



THE LINUX FOUNDATION



NORTH AMERICA

Harnessing git's Superpowers for Code Navigation and Debugging

Matheus Bernardino

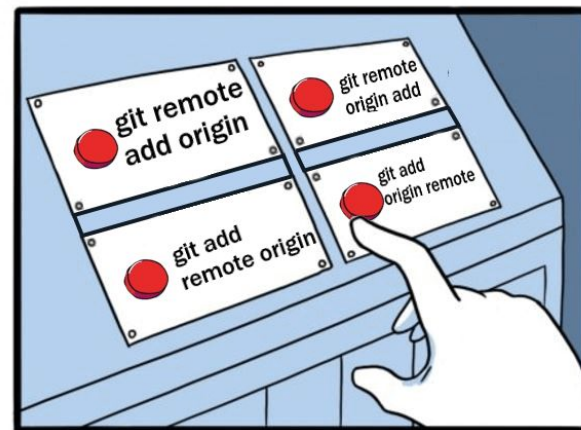
Software Engineer @ Qualcomm

<https://matheustavares.dev/>





<https://programmerhumor.io/>



JAKE-CLARK.TUMBLR

Part I : a peek under the hood



OPEN SOURCE SUMMIT

THE LINUX FOUNDATION

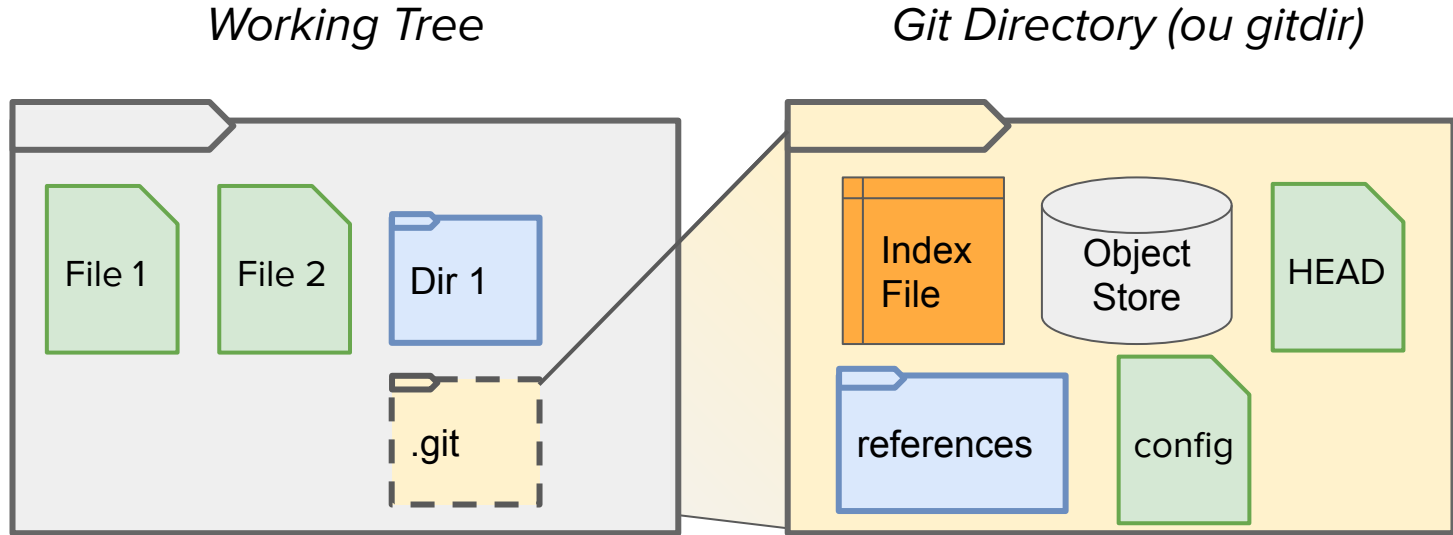
NORTH AMERICA



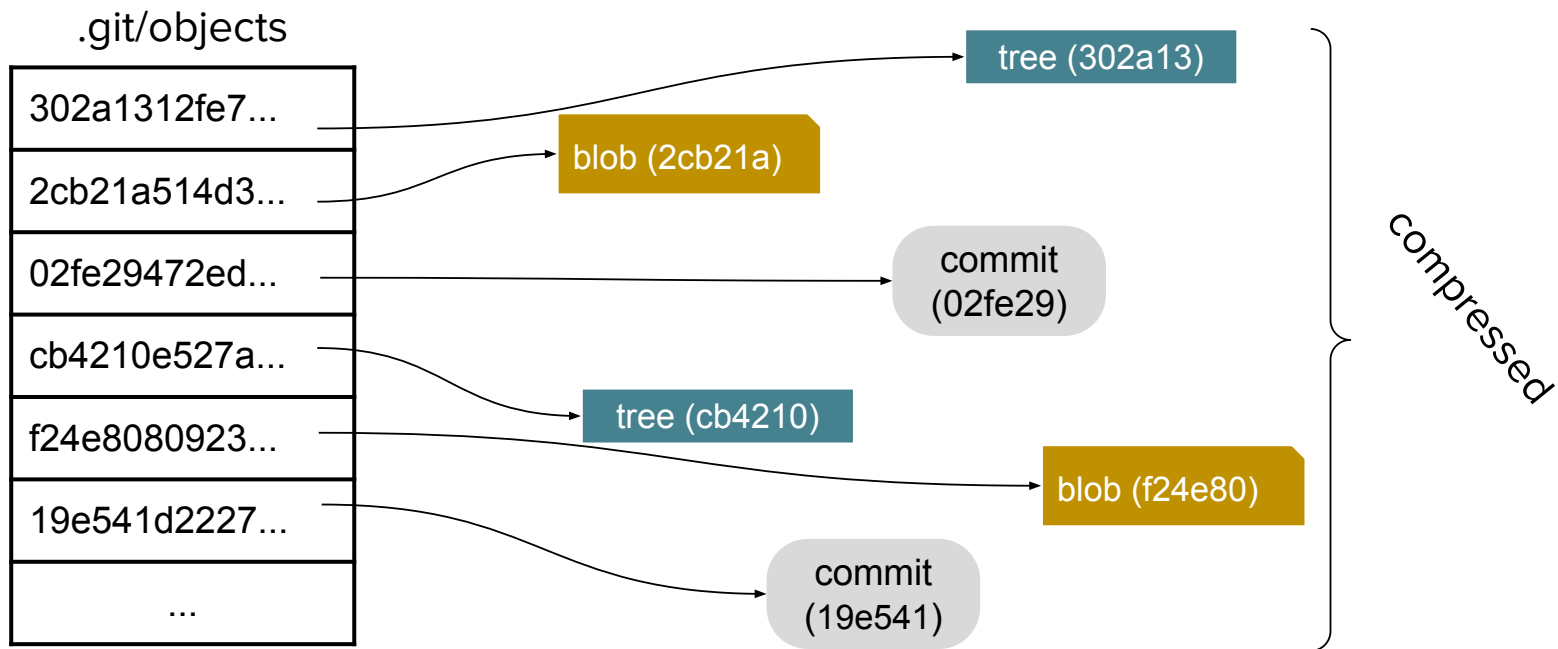
Embedded Linux
Conference



Repository layout



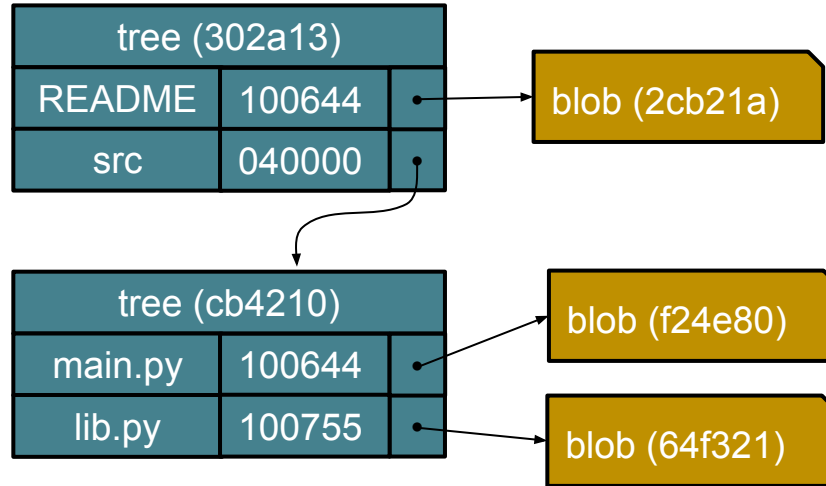
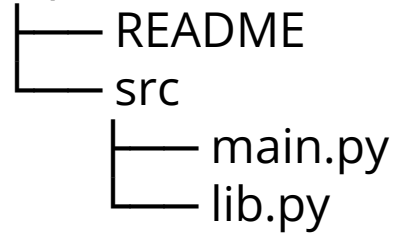
On-disk hash set



