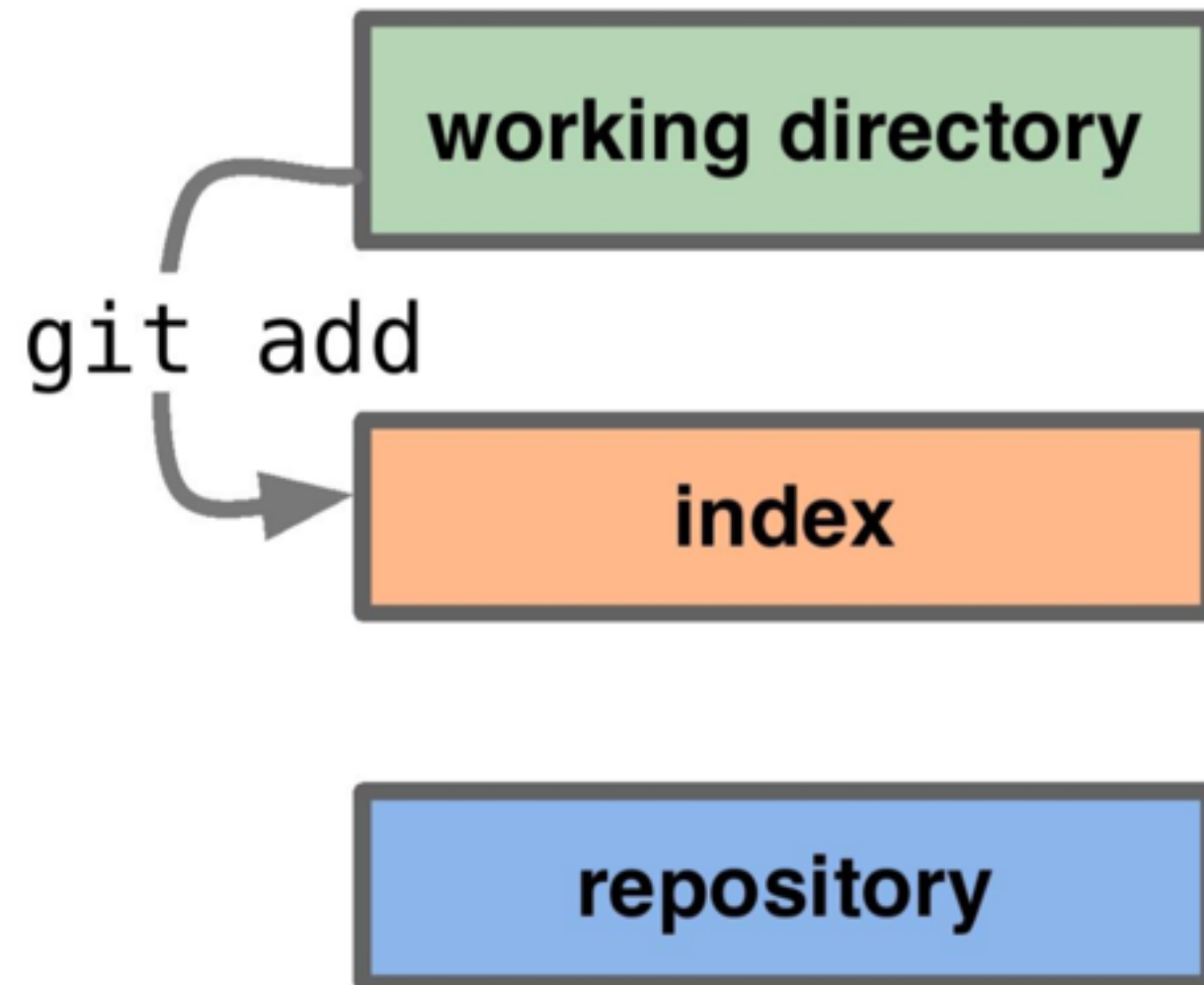
```
$ git init
```

```
$ ls .git
```

git add

```
$ vi README.md
```

```
$ git add README.md
```

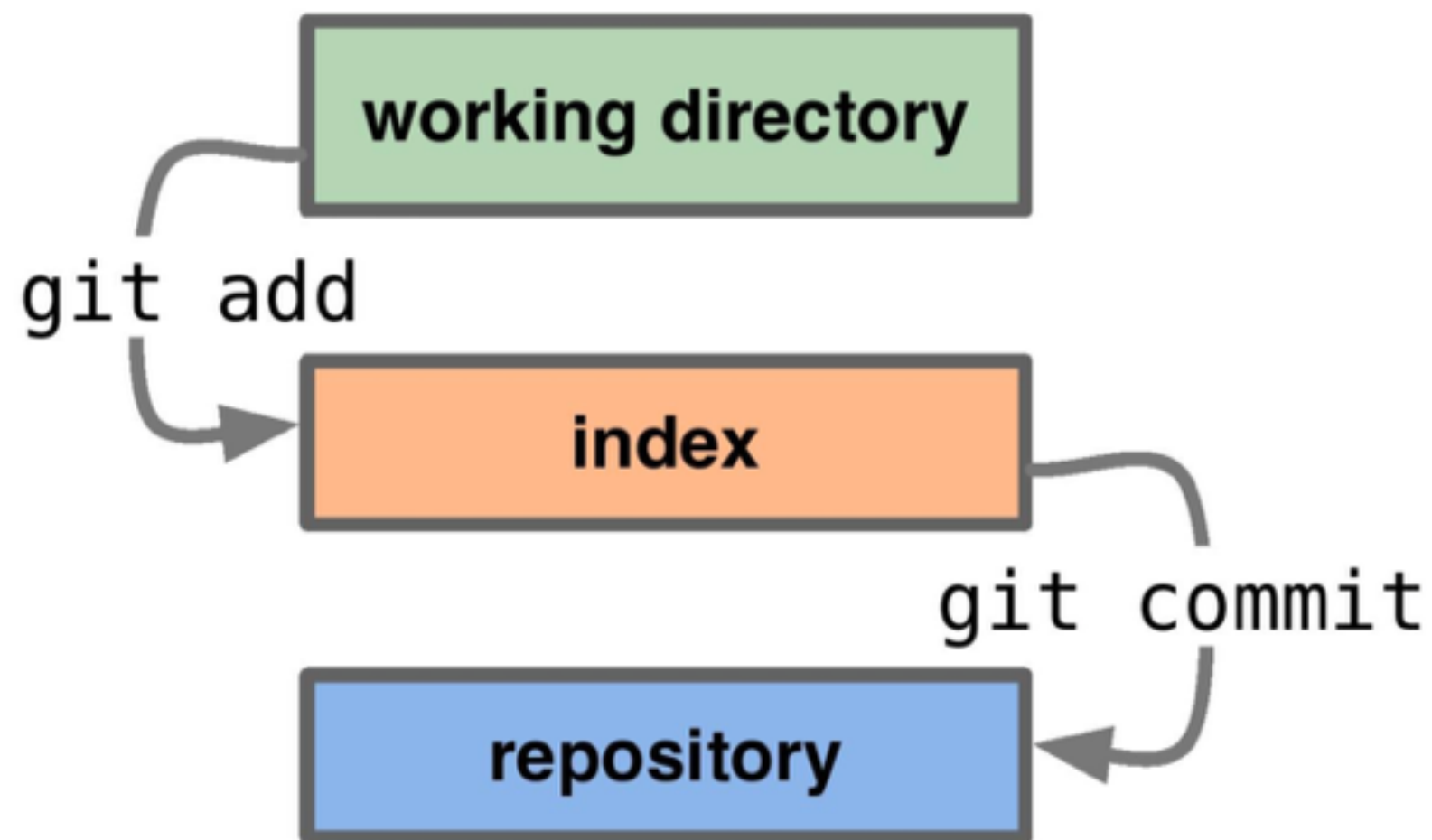


git status

```
$ git status
```

git commit

```
$ git commit -m "first commit"
```



git log

```
$ git log
```


git push

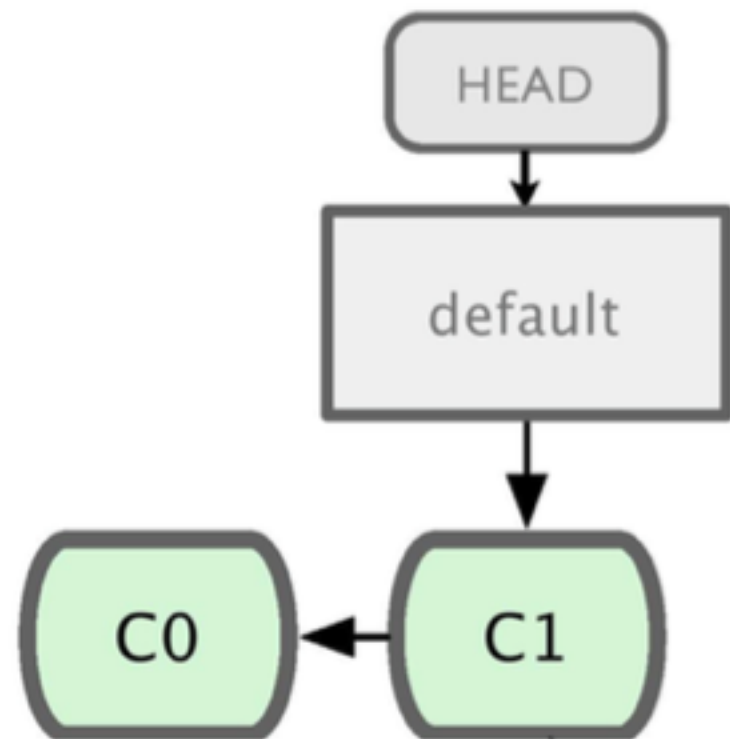
```
$ git remote add origin http://xxxxx/reponame.git $ git  
push -u origin master
```

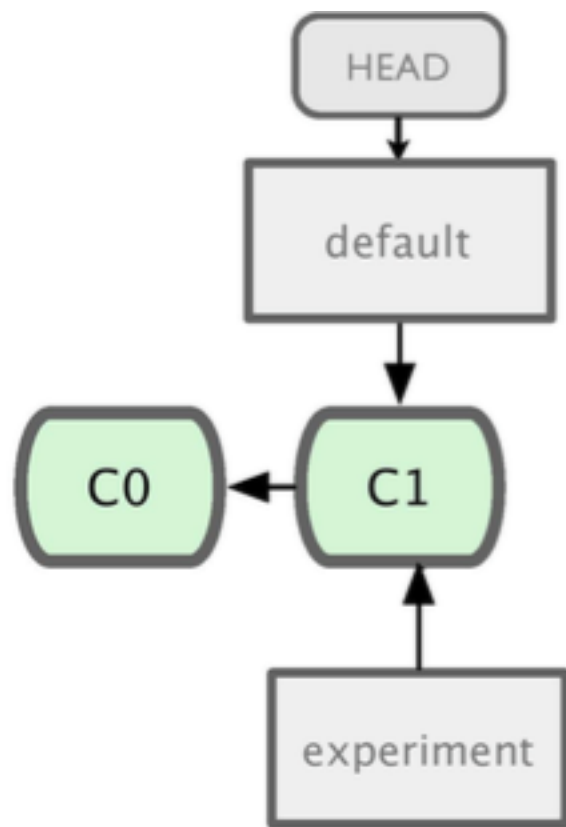
i want to ignore some files!

```
$ .gitignore
```

