

GIT

THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.

COOL. HOW DO WE USE IT?

NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.



VCS -> CVCS -> DVCS

advantages/differences of git

- decentralized
- snapshots, not differences
- operations are local
 - entire history is local, working on trains/airplanes -> off the grid
- integrity through hashing
 - stores everything by hash not filename
- hard to loose data

configuration

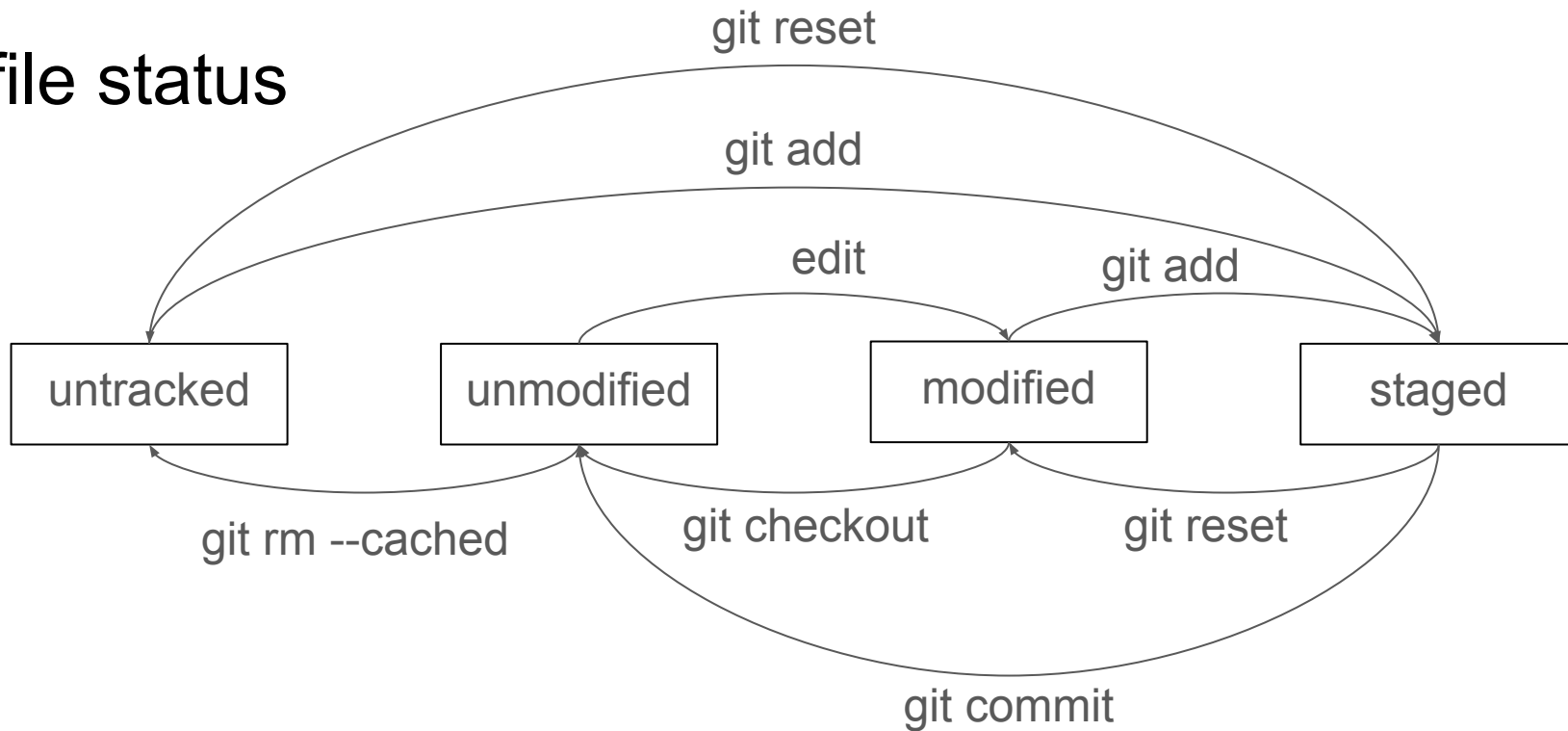
git config (--global | --system) user.name "Flo Sattler"

- /etc/gitconfig | system wide
- ~/.gitconfig | user wide
- .git/config | project wide

- user.name
- user.email
- core.editor vim
- commit.template
- core.excludesfile ~/.gitignore_global
- alias.fancyLog "!tig"

