

Introduction GIT

Jun-Ru Chang

jrjang@gmail.com



Outline

- Introduction VCS
- Introduction GIT

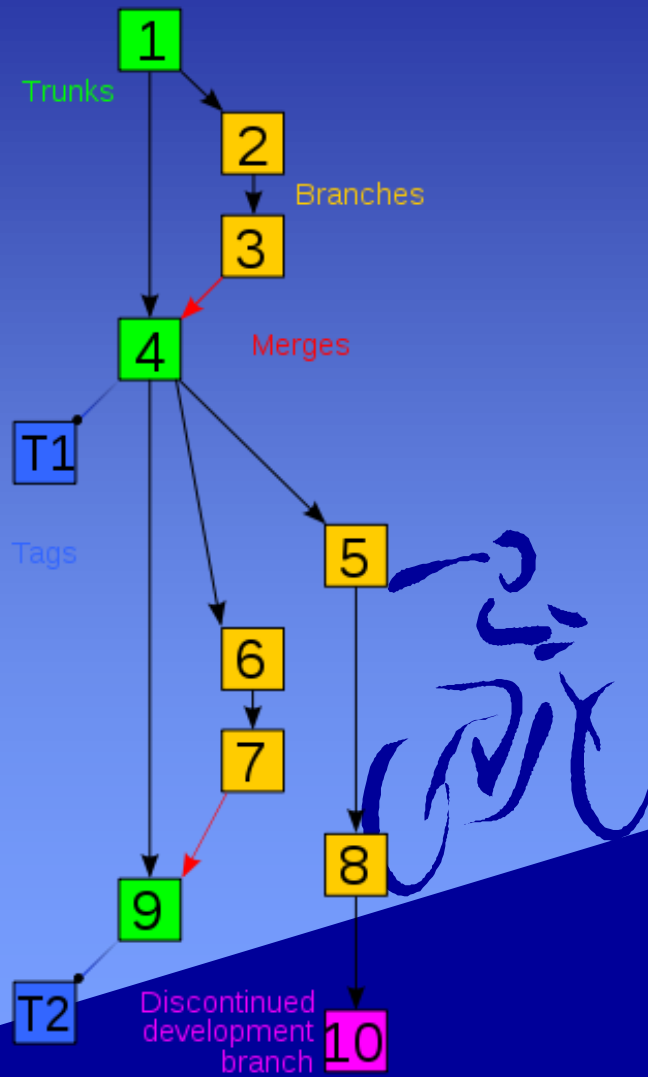


Introduction VCS

- VCS: Version Control System
 - We will be known forever by the tracks we leave
 - Automatic backup
 - Sharing on multiple computers
 - Version control and branching
 - Logging where be changed