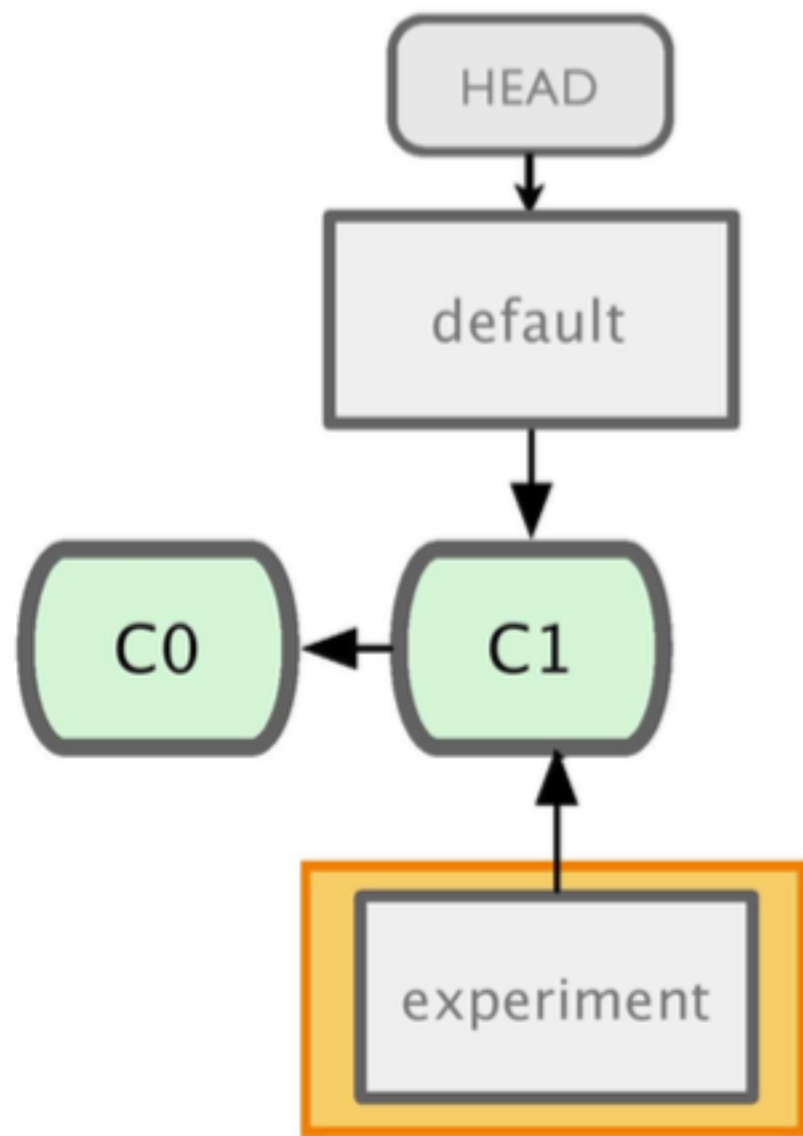
Branching

`git branch && git checkout`

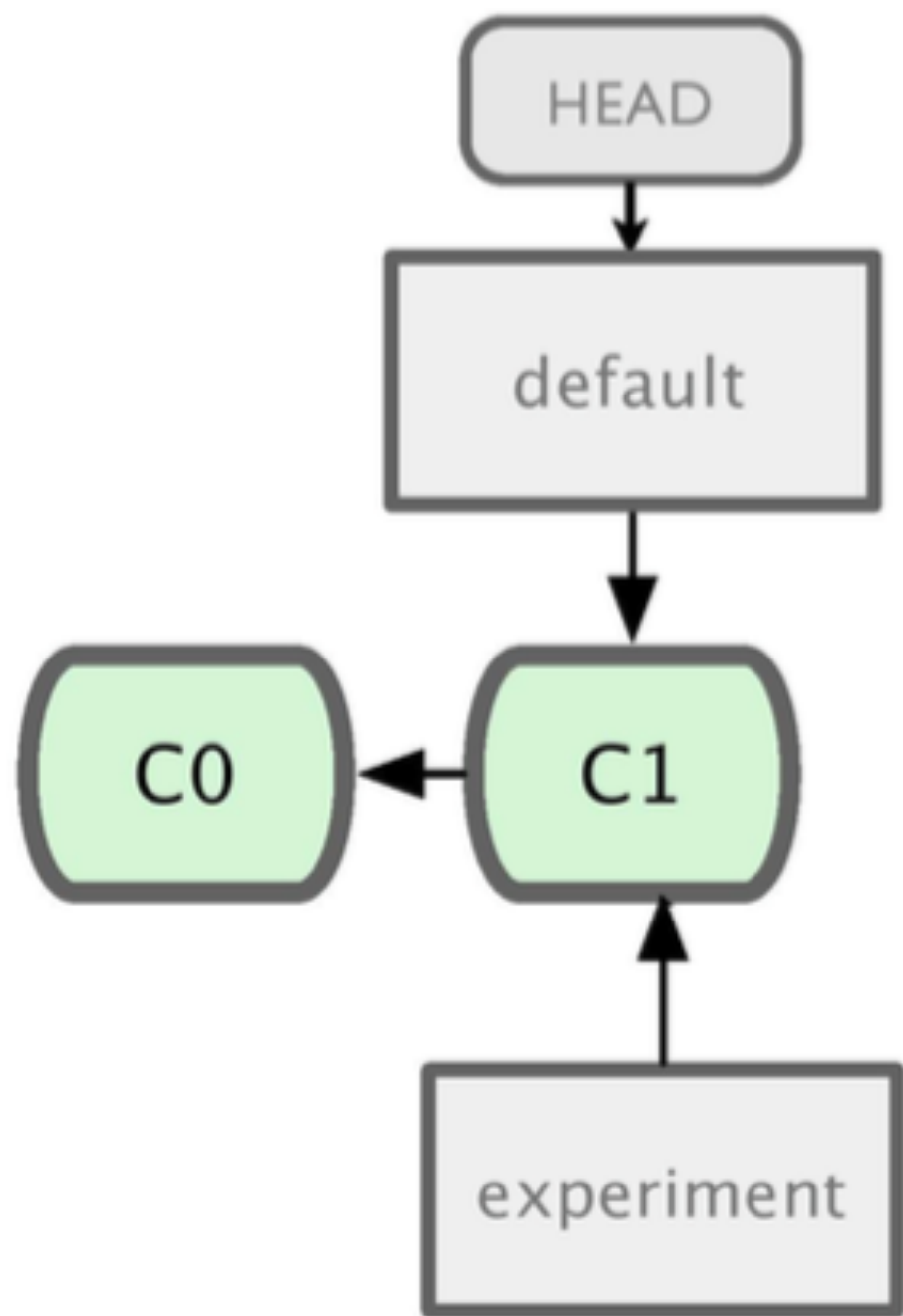




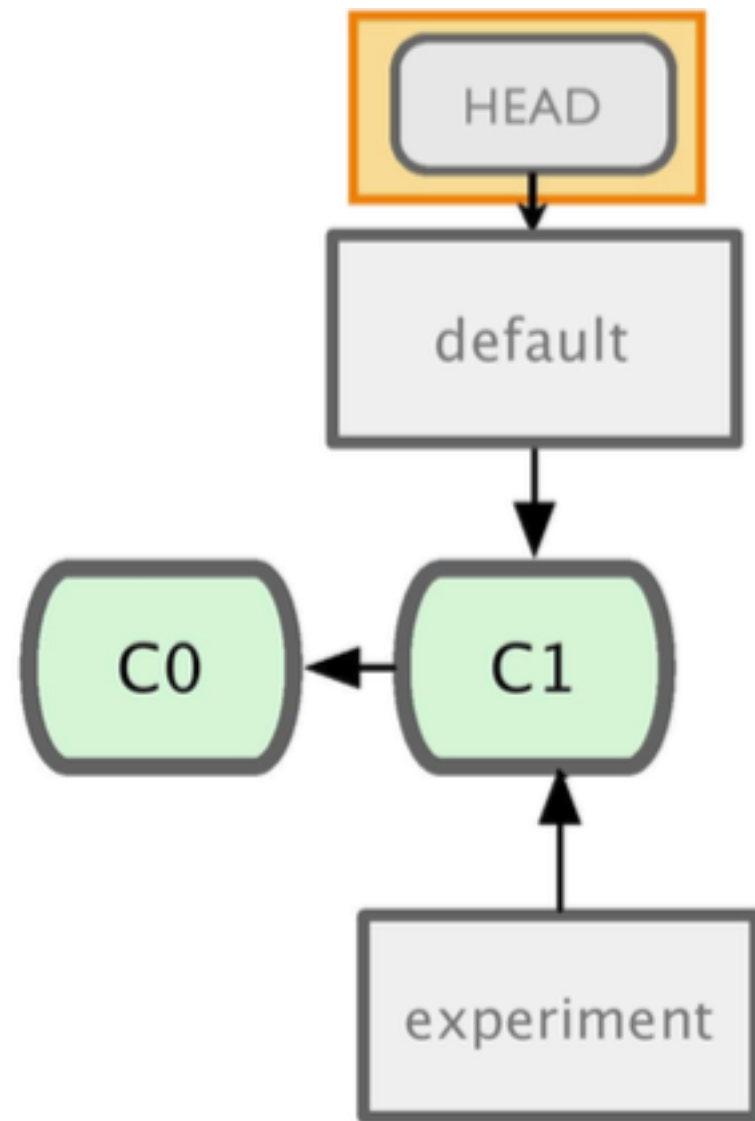
git branch experiment



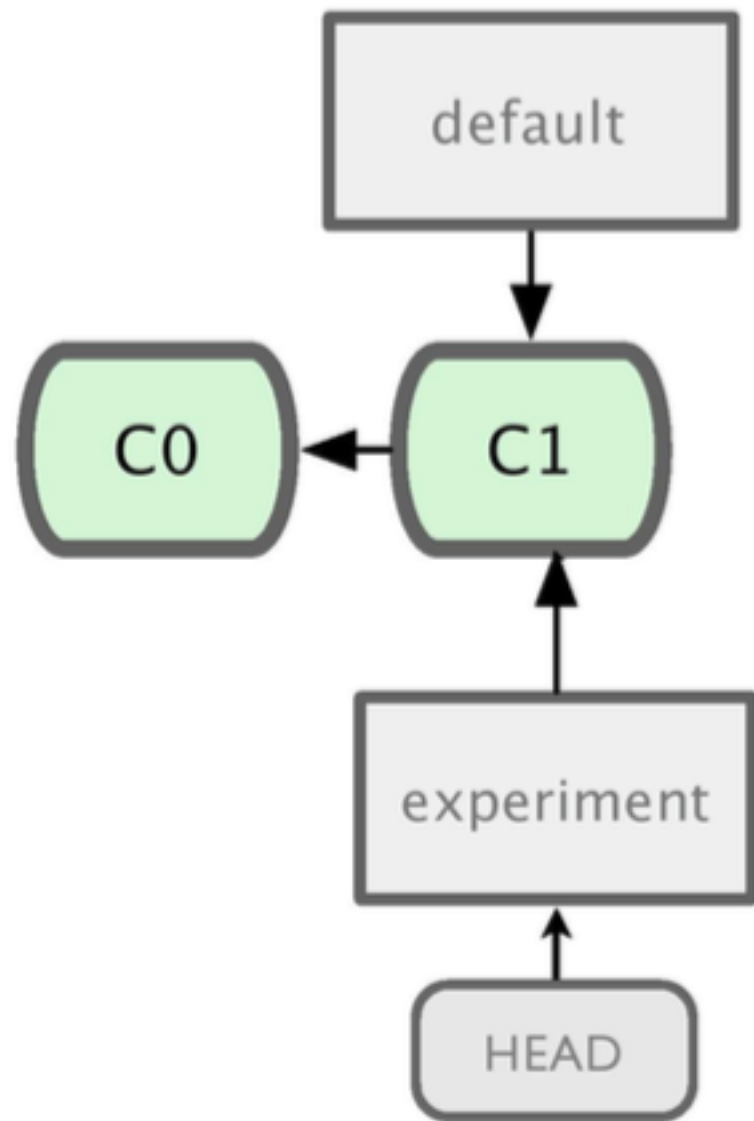
git branch experiment



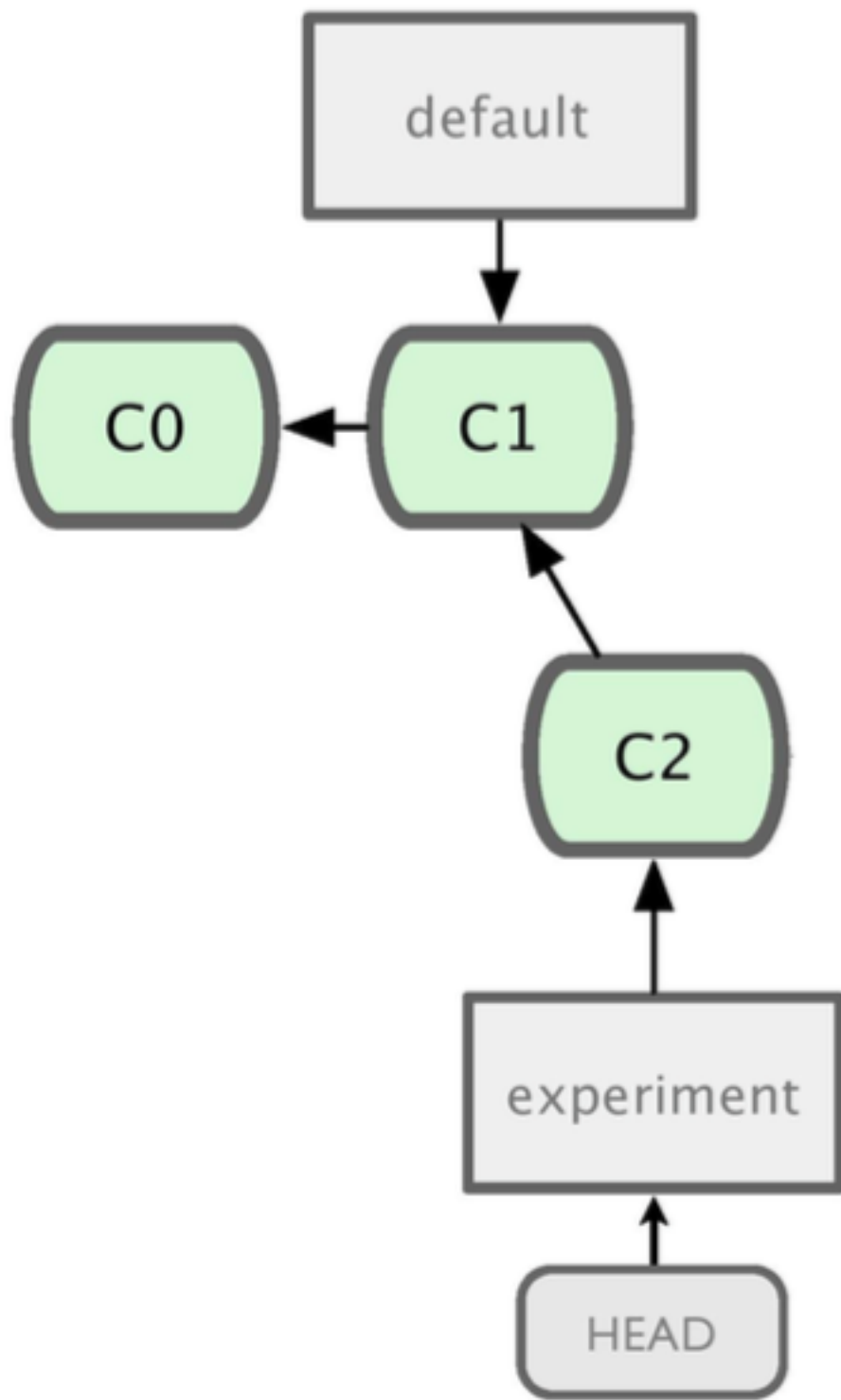
```
$ git branch  
* default  
experiment
```

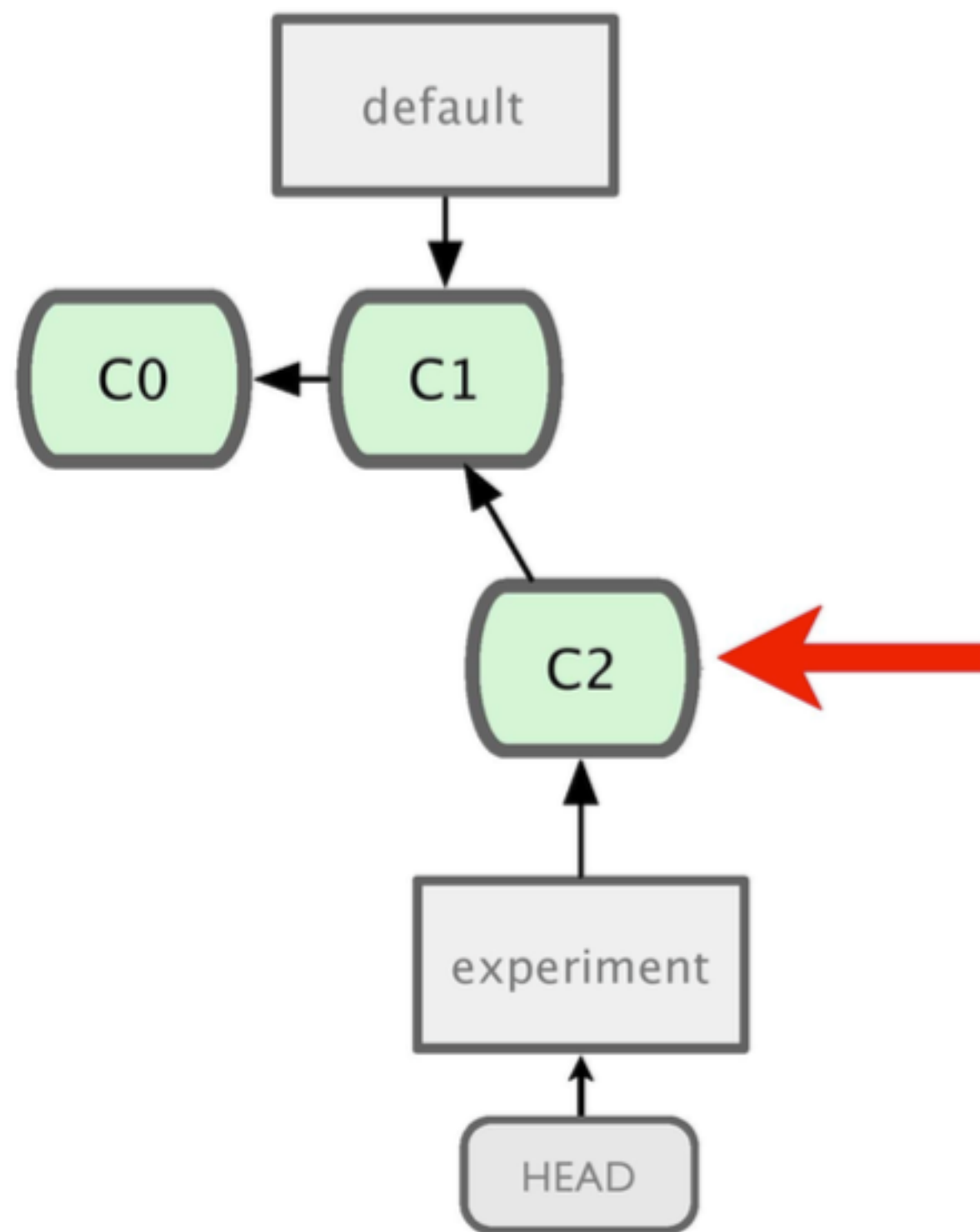
```
$ git branch
* default
experiment
```