git config --list

file status



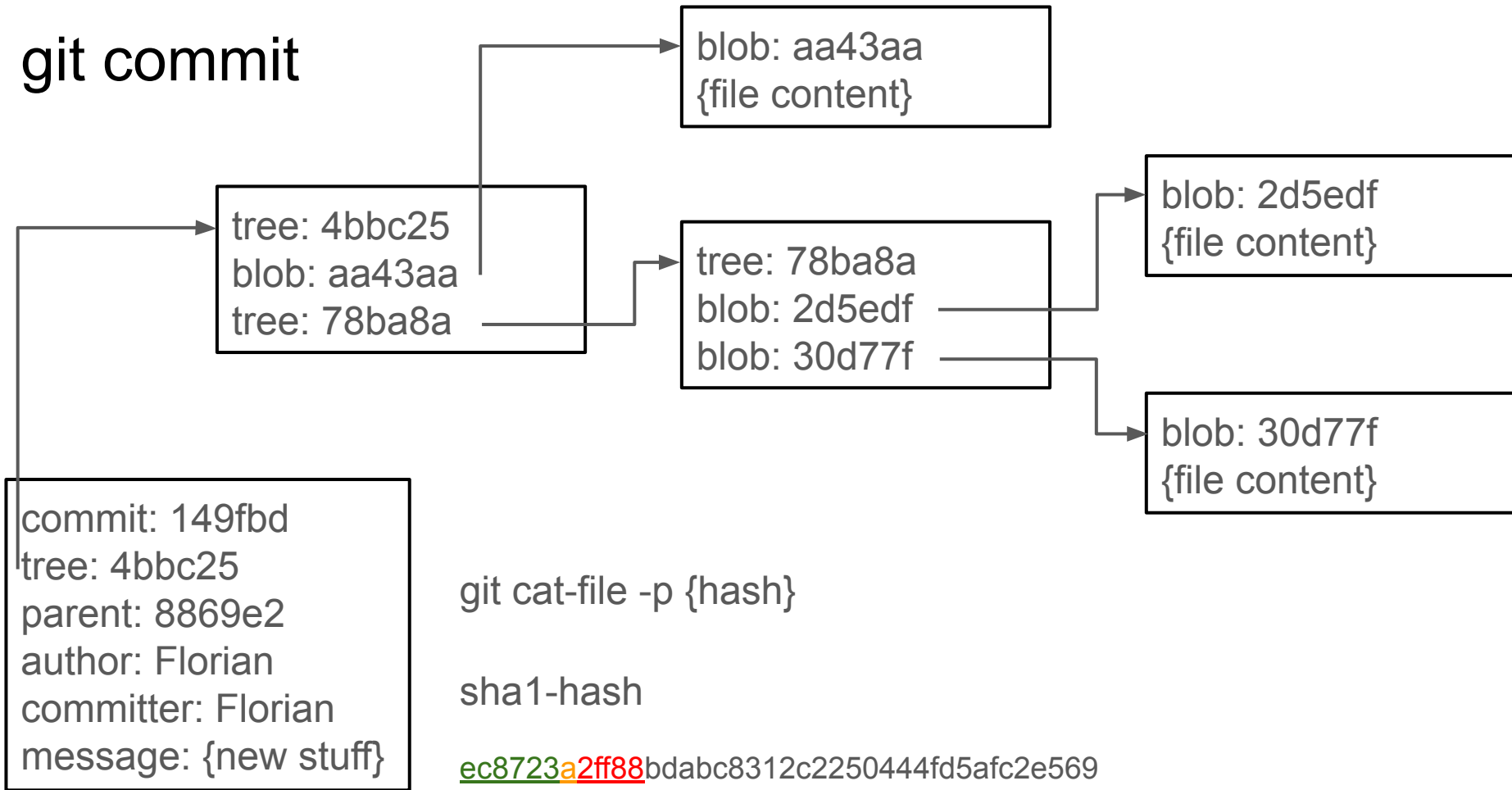
`git diff` | changed and not staged

`git diff --staged` | compares staged with last commit

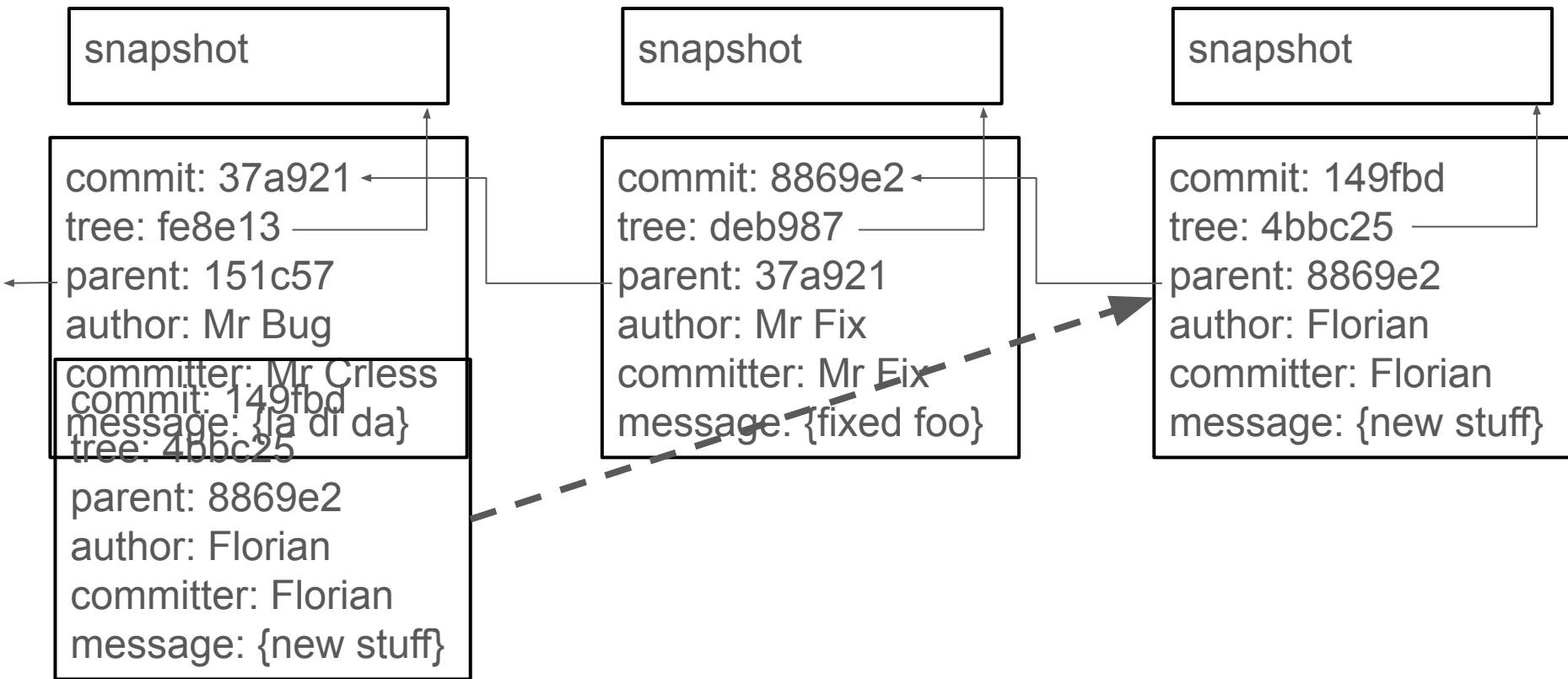
basic commands

- `git init` | `git clone <URL>`
- `git status`
- `git add` | start tracking/stage file
 - `-p` | patch mode, select specific hunks to add
- `git commit` | commit staged files
 - `-v` | show diff
 - `-m` | message
 - `-a` | stage all
- `git rm <file>` | remove file from tracking
 - `--cached` | keep file
- `git mv <file> <file>` | mv/rename file (mv, git rm, git add)
- `git reset [--hard] HEAD` | reset HEAD to different state

git commit



The history





[mips][microMIPS] Add CodeGen support for ADD, ADDIU*, ADDU* and DADD...

...



bd8d19a



Zlatko Buljan committed 8 days ago

...* instructions

Differential Revision: <http://reviews.llvm.org/D16454>

git-svn-id: <https://llvm.org/svn/llvm-project/llvm/trunk@265772> 91177308-0d34-0410-b5e6-96231b3b80d8



Use std::fill to simplify some code. NFC

...



b033fbe



Craig Topper committed 8 days ago

git-svn-id: <https://llvm.org/svn/llvm-project/llvm/trunk@265771> 91177308-0d34-0410-b5e6-96231b3b80d8



[IFUNC] Fix ifunc-asm.ll test

...



b34a81f



Dmitry Polukhin committed 8 days ago

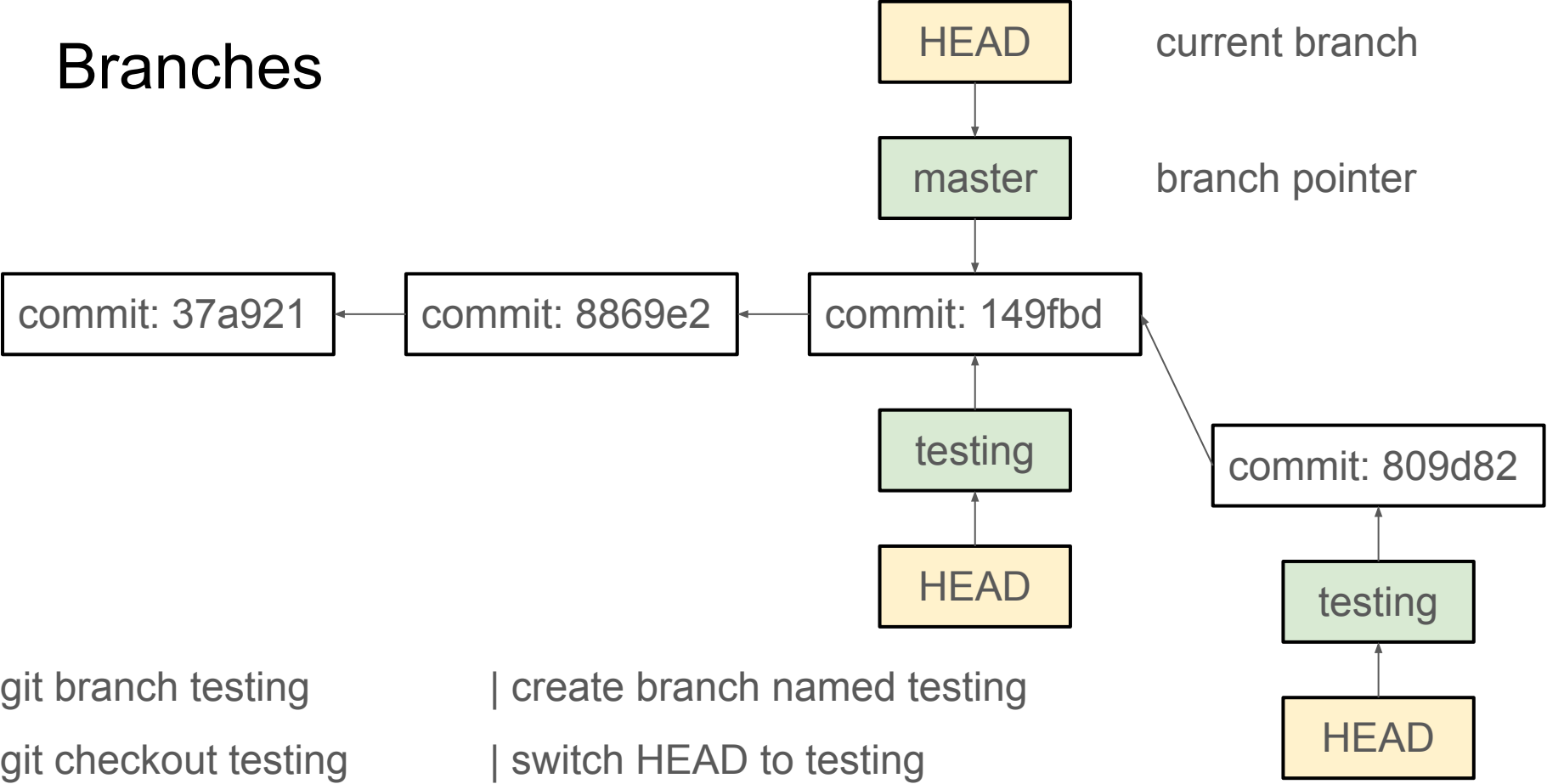
It seems that llc cannot be called used in assembler tests so test that checks asm for particular target needs to be moved to codegen.

git-svn-id: <https://llvm.org/svn/llvm-project/llvm/trunk@265770> 91177308-0d34-0410-b5e6-96231b3b80d8

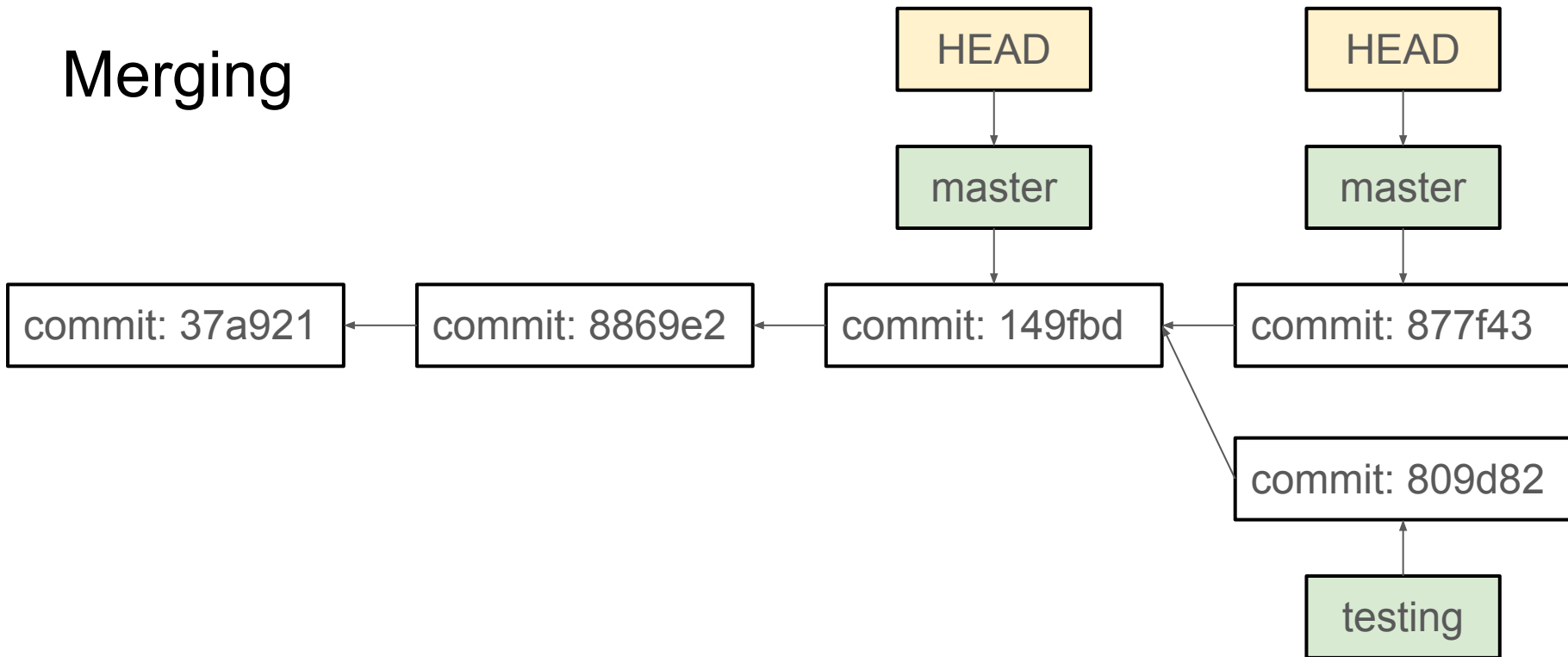
The history (git log)

- `-p` | show diff
- `-<n>` | last n entries
- `--stat` | states, like insertions/deletions
- `--pretty=` | formatting
 - `oneline`
 - `short`
 - `fuller`
 - `format:""`
- `--graph`
- `--since/until=3.weeks` | time range
- `-S<string>` | shows commits that add/delete the string