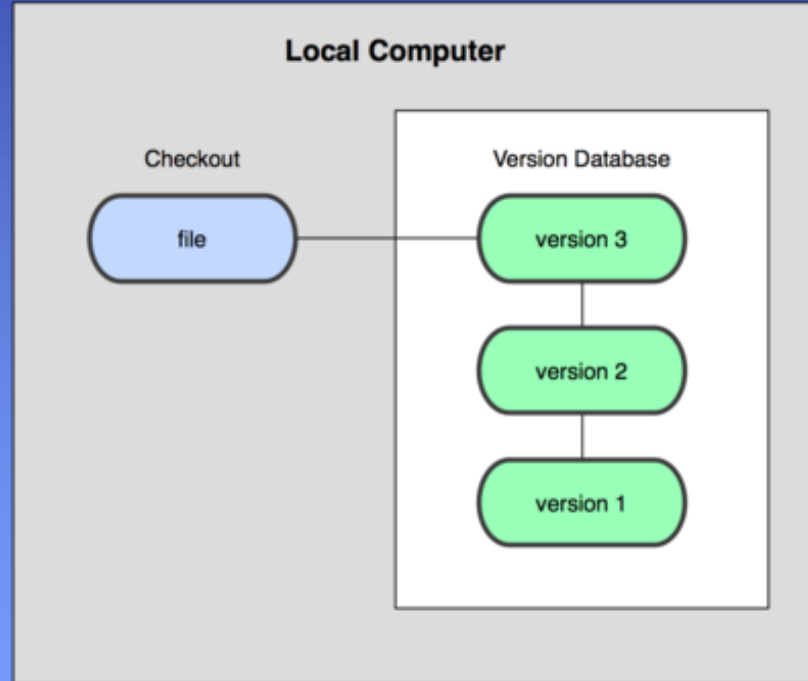


Introduction VCS



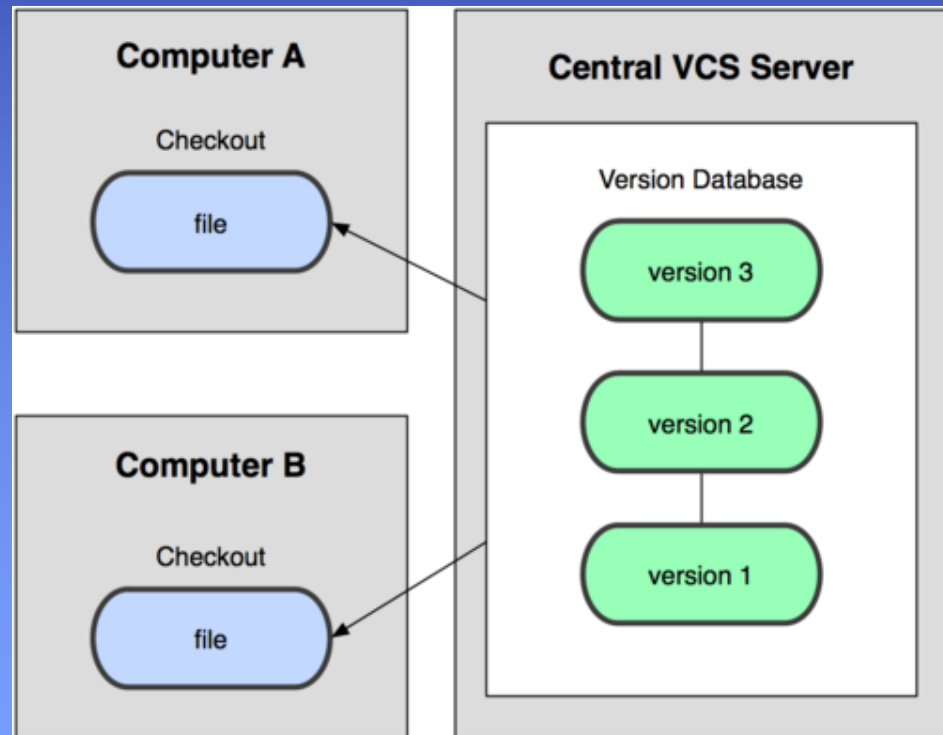
Introduction VCS

- Local VCS



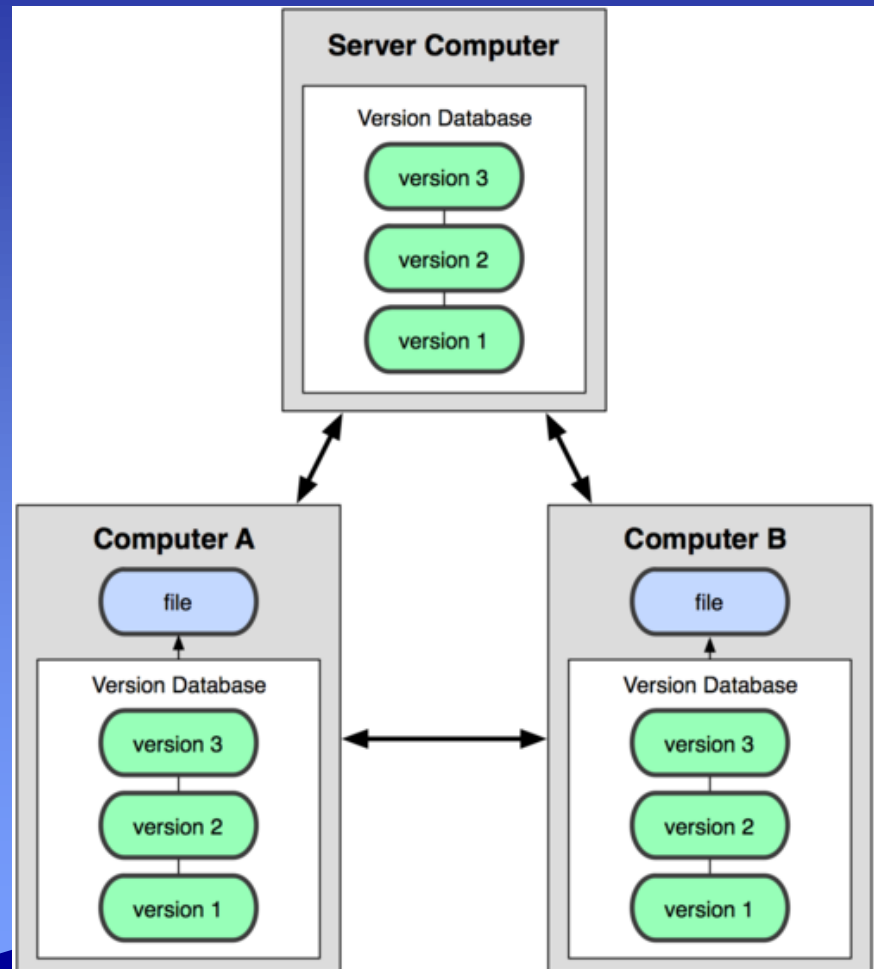
Introduction VCS

- Centralized VCS
 - Subversion (SVN)
 - checkout
 - update
 - commit

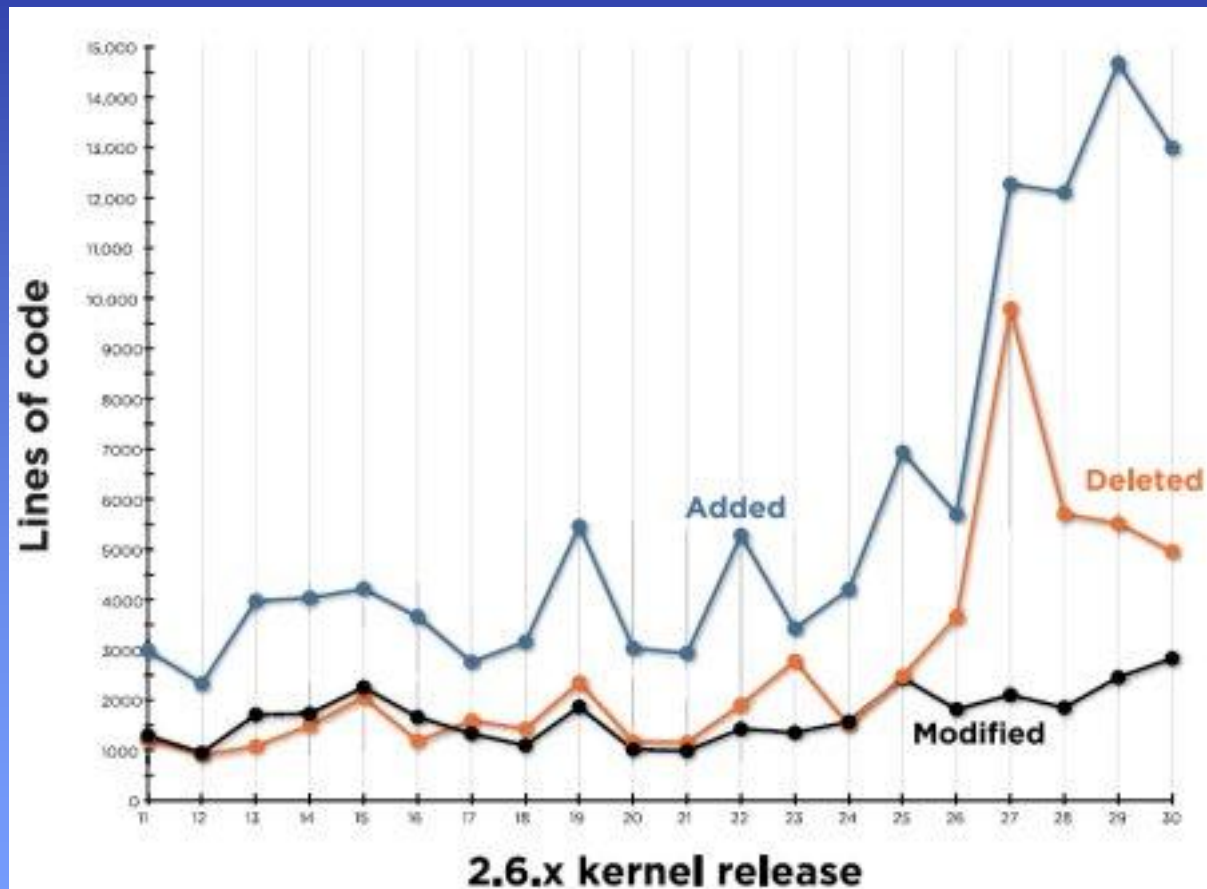


Introduction VCS

- Distributed VCS
 - GIT



Introduction GIT



Introduction GIT

- Linus Torvalds
- Originally using VCS developed by BitKeeper
- Feature
 - Fast
 - Decentralize revision control