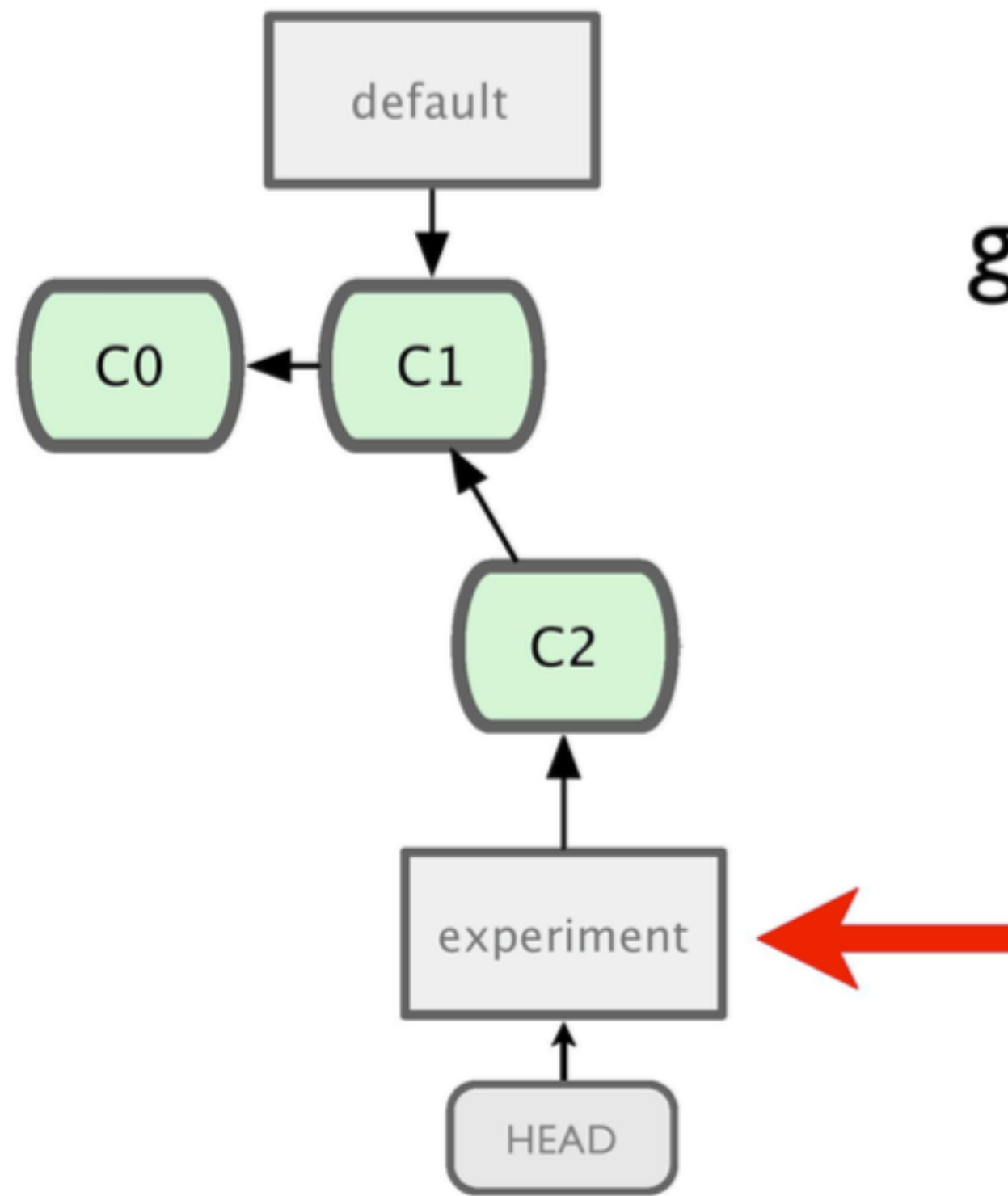
git checkout experiment



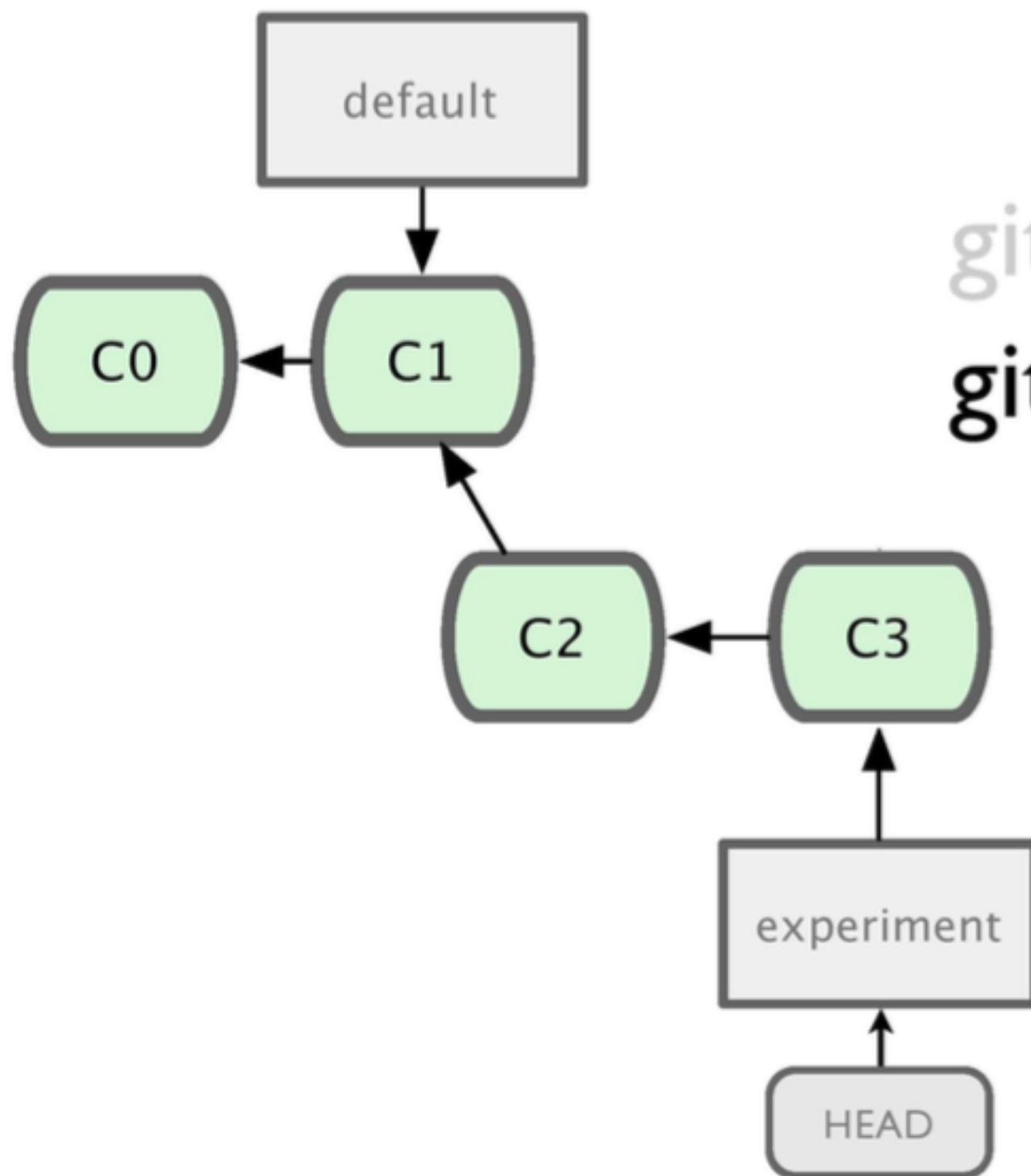
git commit



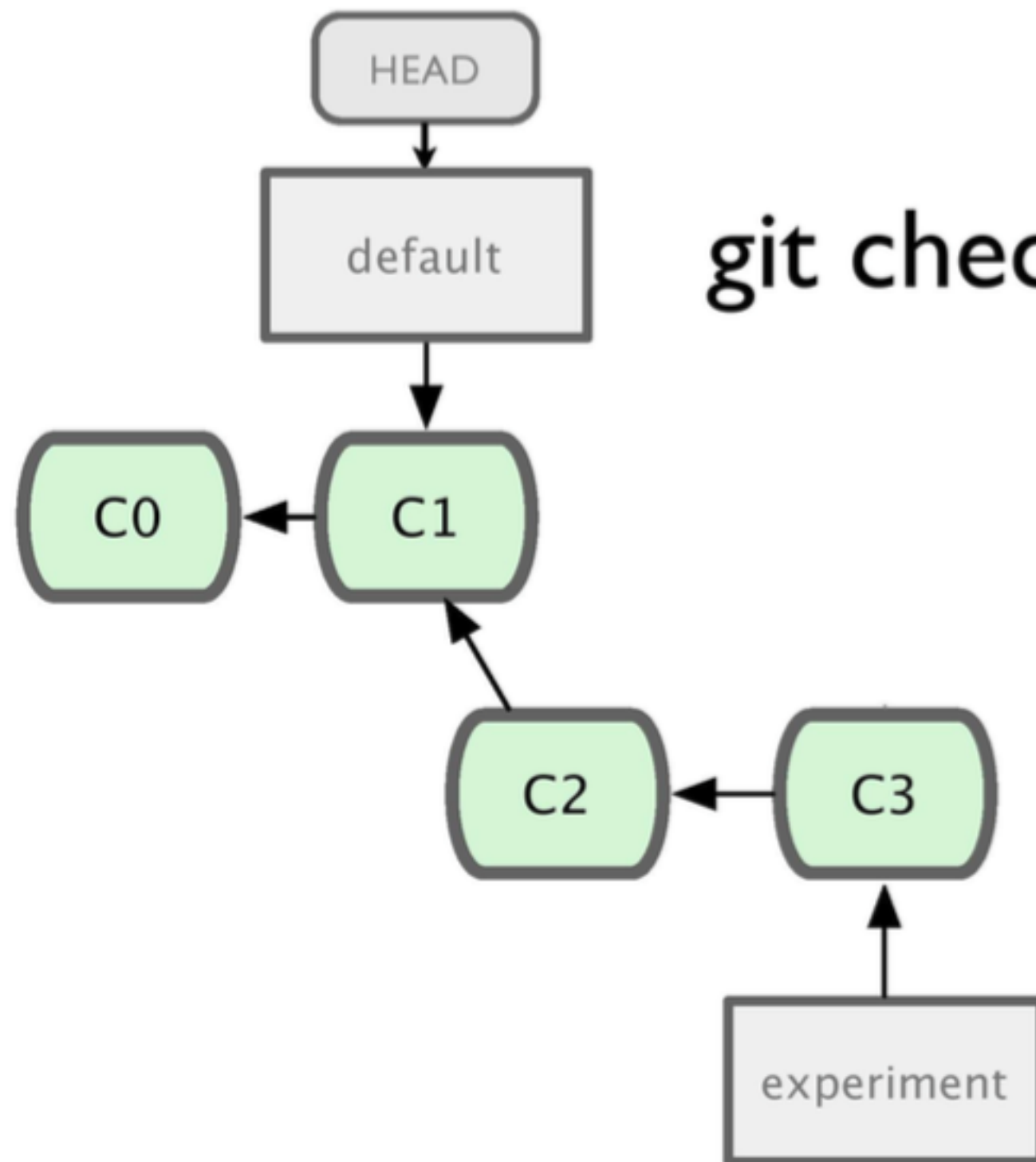
git commit



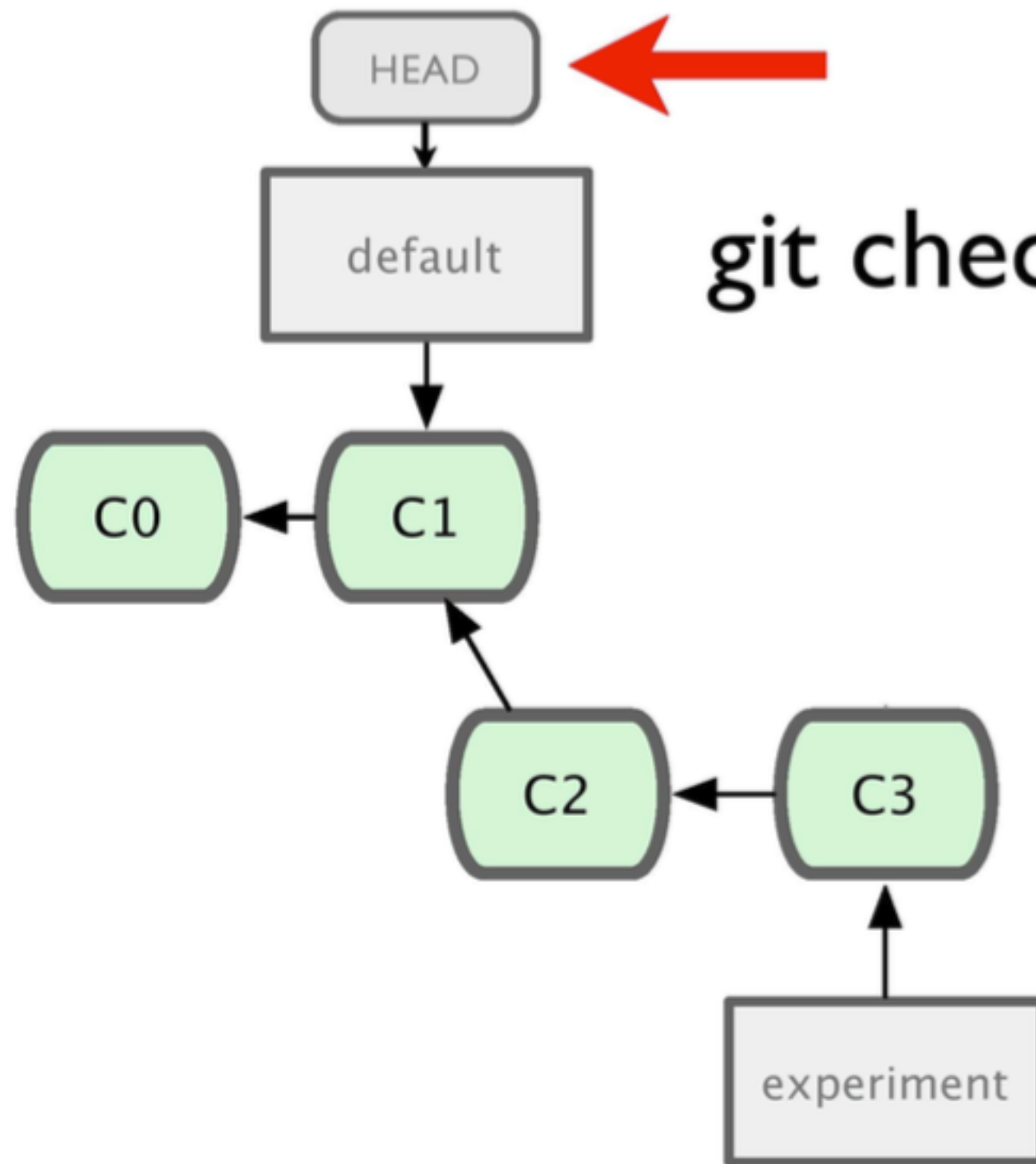
git commit



git commit
git commit

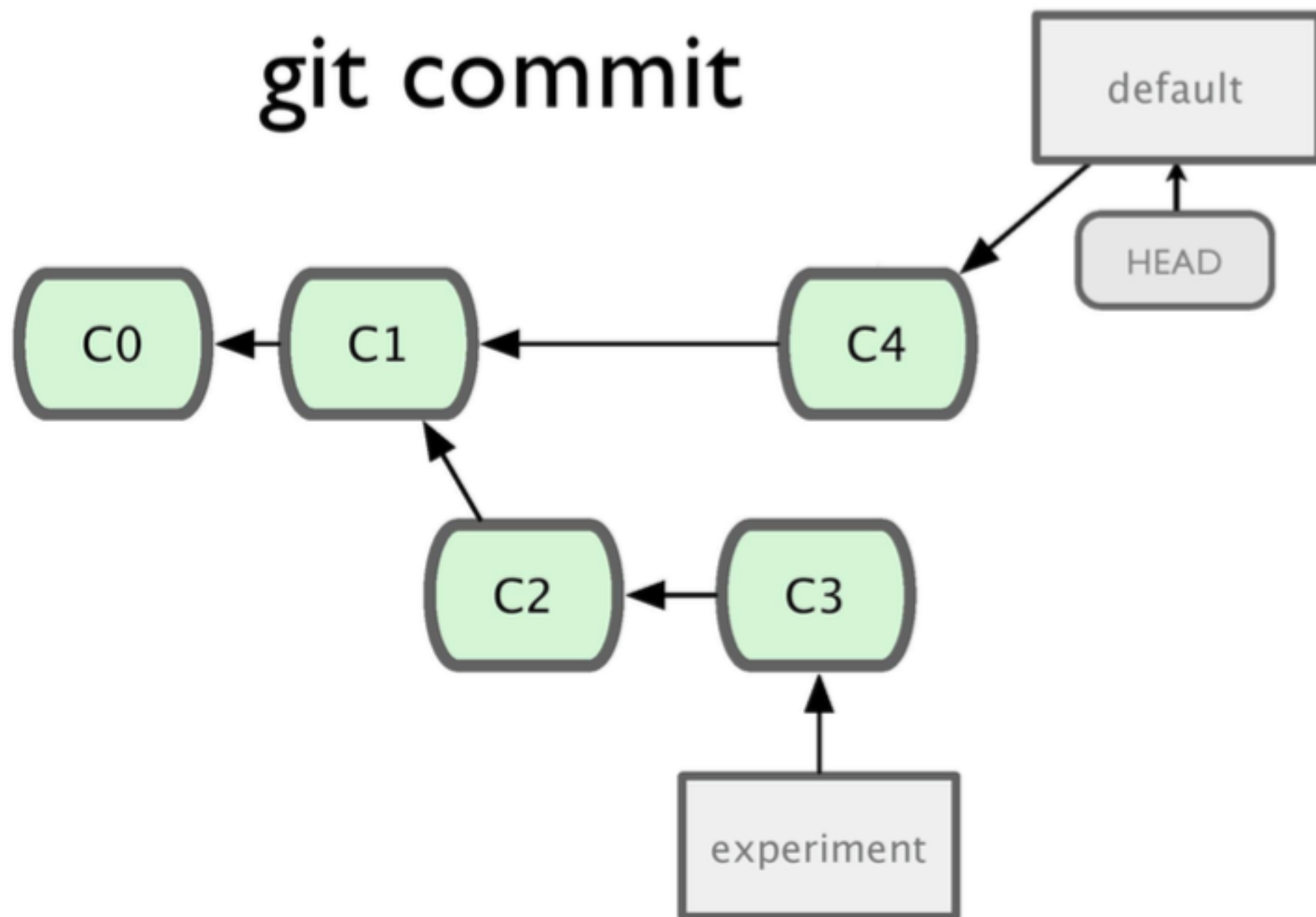


git checkout default

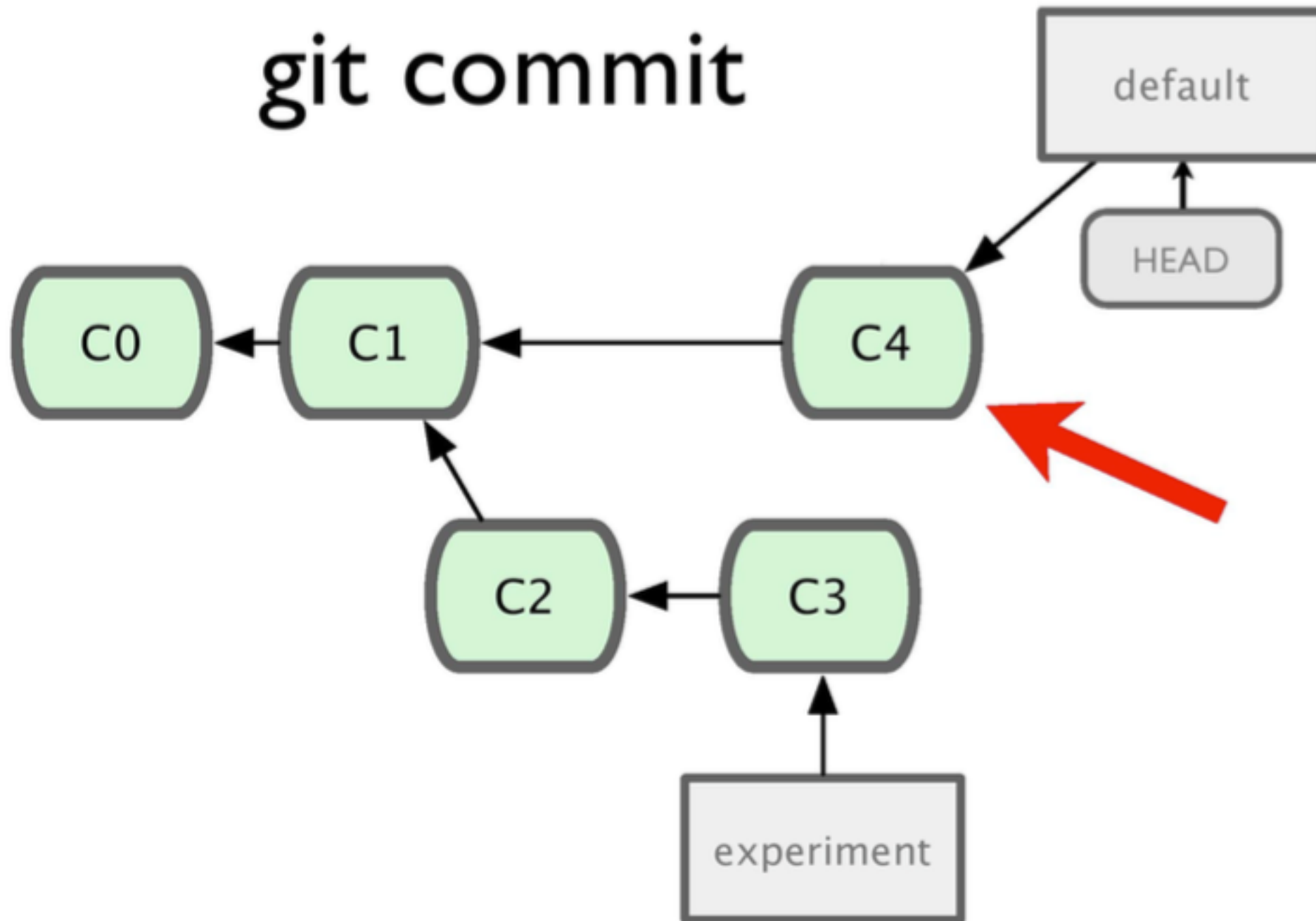


