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.
https://xkcd.com/1597/

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

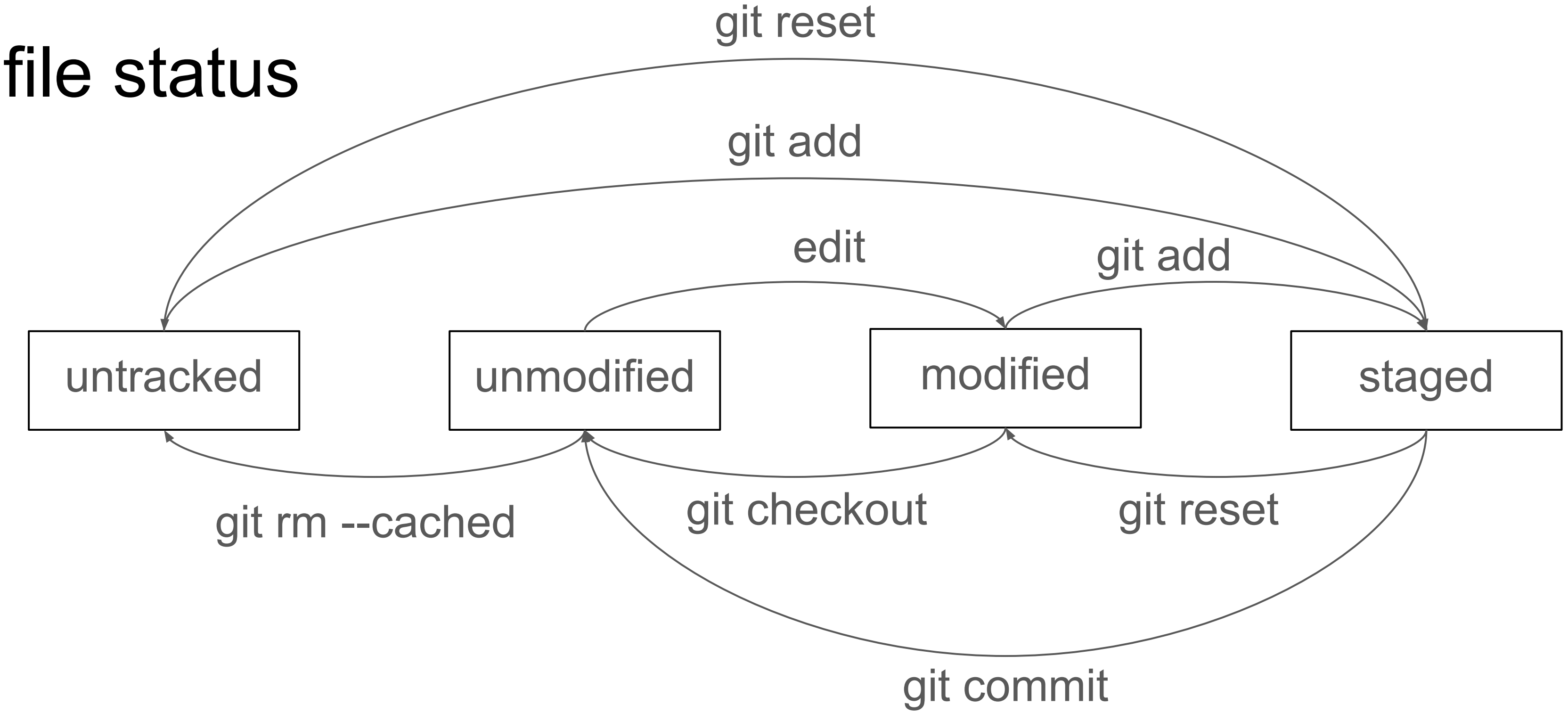
configuration
git config (--global | --system) user.name “Flo Sattler”

- /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

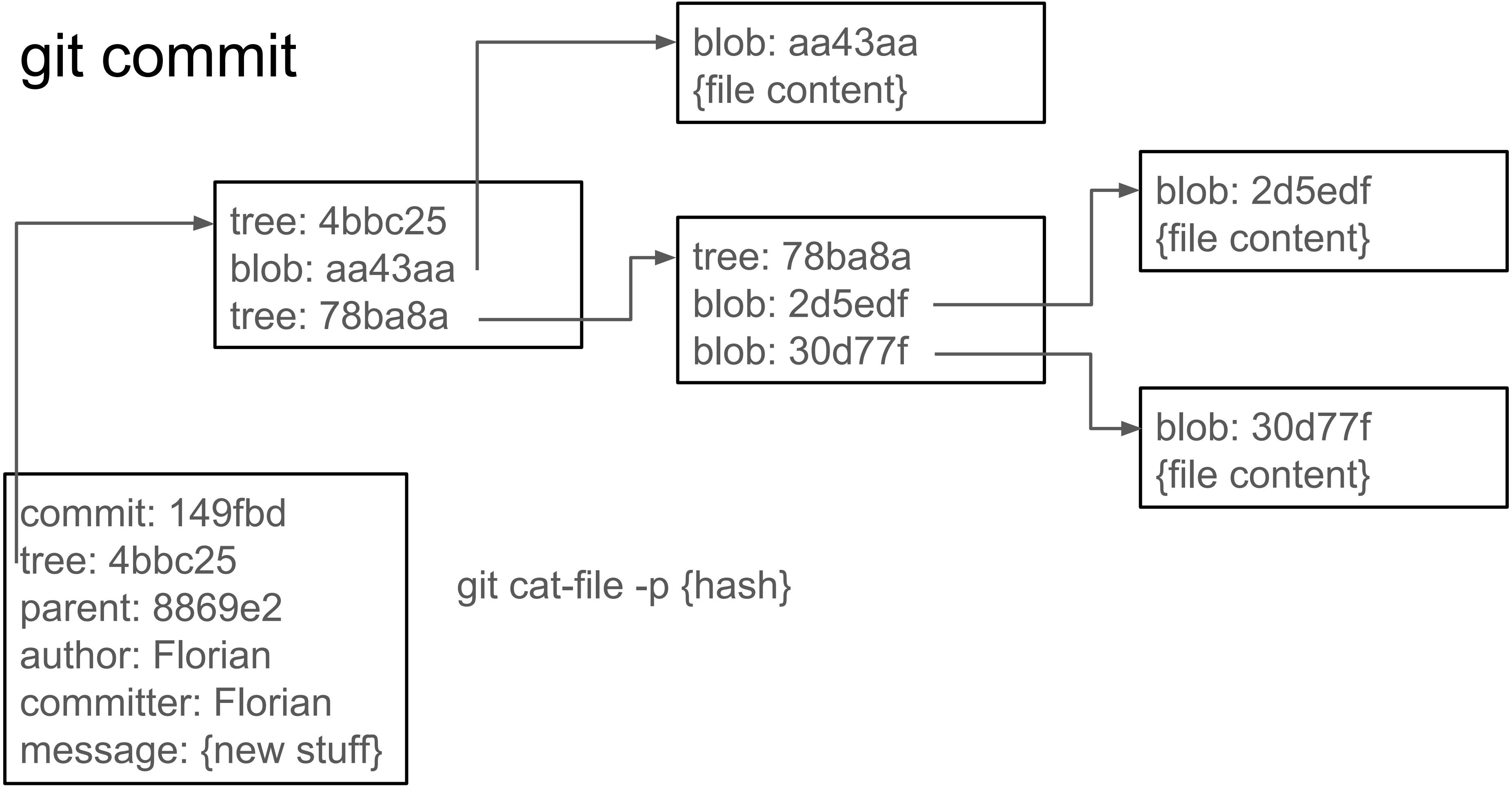


git diff	| changed and not staged
git diff --staged	| compares staged with last commit

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



commit: 149fbd
tree: 4bbc25
parent: 8869e2
author: Florian
committer: Florian
message: {new stuff}

tree: 4bbc25
blob: aa43aa
tree: 78ba8a

blob: aa43aa
{file content}

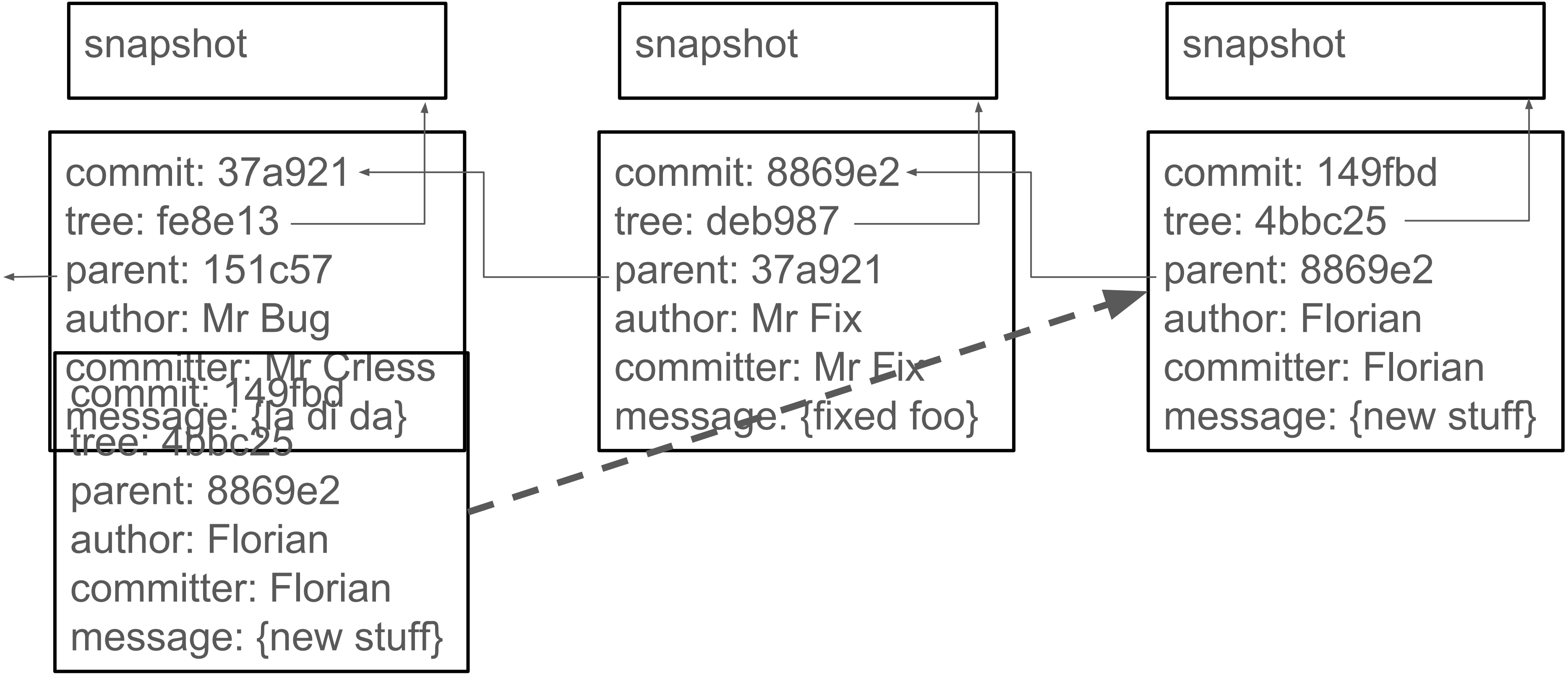
tree: 78ba8a
blob: 2d5edf
blob: 30d77f

blob: 2d5edf
{file content}

blob: 30d77f
{file content}

git cat-file -p {hash}

The history



snapshot

commit: 37a921
tree: fe8e13
parent: 151c57
author: Mr Bug
committer: Mr Crless
message: {la di da}

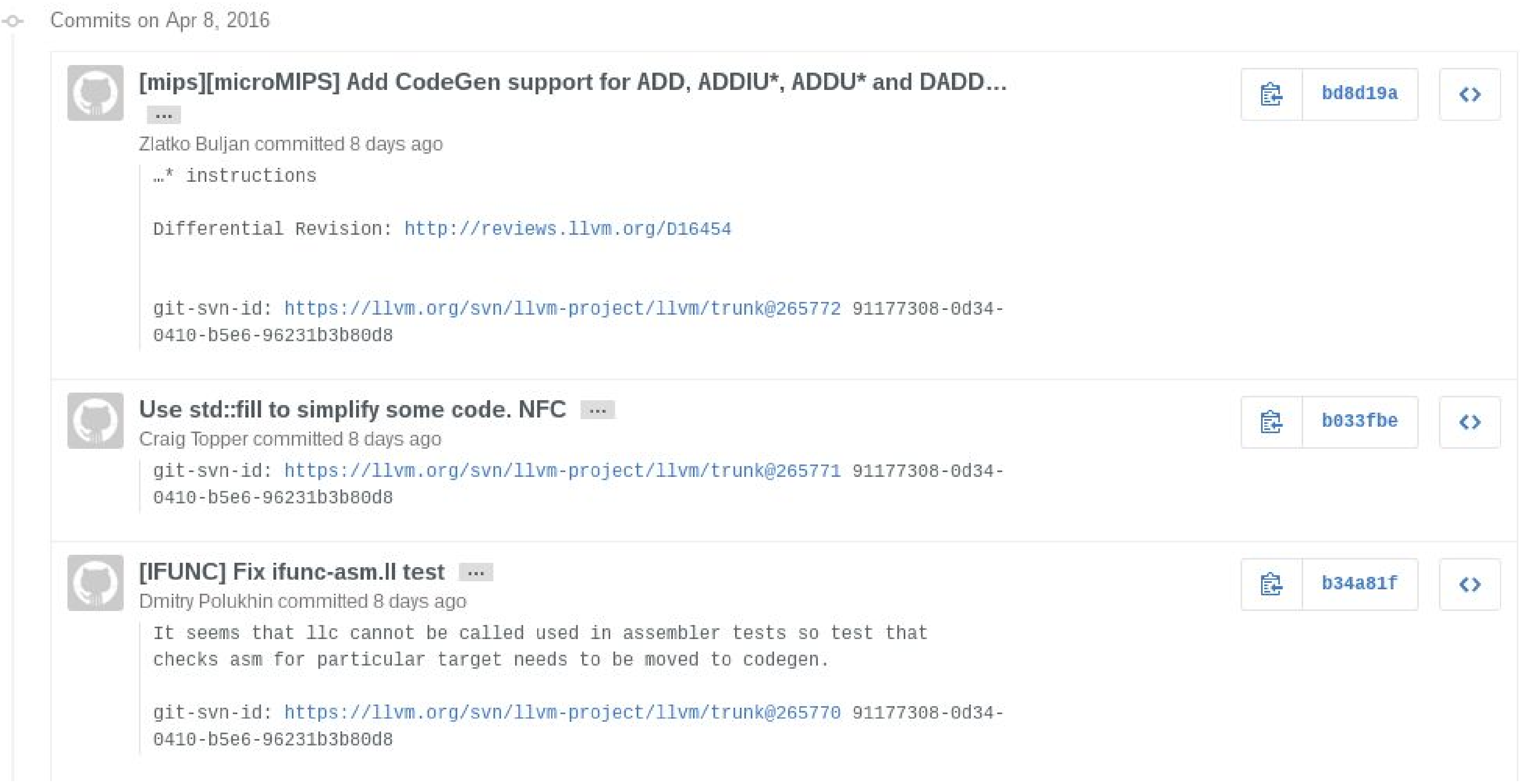
snapshot

commit: 8869e2
tree: deb987
parent: 37a921
author: Mr Fix
committer: Mr Fix
message: {fixed foo}

snapshot

commit: 149fbd
tree: 4bbc25
parent: 8869e2
author: Florian
committer: Florian
message: {new stuff}

commit: 149fbd
tree: 4bbc25
parent: 8869e2
author: Florian
committer: Florian
message: {new stuff}