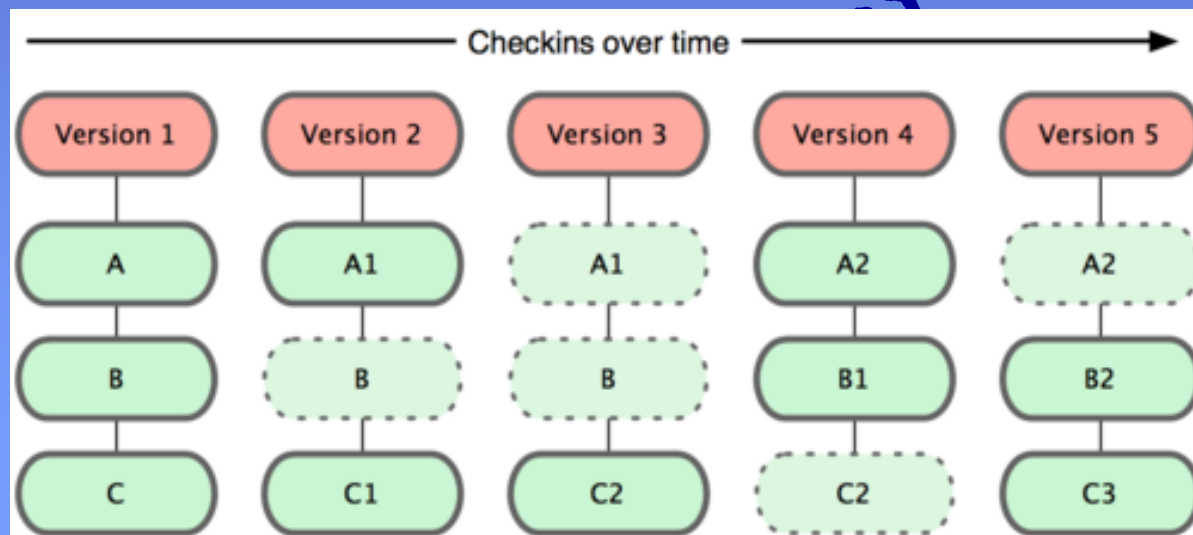
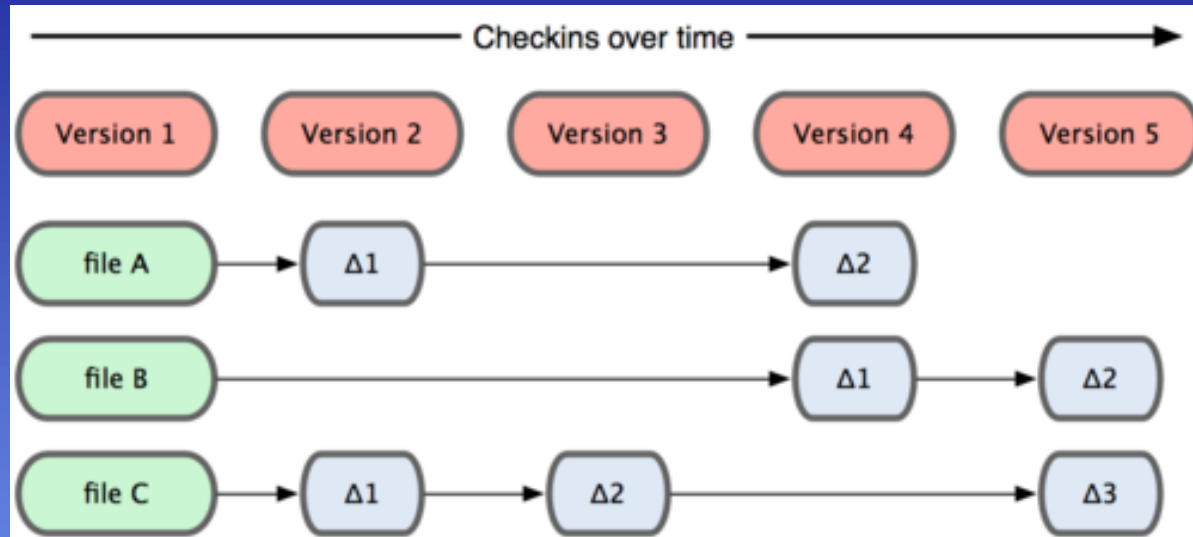


Introduction GIT

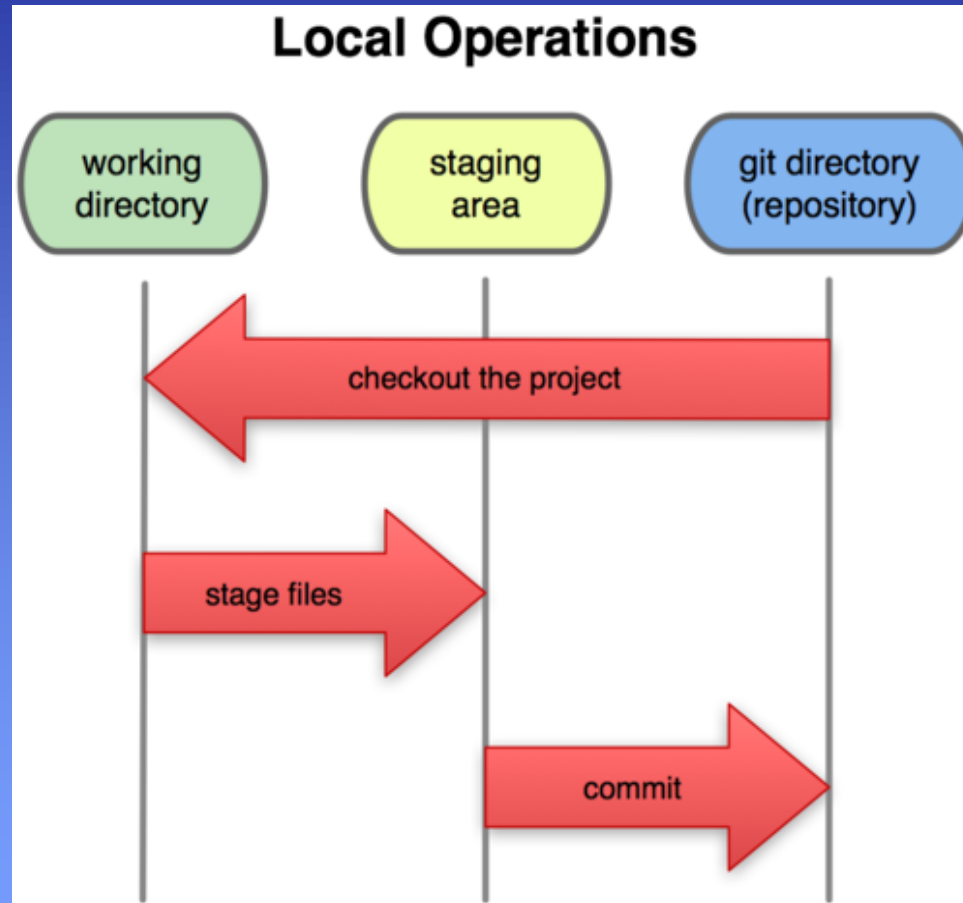
- Git basics
 - Snapshot, not differences
 - Nearly every operation is local
 - Three stages



Introduction GIT

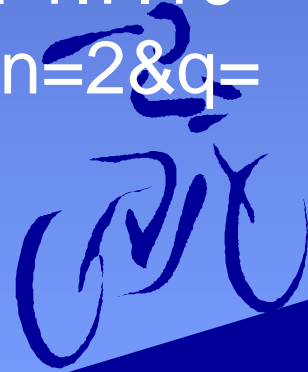


Introduction GIT



Introduction GIT

- Github
 - <https://github.com/>
- msysgit
 - <http://code.google.com/p/msysgit/downloads/detail?name=PortableGit-1.7.10-preview20120409.7z&can=2&q=>