Object relations

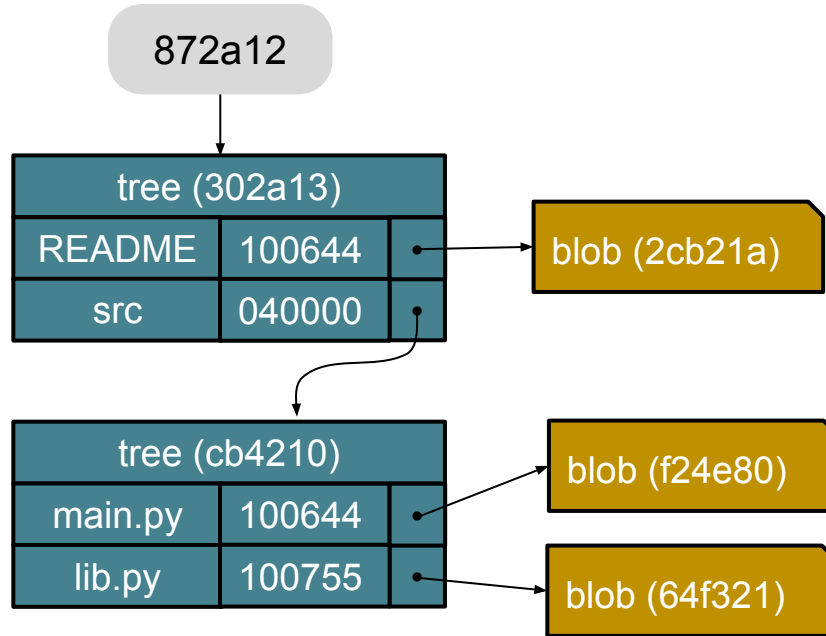
\$ tree repo

repo



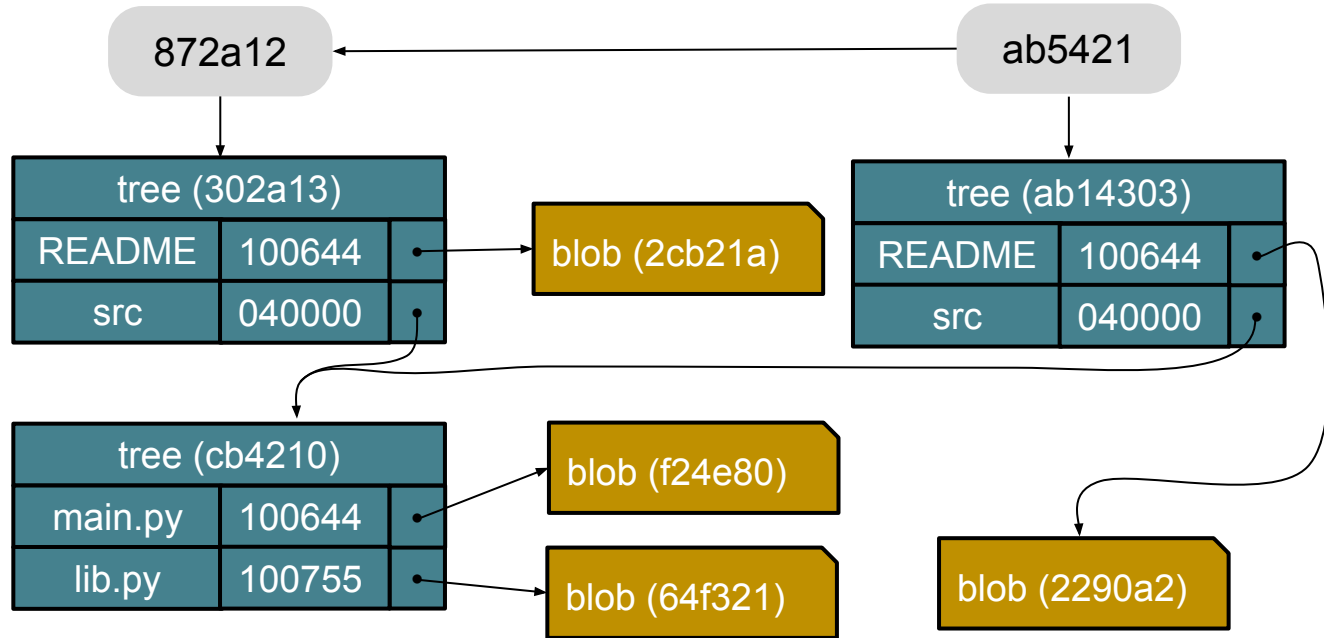
Object relations

A commit is a snapshot



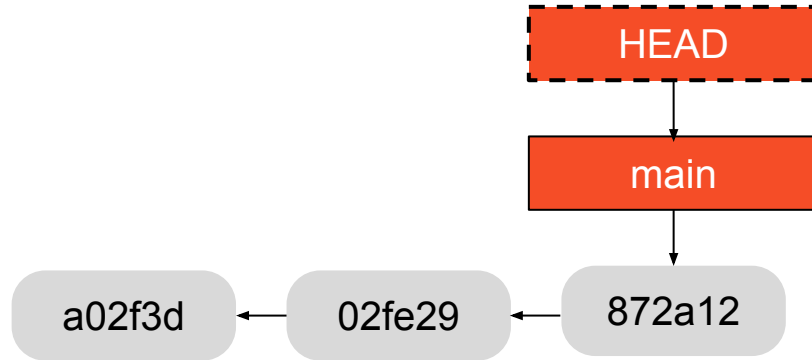
Object relations

```
$ vim README  
$ git commit
```