Commits on Apr 8, 2016

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

Zlatko Buljan committed 8 days ago

bd8d19a

```
…* 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 …

Craig Topper committed 8 days ago

b033fbe

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

[IFUNC] Fix ifunc-asm.ll test …

Dmitry Polukhin committed 8 days ago

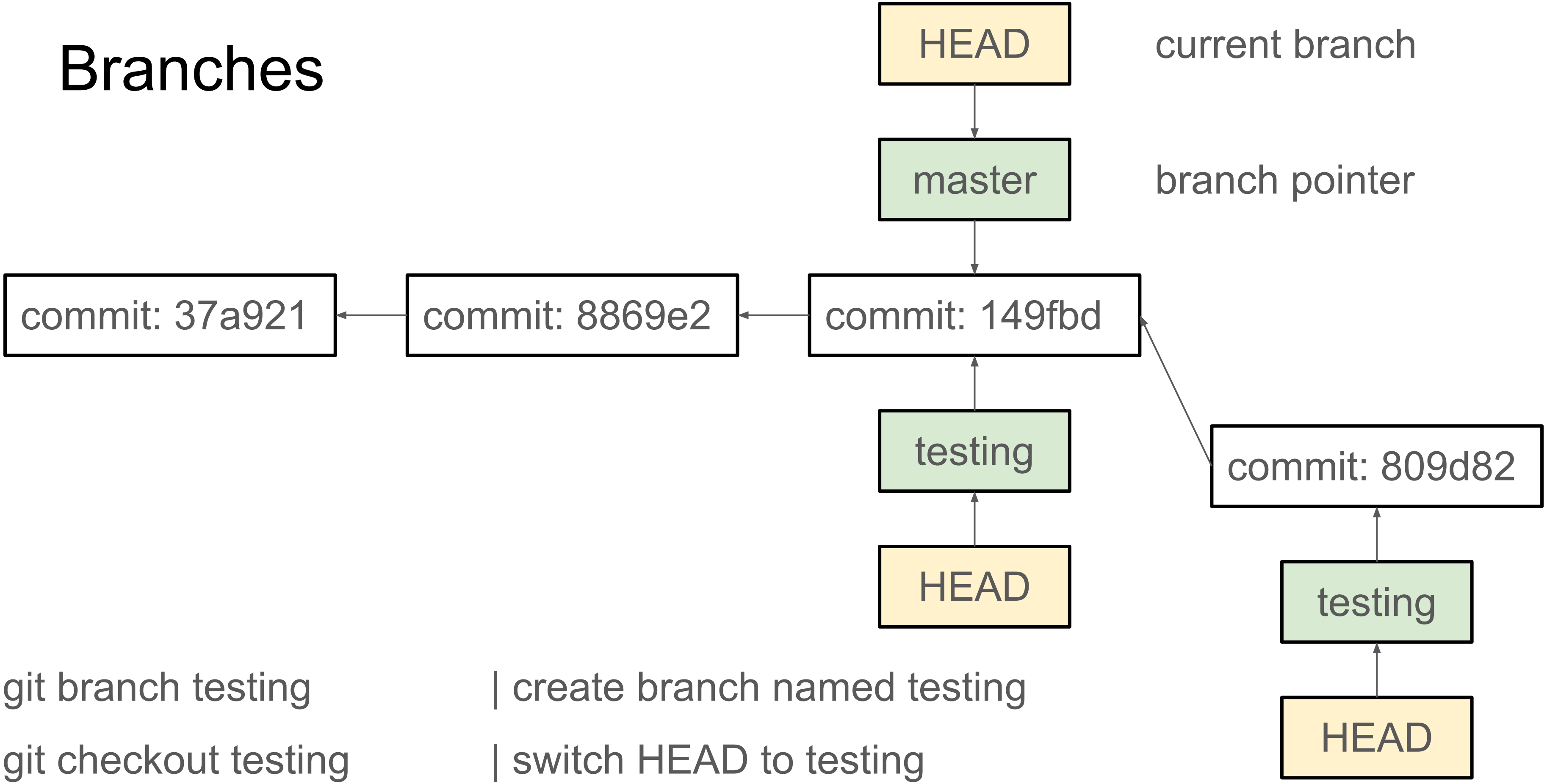
b34a81f

```
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= | formating
  - oneline
  - short
  - fuller
  - format:””
- --graph
- --since/until=3.weeks | time range
- -S<string> | shows commits that add/delete the string

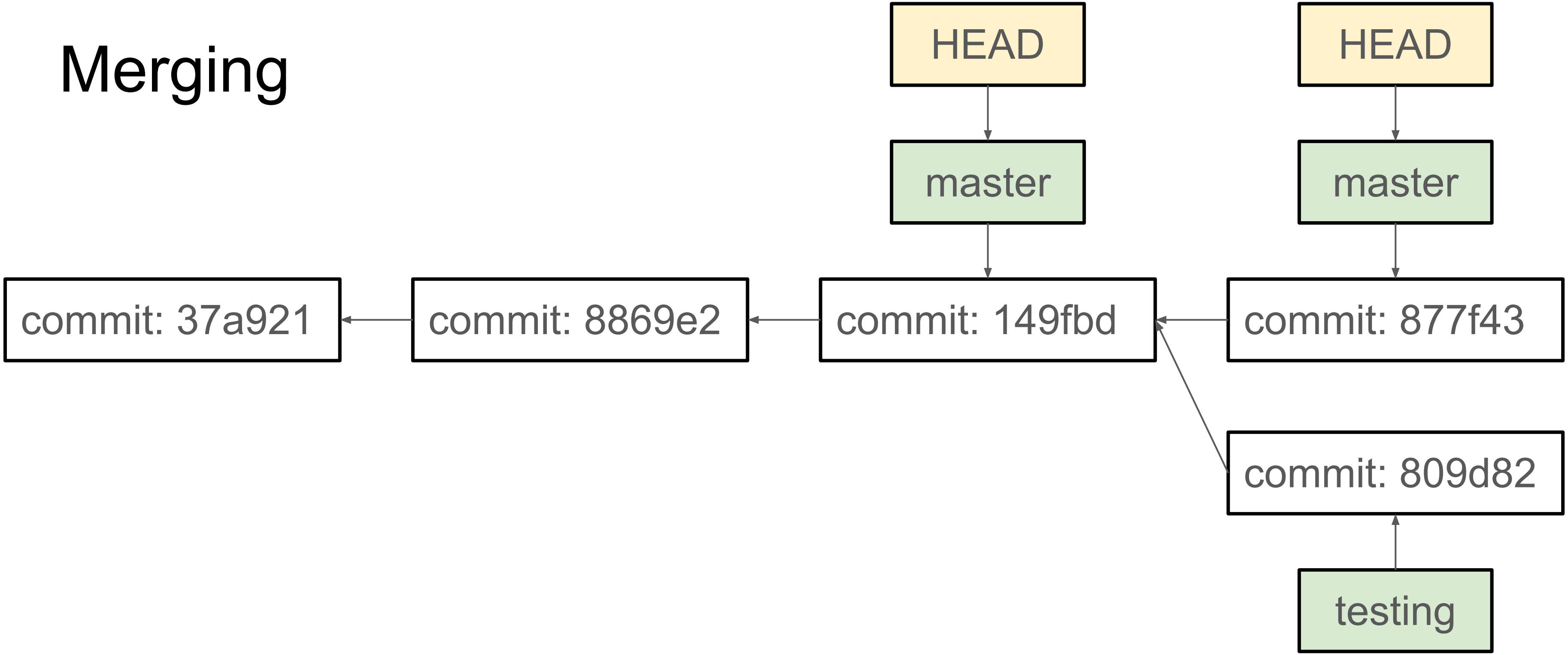
Branches



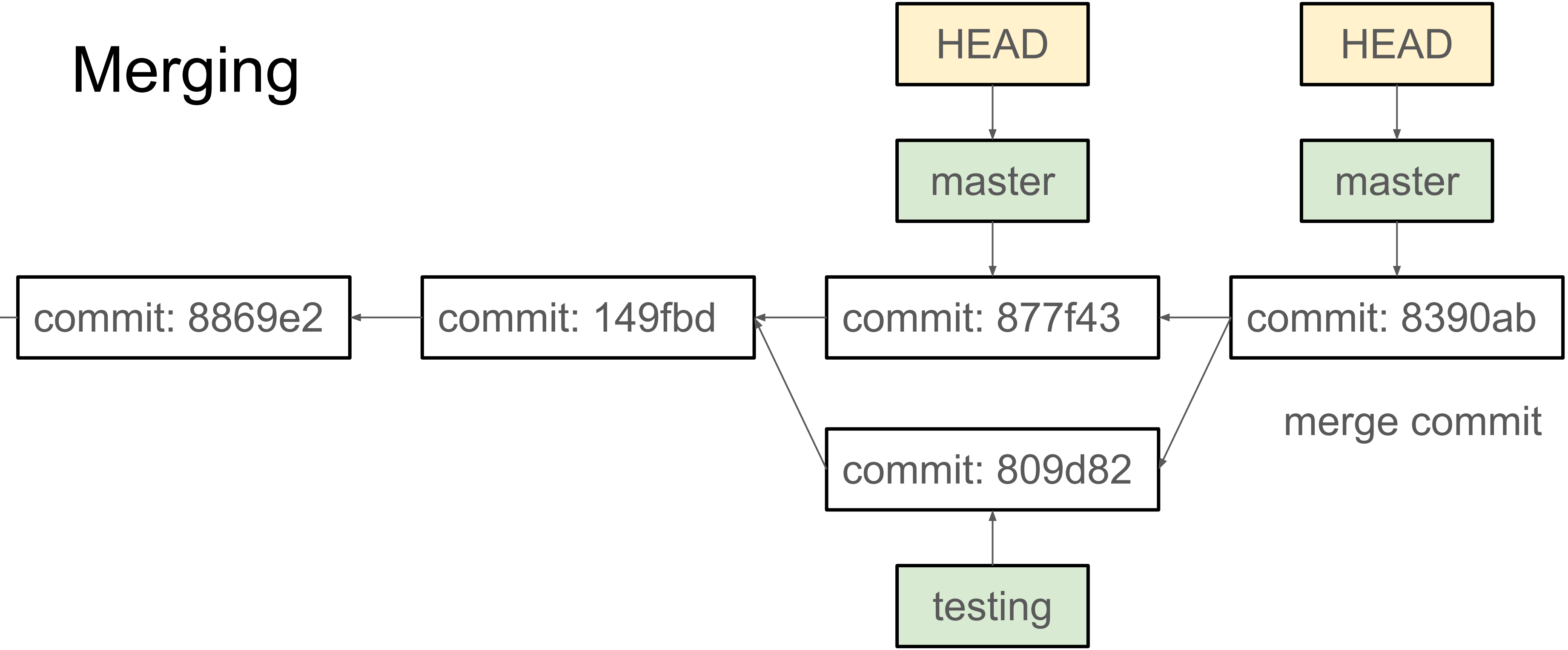
git branch testing | create branch named testing

git checkout testing | switch HEAD to testing

Merging


HEAD
master
HEAD
master
commit: 37a921
commit: 8869e2
commit: 149fbd
commit: 877f43
commit: 809d82
testing

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

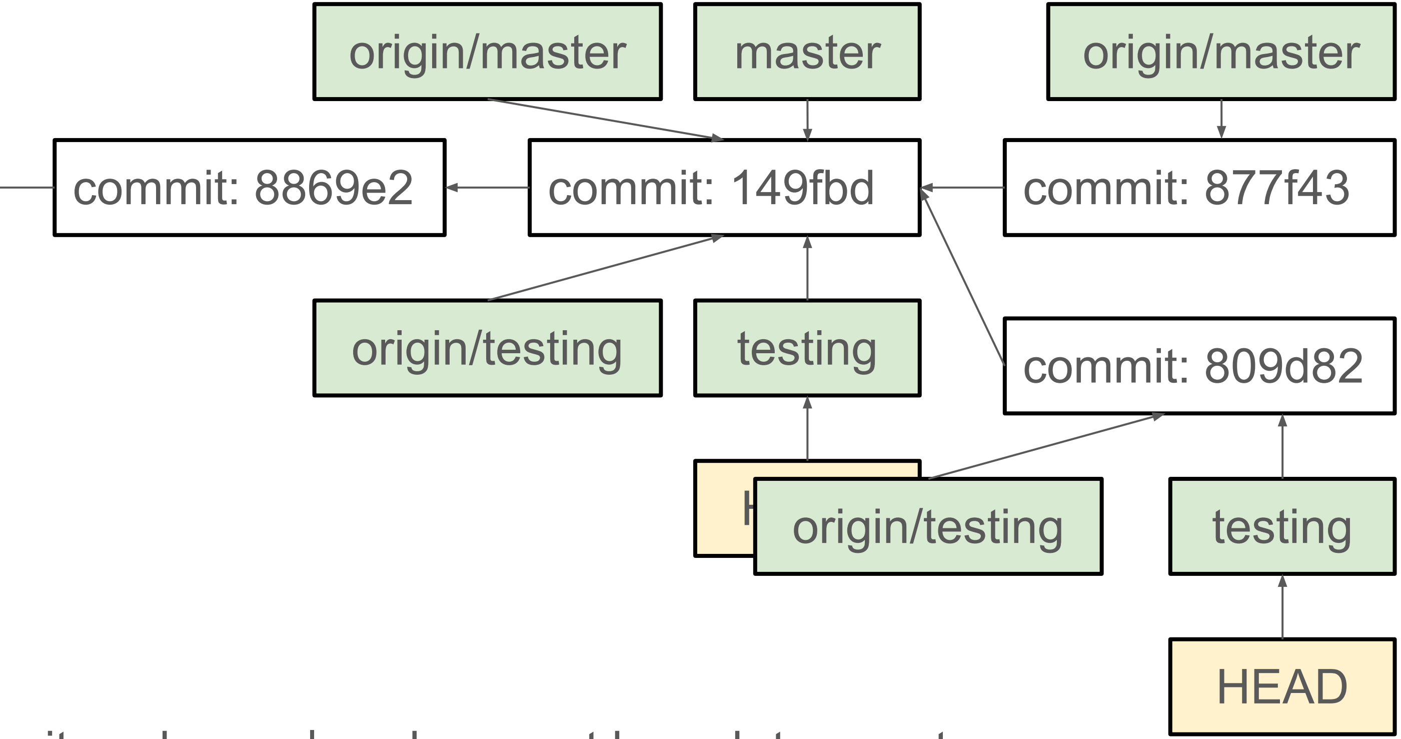
branches: (git branch -vv)