Introduction GIT

- Register account on Github
- Unzip msysgit
- Generate key
 - `$ ssh-keygen -t rsa -C "user@email"`
- Import public key into Github
 - `$ ssh -T git@github.com`
- Create a new repository




Introduction GIT

- `$ git config --global user.name "user1"`
- `$ git config --global user.email "user1@email"`



Introduction GIT

- `$ mkdir test; cd test`
 - Getting a repository
 - importing existing project or directory into git
`$ git init`
`$ git remote add origin`
`git@github.com:user1/test.git`
 - cloning an existing git repository from another server
`$ git clone`
`git@github.com:jrjang/ppt.git`
- 

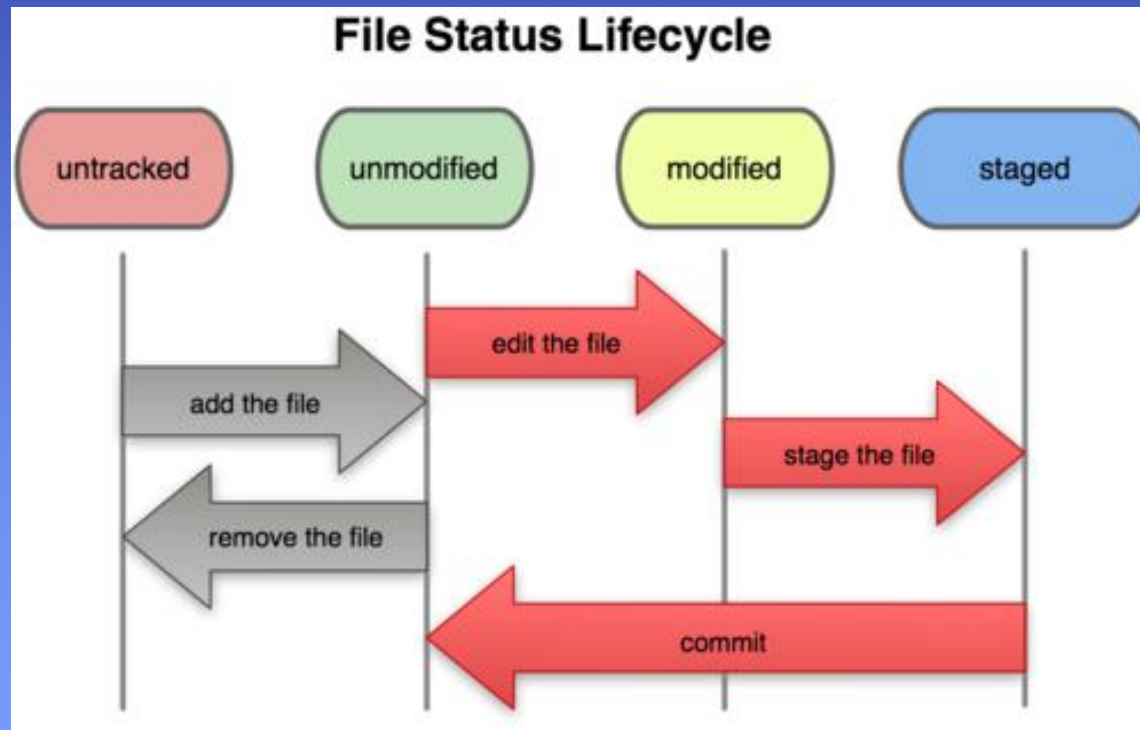
Introduction GIT

- git add
- git rm
- git mv
- git diff
- git commit
- git log

- ```
$ touch test1.c
$ touch test2.c
$ git add test1.c
test2.c
$ git commit -c "xxx"
$ git rm test2.c
$ git mv test1.c test.c
$ git status
$ git commit -c "yyy"
$ git log
$ echo "test" > test.c
$ git diff
```

# Introduction GIT

- git status



# Introduction GIT

- `git commit --amend`
- `git reset HEAD file`
- `$ touch test3.c`  
`$ git add test3.c`  
`$ git reset HEAD test3.c`



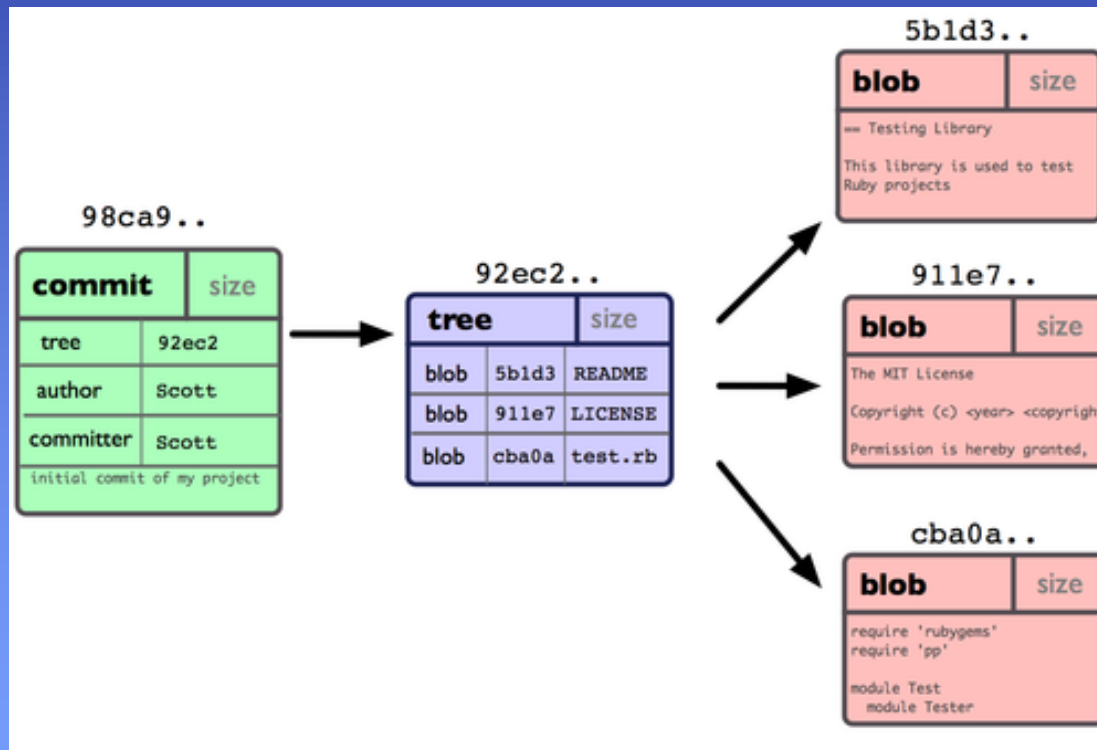
# Introduction GIT

- Working with remotes
  - git pull
  - git push
- \$ git push origin master
- \$ git pull git@github.com:user1/test.git