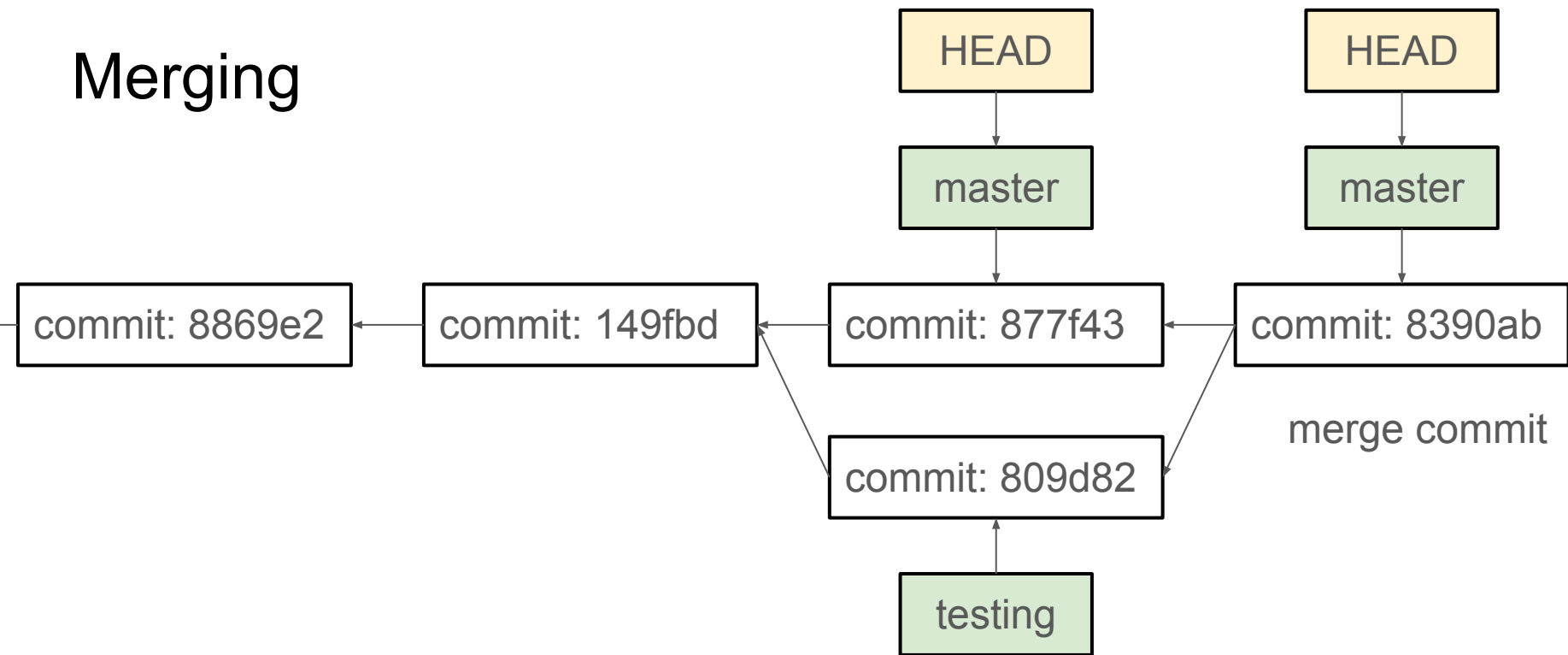
Branches



Merging



Merging



git merge testing

| merge branch testing into master

remote interaction

- git remote
 - -v | show URLs
 - add [r-name] [url] | add new remote
 - show [r-name] | shows infos of remote
 - rm [r-name] | remove remote
 - rename [r-old] [r-new] | local rename name of the remote
- git fetch [r-name] | gets objects/refs from remote
- git pull | gets o/r from remote and merges them into own
- git push [r-name] [branch] | pushes local o/r to remote

origin ssh://gitblit/~vulder/foo.git (fetch)

origin ssh://gitblit/~vulder/foo.git (push)

github git@github.com:vulder/foo.git (fetch)

github git@github.com:vulder/foo.git (push)

remote interaction

cloning/push/pull:

- https with logging
 - git config credential.helper 'cache [opts]'

```
git config credential.helper cache --timeout=300
git credential-cache exit
```

```
github https://github.com/vulder/foo.git (fetch)
github https://github.com/vulder/foo.git (push)
```

- ssh with key auth
 - add pub key to server
 - .ssh/config +priv key

```
Host github.com
IdentityFile ~/.ssh/id_rsa
```

```
github git@github.com:vulder/foo.git (fetch)
github git@github.com:vulder/foo.git (push)
```

remote branches

client

origin git@github.com:vulder/foo.git

branches: (git branch -a)

- master
- remotes/origin/HEAD -> origin/master
- remotes/origin/master
- remotes/origin/testing

git branch testing origin/testing

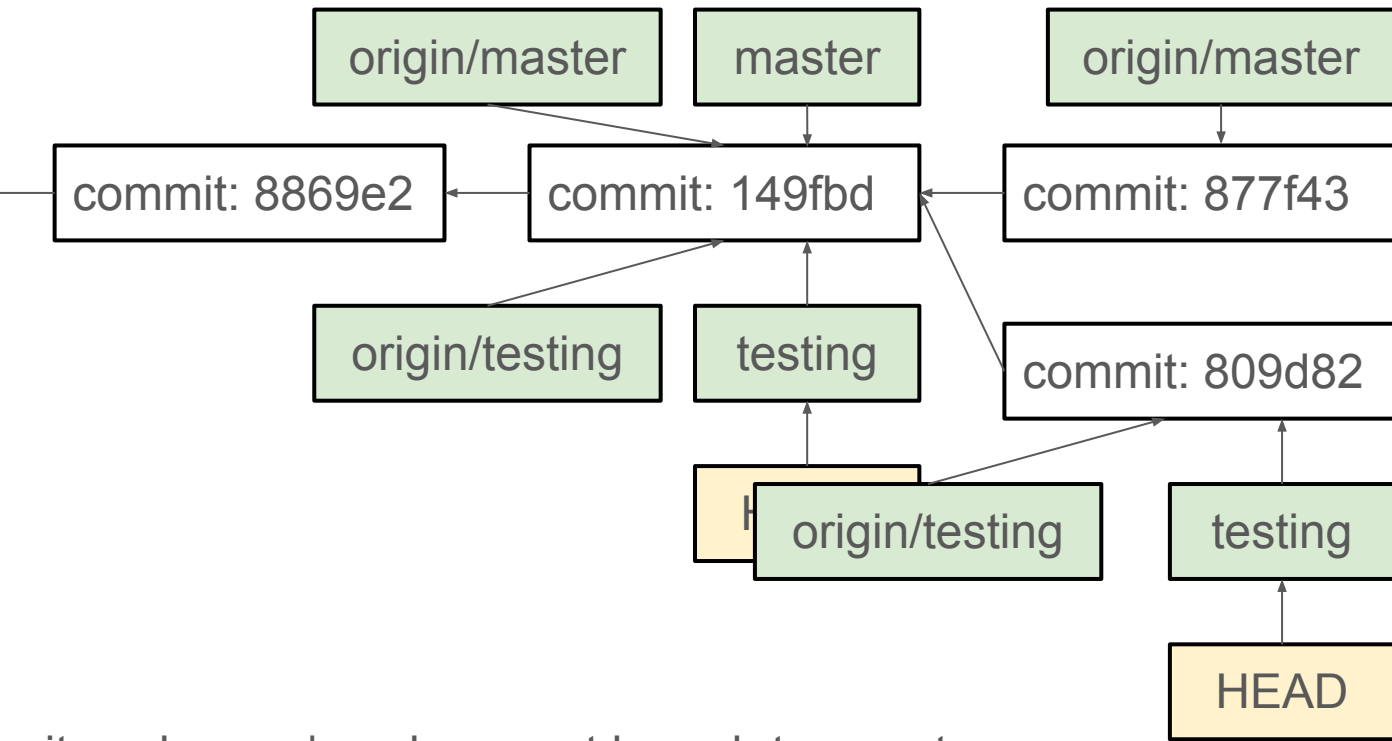
branches: (git branch -vv)

- master 149fbd [origin/master] {cmessage}
- testing 149fbd [origin/testing] {cmessage}

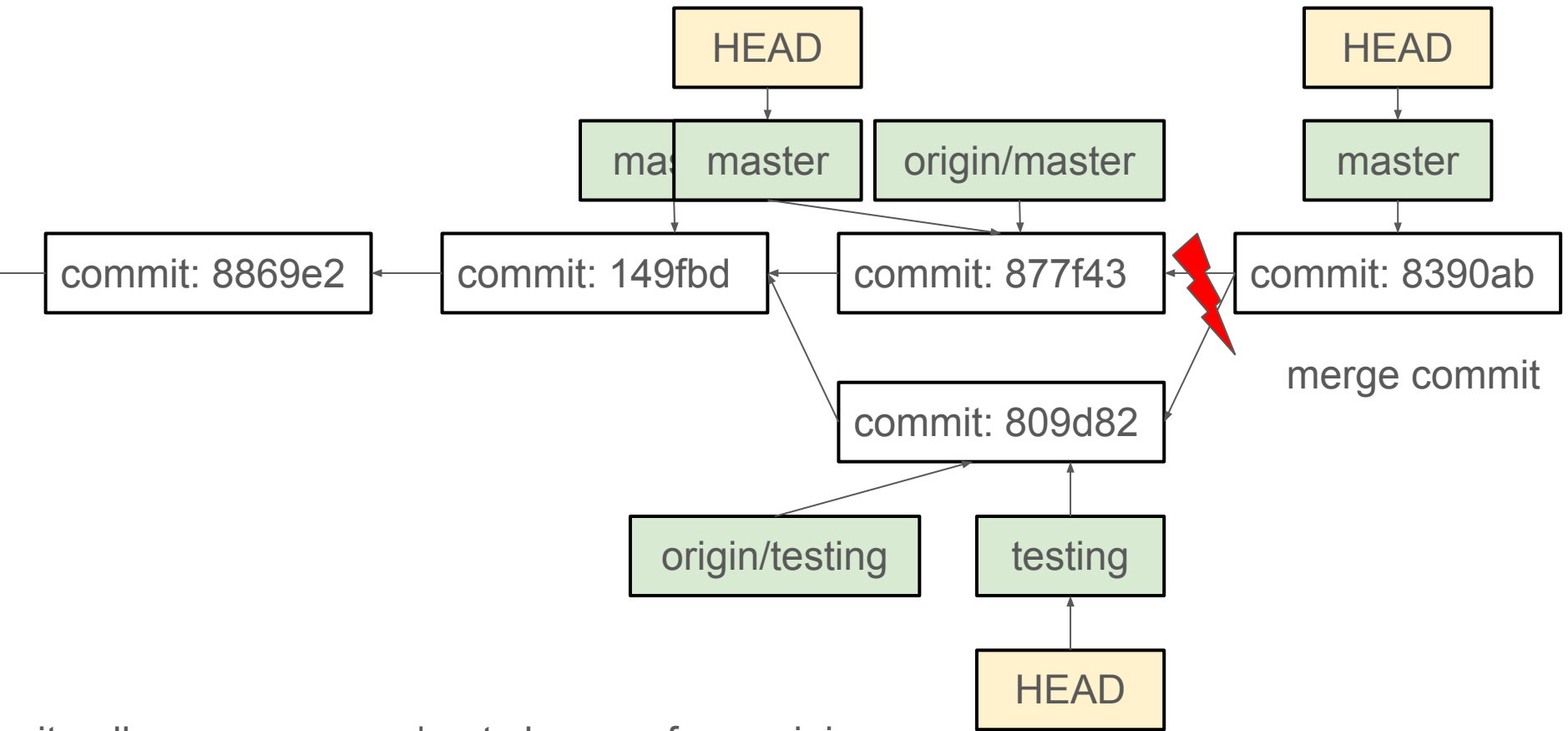
github.com

branches:

- master
- testing



git push | push current branch to remote



merge commit

git pull
git checkout master
git merge testing

| get changes from origin
| switch to brache master
| merge testing into master

MERGE CONFLICT?