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

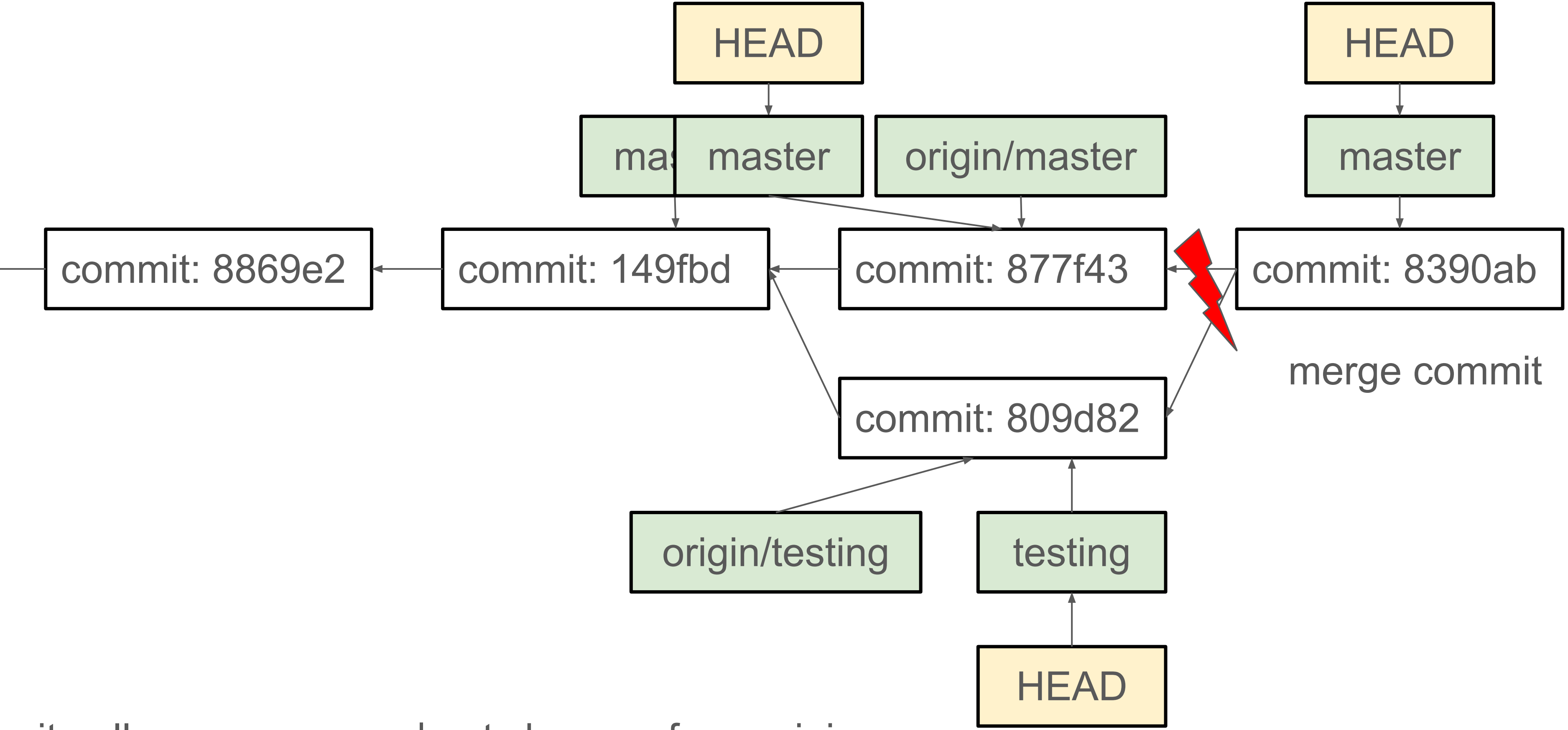
github.com

branches:

- ➔ master
- ➔ testing



git push | push current branch to remote

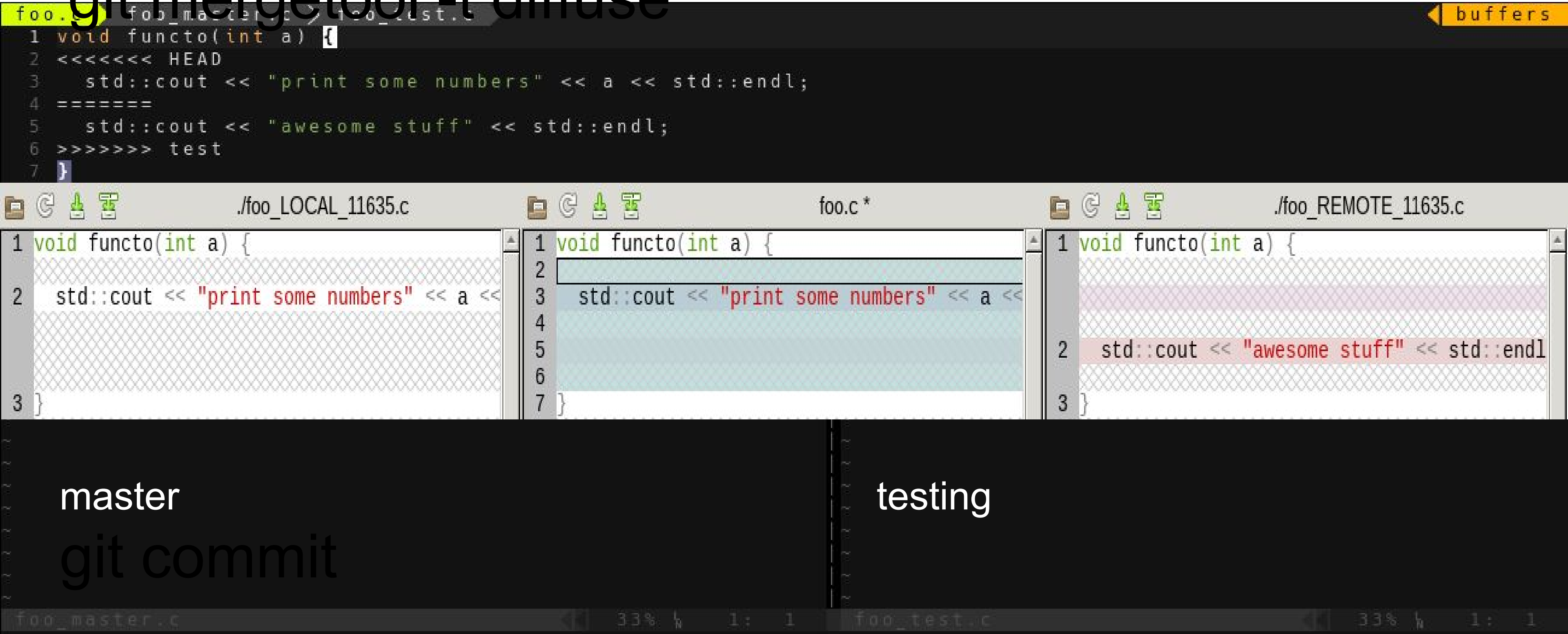


git pull | get changes from origin
git checkout master | switch to brache master
git merge testing | merge testing into master

http://www.columbia.edu/~zjn2101/intermediate-git/meme-merge.jpg

resolving conflicts (git merge testing)

git mergetool -t diffuse



```
void functo(int a) {
<<<<<<< HEAD
  std::cout << "print some numbers" << a << std::endl;
=======
  std::cout << "awesome stuff" << std::endl;
>>>>>>> test
}
```

```
void functo(int a) {
  std::cout << "print some numbers" << a <<
}
```

```
void functo(int a) {

  std::cout << "print some numbers" << a <<



}
```

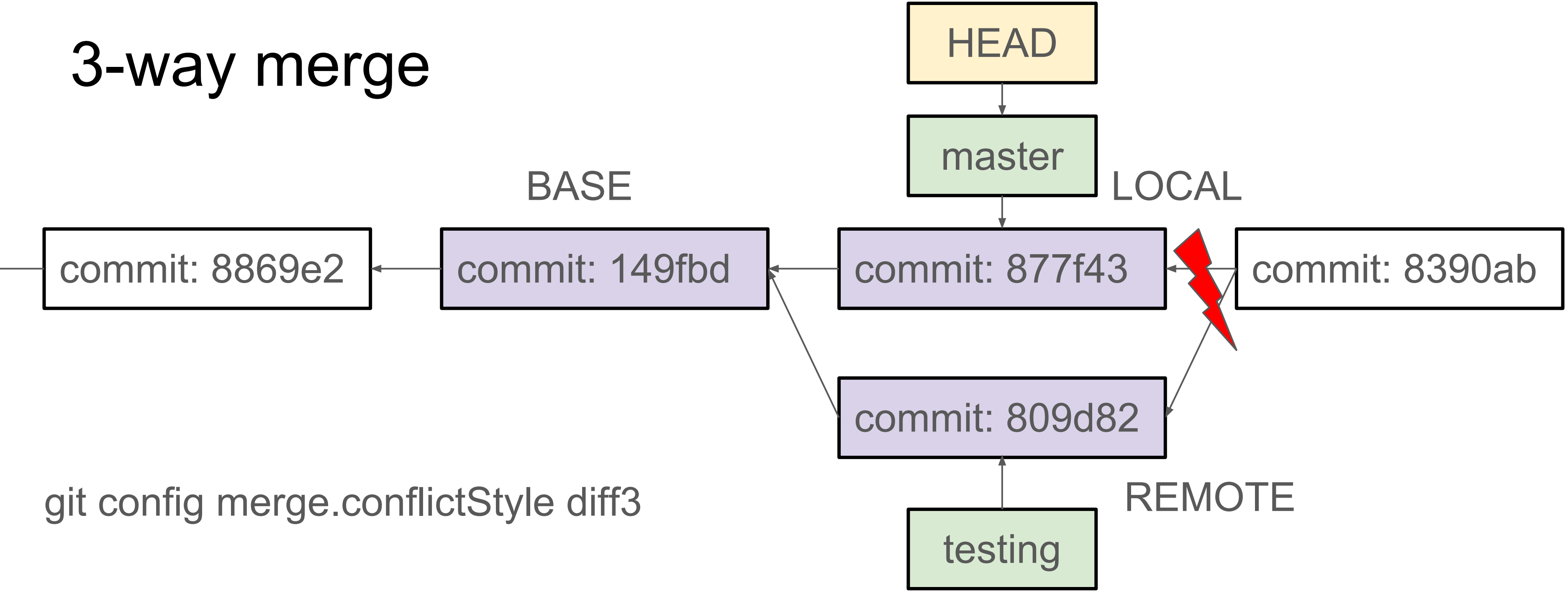
```
void functo(int a) {
  std::cout << "awesome stuff" << std::endl
}
```

master

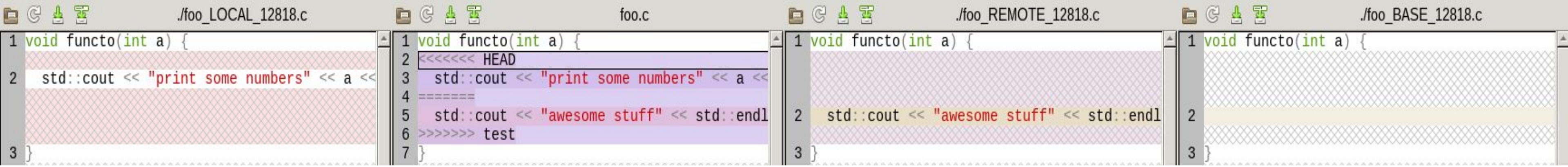
testing

git commit

3-way merge



git config merge.conflictStyle diff3



./foo_LOCAL_12818.c

```
void functo(int a) {
  std::cout << "print some numbers" << a <<
}
```

foo.c

```
void functo(int a) {
<<<<<<< HEAD
  std::cout << "print some numbers" << a <<
=======
  std::cout << "awesome stuff" << std::endl
>>>>>>> test
}
```

./foo_REMOTE_12818.c

```
void functo(int a) {
  std::cout << "awesome stuff" << std::endl
}
```