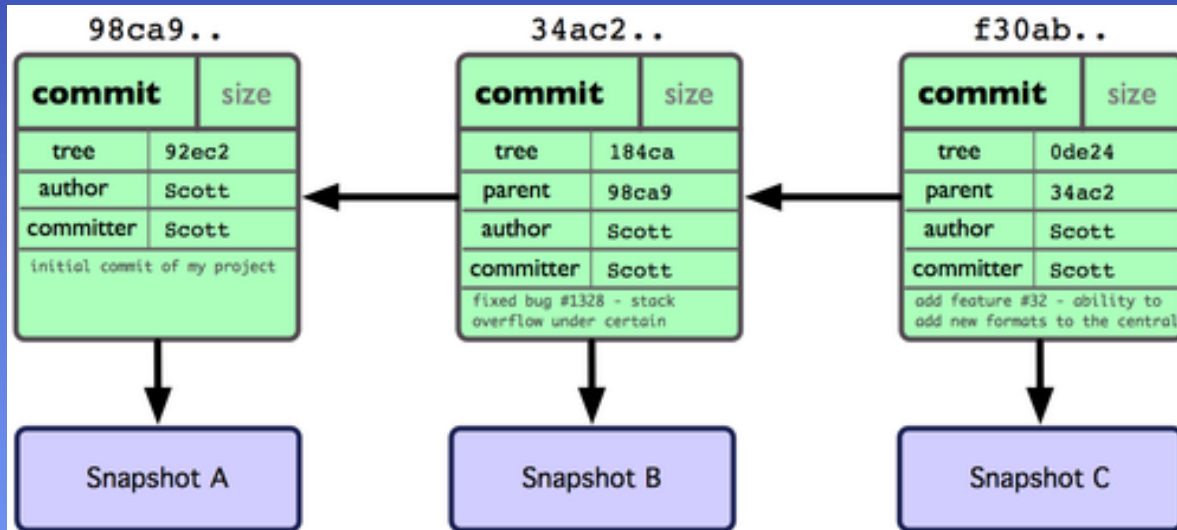


# Introduction GIT

- What a branch is

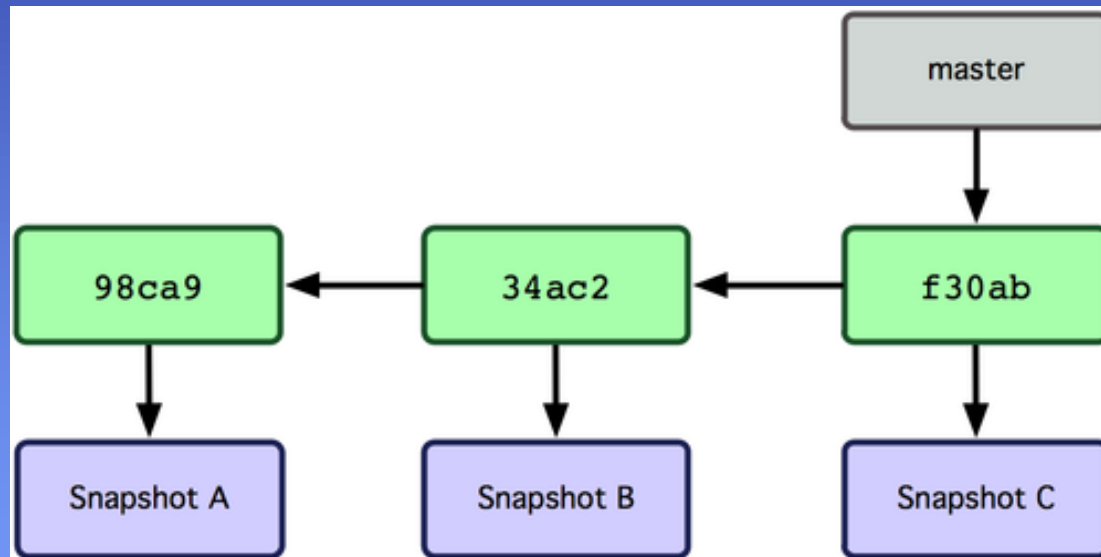


# Introduction GIT



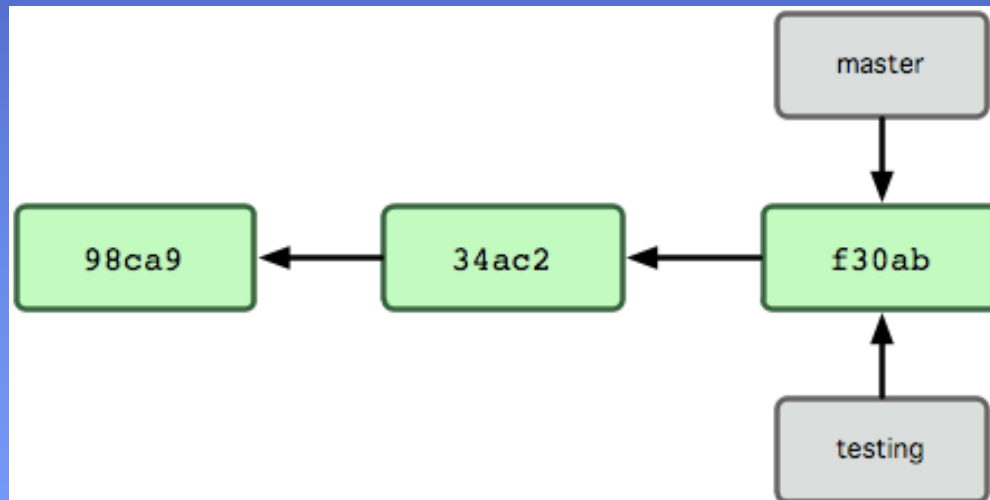
*Handwritten signature*