**WHY SHOULD THAT
MATTER?**

resolving conflicts (git merge testing)

git mergetool -t diffuse

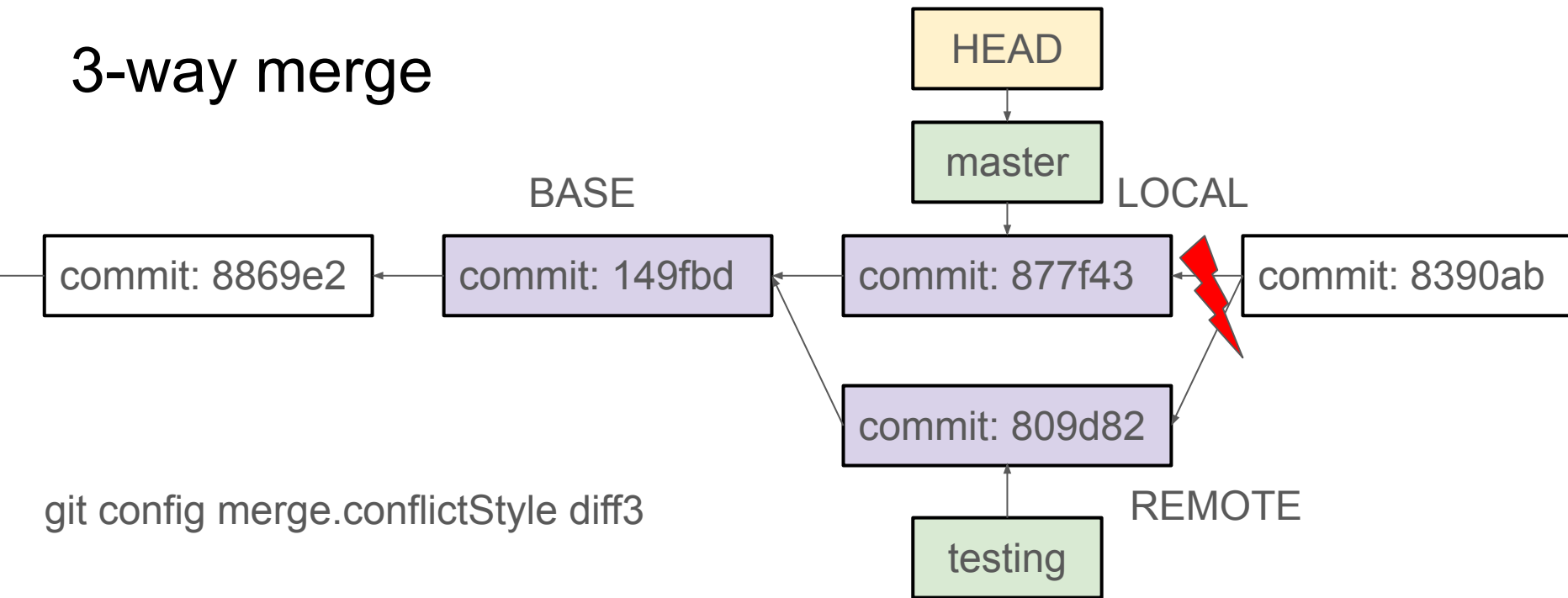
```
foo... foo_master: foo_test... foo_test... buffers
1 void funcno(int a) {
2 <<<<<< HEAD
3     std::cout << "print some numbers" << a << std::endl;
4 =====
5     std::cout << "awesome stuff" << std::endl;
6 >>>>>> test
7 }
```

./foo_LOCAL_11635.c	foo.c *	./foo_REMOTE_11635.c
1 void funcno(int a) {	1 void funcno(int a) {	1 void funcno(int a) {
2 std::cout << "print some numbers" << a <<	2	2
3 }	3 std::cout << "print some numbers" << a <<	2 std::cout << "awesome stuff" << std::endl
	4	3 }
	5	
	6	
	7 }	

master
git commit

testing

3-way merge



git config merge.conflictStyle diff3

./foo_LOCAL_12818.c	foo.c	./foo_REMOTE_12818.c	./foo_BASE_12818.c
<pre>1 void functo(int a) { 2 std::cout << "print some numbers" << a << 3 }</pre>	<pre>1 void functo(int a) { 2 <<<<<< HEAD 3 std::cout << "print some numbers" << a << 4 ===== 5 std::cout << "awesome stuff" << std::endl 6 >>>>>> test 7 }</pre>	<pre>1 void functo(int a) { 2 std::cout << "awesome stuff" << std::endl 3 }</pre>	<pre>1 void functo(int a) { 2 3 }</pre>

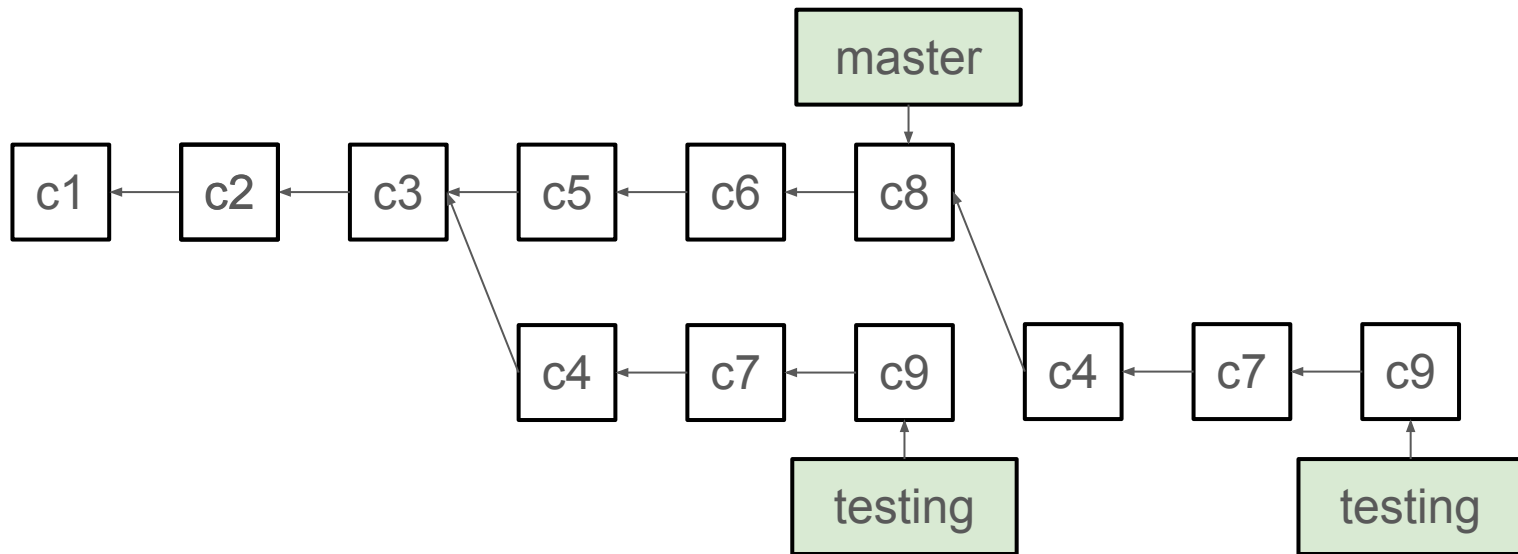
git merge

- `-no-commit` | merge but not autocommit
- `--ff-only` | only merge if fast forward or HEAD up to date
- `--no-ff` | create merge commit
- `-S` | GPG sign
- `--squash` | merge, create working tree and update state, but don't commit