./foo_BASE_12818.c

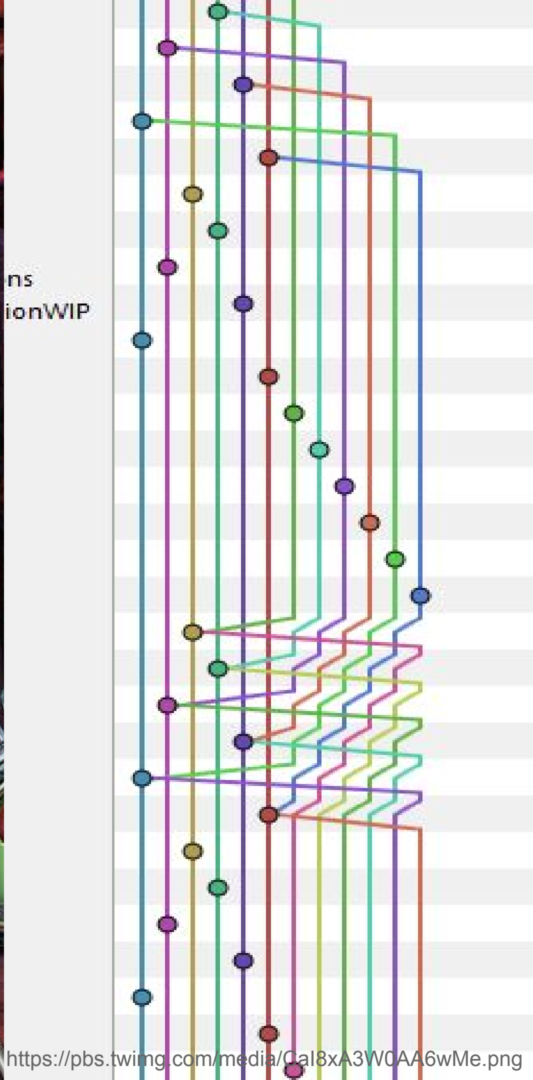
```
void functo(int a) {

}
```

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

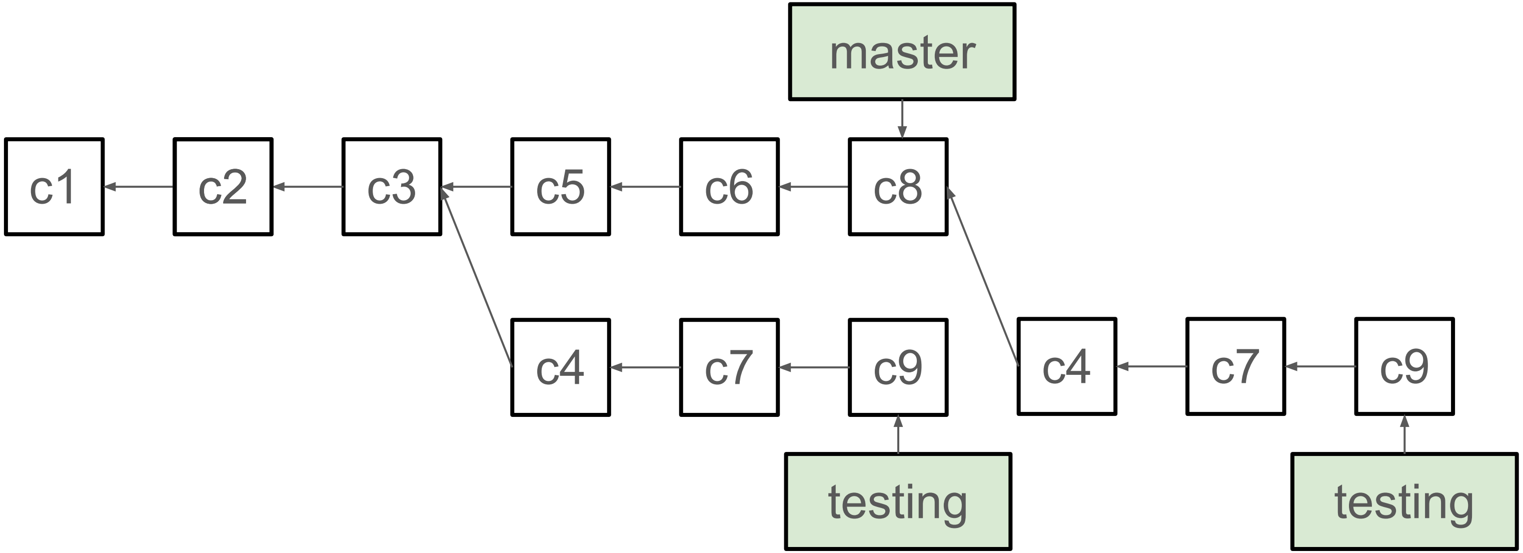
when git turns into guitar hero

https://i.kinja-img.com/gawker-media/image/upload/hvknczj1c4qqn5edap0e.jpg



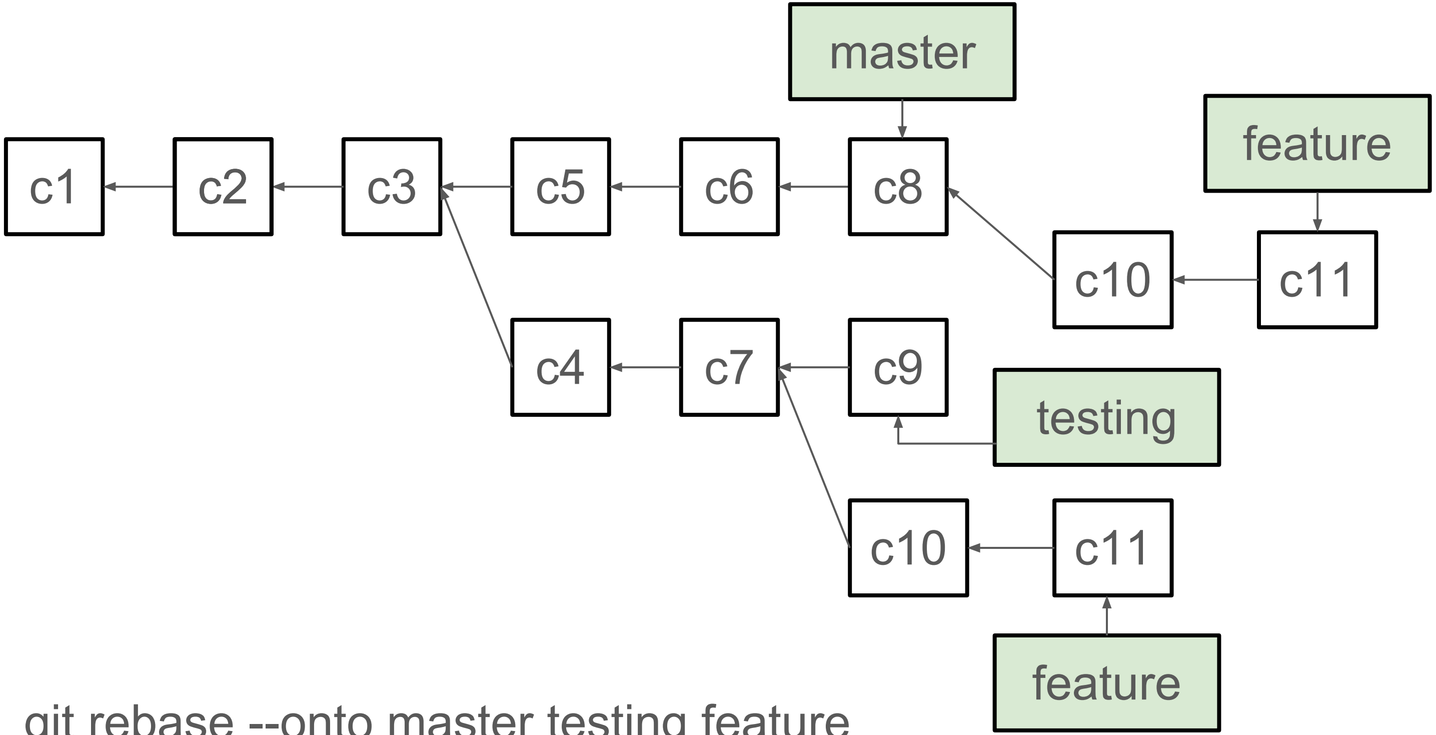
https://pbs.twimg.com/media/Cal8xA3W0AA6wMe.png

rebase



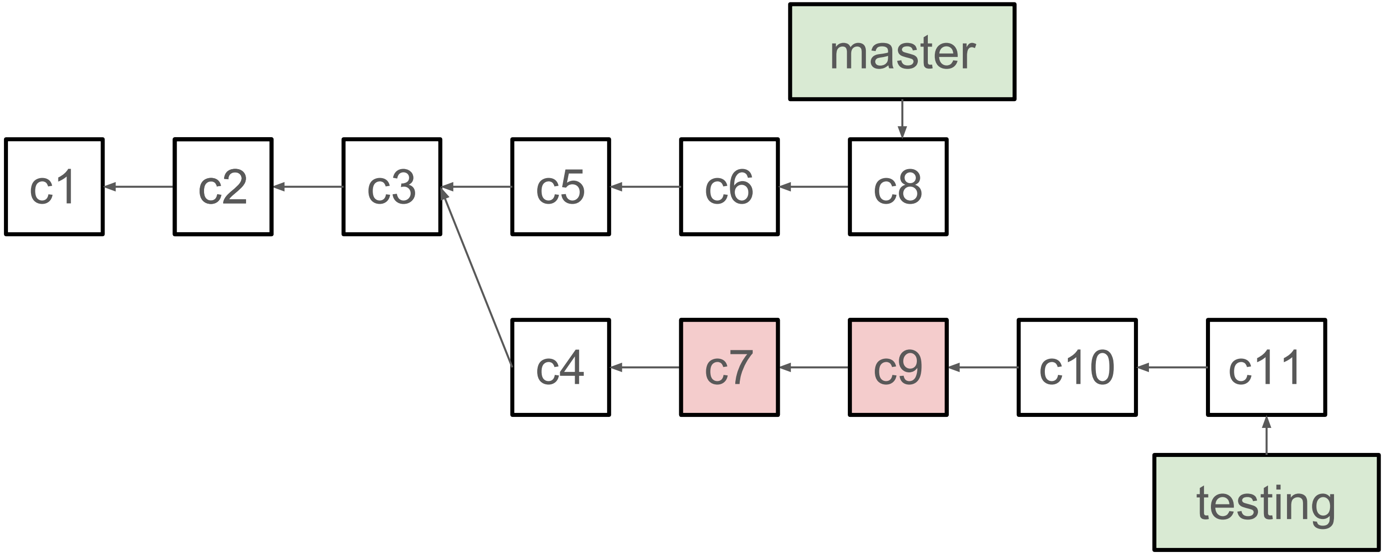
git rebase master

rebase II



git rebase --onto master testing feature

rebase III



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

specifying revisions

<rev>: sha1-hash

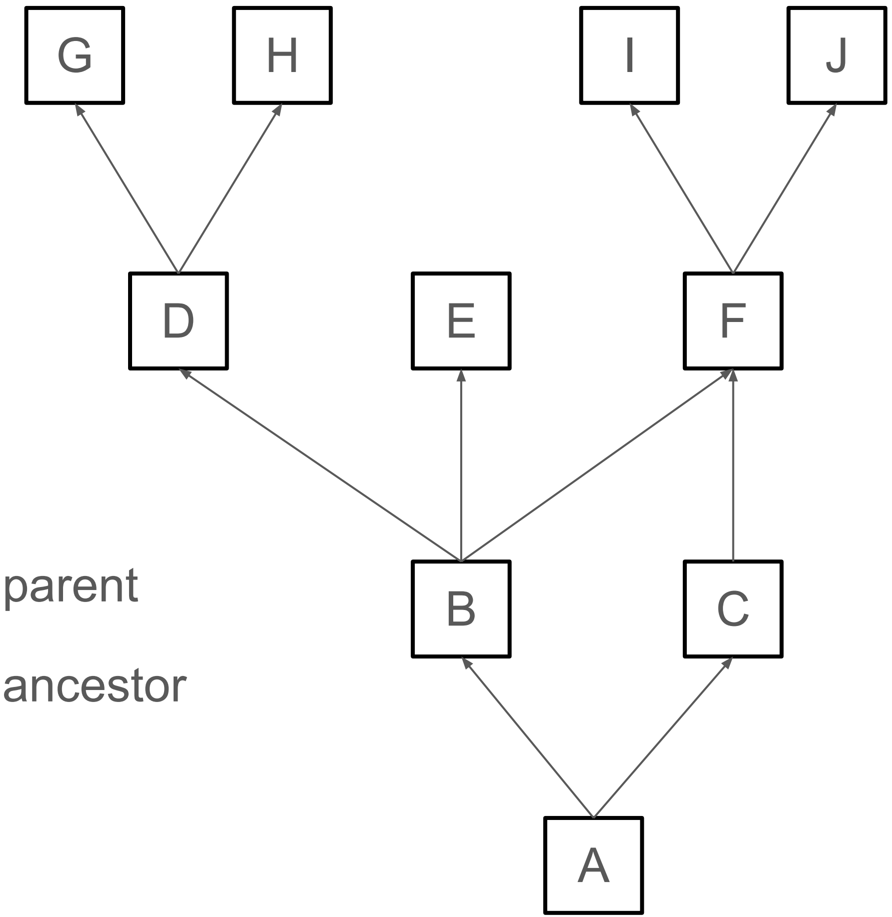
ec8723a2ff88bdabc8312c2250444fd5afc2e569

<refname>:

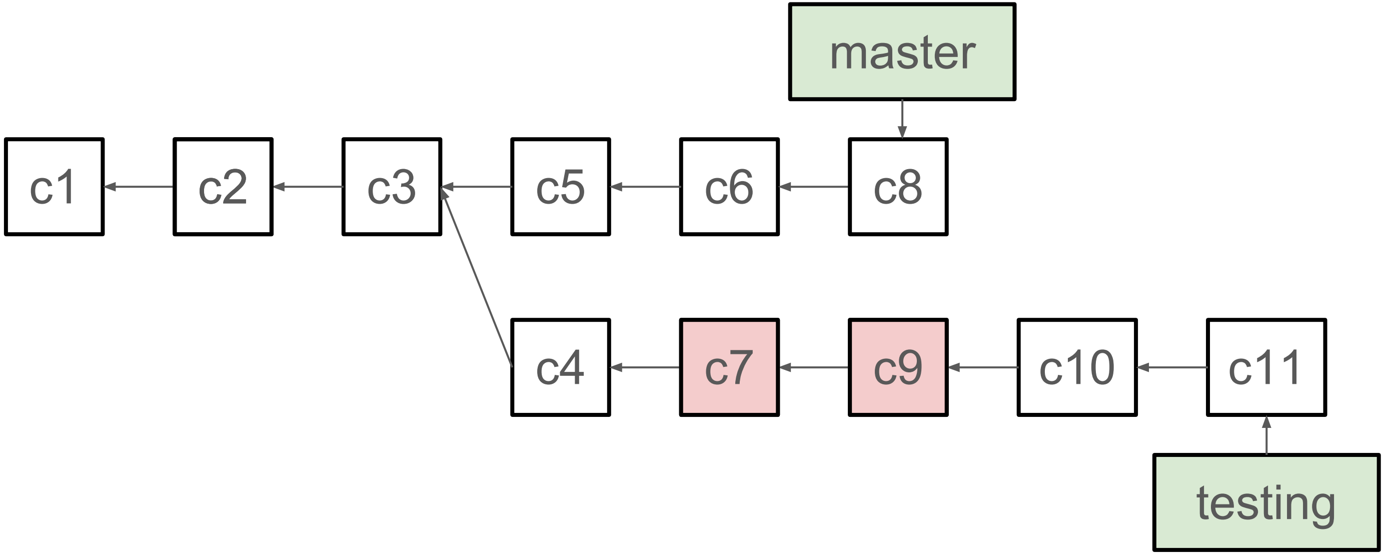
master, heads/master

A	A^0		
B	A^	A^1	A~1
F	B^3	A^^3	
G	A^^^	A~3	

^<n>	n’th parent
~<n>	n’th ancestor

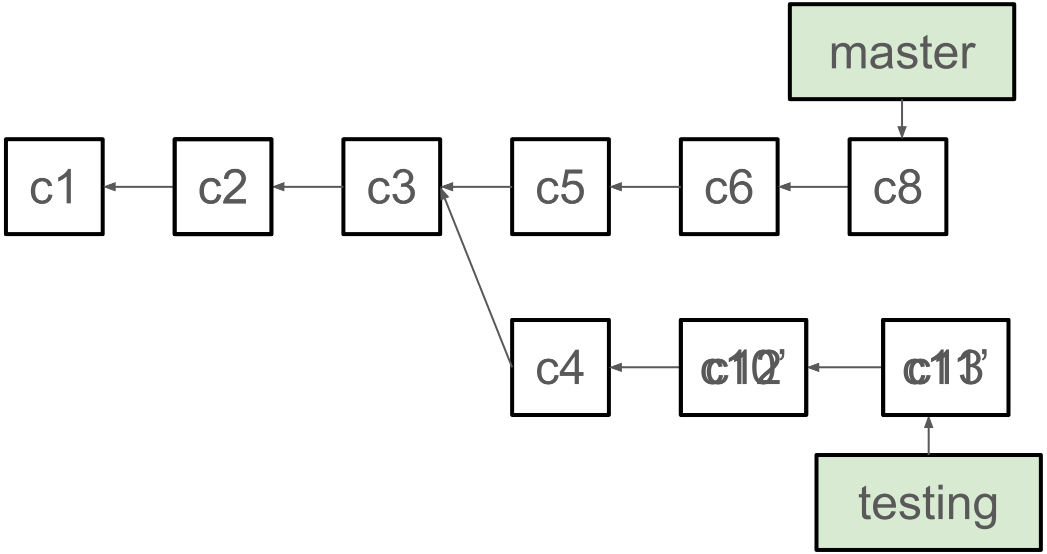


rebase



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

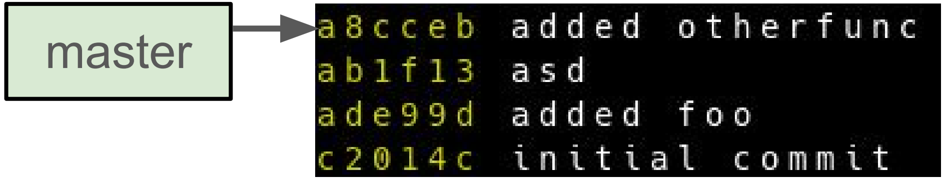
rebase



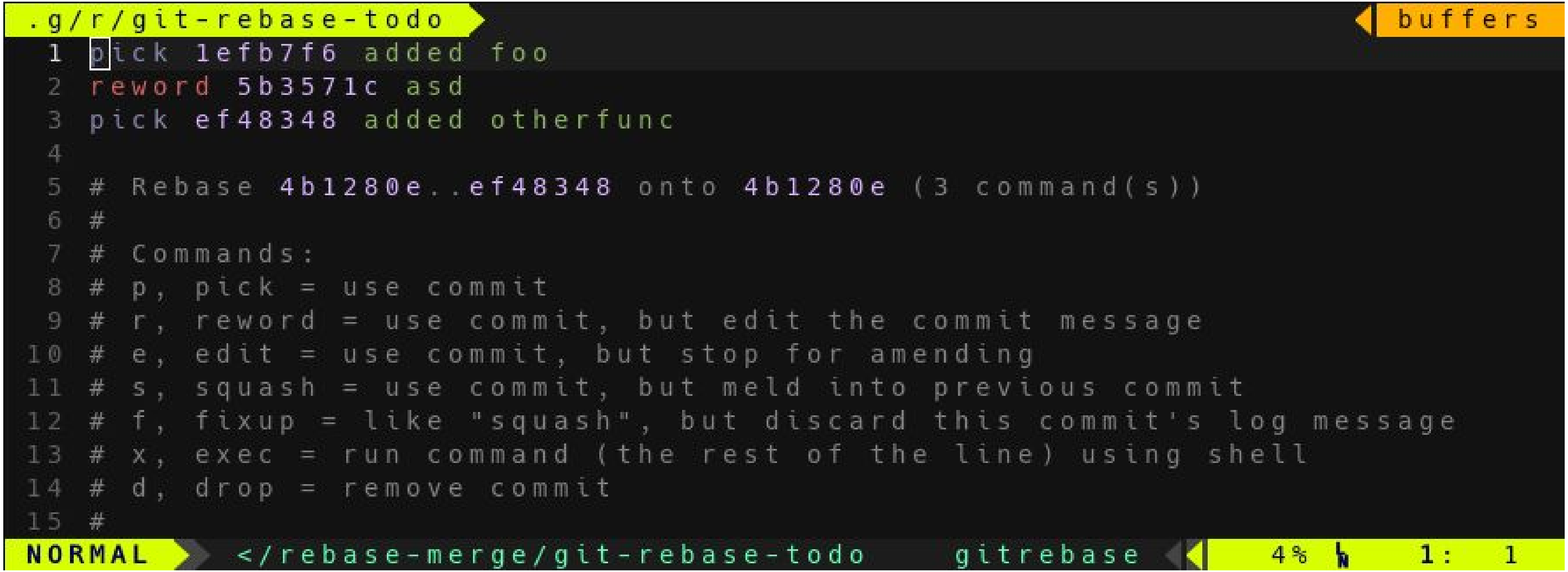
git rebase --onto testing~4 testing~2 testing

rebase interactive

git rebase -i master~3



```
a8cceb added otherfunc
ab1f13 asd
ade99d added foo
c2014c initial commit
```