git checkout default

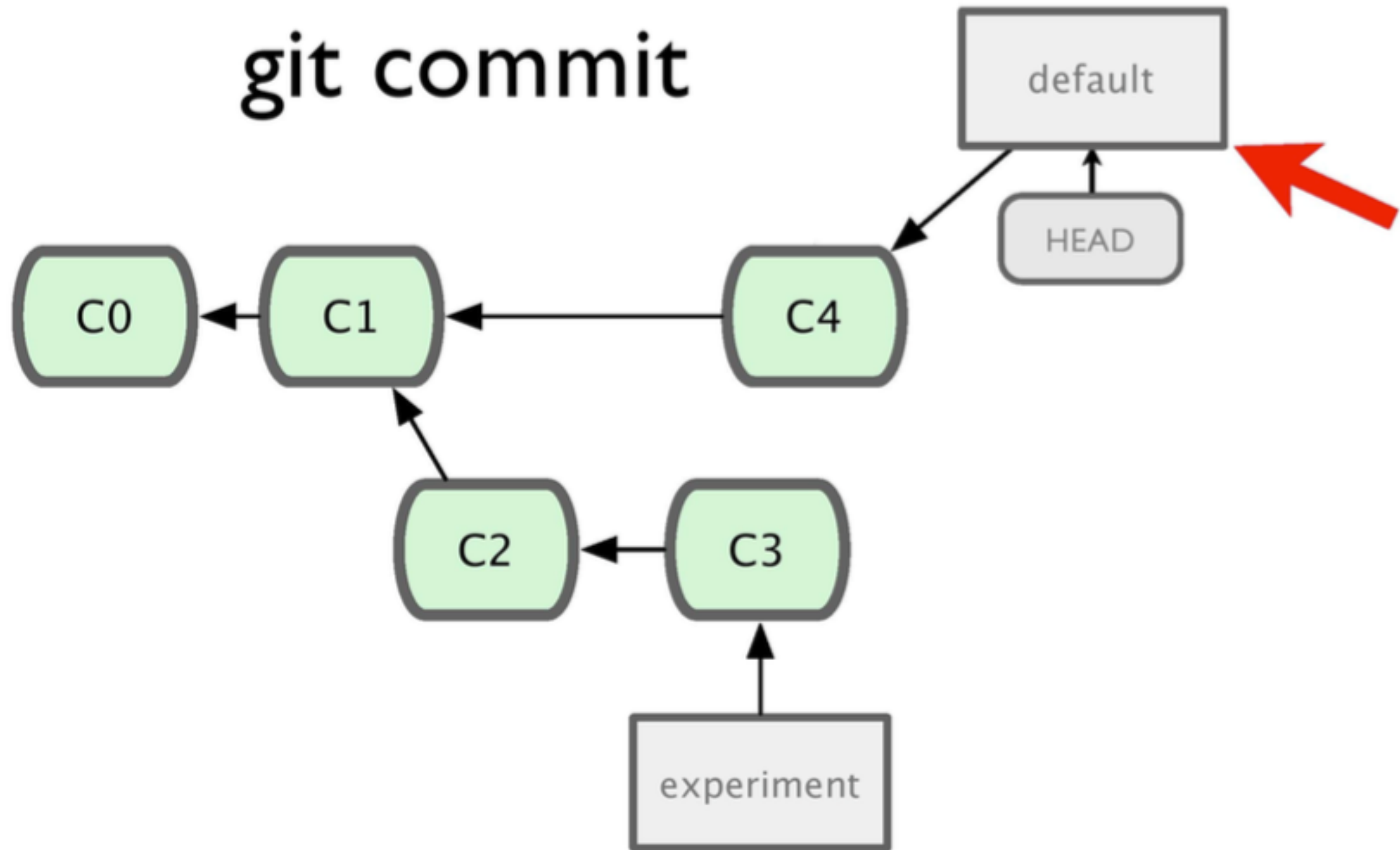
git commit

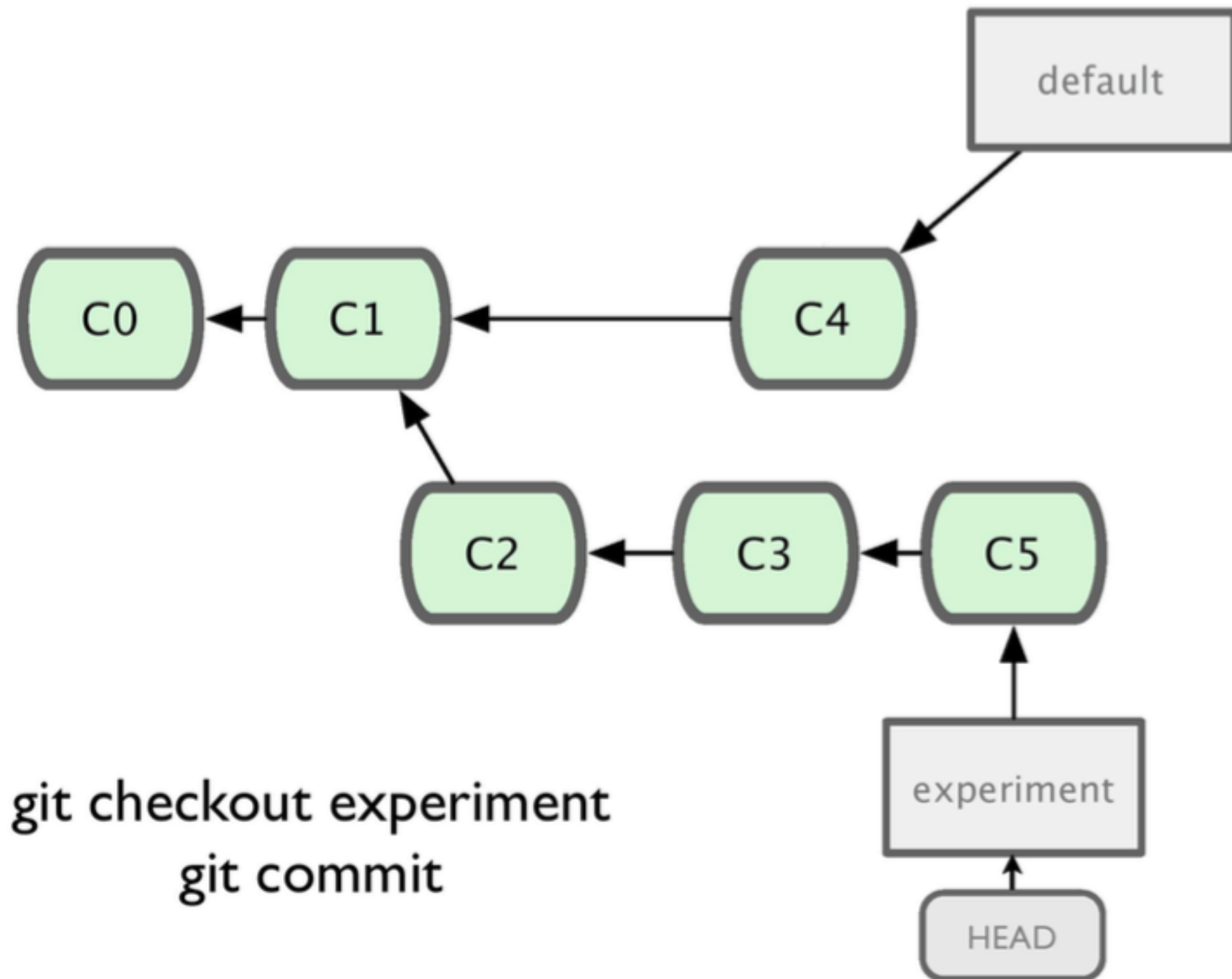


git commit

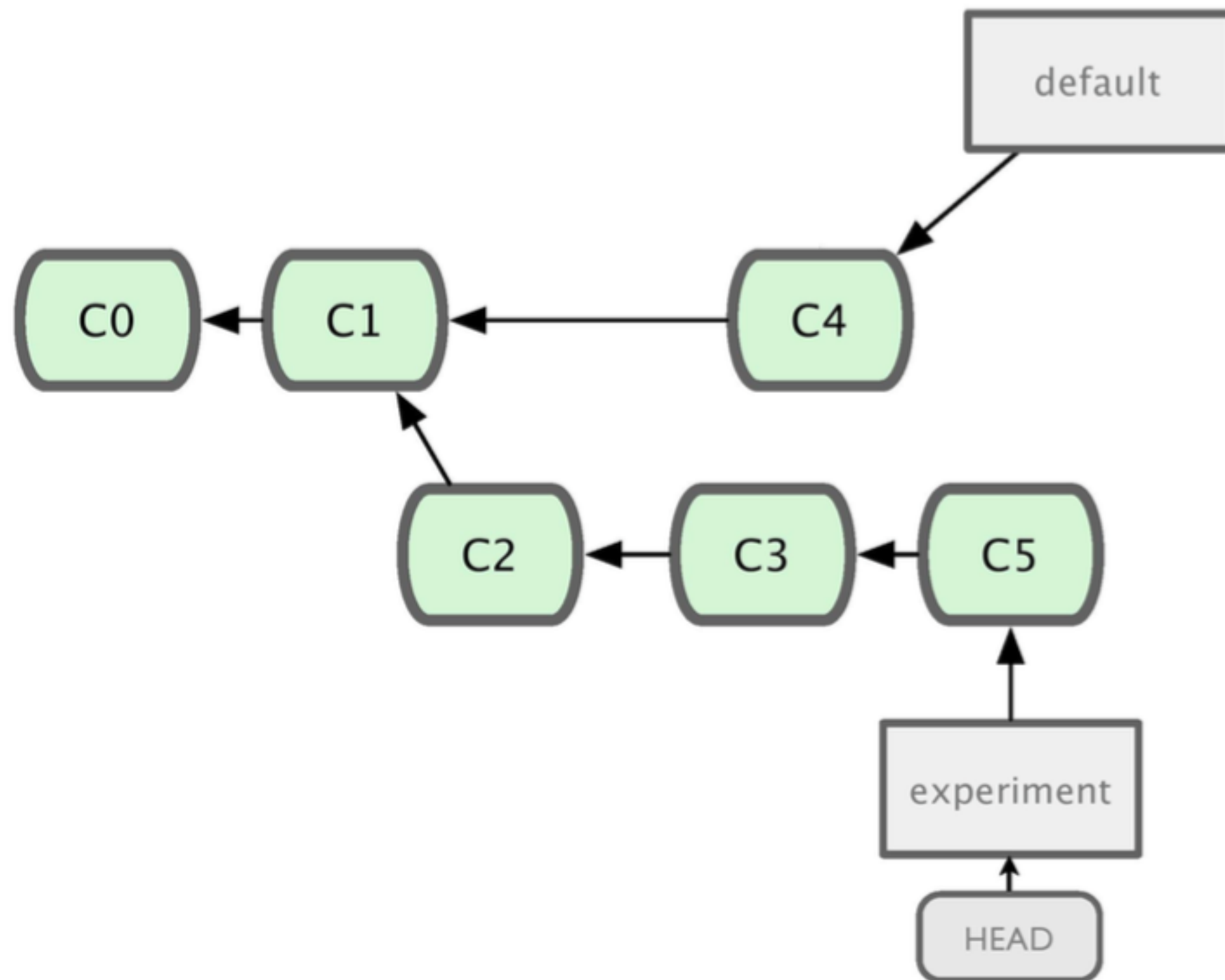


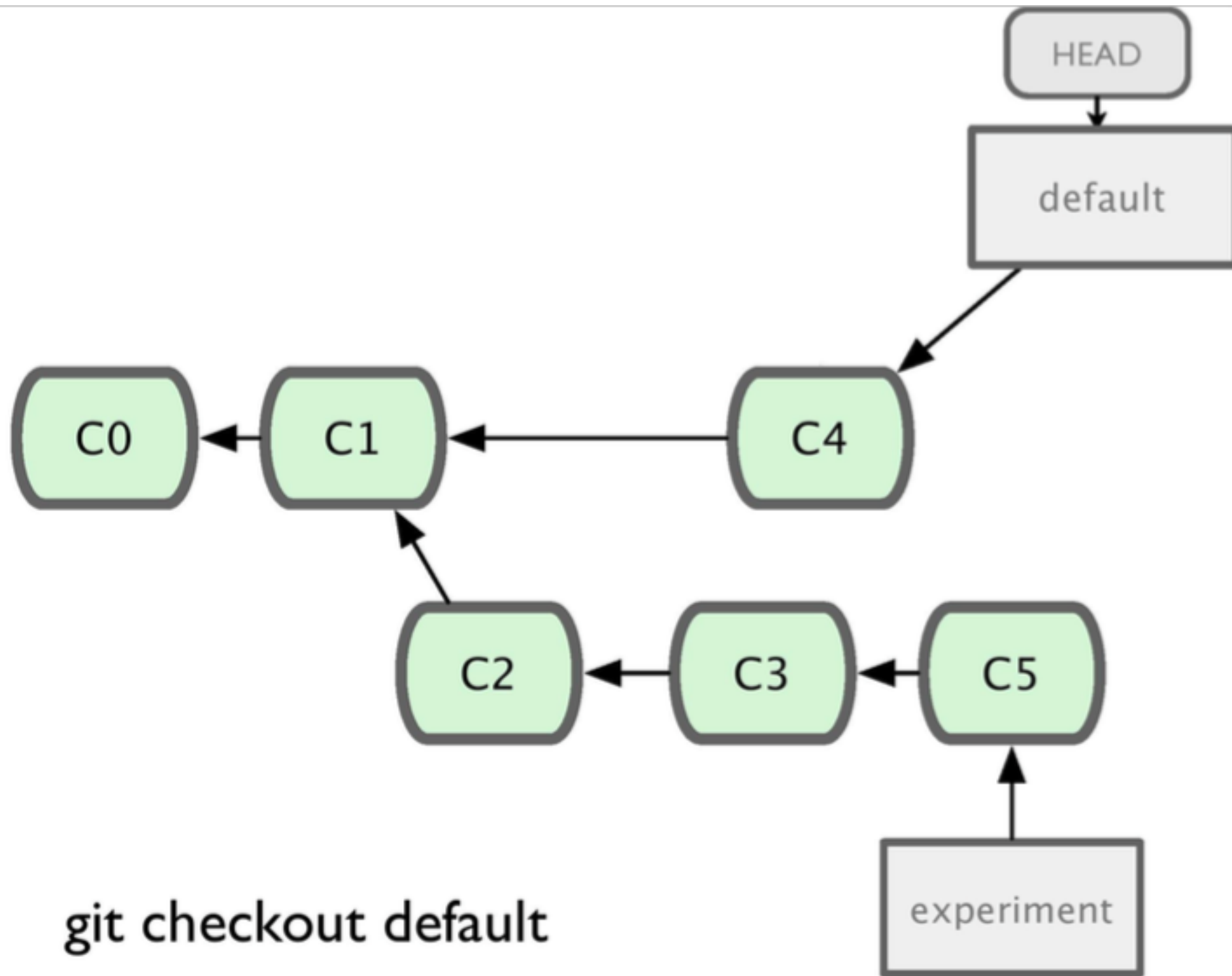
git commit



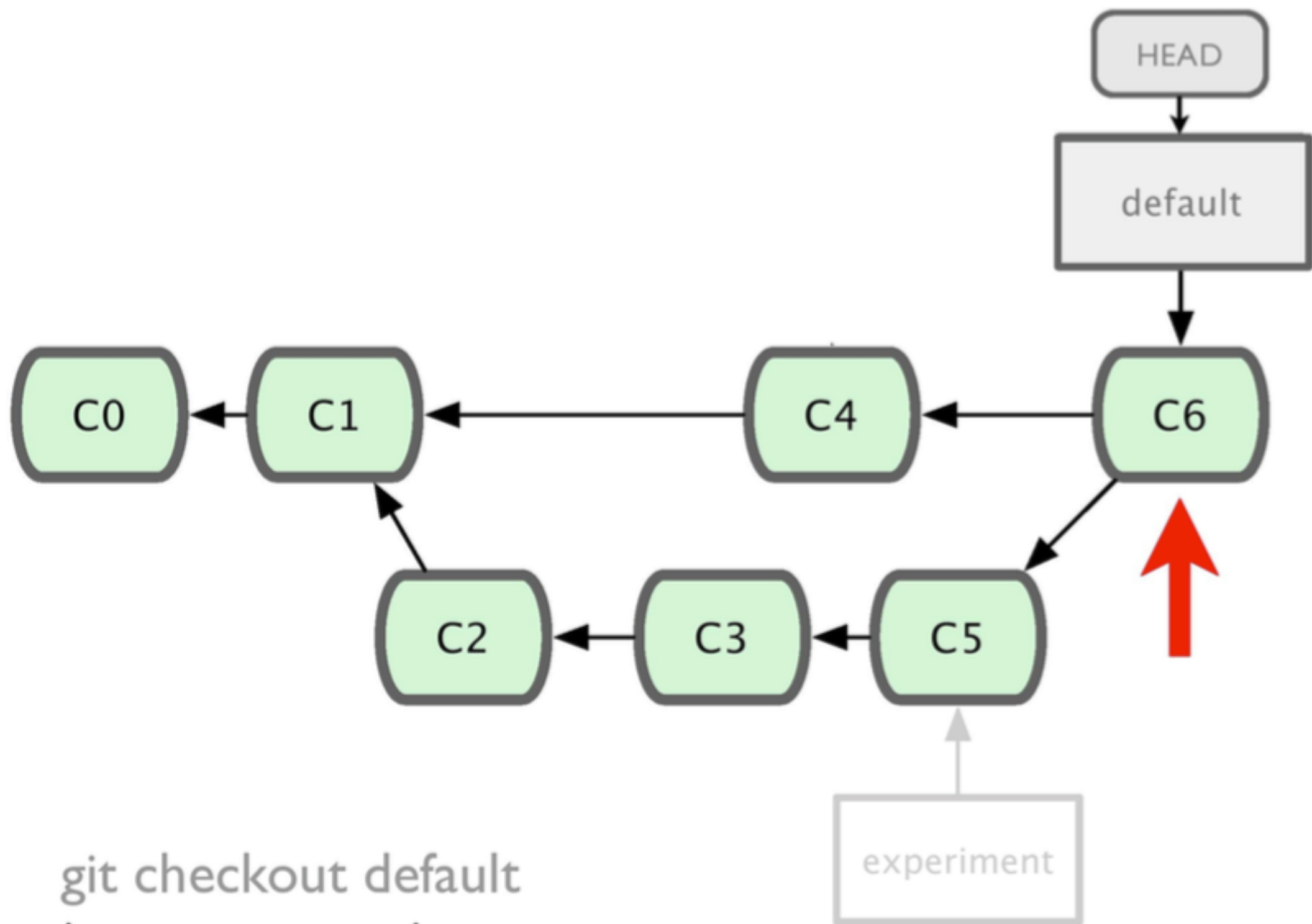


Git Merge



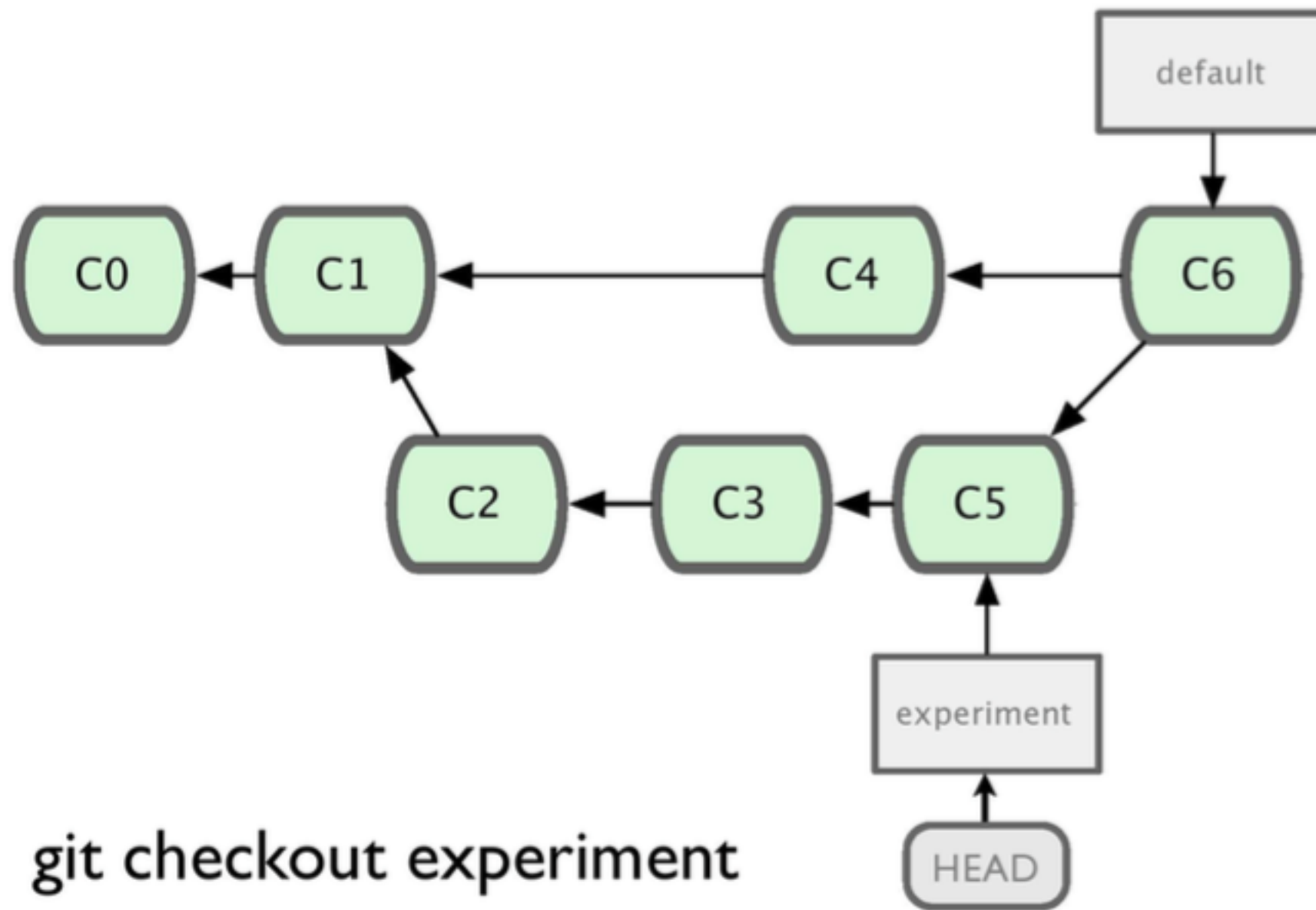