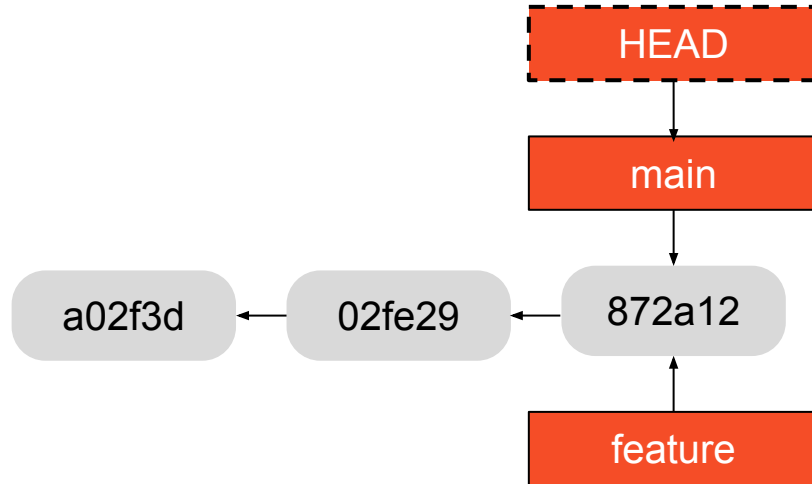
refs & commits

- branches
- tags
- HEAD



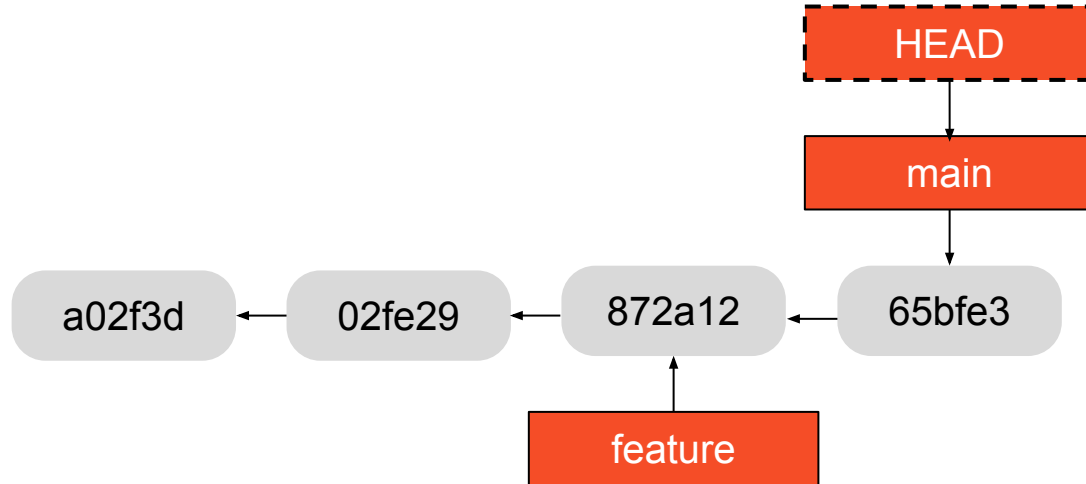
refs & commits

\$ git branch feature