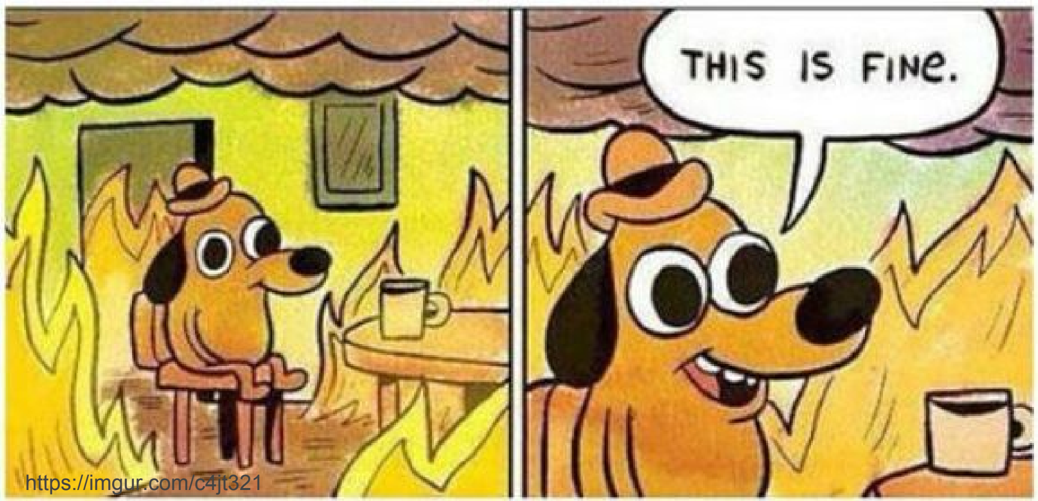


```
pick 1efb7f6 added foo
reword 5b3571c asd
pick ef48348 added otherfunc

# Rebase 4b1280e..ef48348 onto 4b1280e (3 command(s))
#
# Commands:
# p, pick = use commit
# r, reword = use commit, but edit the commit message
# e, edit = use commit, but stop for amending
# s, squash = use commit, but meld into previous commit
# f, fixup = like "squash", but discard this commit's log message
# x, exec = run command (the rest of the line) using shell
# d, drop = remove commit
#
```

rebase

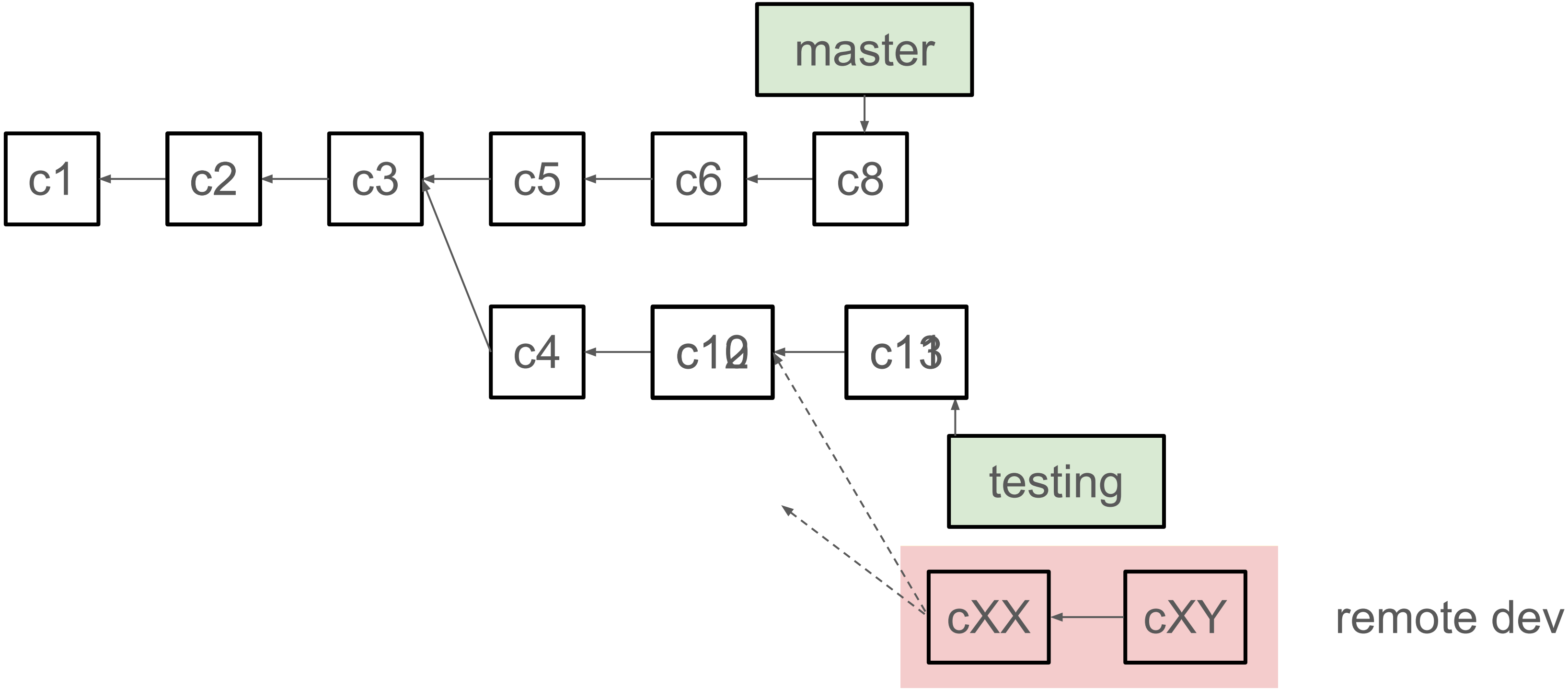
```
git rebase add

git rebase --continue

git rebase --abort
```

https://imgur.com/XFQLB



https://imgur.com/c4jt321

rebase problem

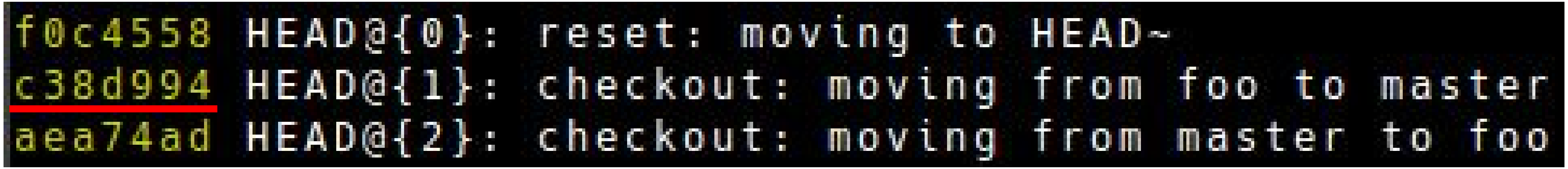


git config pull.rebase true

YOU SHALL NOT
REBASE PUSHED COMMITS
memegenerator.net

git reflog

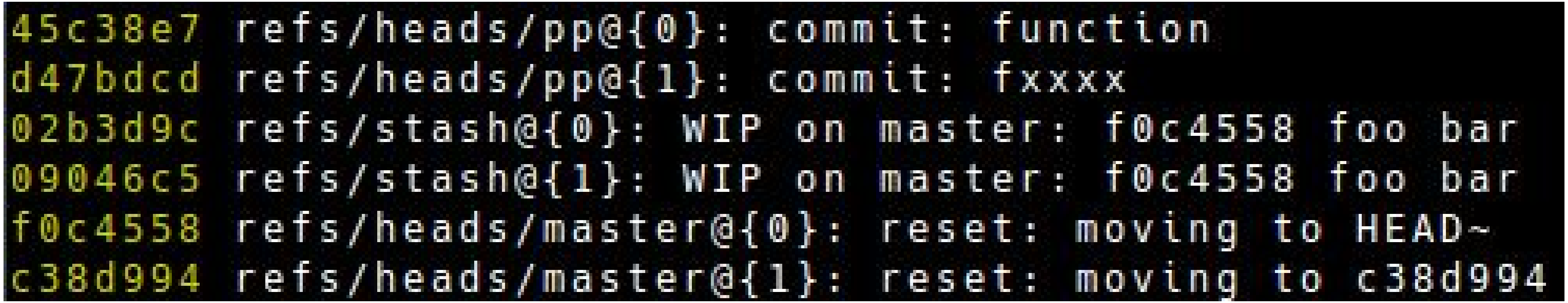
logs the states where branches/HEAD was before a change occurred



```
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
```

tags (git tag)

lightweight tag: git tag [new tag name]

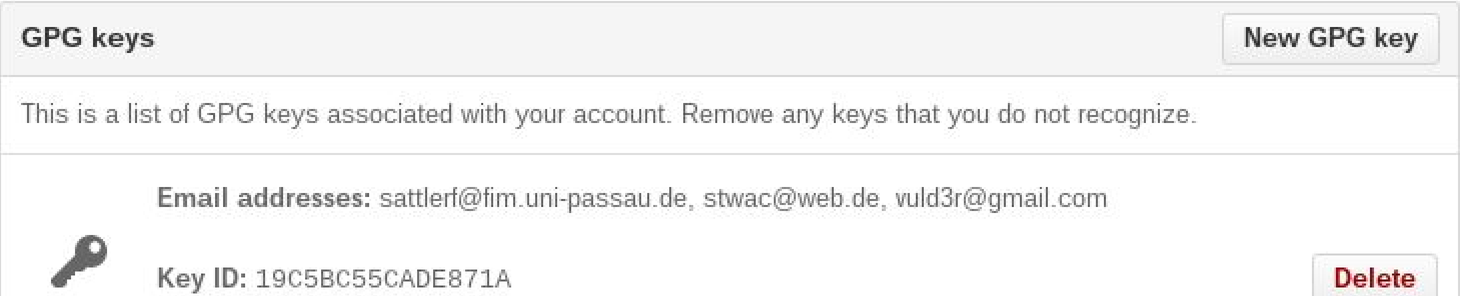
annotated tag: git tag -a [tag] -m [message]

+ sign tag with -s

default email key
Conf: user.signingKey = [keyid]

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

github - GPG support



GPG keys

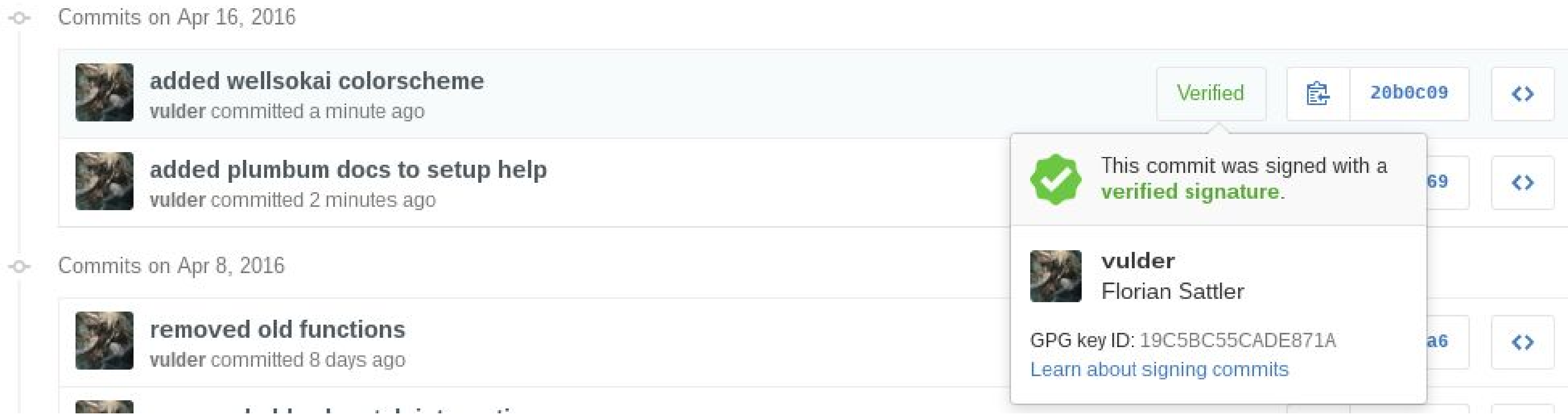
New GPG key

This is a list of GPG keys associated with your account. Remove any keys that you do not recognize.

Email addresses: sattlerf@fim.uni-passau.de, stwac@web.de, vuld3r@gmail.com

Key ID: 19C5BC55CADE871A

Delete



Commits on Apr 16, 2016

added wellsokai colorscheme
vulder committed a minute ago

Verified 20b0c09

added plumbum docs to setup help
vulder committed 2 minutes ago

69

Commits on Apr 8, 2016

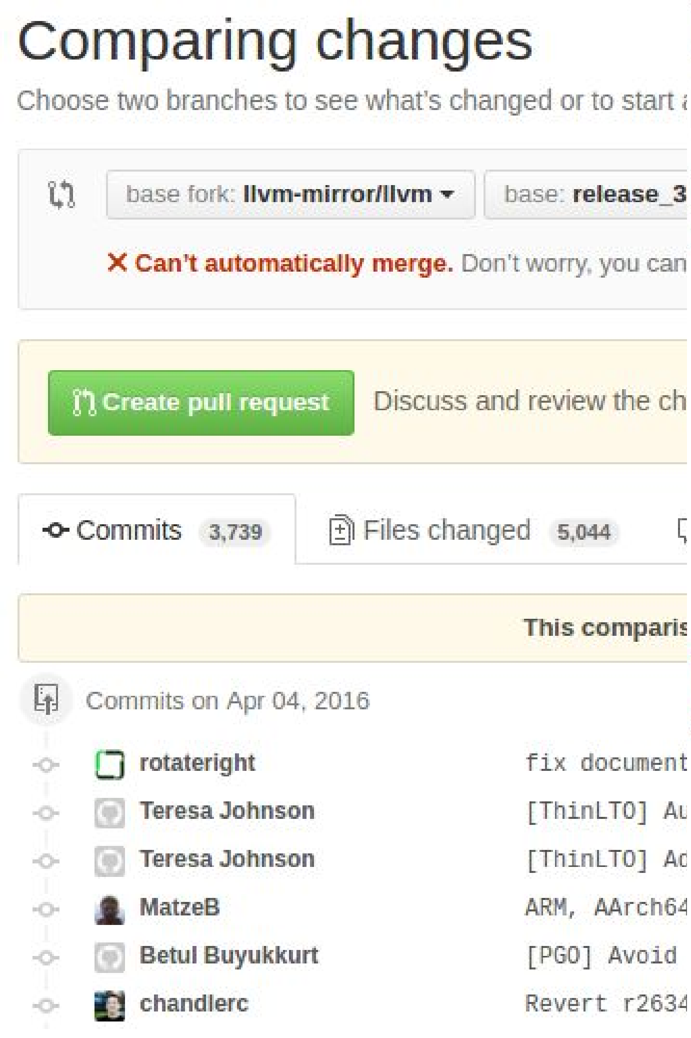
removed old functions
vulder committed 8 days ago

a6

This commit was signed with a **verified signature**.