# Introduction GIT



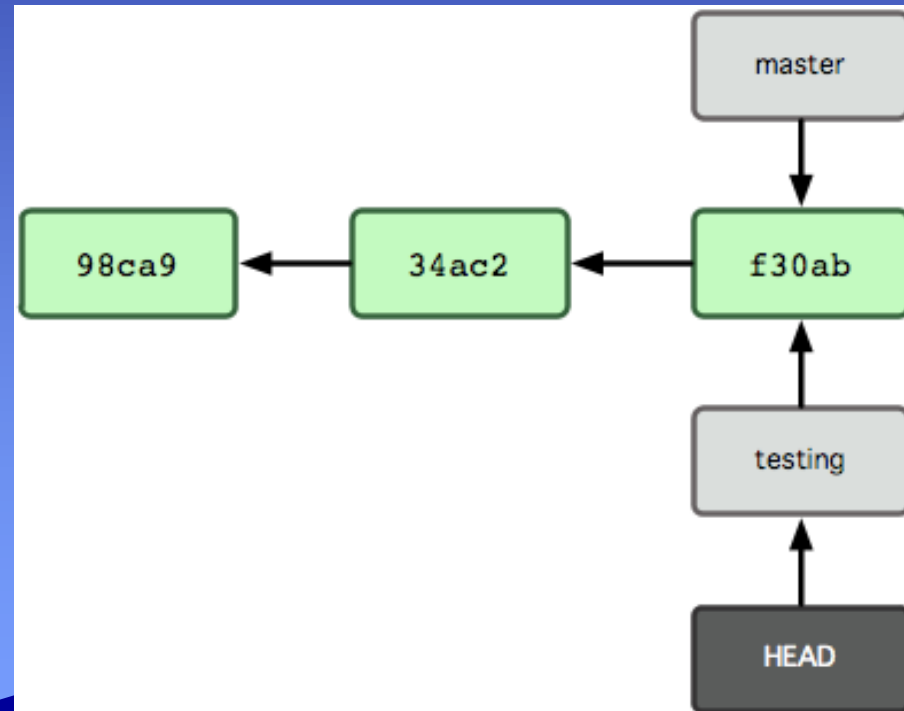
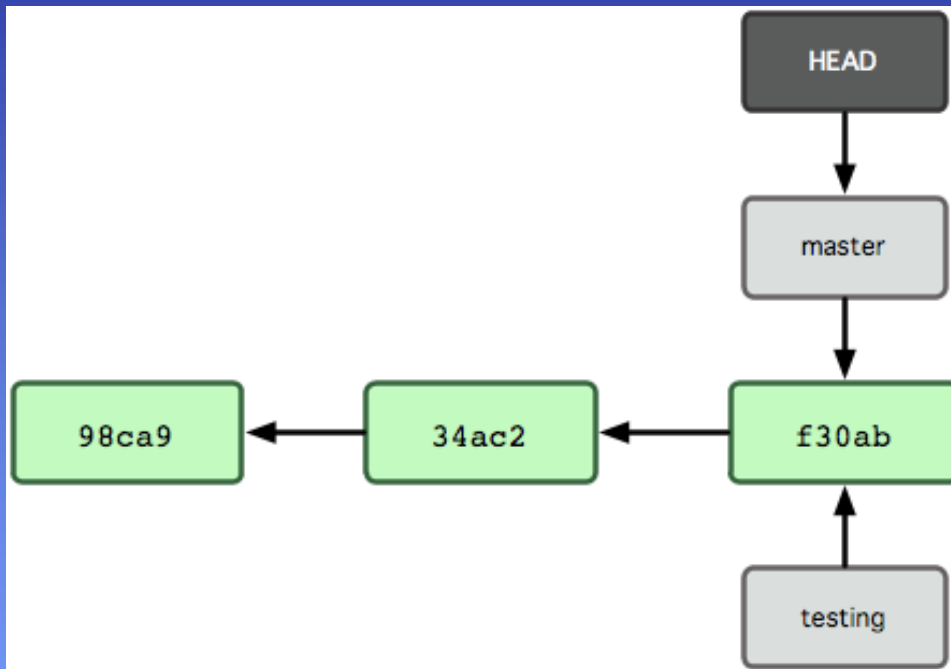
# Introduction GIT

- git checkout
- git branch

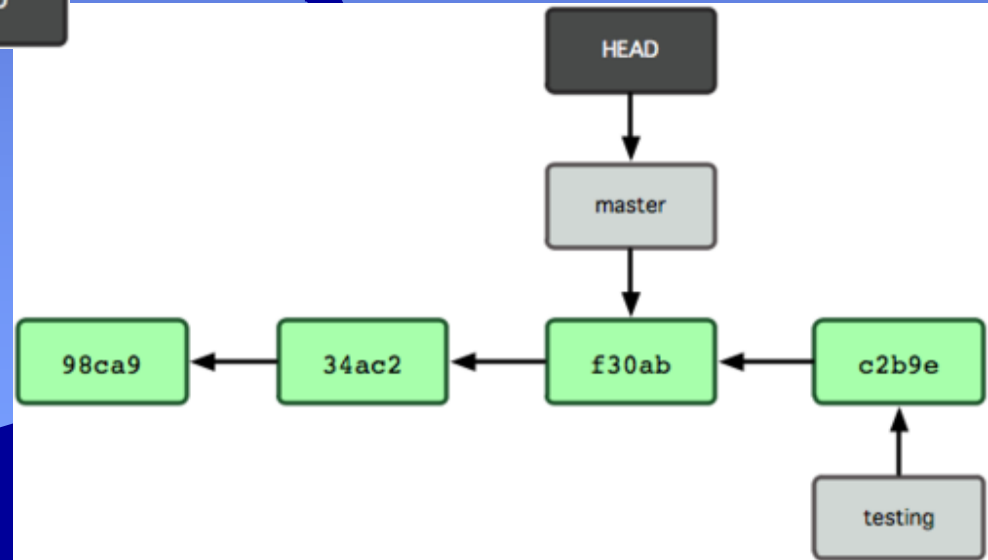
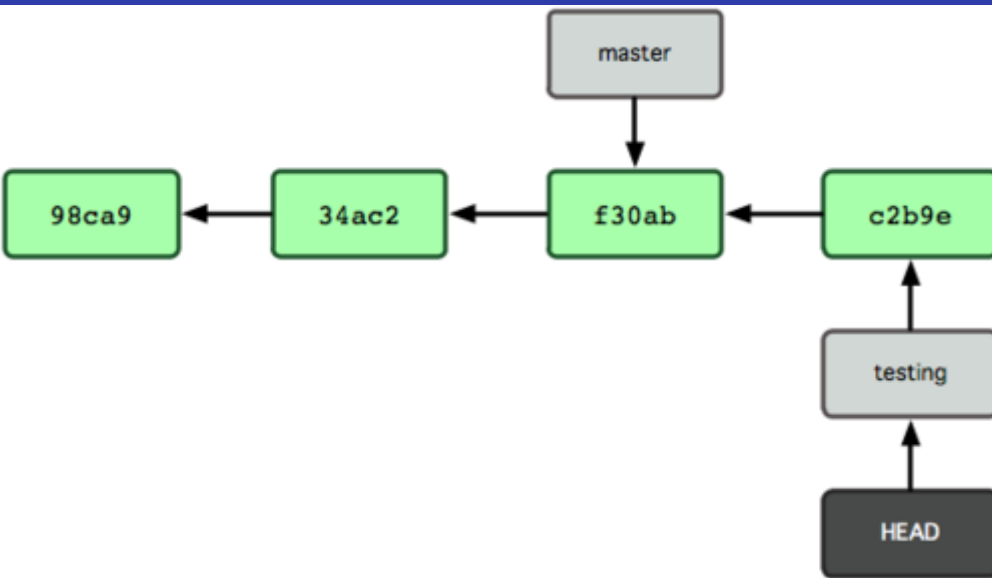