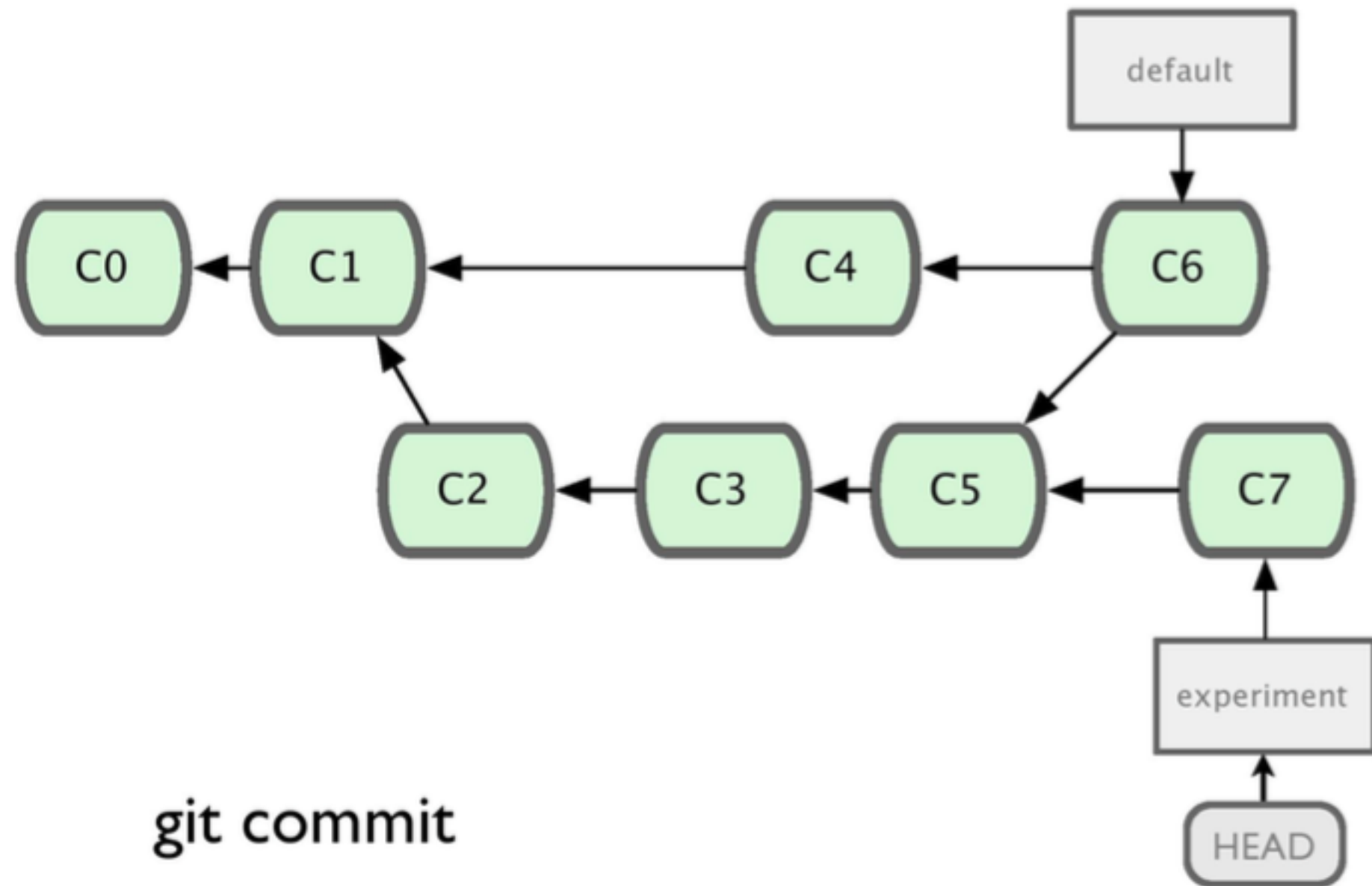
git checkout default

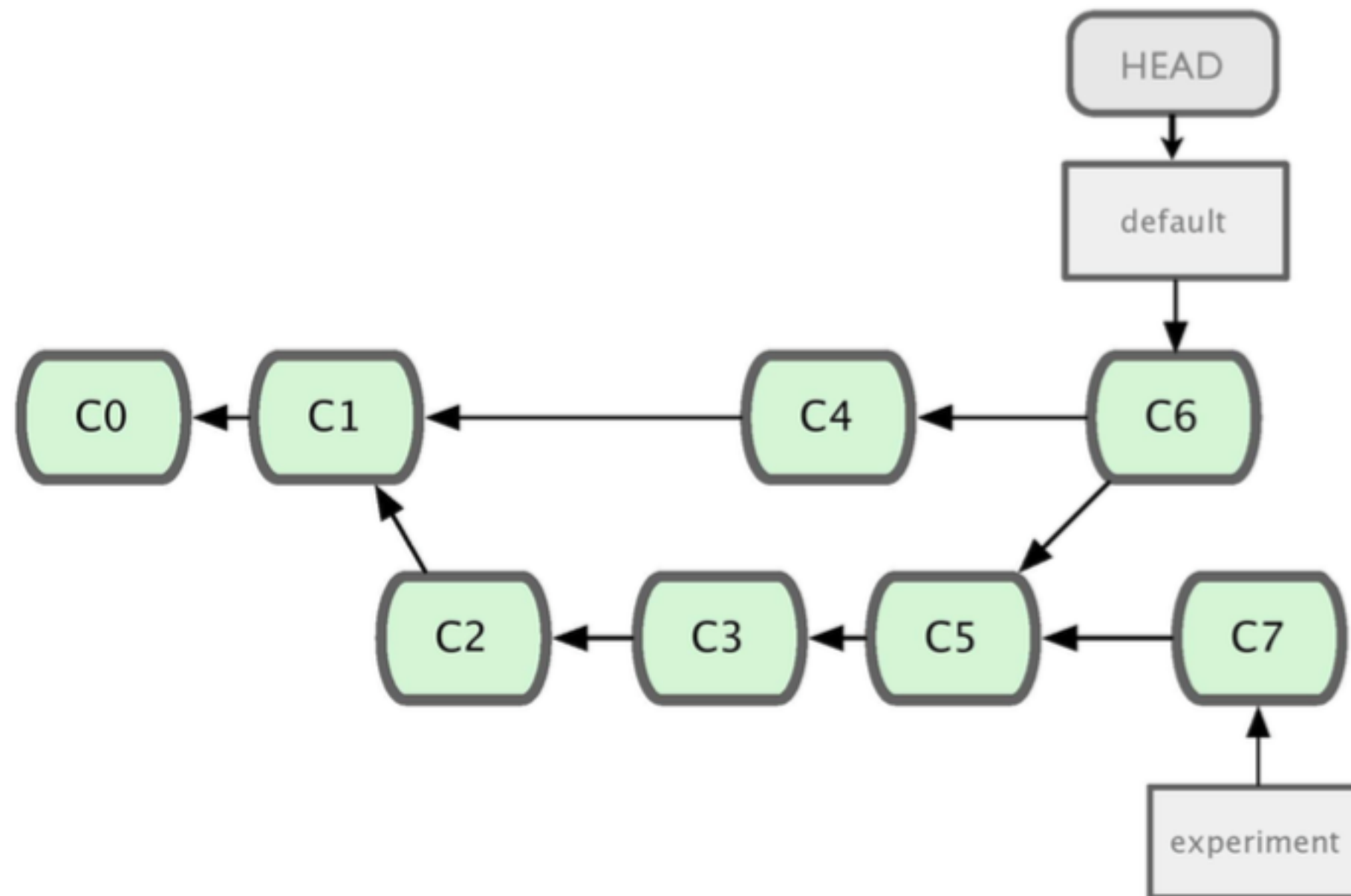


git checkout default
git merge experiment

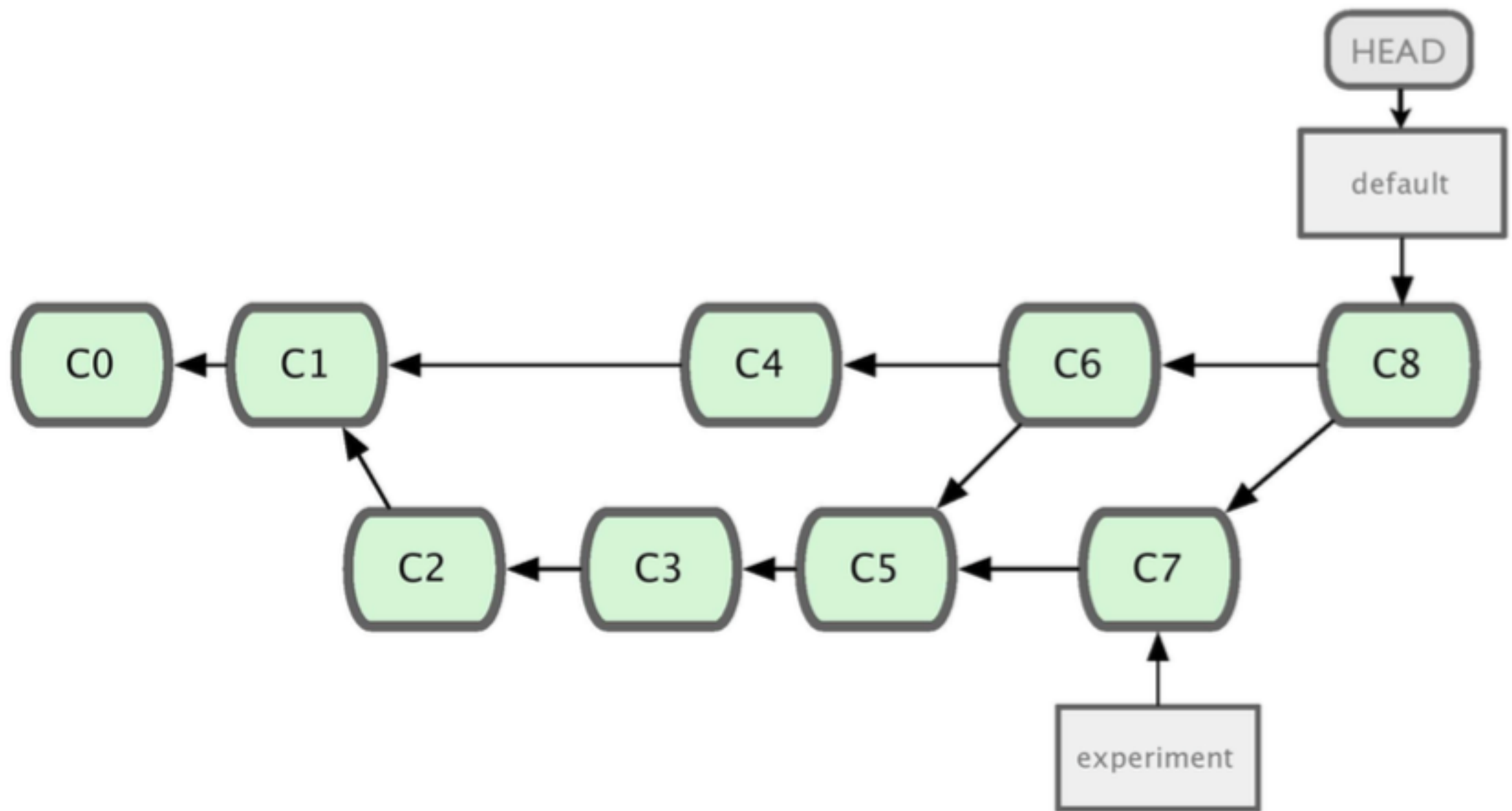








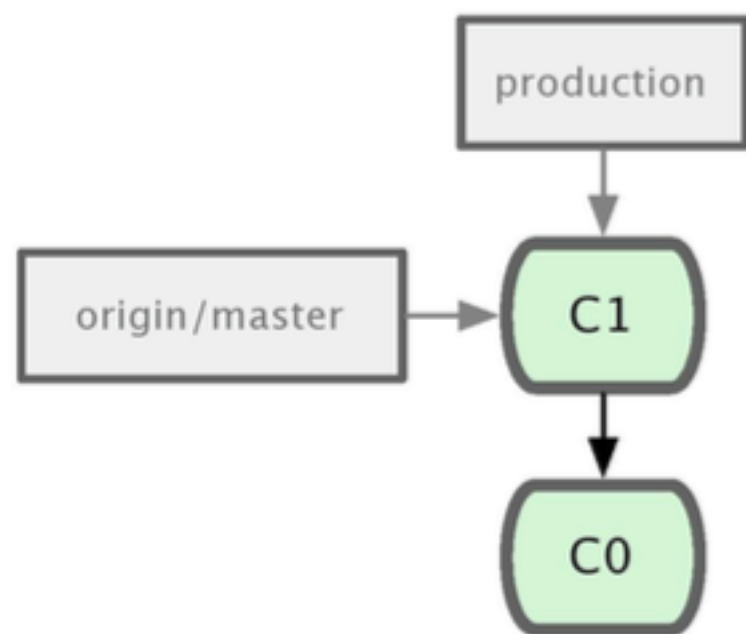
git checkout default



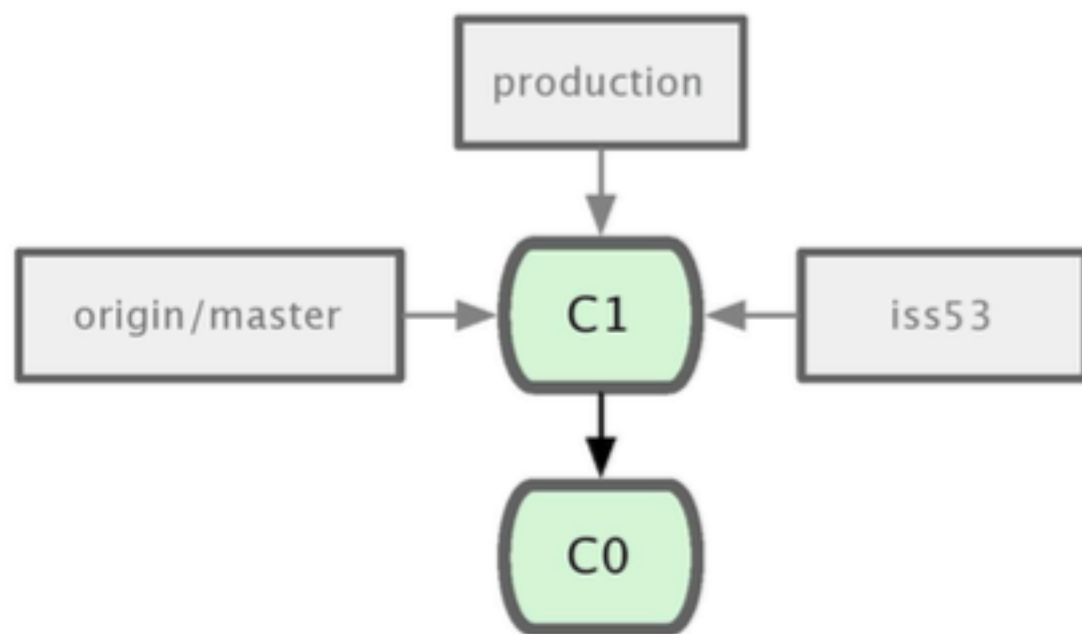
git merge experiment

Non Linear Development