vulder
Florian Sattler

GPG key ID: 19C5BC55CADE871A
Learn about signing commits



Comparing changes

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

base fork: llvm-mirror/llvm ▾ | base: release_3

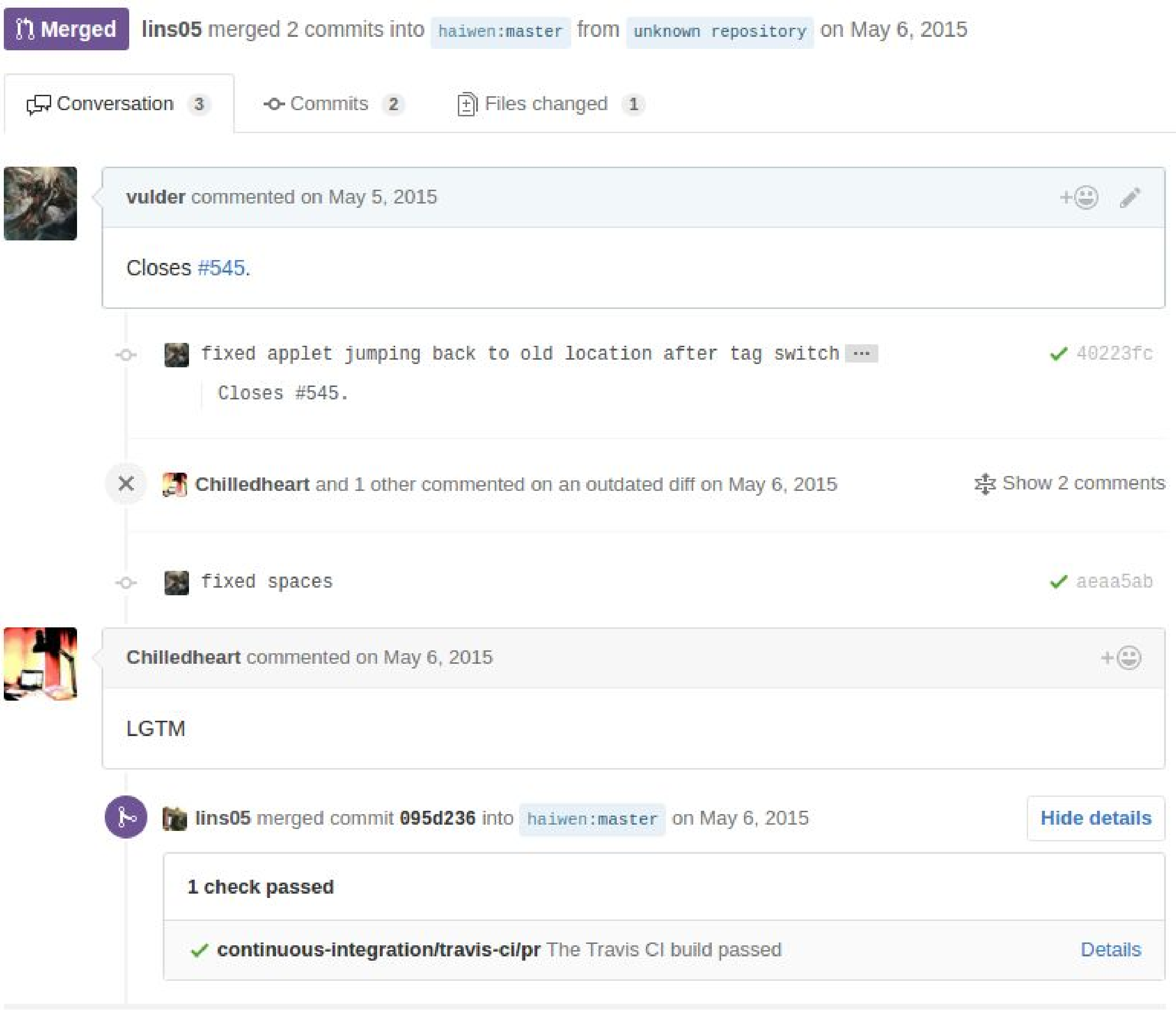
✕ Can't automatically merge. Don't worry, you can

Discuss and review the ch

This comparis

Commits on Apr 04, 2016

rotateright	fix document
Teresa Johnson	[ThinLTO] Au
Teresa Johnson	[ThinLTO] Ad
MatzeB	ARM, AArch64
Betul Buyukkurt	[PGO] Avoid
chandlerc	Revert r2634



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

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

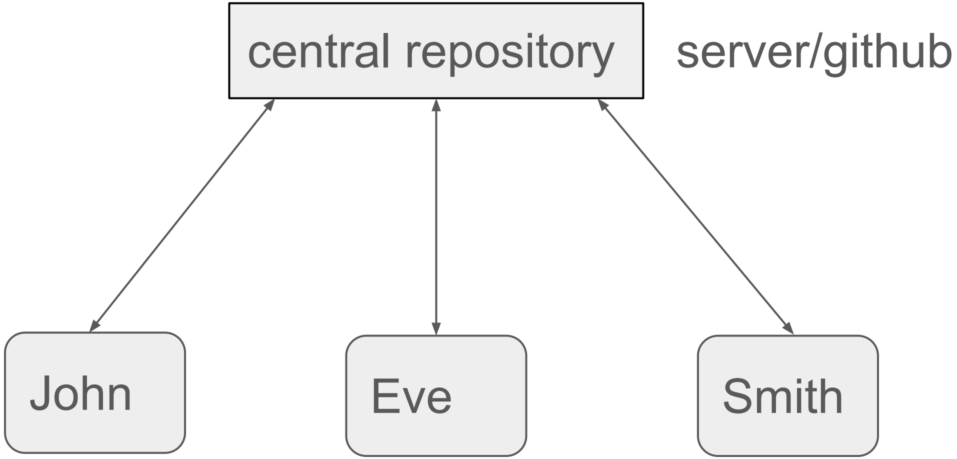
LGTM

lins05 merged commit **095d236** into `haiwen:master` on May 6, 2015 — Hide details

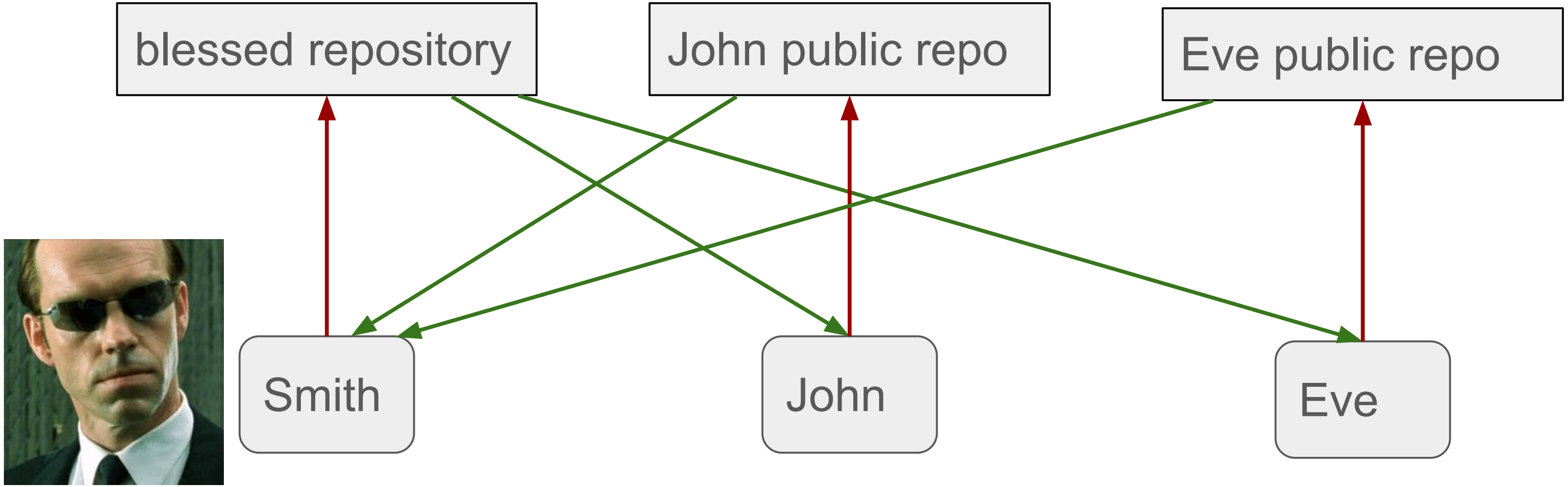
1 check passed

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

centralized workflow

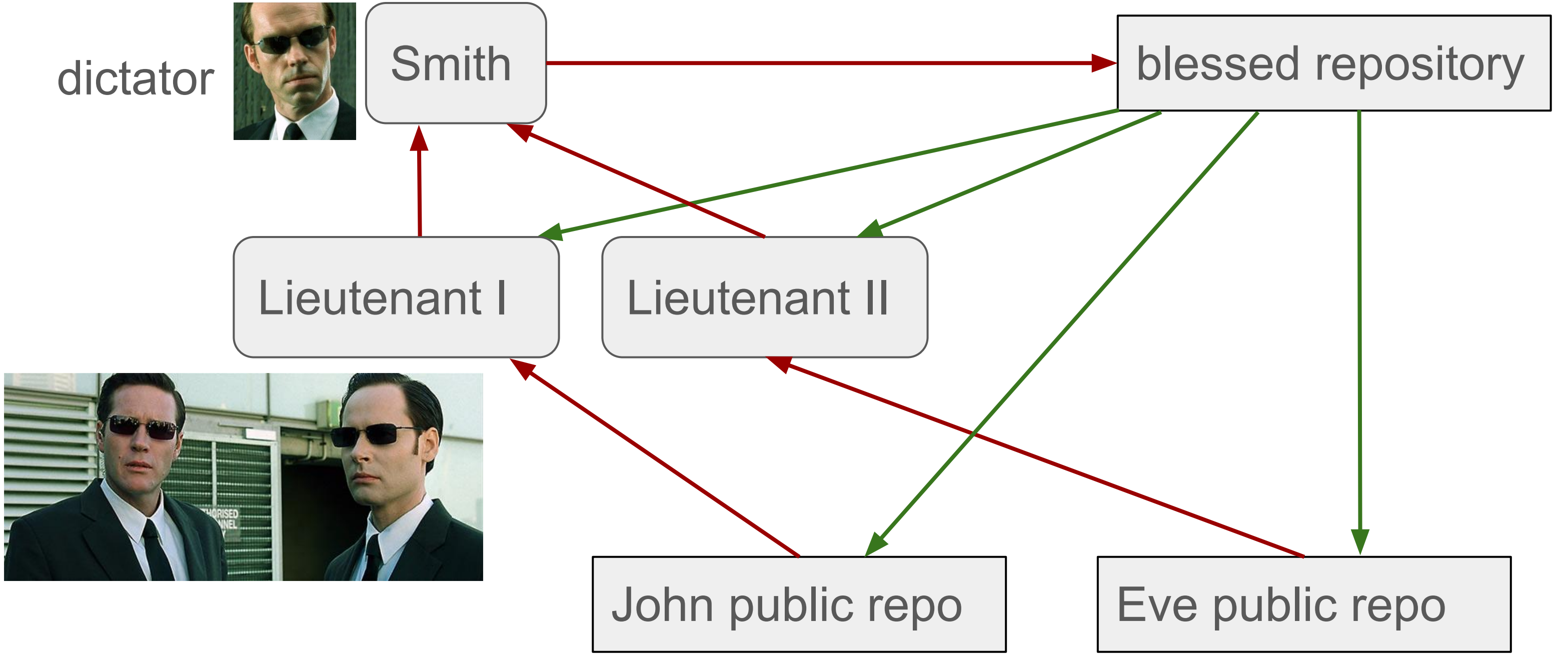


integration-manager workflow



integration manager

dictator and lieutenants workflow



dictator

Smith

blessed repository

Lieutenant I

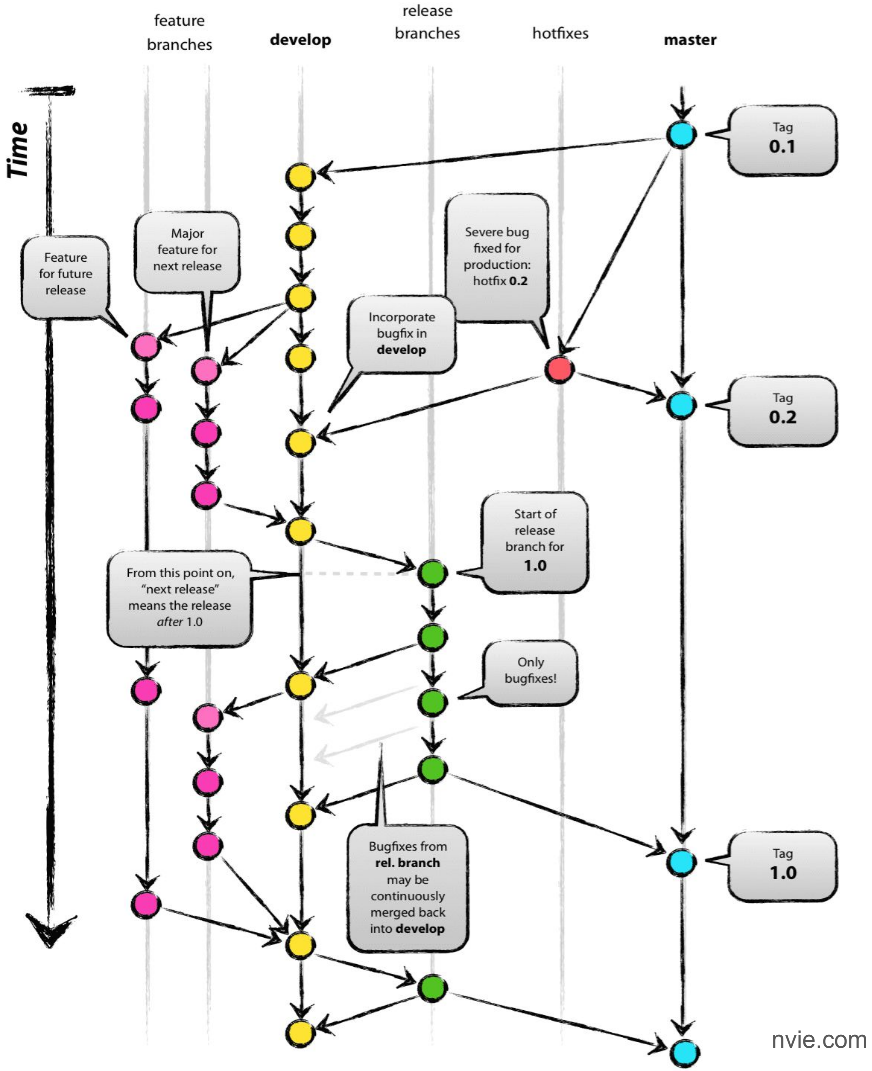
Lieutenant II

John public repo

Eve public repo

git flow (https://github.com/nvie/gitflow)

- master
- develop
- feature
- release
- hotfix


feature branches
develop
release branches
hotfixes
master
Time
Tag 0.1
Feature for future release
Major feature for next release
Severe bug fixed for production: hotfix 0.2
Incorporate bugfix in develop
Tag 0.2
Start of release branch for 1.0
From this point on, "next release" means the release after 1.0
Only bugfixes!
Bugfixes from rel. branch may be continuously merged back into develop
Tag 1.0
nvie.com