when git turns into guitar hero

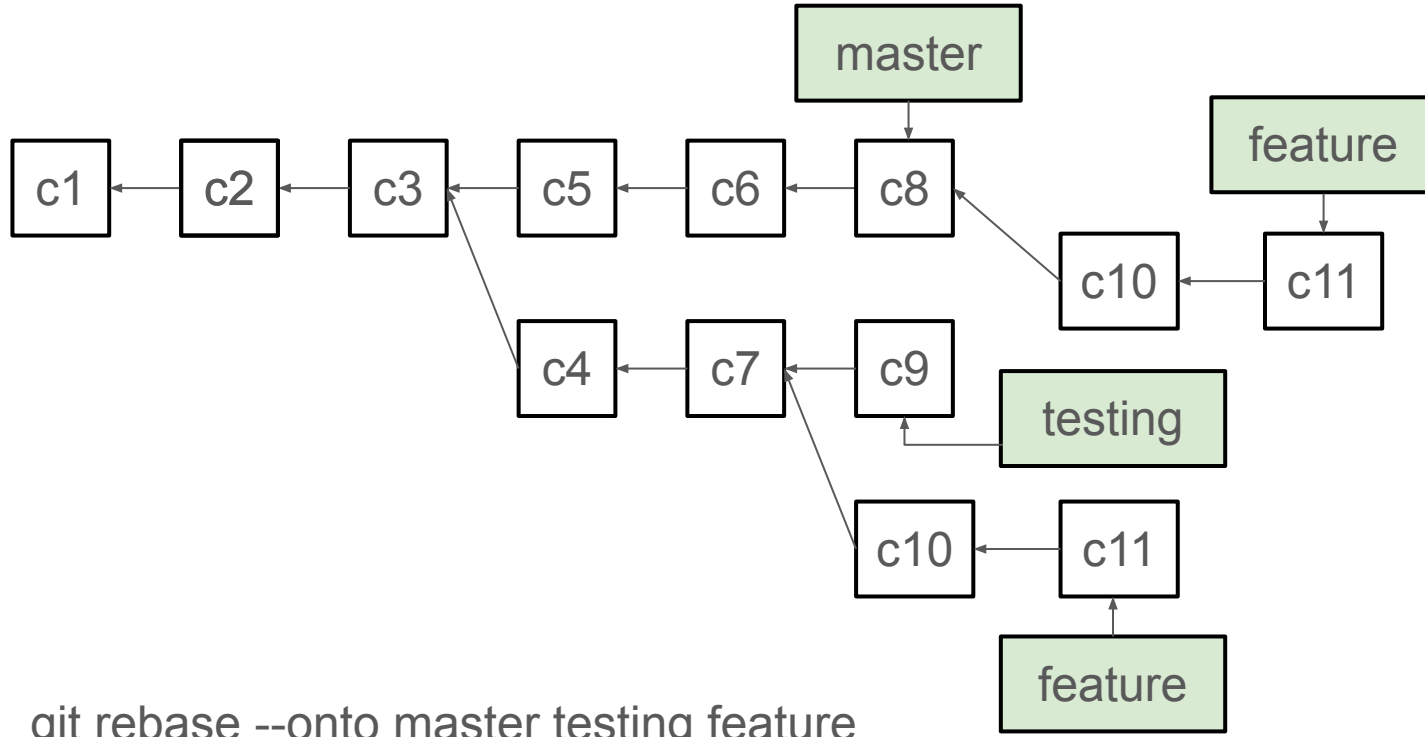


rebase

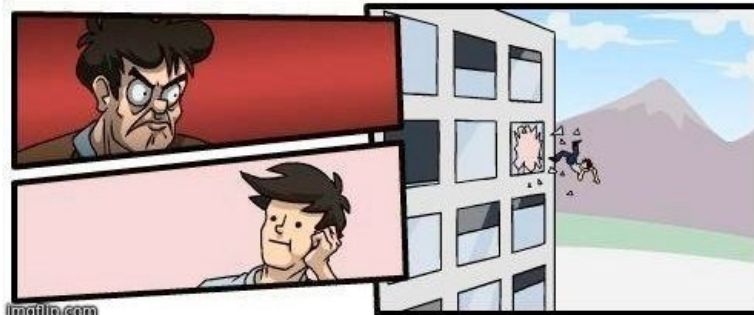


git rebase master

rebase II



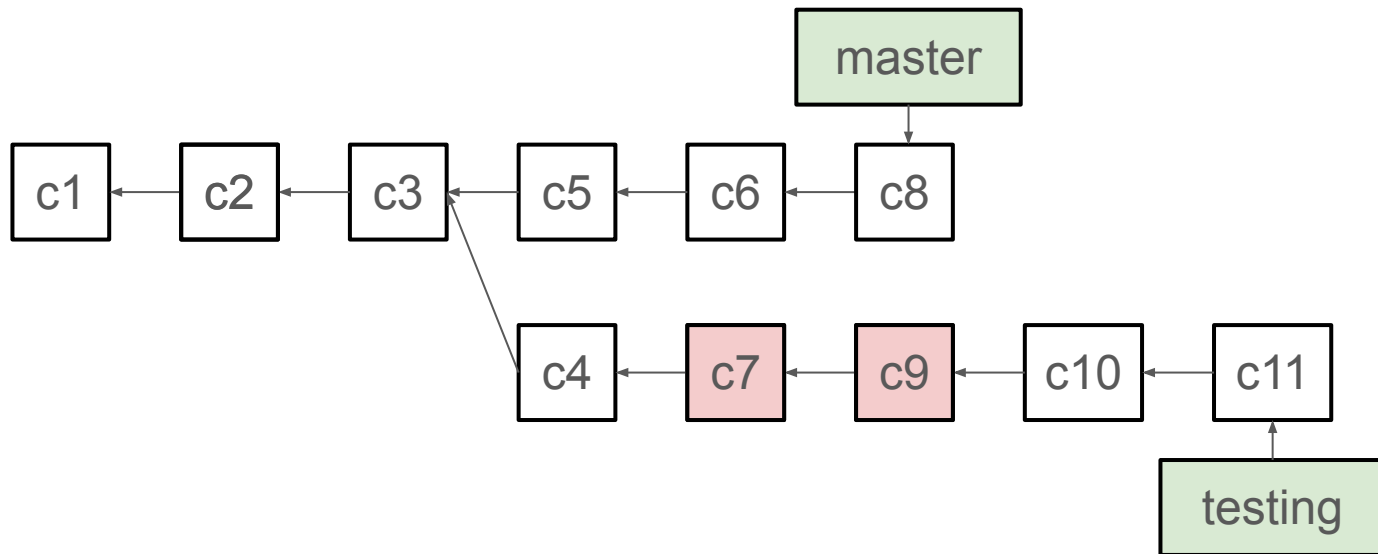
git rebase --onto master testing feature



git-rerere:
Reuse recorded resolution of conflicted merges

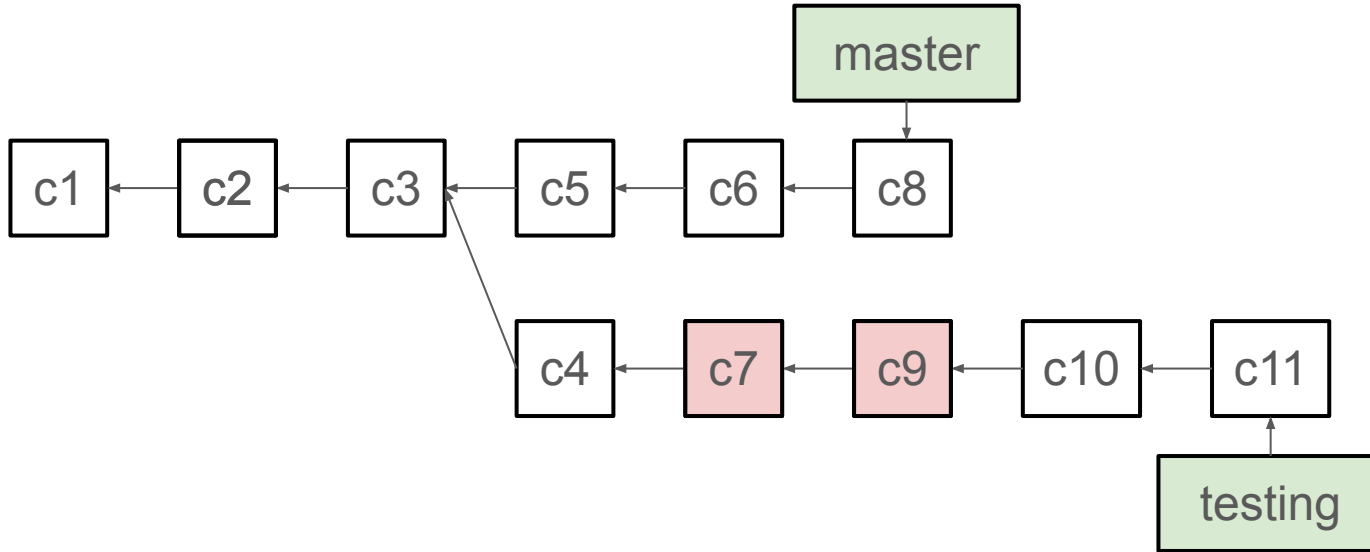
git config --global rerere.enabled 1

rebase III



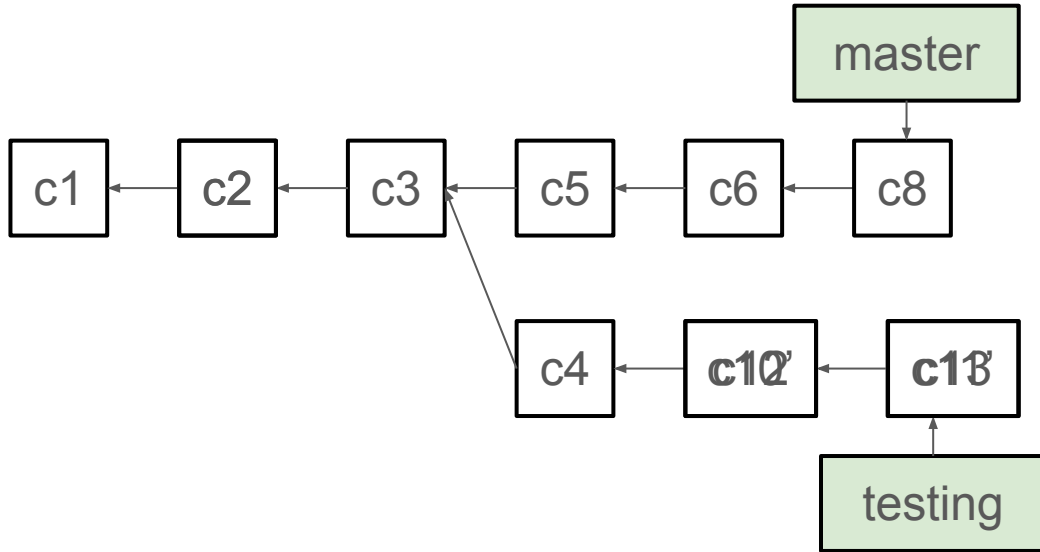
```
git rebase --onto testing~4 testing~2 testing
```

rebase



`git rebase --onto testing~4 testing~2 testing`

rebase



git rebase --onto testing~4 testing~2 testing

rebase interactive

git rebase -i master~3

master

a8cceb added otherfunc
ab1f13 asd
ade99d added foo
c2014c initial commit

.g/r/git-rebase-todo

buffers

```
1 pick 1efb7f6 added foo
2 reword 5b3571c asd
3 pick ef48348 added otherfunc
4
5 # Rebase 4b1280e..ef48348 onto 4b1280e (3 command(s))
6 #
7 # Commands:
8 # p, pick = use commit
9 # r, reword = use commit, but edit the commit message
10 # e, edit = use commit, but stop for amending
11 # s, squash = use commit, but meld into previous commit
12 # f, fixup = like "squash", but discard this commit's log message
13 # x, exec = run command (the rest of the line) using shell
14 # d, drop = remove commit
15 #
```