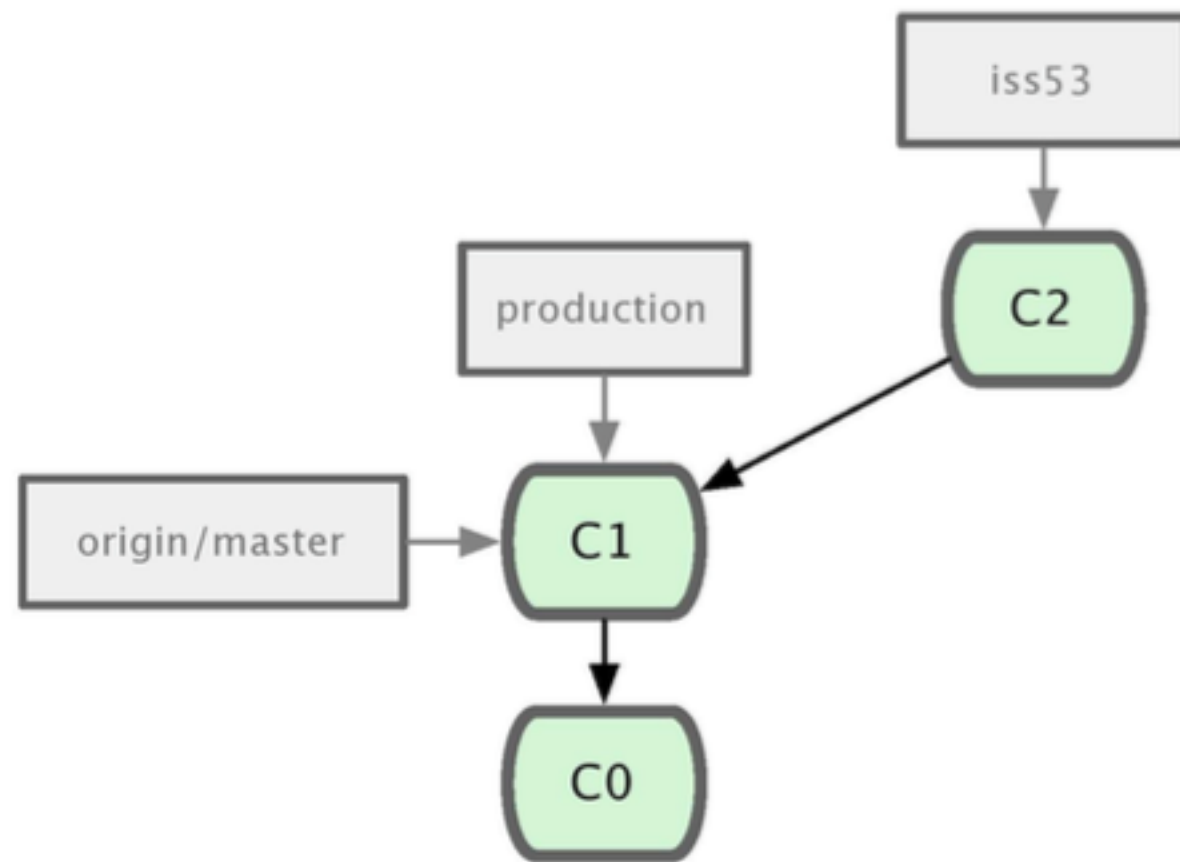
- clone the code that is in production
- create a branch for issue #53 (iss53)
- work for 10 minutes
- someone ask for a hotfix for issue #102
- checkout `production`
- create a branch (iss102)
- fix the issue
- checkout `production` mege `iss102`
- push `production`
- checkout `iss53` and keep working