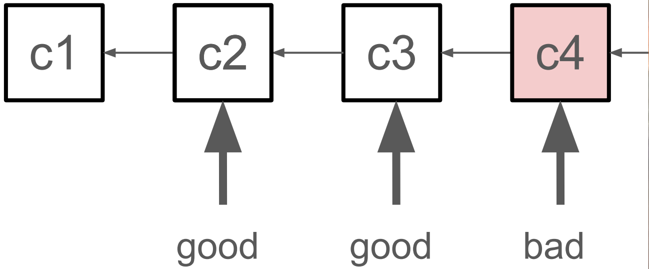
git stash stash@{0}

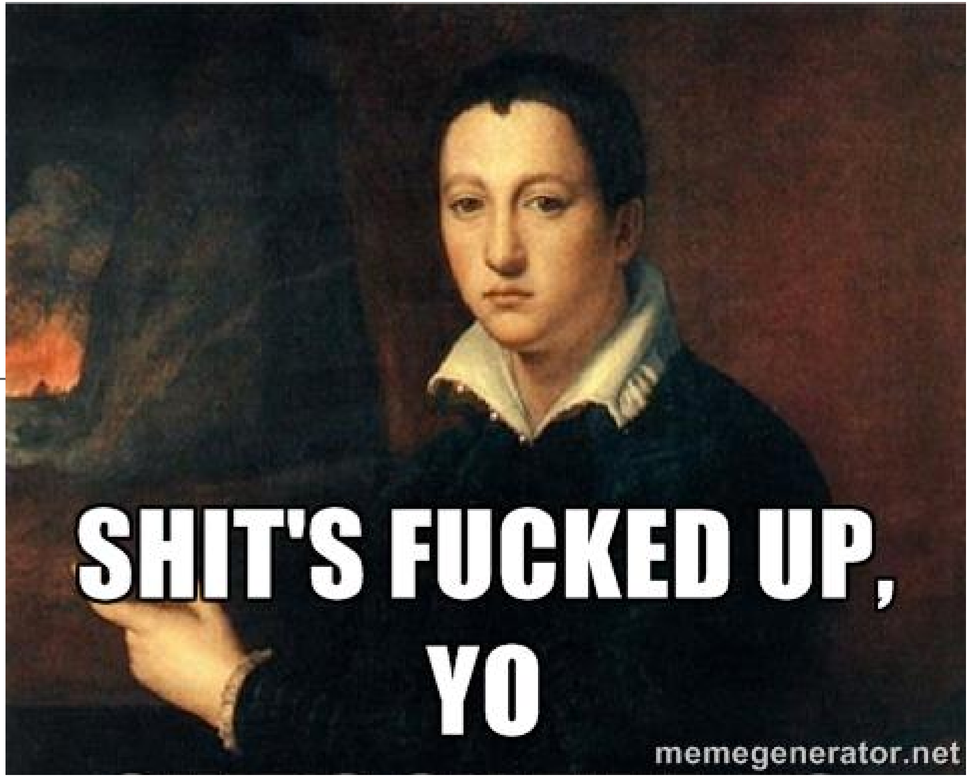
- list | list all stashed states
- show <stash> | show state of stash
- pop/apply | {remove} state and apply it to workdir
- branch <stash> | create branch were shash was recorded + apply
- drop <stash> | drop state from stash

- -u | include untracked files
- -p | patch

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

git bisect

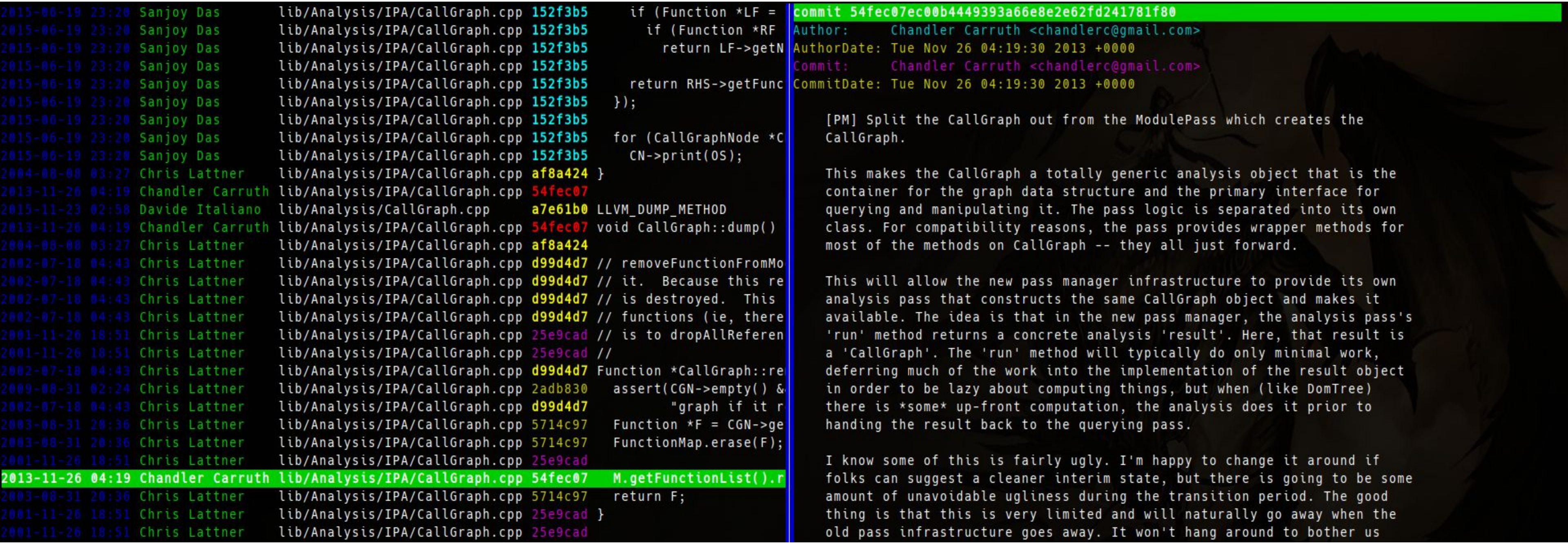

c1
c2
c3
c4
good
good
bad


SHIT'S FUCKED UP, YO
memegenerator.net

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

git blame {file}



```
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
Commit:     Chandler Carruth <chandlerc@gmail.com>
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
```

tool: tig

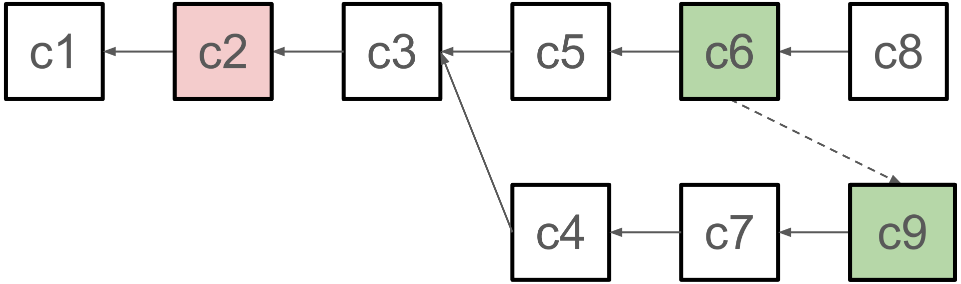
YOU SAID IT WASN'T YOUR FAULT
GIT BLAME DETERMINED THAT WAS A LIE
memegenerator.net

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

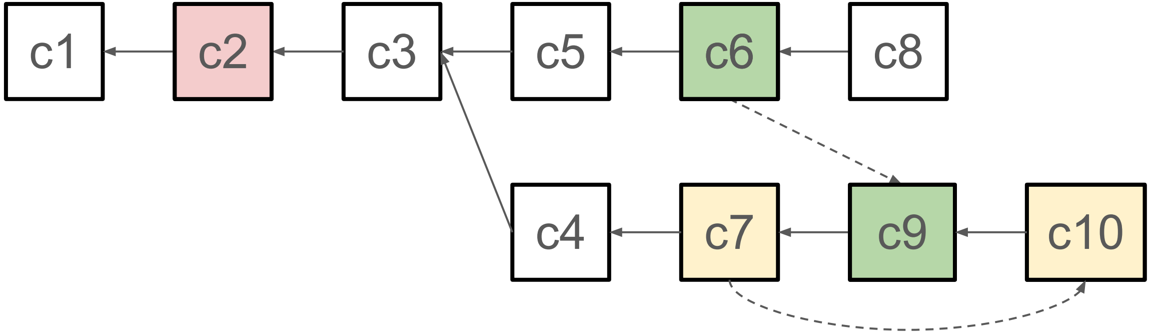


git cherry-pick c6

● -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 filter-branch

- git filter-branch --tree-filter ‘rm -rf FILE’ master
- git filter-branch --msg-filter ‘sed -e “/BADWORD/d”’
- git filter-branch --env-filter ‘

  if test ”$GIT_AUTHOR_EMAIL” = “bar@foo.org”

  then … ’

old commits saved to refs/original/*

-f | overwrite saved commits

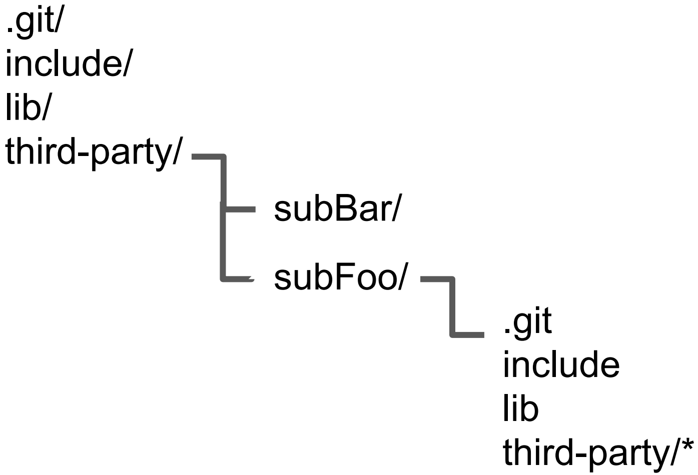
git update-ref -d refs/original/refs/heads/master

BFG Repo-Cleaner (https://rtyley.github.io/bfg-repo-cleaner/)

alias java -jar bfg.jar = bfg

- bfg --delete-files id_{rsa,dsa} repo.git | delete key files
- bfg --strip-blobs-bigger-than 50M repo.git | remove big files
  - --protect-blobs-from master, next | restrict commands
- bfg --replace-text password.txt repo.git | remove all on pw list

git submodule



.git/
include/
lib/
third-party/
subBar/
subFoo/
.git
include
lib
third-party/*

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

git-large file storage

- git lfs track <file type>
  - ‘*.pdf’
  - ‘bigStuff/’
- git lfs track
- git lfs ls-files

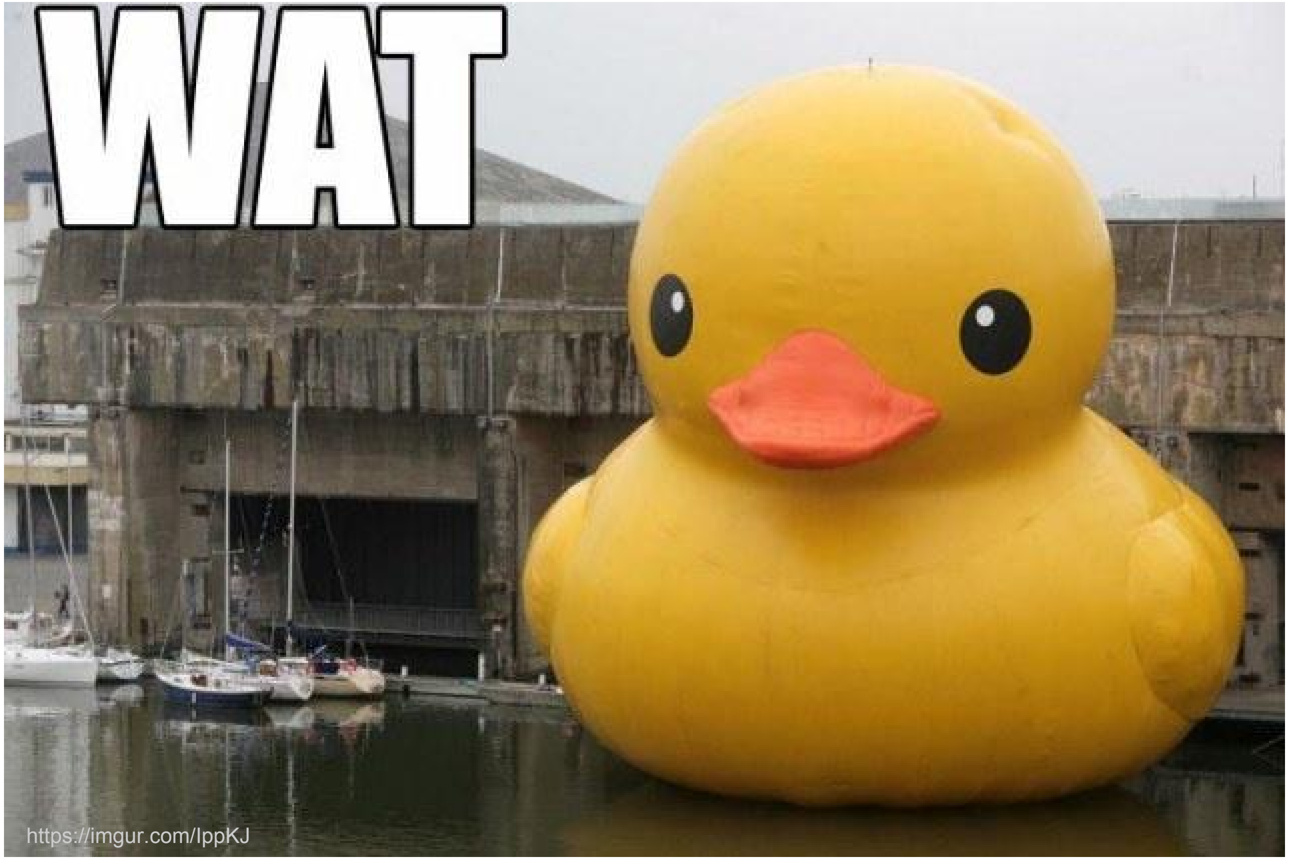
git add {file} | file replaced by pointer