NORMAL > </rebase-merge/git-rebase-todo

gitrebase

4% 1: 1

rebase

git rebase add

git rebase --continue

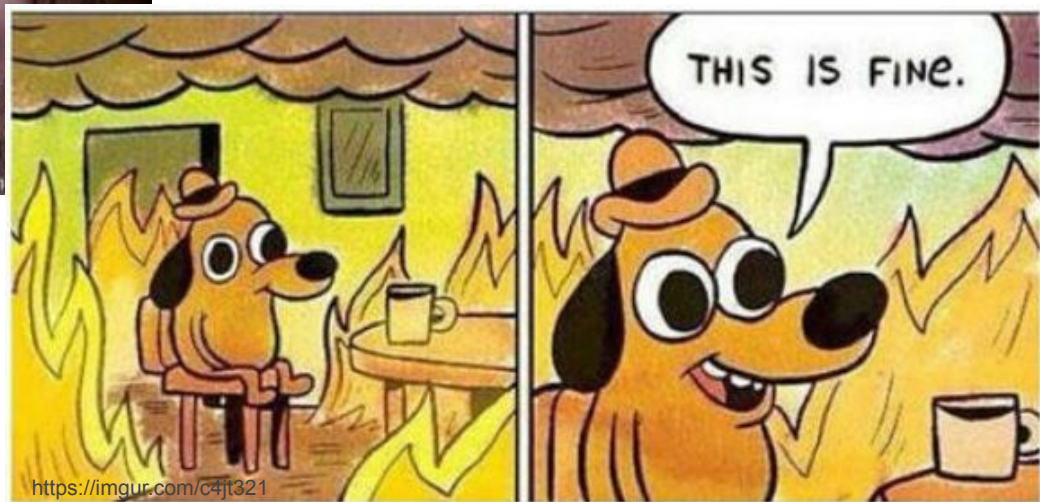
git rebase --abort

**PUSH REJECTED, REBASE OR
MERGE**



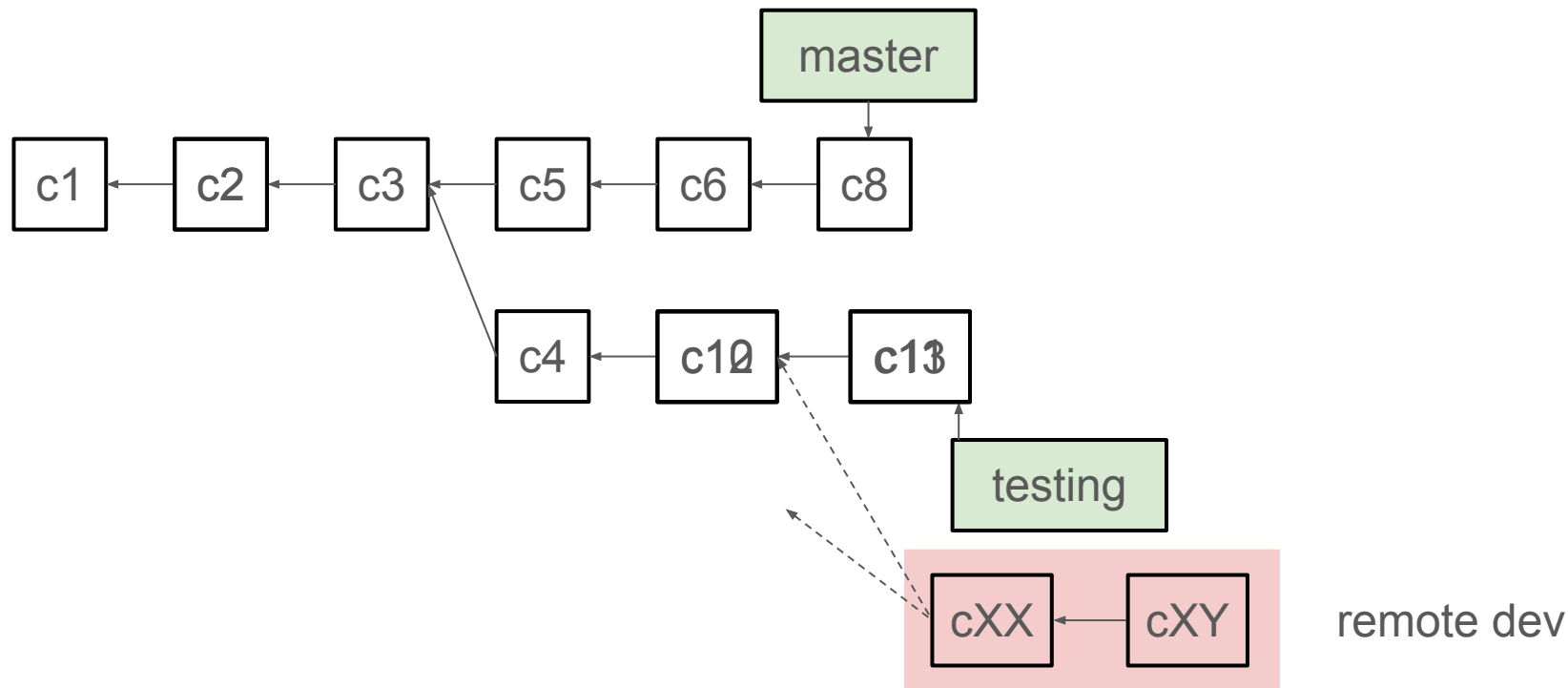
GIT PUSH --FORCE

<https://imgur.com/XFQLB>



<https://imgur.com/c4jt321>

rebase problem



```
git config pull.rebase true
```


YOU SHALL NOT



REBASE PUSHED COMMITS

git reflog

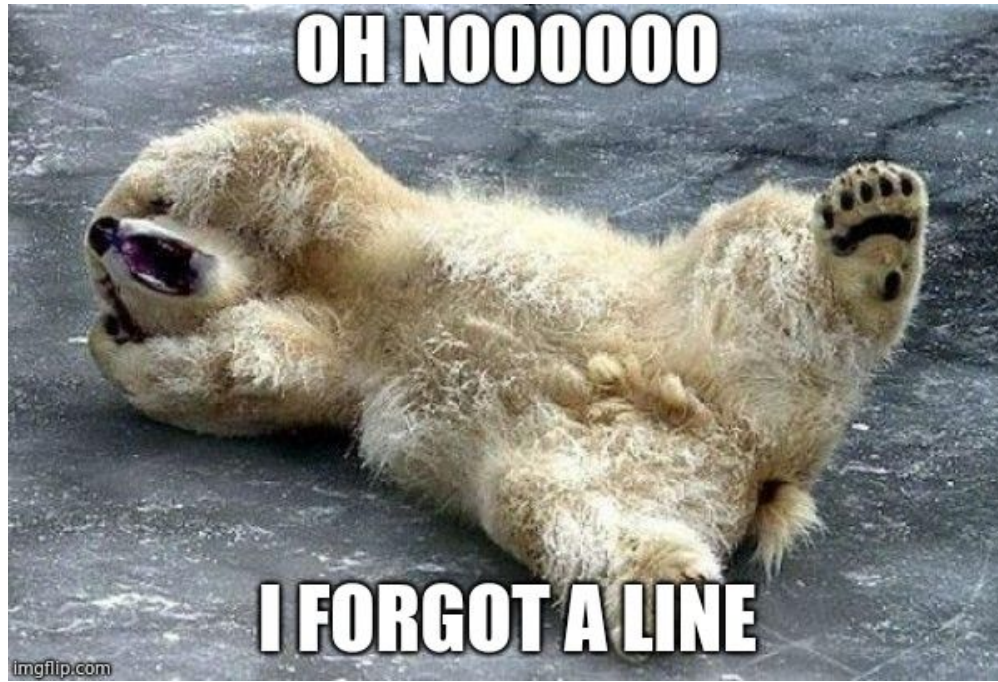
logs the states where branches/HEAD was before a change occurred

```
f0c4558 HEAD@{0}: reset: moving to HEAD~  
c38d994 HEAD@{1}: checkout: moving from foo to master  
aea74ad HEAD@{2}: checkout: moving from master to foo
```

git reset --hard c38d994

repo state before “reset --hard HEAD~”

```
45c38e7 refs/heads/pp@{0}: commit: function  
d47bdcd refs/heads/pp@{1}: commit: fxxxx  
02b3d9c refs/stash@{0}: WIP on master: f0c4558 foo bar  
09046c5 refs/stash@{1}: WIP on master: f0c4558 foo bar  
f0c4558 refs/heads/master@{0}: reset: moving to HEAD~  
c38d994 refs/heads/master@{1}: reset: moving to c38d994
```



```
git commit --fixup <hash>
```

...

```
git rebase --interactive --autosquash master
```

tags (git tag)

lightweight tag: `git tag [new tag name]`

annotated tag: `git tag -a [tag] -m [message]`

+ sign tag with -s

default email key

Conf: `user.signingKey = [keyid]`

`git push [remote] [tag-name]`

`git push [remote] --tags`

Comparing changes

Choose two branches to see what's changed or to start a pull request



base fork: **llvm-mirror/llvm**

base: **release_3**

✗ Can't automatically merge. Don't worry, you can



Create pull request

Discuss and review the changes

Commits 3,739

Files changed 5,044

This comparison



Commits on Apr 04, 2016

- rotateright fix document
- Teresa Johnson [ThinLT0] Au
- Teresa Johnson [ThinLT0] Ac
- MatzeB ARM, AArch64
- Betul Buyukkurt [PG0] Avoid
- chandlerc Revert r2634

Merged **lins05** merged 2 commits into **haiwen:master** from **unknown repository** on May 6, 2015

Conversation 3

Commits 2

Files changed 1



vulder commented on May 5, 2015



Closes #545.



fixed applet jumping back to old location after tag switch

✓ 40223fc

Closes #545.