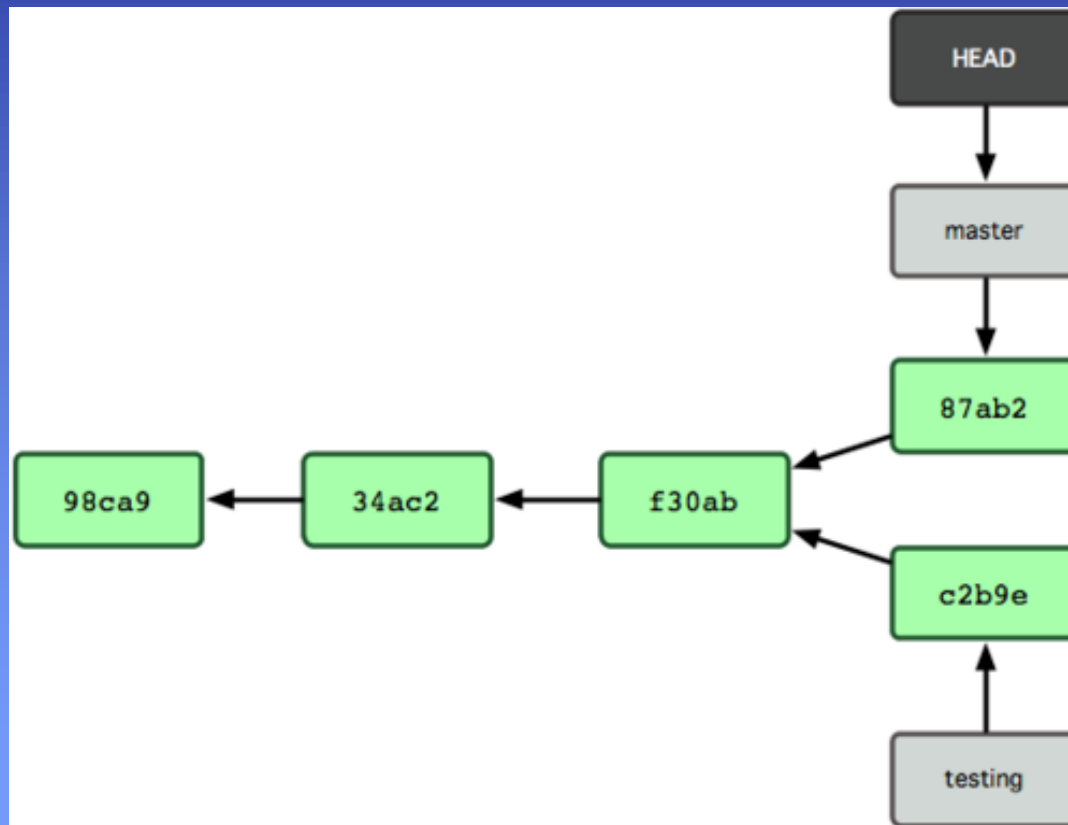
# Introduction GIT



# Introduction GIT



# Introduction GIT



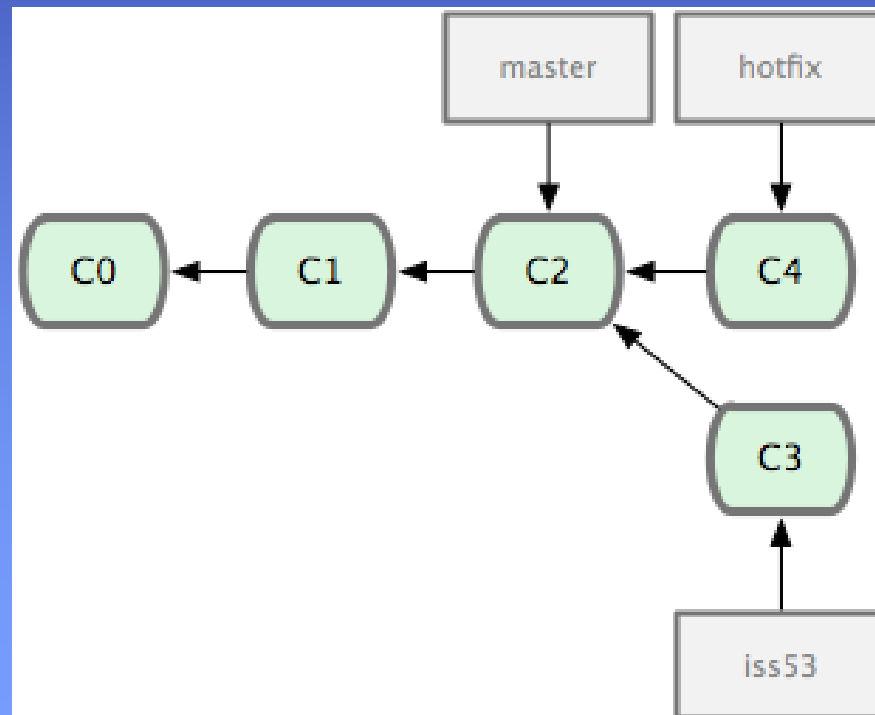
# Introduction GIT

- \$ git checkout -b testing  
\$ git branch  
\$ touch haha.c  
\$ git add haha.c  
\$ git commit -c "in testing"  
\$ git checkout master  
\$ touch hehe.c  
\$ git add hehe.c  
\$ git commit -c "in master"  
\$ git show-branch