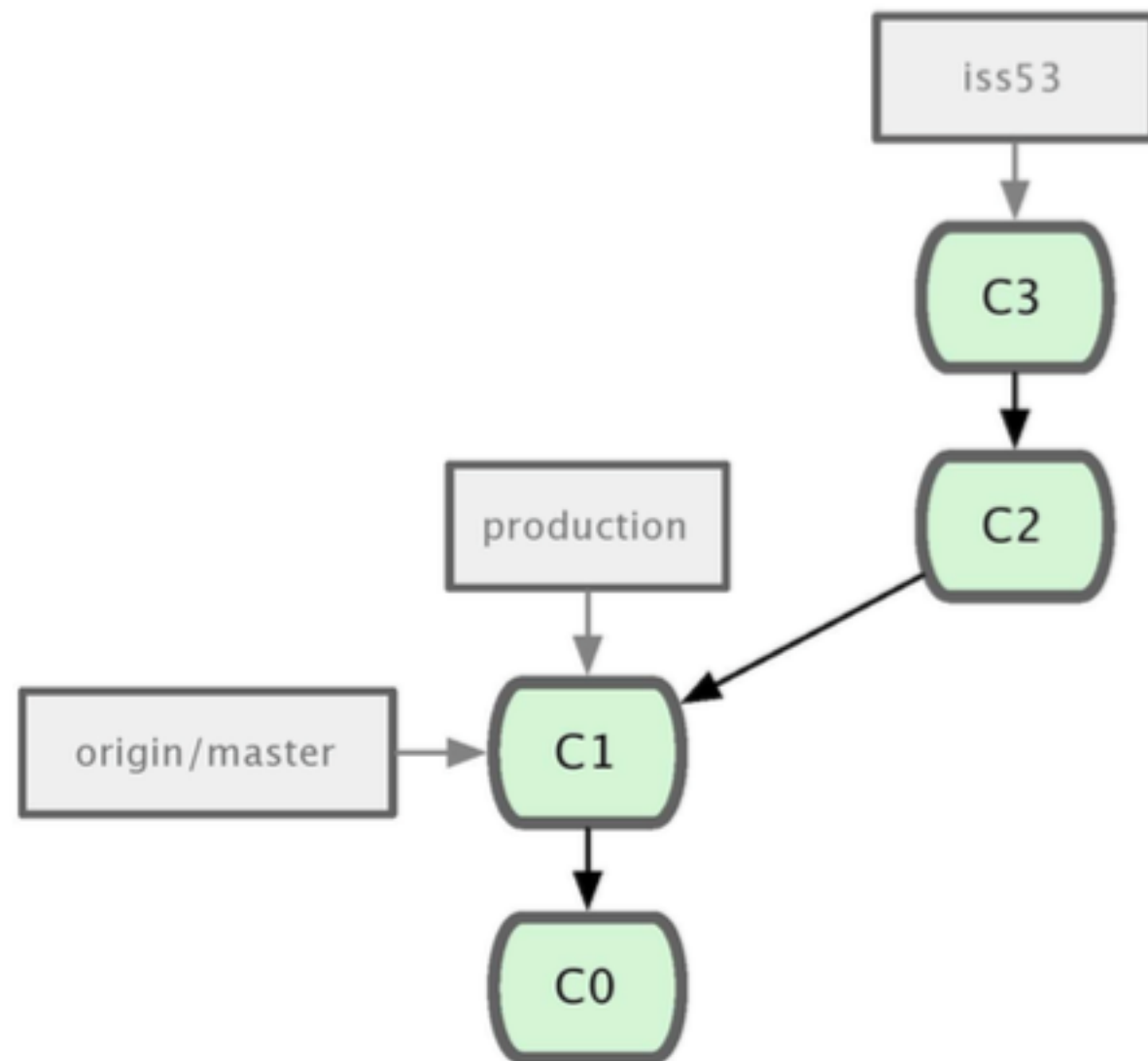
git clone



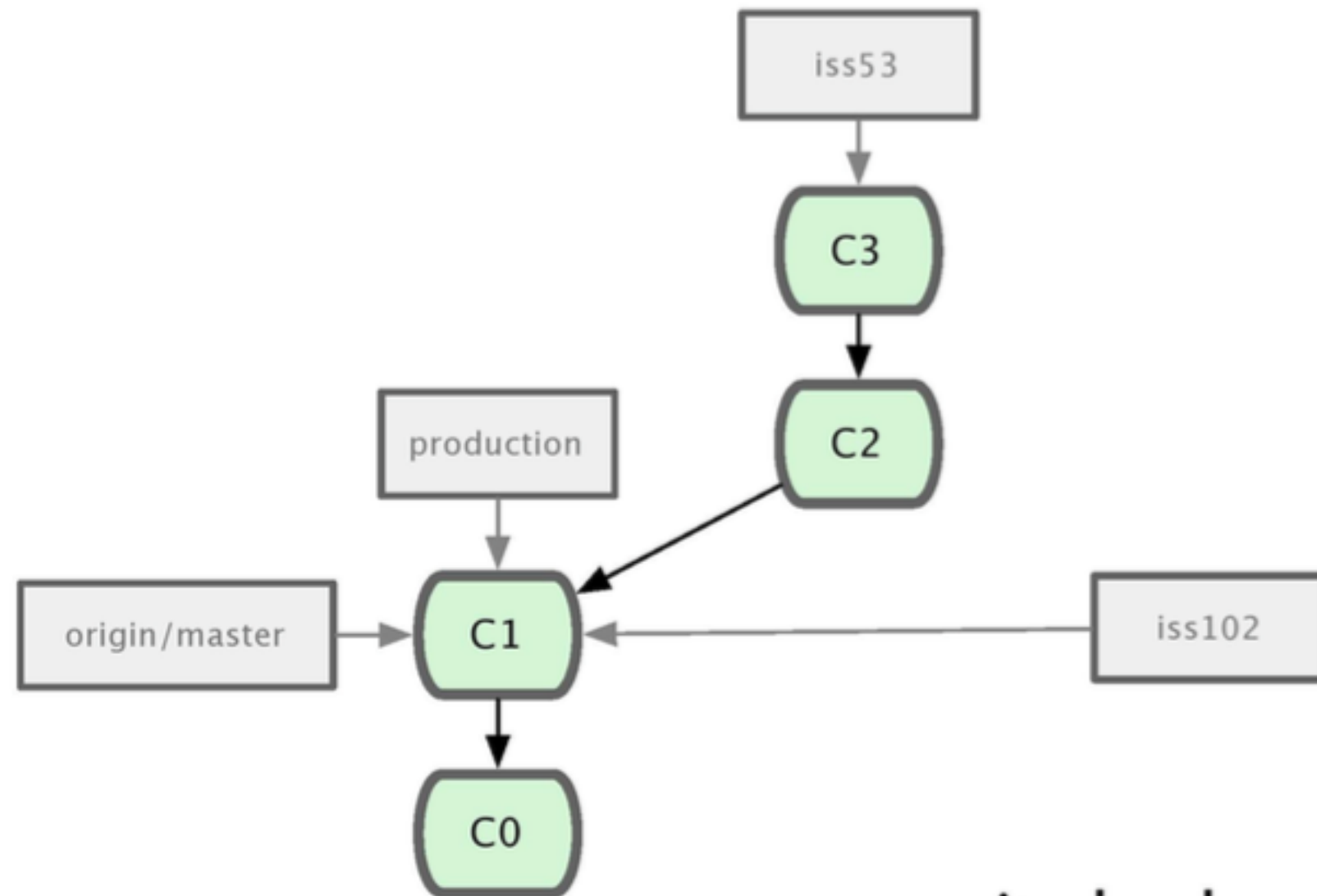
git checkout -b iss53



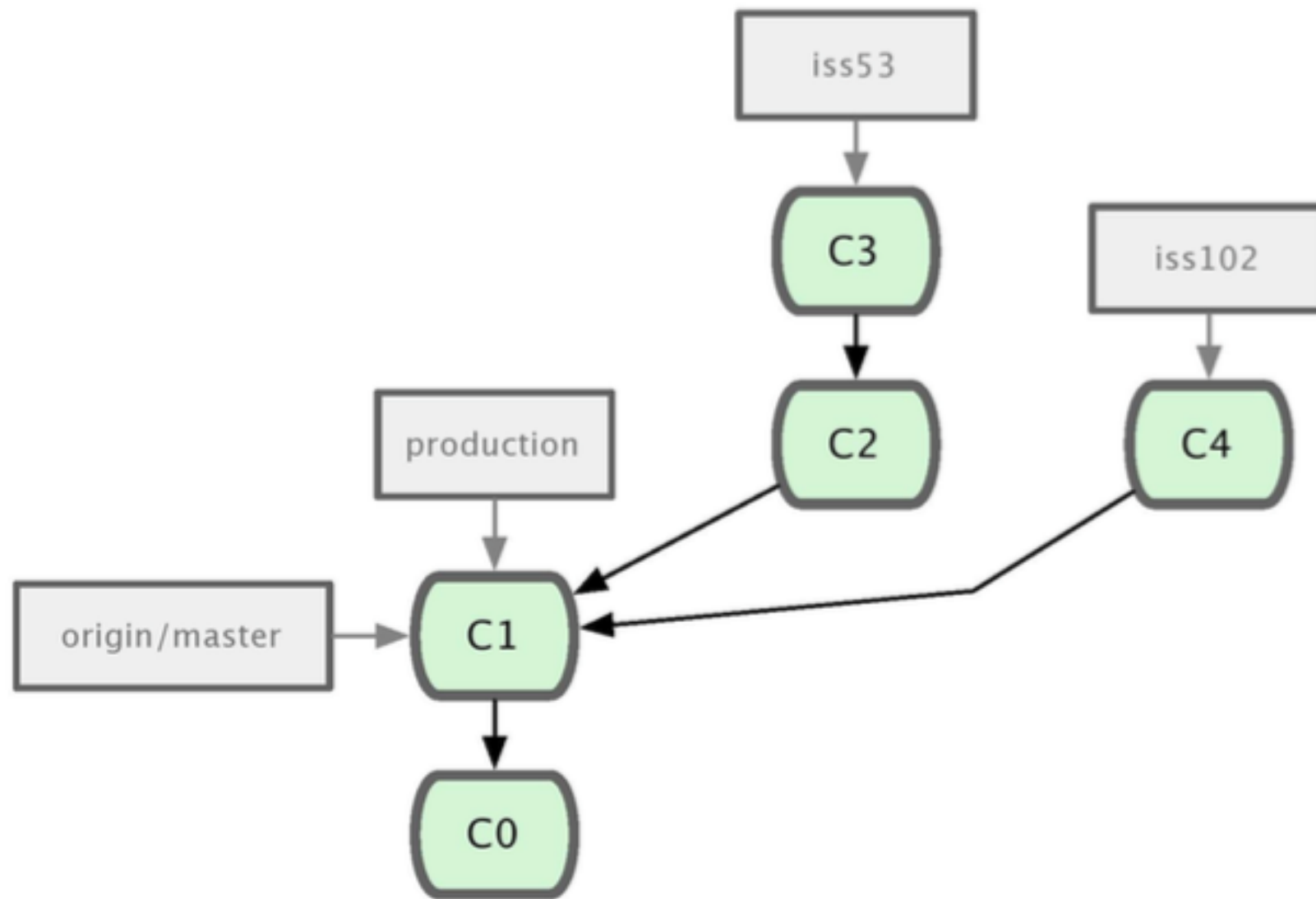
git commit



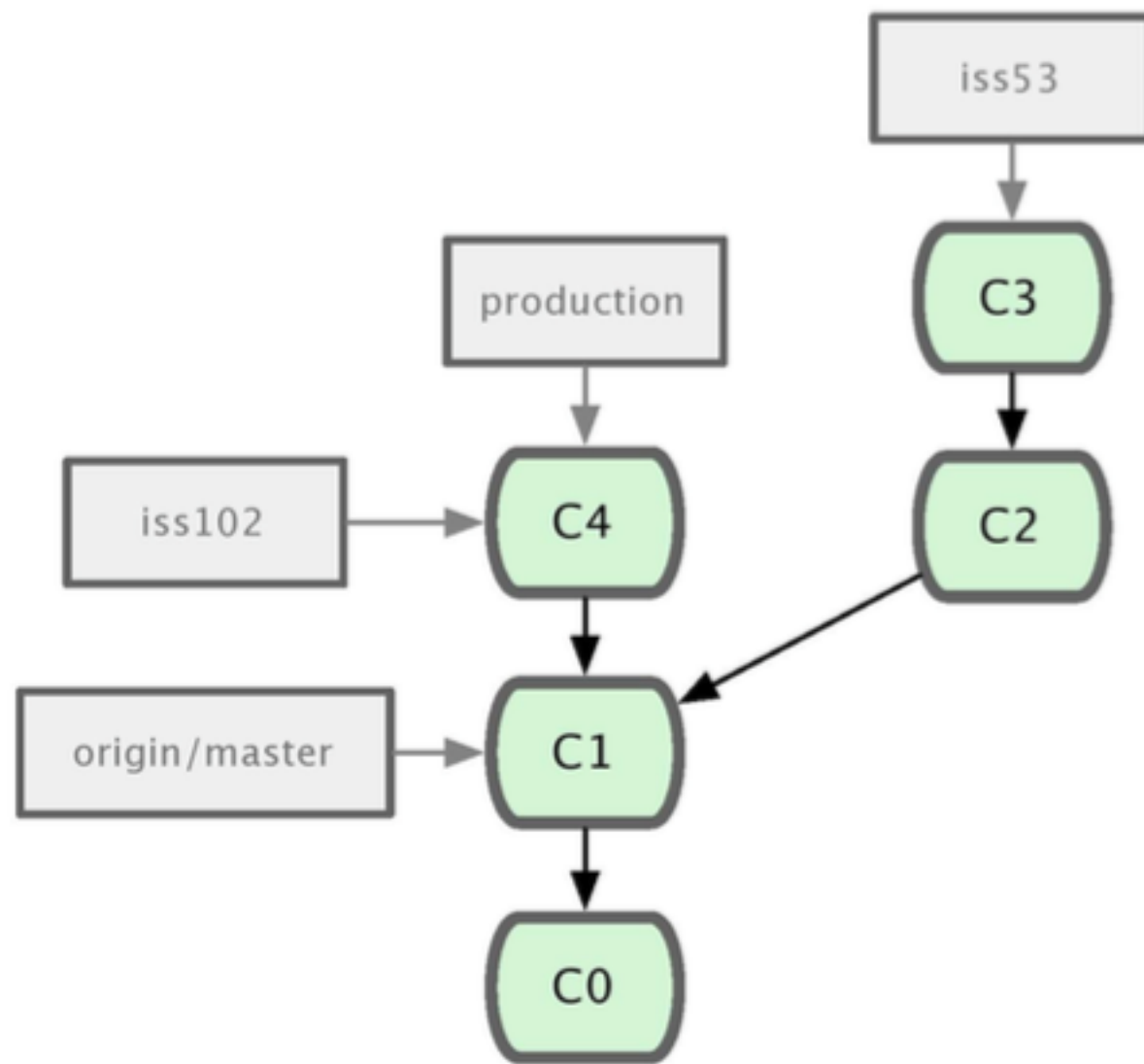
git commit



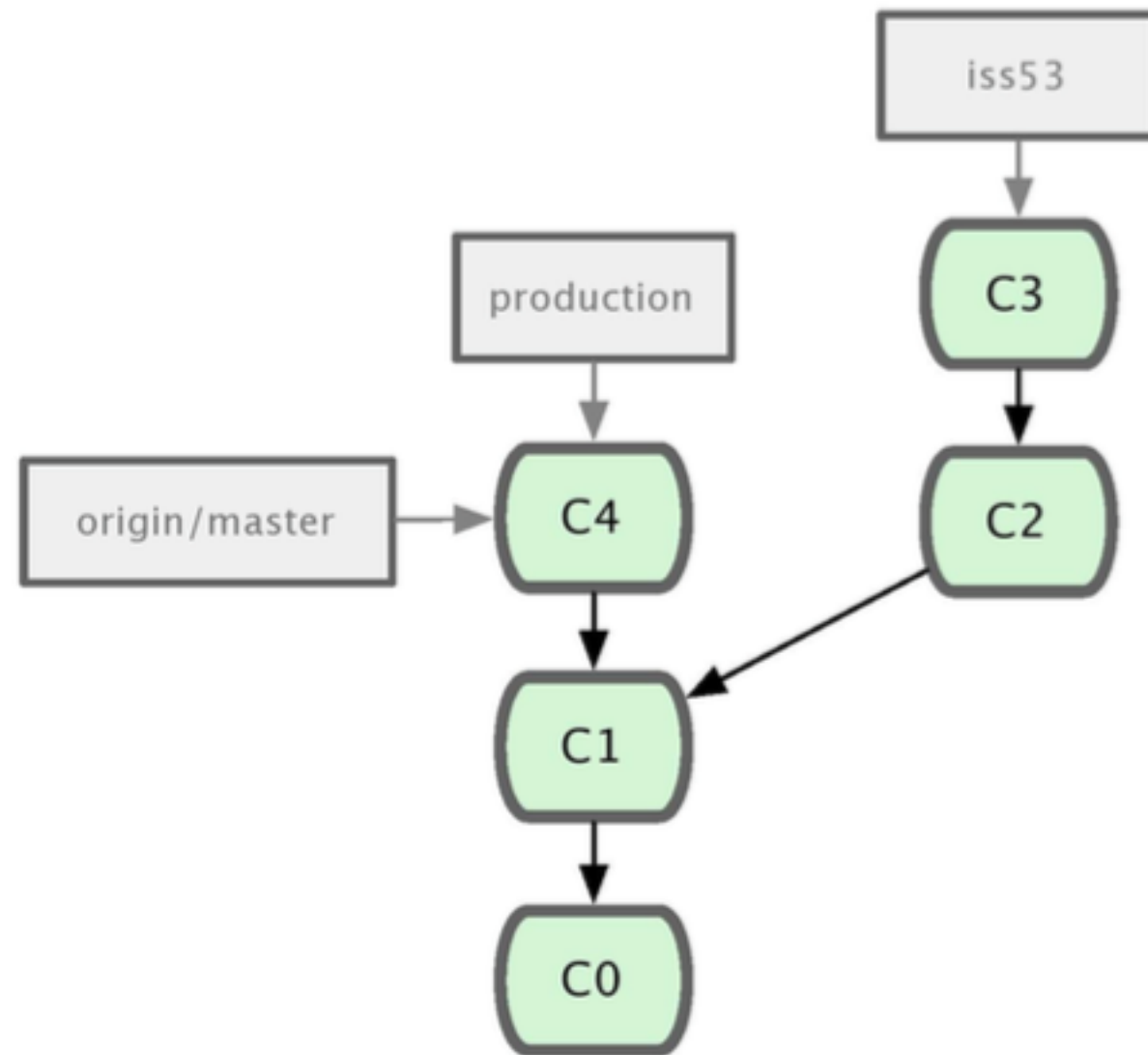
git checkout production
git checkout -b iss102



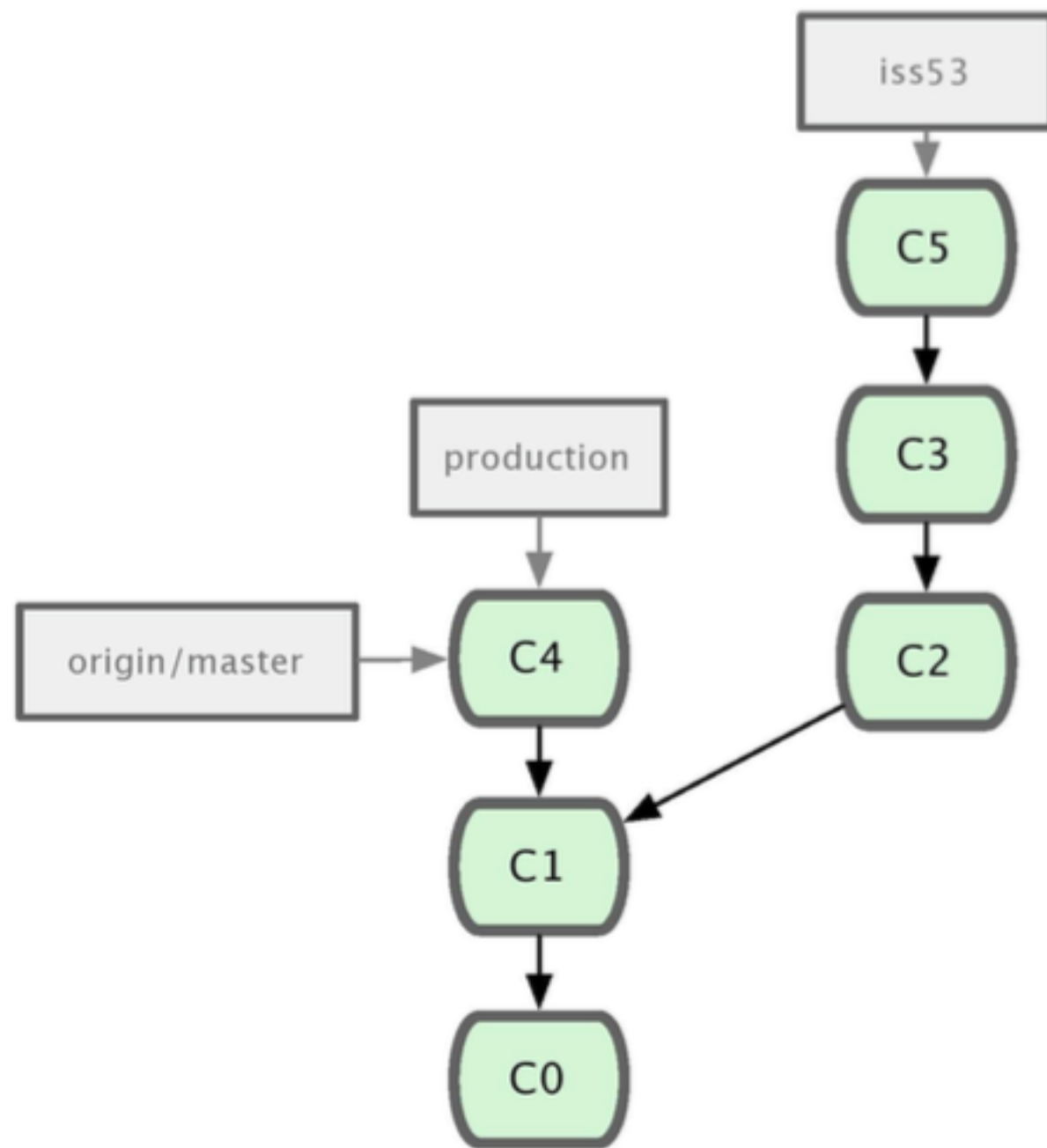
git commit



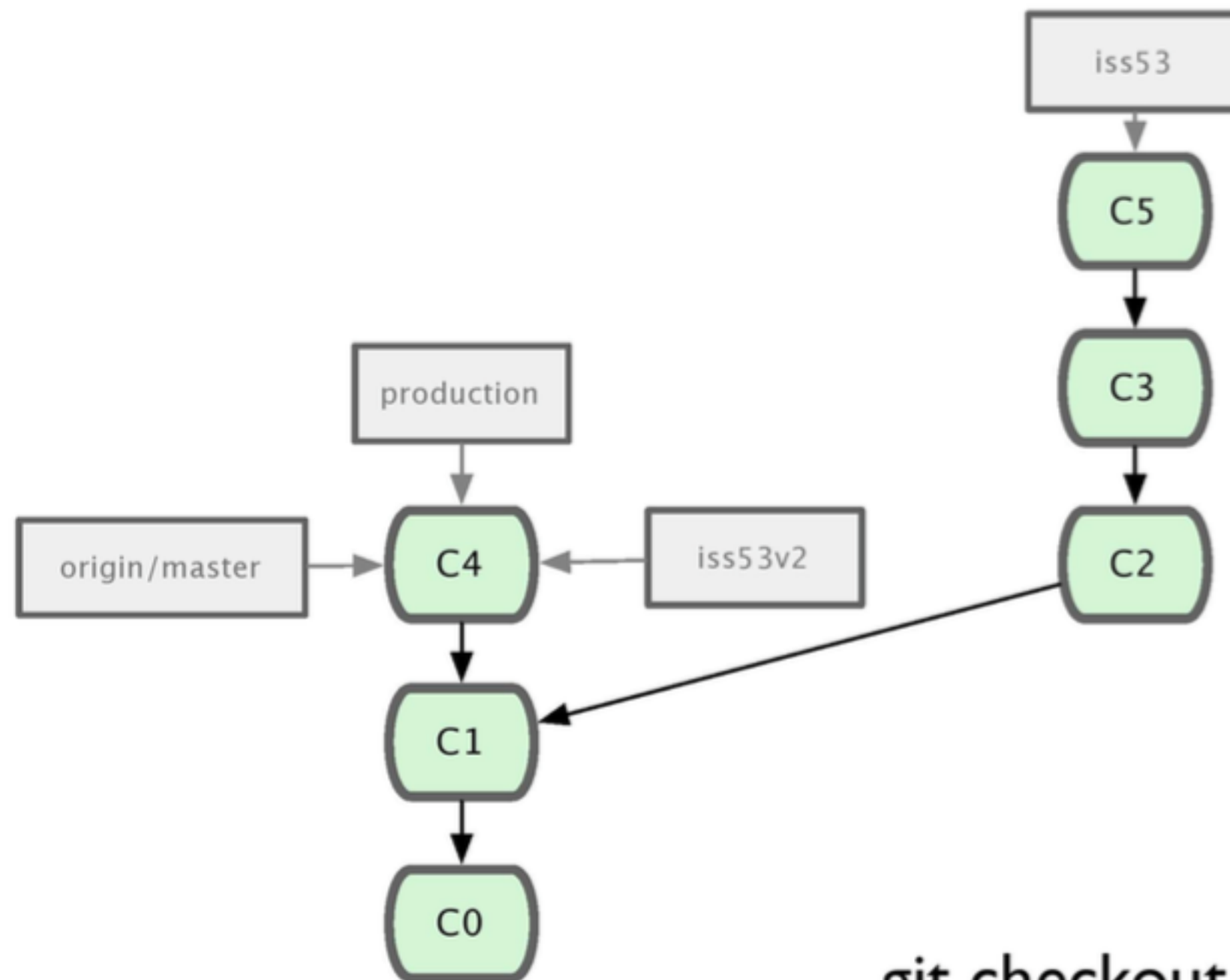
git checkout production
git merge iss102



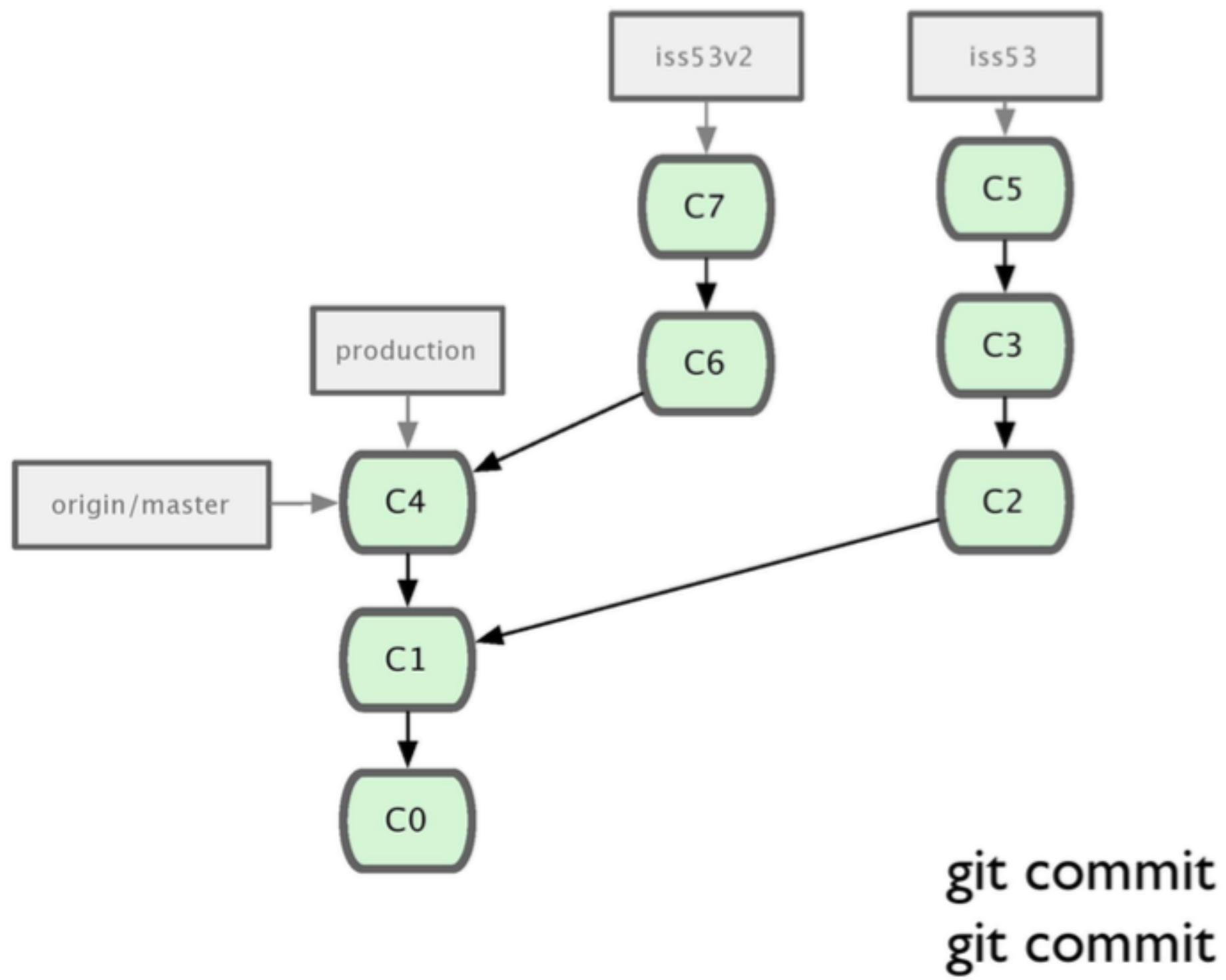
git push



git checkout iss53
git commit



git checkout production
git checkout -b iss53v2



Remote

Showing remote

```
git remote
```

```
git remote -v
```

More Remote commands

#add remote repository

```
git remote add <remote-name> <URL>
```

#fetch and pull from remote

```
git remote <remote-name>
```

#push to remote

```
git push <remote-name> <branch>
```

#inspecting remote

```
git remote show <remote-name>
```

Rename & Delete remotes

#rename remote

```
git remote rename <remote-name> <remote-newname>
```

#remove remote

```
git remote rm <remote-name>
```

Further Reading

- <https://medium.com/@adeyahya/belajar-git-f296fa656c17#.jy0a4unrg>
- <https://kelaskita.com/kelaskita/kelas/mengenal-git/>
- <https://codesaya.com/git/>

a productive team with git



Thank you
any question?