Chilledheart and 1 other commented on an outdated diff on May 6, 2015

Show 2 comments



fixed spaces

✓ aeaa5ab



Chilledheart commented on May 6, 2015



LGTM



lins05 merged commit **095d236** into **haiwen:master** on May 6, 2015

Hide details

1 check passed

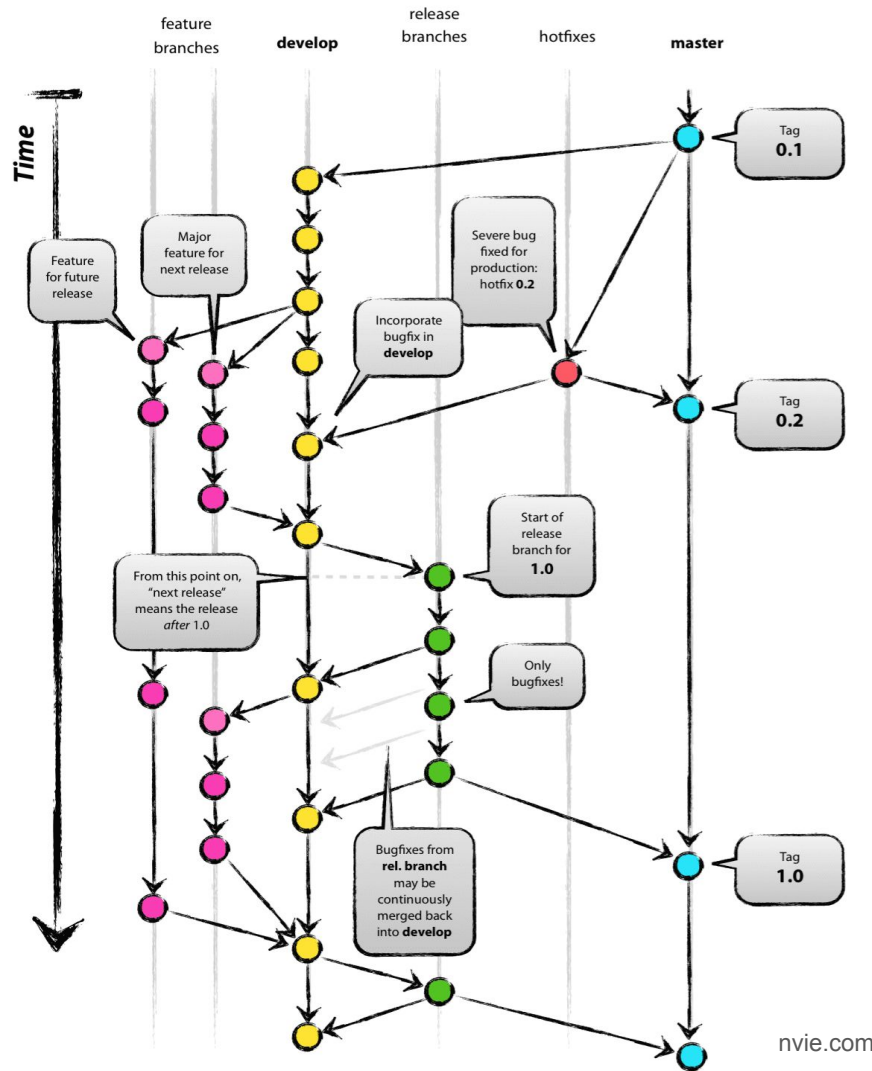
✓ **continuous-integration/travis-ci/pr** The Travis CI build passed

Details

Workflow: git flow

(<https://github.com/nvie/gitflow>)

- master
- develop
- feature
- release
- hotfix

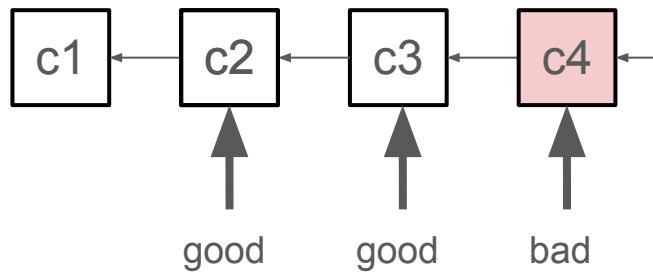


git stash stash@{0}

- list | list all stashed states
 - show <stash> | show state of stash
 - pop/apply | {remove} state and apply it to workdir
 - branch <stash> | create branch where stash was recorded + apply
 - drop <stash> | drop state from stash
-
- -u | include untracked files
 - -p | patch

git stash show -p stash@{0} | git apply -R

git bisect



git bisect

- start | start git bisect
- good | mark commit as good
- bad | mark commit as bad
- skip | skip commit
- reset | end git bisect
- visualize | show remaining commits
- log | show bisect log
- replay (log -> reset -> replay) | replay bisect log
- run {script} | run script to check if commit is good/bad

tool: tig

```

2015-06-19 23:20 Sanjoy Das lib/Analysis/IPA/CallGraph.cpp 152f3b5 if (Function *LF =
2015-06-19 23:20 Sanjoy Das lib/Analysis/IPA/CallGraph.cpp 152f3b5 if (Function *RF
2015-06-19 23:20 Sanjoy Das lib/Analysis/IPA/CallGraph.cpp 152f3b5 return LF->getN
2015-06-19 23:20 Sanjoy Das lib/Analysis/IPA/CallGraph.cpp 152f3b5
2015-06-19 23:20 Sanjoy Das lib/Analysis/IPA/CallGraph.cpp 152f3b5 return RHS->getFunc
2015-06-19 23:20 Sanjoy Das lib/Analysis/IPA/CallGraph.cpp 152f3b5 });
2015-06-19 23:20 Sanjoy Das lib/Analysis/IPA/CallGraph.cpp 152f3b5
2015-06-19 23:20 Sanjoy Das lib/Analysis/IPA/CallGraph.cpp 152f3b5 for (CallGraphNode *C
2015-06-19 23:20 Sanjoy Das lib/Analysis/IPA/CallGraph.cpp 152f3b5 CN->print(OS);
2004-08-08 03:27 Chris Lattner lib/Analysis/IPA/CallGraph.cpp af8a424 }
2013-11-26 04:19 Chandler Carruth lib/Analysis/IPA/CallGraph.cpp 54fec07
2015-11-23 02:58 Davide Italiano lib/Analysis/CallGraph.cpp a7e61b0 LLVM_DUMP_METHOD
2013-11-26 04:19 Chandler Carruth lib/Analysis/IPA/CallGraph.cpp 54fec07 void CallGraph::dump()
2004-08-08 03:27 Chris Lattner lib/Analysis/IPA/CallGraph.cpp af8a424
2002-07-18 04:43 Chris Lattner lib/Analysis/IPA/CallGraph.cpp d99d4d7 // removeFunctionFromMo
2002-07-18 04:43 Chris Lattner lib/Analysis/IPA/CallGraph.cpp d99d4d7 // it. Because this re
2002-07-18 04:43 Chris Lattner lib/Analysis/IPA/CallGraph.cpp d99d4d7 // is destroyed. This
2002-07-18 04:43 Chris Lattner lib/Analysis/IPA/CallGraph.cpp d99d4d7 // functions (ie, there
2001-11-26 18:51 Chris Lattner lib/Analysis/IPA/CallGraph.cpp 25e9cad // is to dropAllReferen
2001-11-26 18:51 Chris Lattner lib/Analysis/IPA/CallGraph.cpp 25e9cad //
2002-07-18 04:43 Chris Lattner lib/Analysis/IPA/CallGraph.cpp d99d4d7 Function *CallGraph::re
2009-08-31 02:24 Chris Lattner lib/Analysis/IPA/CallGraph.cpp 2adb830 assert(CGN->empty()) &
2002-07-18 04:43 Chris Lattner lib/Analysis/IPA/CallGraph.cpp d99d4d7 "graph if it r
2003-08-31 20:36 Chris Lattner lib/Analysis/IPA/CallGraph.cpp 5714c97 Function *F = CGN->ge
2003-08-31 20:36 Chris Lattner lib/Analysis/IPA/CallGraph.cpp 5714c97 FunctionMap.erase(F);
2001-11-26 18:51 Chris Lattner lib/Analysis/IPA/CallGraph.cpp 25e9cad
2013-11-26 04:19 Chandler Carruth lib/Analysis/IPA/CallGraph.cpp 54fec07 M.getFunctionList().r
2003-08-31 20:36 Chris Lattner lib/Analysis/IPA/CallGraph.cpp 5714c97 return F;
2001-11-26 18:51 Chris Lattner lib/Analysis/IPA/CallGraph.cpp 25e9cad }
2001-11-26 18:51 Chris Lattner lib/Analysis/IPA/CallGraph.cpp 25e9cad

```

```
commit 54fec07ec00b4449393a66e8e2e62fd241781f80
```

Author: Chandler Carruth <chandlerc@gmail.com>

AuthorDate: Tue Nov 26 04:19:30 2013 +0000

CommitDate: Tue Nov 26 04:19:30 2013 +0000

```
[PM] Split the CallGraph out from the ModulePass which creates the
CallGraph.
```

This makes the CallGraph a totally generic analysis object that is the container for the graph data structure and the primary interface for querying and manipulating it. The pass logic is separated into its own class. For compatibility reasons, the pass provides wrapper methods for most of the methods on CallGraph -- they all just forward.

This will allow the new pass manager infrastructure to provide its own analysis pass that constructs the same CallGraph object and makes it available. The idea is that in the new pass manager, the analysis pass's 'run' method returns a concrete analysis 'result'. Here, that result is a 'CallGraph'. The 'run' method will typically do only minimal work, deferring much of the work into the implementation of the result object in order to be lazy about computing things, but when (like DomTree) there is *some* up-front computation, the analysis does it prior to handing the result back to the querying pass.

I know some of this is fairly ugly. I'm happy to change it around if folks can suggest a cleaner interim state, but there is going to be some amount of unavoidable ugliness during the transition period. The good thing is that this is very limited and will naturally go away when the old pass infrastructure goes away. It won't hang around to bother us

**YOU SAID IT WASN'T YOUR
FAULT**

**GIT BLAME DETERMINED THAT
WAS A LIE**

project multitasking -
Houston, we have a problem!



git worktree

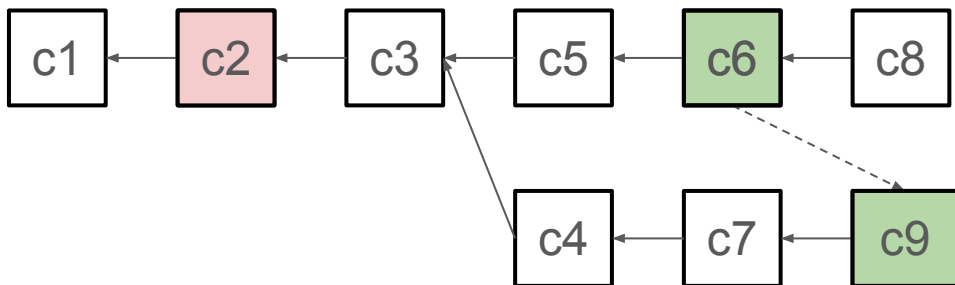
- `git worktree add <PATH> <HASH>`
- `git worktree list`
- `git worktree prune`

git apply / format-patch

1. create branch
2. fix bug
3. git format-patch master | creates patch file
 - a. git format-patch master --stdout
 - b. git format-patch master -o FOLDER
4. git send-mail FOLDER/* --to=foo@bar.org
5. git apply *.patch
 - a. --stat | info about patch
 - b. --check | check if patches applies cleanly
6. git am --signoff < 0001-foo.patch | apply patch