.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

digging deeper .git/objects/

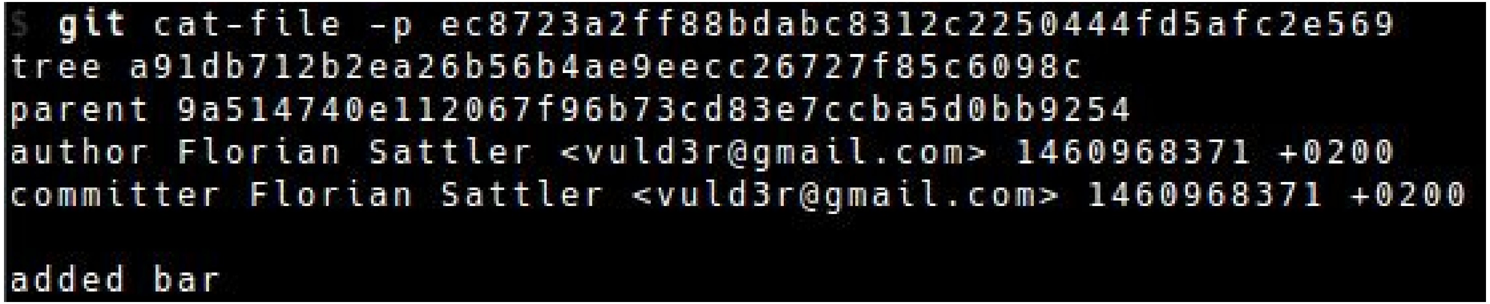
- 02
- 9a
- d4
- f7
- ec



- ❖ ec
  - ➢ 8723a2ff88bdabc8312c2250444fd5afc2e569

ec8723a2ff88bdabc8312c2250444fd5afc2e569

git cat-file -p ec8723a2ff88bdabc8312c2250444fd5afc2e569



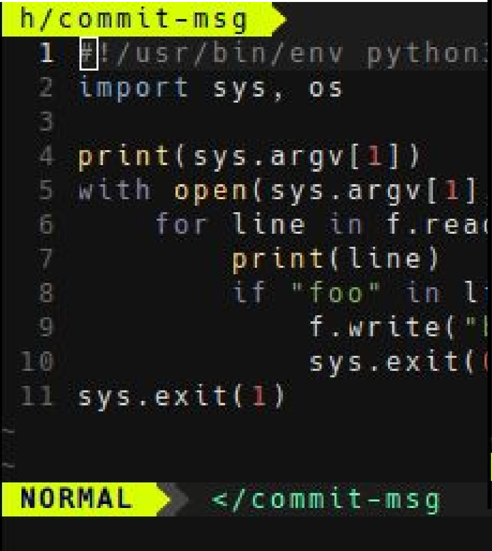
```
$ git cat-file -p ec8723a2ff88bdabc8312c2250444fd5afc2e569
tree a91db712b2ea26b56b4ae9eecc26727f85c6098c
parent 9a514740e112067f96b73cd83e7ccba5d0bb9254
author Florian Sattler <vuld3r@gmail.com> 1460968371 +0200
committer Florian Sattler <vuld3r@gmail.com> 1460968371 +0200

added bar
```

git gc | clean and optimize local repo

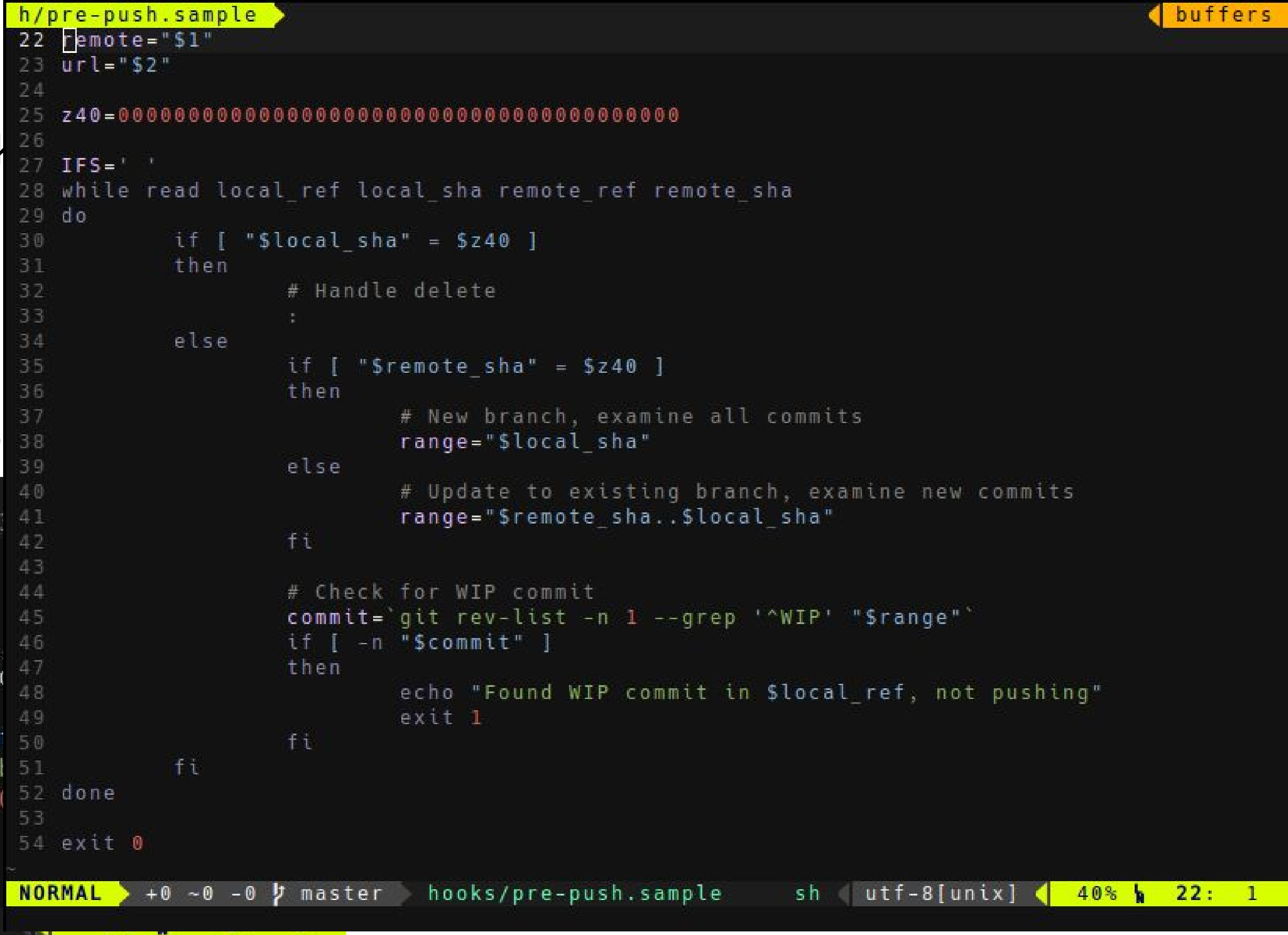
- ❖ info
- ❖ pack
  - ➢ pack-6d505071a5d9cfe8cae32a8b90057c16c4a378ef.idx | offsets
  - ➢ pack-6d505071a5d9cfe8cae32a8b90057c16c4a378ef.pack | objects

hooks (.git/hook

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



```
#!/usr/bin/env python
import sys, os

print(sys.argv[1])
with open(sys.argv[1]
    for line in f.read
        print(line)
        if "foo" in li
            f.write("
            sys.exit(
sys.exit(1)
```



```
remote="$1"
url="$2"

z40=0000000000000000000000000000000000000000

IFS=' '
while read local_ref local_sha remote_ref remote_sha
do
        if [ "$local_sha" = $z40 ]
        then
                # Handle delete
                :
        else
                if [ "$remote_sha" = $z40 ]
                then
                        # New branch, examine all commits
                        range="$local_sha"
                else
                        # Update to existing branch, examine new commits
                        range="$remote_sha..$local_sha"
                fi

                # Check for WIP commit
                commit=`git rev-list -n 1 --grep '^WIP' "$range"`
                if [ -n "$commit" ]
                then
                        echo "Found WIP commit in $local_ref, not pushing"
                        exit 1
                fi
        fi
done

exit 0
```

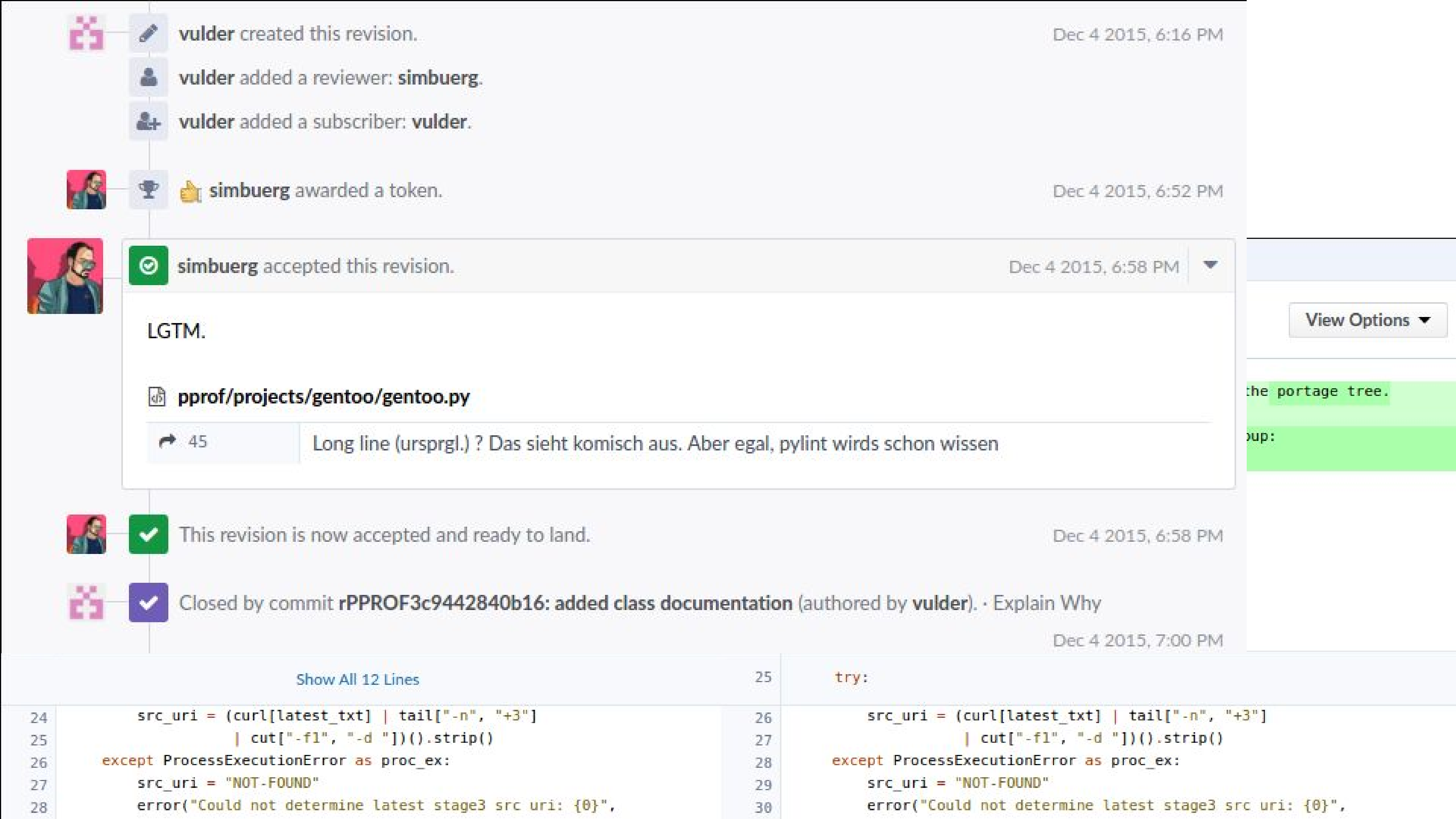
git hosting

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

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

```
he portage tree.
up:
```

```
    try:
```

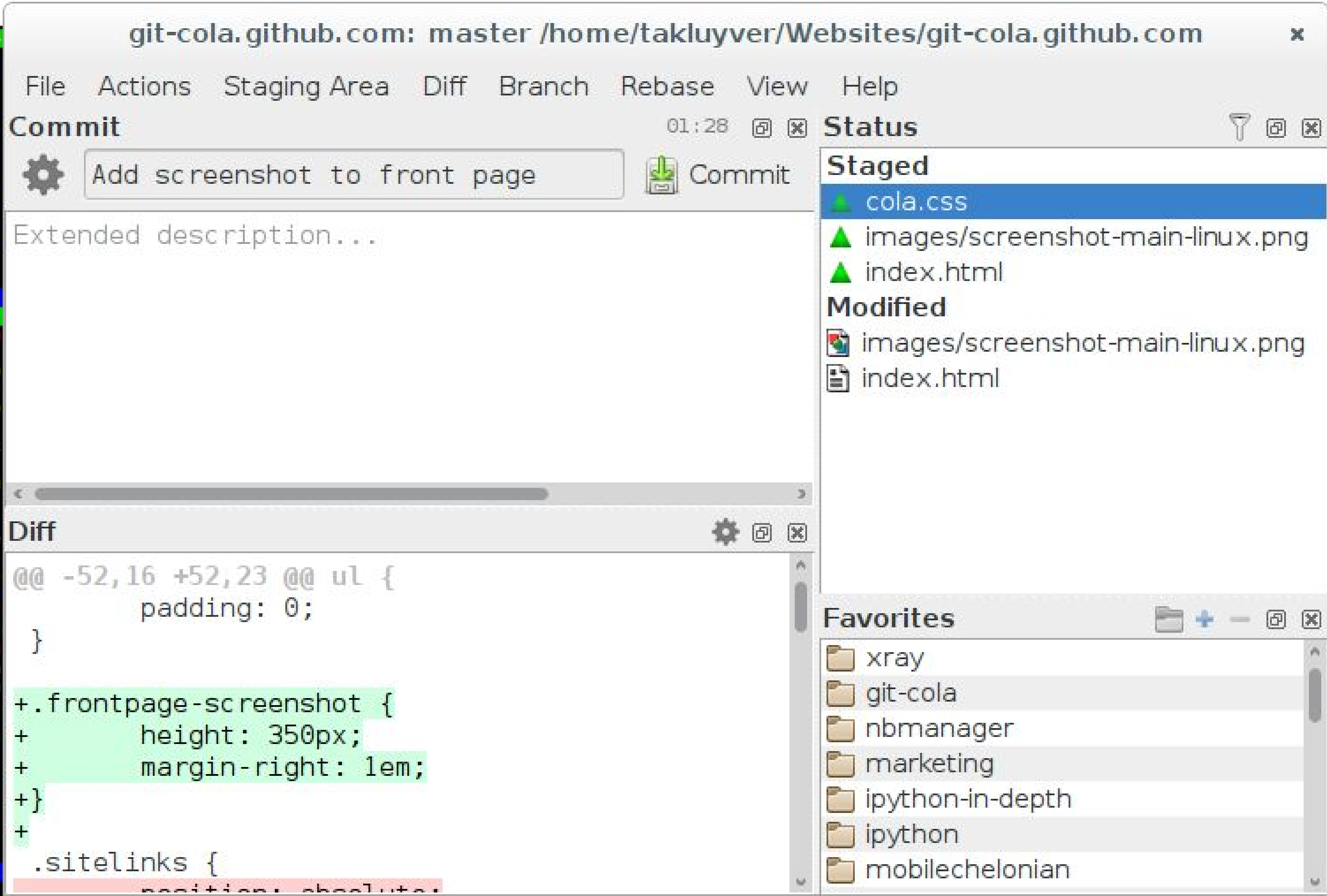
```
        src_uri = (curl[latest_txt] | tail["-n", "+3"]
                   | cut["-f1", "-d "])().strip()
    except ProcessExecutionError as proc_ex:
        src_uri = "NOT-FOUND"
        error("Could not determine latest stage3 src uri: {0}",
```

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

tooling

- gitk
- tig
- sourcetree
- git-cola

```
2016-04
2016-04
2016-04
2016-04
2016-04
2016-04
2016-04
2016-04
2016-04
2016-04
2016-04
2016-04

[main]
commit
Author:
AuthorD
Commit:
CommitD

     fxx

     Sig
---
 foo.c
 1 file

diff --
index b
--- a/f
+++ b/f
@@ -5,3
 int ba
   retu
 }
+
+void f
+}

[diff]
```



Add screenshot to front page

Extended description...

Staged
- cola.css
- images/screenshot-main-linux.png
- index.html

Modified
- images/screenshot-main-linux.png
- index.html

```
@@ -52,16 +52,23 @@ ul {
        padding: 0;
 }

+.frontpage-screenshot {
+       height: 350px;
+       margin-right: 1em;
+}
+
 .sitelinks {
        position: absolute;
```

Favorites:
- xray
- git-cola
- nbmanager
- marketing
- ipython-in-depth
- ipython
- mobilechelonian

tooling

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

gitattributes

.git/info/attributes | path/file spec settings

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

| word filter

```
#!/bin/bash
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
- git rerere

...

WOW, SUCH AUDIENCE
MUCH QUESTIONS, VERY RELEVANT

GIT'EM ALL