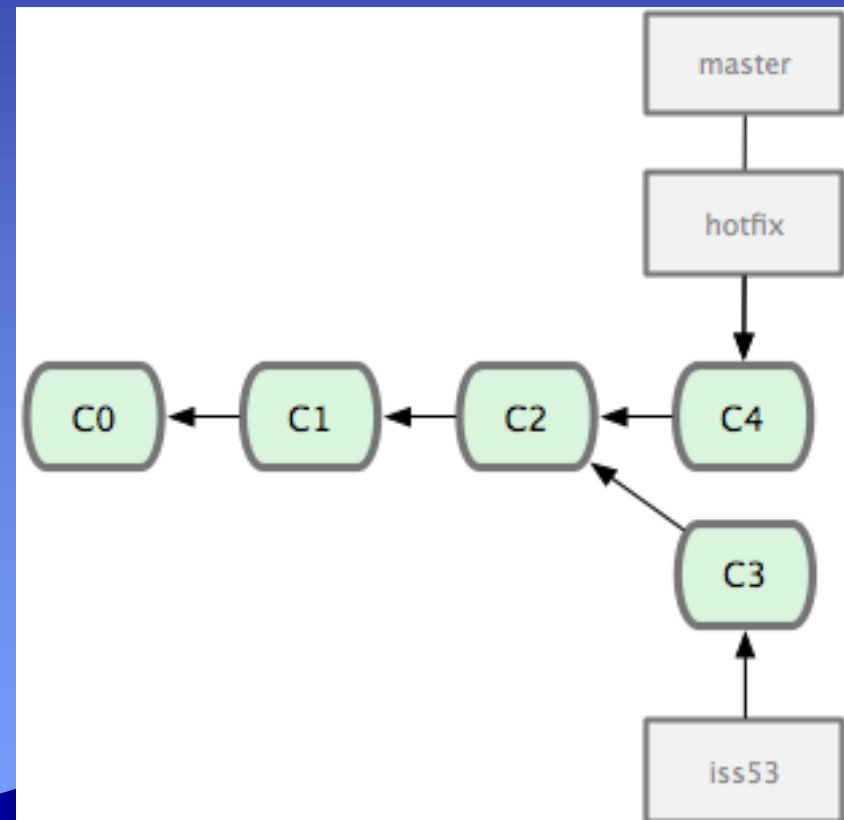
# Introduction GIT

- git checkout master
- git checkout -b hotfix

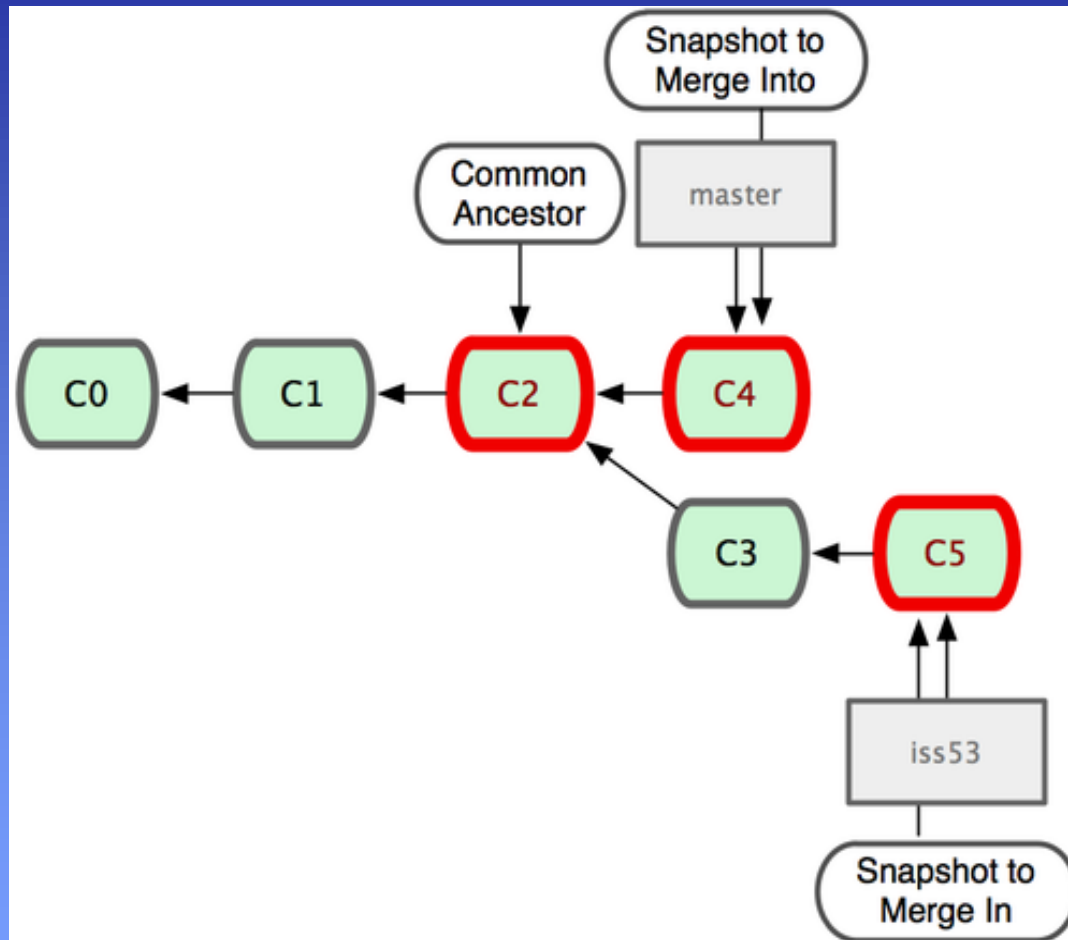


# Introduction GIT

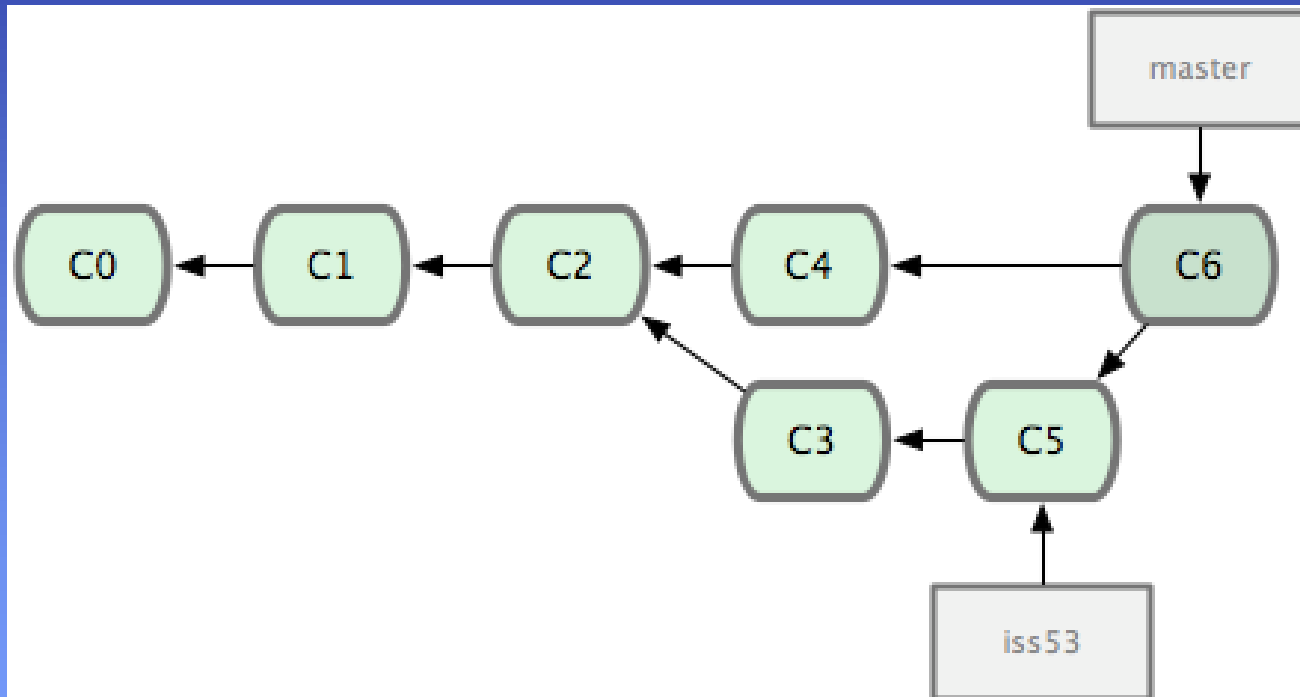
- `$ git checkout master`
- `$ git merge hotfix`
- `$ git branch -D hotfix`



# Introduction GIT



# Introduction GIT





# Introduction GIT

- Conflict
  - git status
  - edit the conflict files
  - git commit

```
<<<<<<<HEAD:test.c
```

```
=====
```

```
>>>>>>>iss53:test.c
```