git cherry-pick

apply changes from existing commit

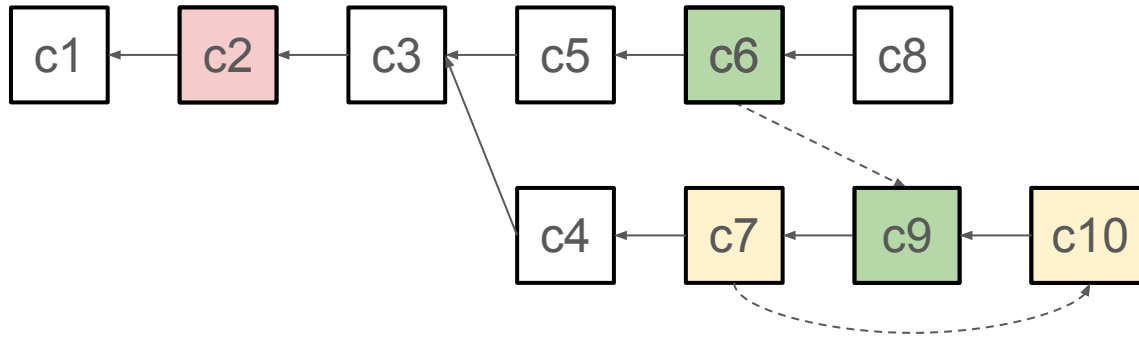


git cherry-pick c6

- -n | no commit
- -e | edit
- -x | add from line

git revert

reverte existing commit

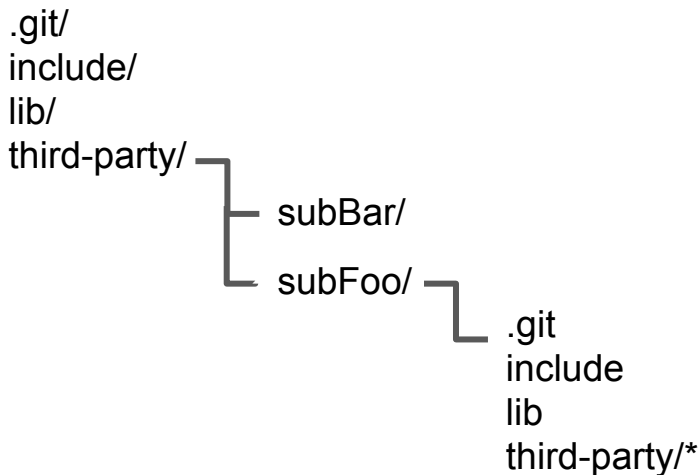


git revert c7

git revert -n master~5..master~2

- -n | no commit
- -e | edit

git submodule



- add <repo>
- status
- init/deinit
- update
- foreach <cmd>
 - --recursive

- | add new sub repo
- | status of submodule
- | initialize the submodule (register/deregister)
- | update registered submodule
- | exec cmd in all submodules
- | traverse recursively into sub...submodules

.git folder

- ❖ config | local configurations
- ❖ FETCH_HEAD | head updated by last fetch
- ❖ HEAD | current ref
- ❖ index | staging area, with meta info
- ❖ hooks | hooks
 - pre-commit
 - commit-msg
- ❖ logs
- ❖ objects | location to store all blobs
- ❖ refs | all refs of the repo
 - heads
 - remotes
 - stash
 - tags

hooks (.git/hooks)

- pre-commit
- commit-msg
- pre-push
- pre-receive/up

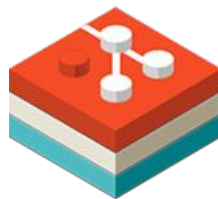
h/commit-msg

```
1 #!/usr/bin/env python
2 import sys, os
3
4 print(sys.argv[1])
5 with open(sys.argv[1]) as f:
6     for line in f.readlines():
7         print(line)
8         if "foo" in line:
9             f.write("bar\n")
10             sys.exit(0)
11 sys.exit(1)
```

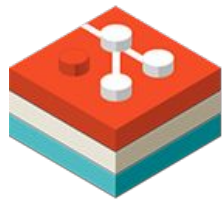
```
NORMAL +0 ~0 -0 ↵ master hooks/pre-push.sample sh utf-8[unix] 40% 22: 1
```

NORMAL > </commit-msg

this is not enough - let's



libgit2



libgit2

```
libgit2.py ➤  
from pygit2 import Repository  
from pygit2 import GIT_SORT_REVERSE  
  
repo = Repository('.git')  
for commit in repo.walk(repo.head.target, GIT_SORT_REVERSE):  
    print(commit.author.name)  
    print(commit.message)
```

Language Bindings



Ruby
Rugged



.Net & Mono
LibGit2Sharp



Objective-C
objective-git



Python
pygit2



Lua
luagit2



Perl
Git::Raw



Parrot VM
parrot-libgit2



Node.js
nodegit



C++ Qt
libqgit2



Go
git2go



PHP
php-git



Erlang
Geef



Chicken Scheme
chicken-git



GObject
libgit2-glib



D
dlibgit

git hosting

provided:

- github.com
- bitbucket.org

self hosted:

- gogs.io
- gitlab.com
- gitblit.com
- phabricator.org



vulder created this revision.

Dec 4 2015, 6:16 PM



vulder added a reviewer: **simbuerg**.



vulder added a subscriber: **vulder**.



simbuerg awarded a token.

Dec 4 2015, 6:52 PM



simbuerg accepted this revision.

Dec 4 2015, 6:58 PM



LGTM.



pprof/projects/gentoo/gentoo.py

45

Long line (ursprgl.) ? Das sieht komisch aus. Aber egal, pylint wirds schon wissen

View Options ▾

the portage tree.

oup:



This revision is now accepted and ready to land.

Dec 4 2015, 6:58 PM



Closed by commit **rPPROF3c9442840b16: added class documentation** (authored by **vulder**). · Explain Why

Dec 4 2015, 7:00 PM

Show All 12 Lines

25

```

24     src_uri = (curl[latest_txt] | tail["-n", "+3"]
25                 | cut["-f1", "-d "])().strip()
26 except ProcessExecutionError as proc_ex:
27     src_uri = "NOT-FOUND"
28     error("Could not determine latest stage3 src uri: {0}",

```

try:

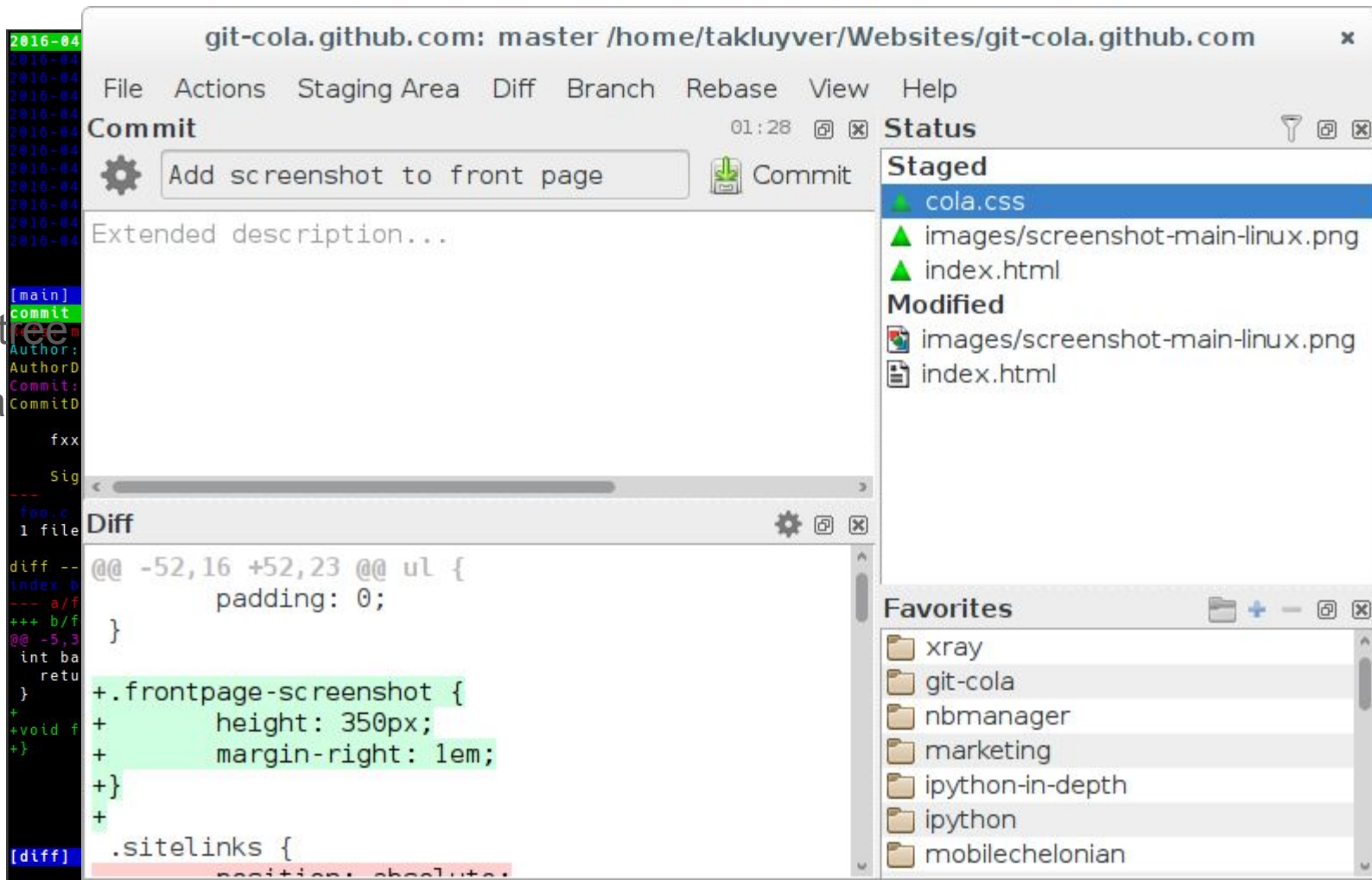
```

26     src_uri = (curl[latest_txt] | tail["-n", "+3"]
27                 | cut["-f1", "-d "])().strip()
28 except ProcessExecutionError as proc_ex:
29     src_uri = "NOT-FOUND"
30     error("Could not determine latest stage3 src uri: {0}",

```

tooling

- gitk
- tig
- sourcetree
- git-cola



tooling

- gitk
- tig
- sourcetree
- git-cola
- docx to text

gitattributes

.git/info/attributes

| path/file spec settings

```
*.docx diff=word
```

| word filter

```
1.  #!/bin/bash  
2.  docx2txt.pl $1 -
```

| script to call docx converter

git config diff.word.textconv docx2txt

infos

@ <https://git-scm.com/book/uz/v2/Customizing-Git-Git-Attributes>

@ <http://docx2txt.sourceforge.net/>

other stuff

- `git clean`
- `git grep`
- `read-tree + subtree`
- `git replace`
- `git fsck`
- `git bundle`
- `git log -L`
- `git describe`
- `git archive`

...



https://miro.medium.com/max/512/1*SptUKxY2gmxa4XC3W1_nRw.jpeg