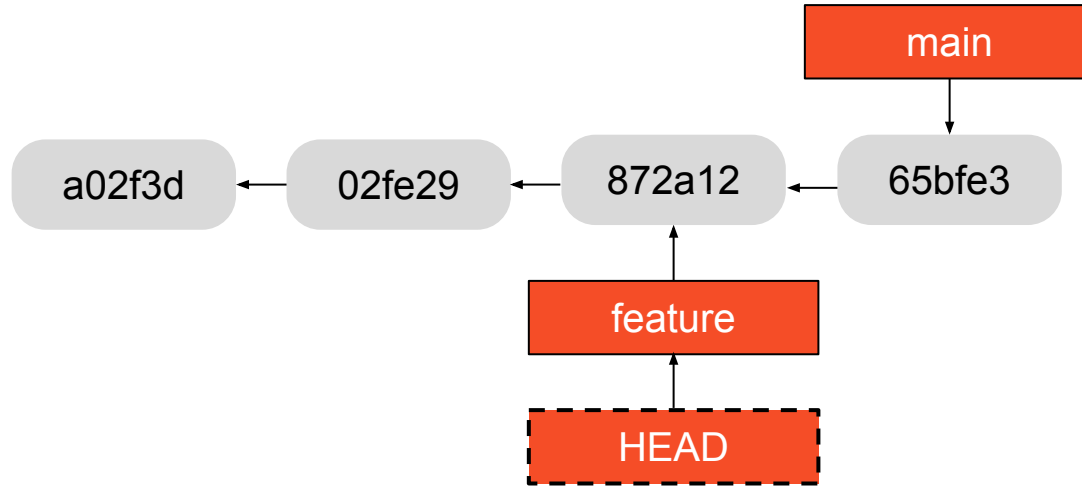
refs & commits

\$ git commit



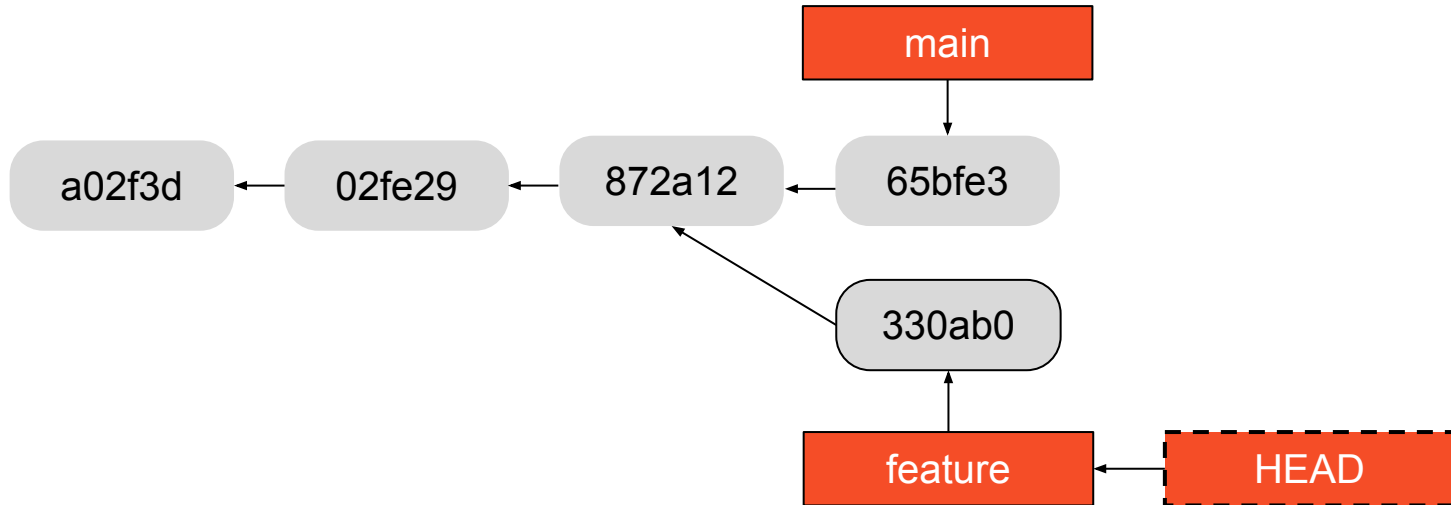
refs & commits

\$ git checkout feature



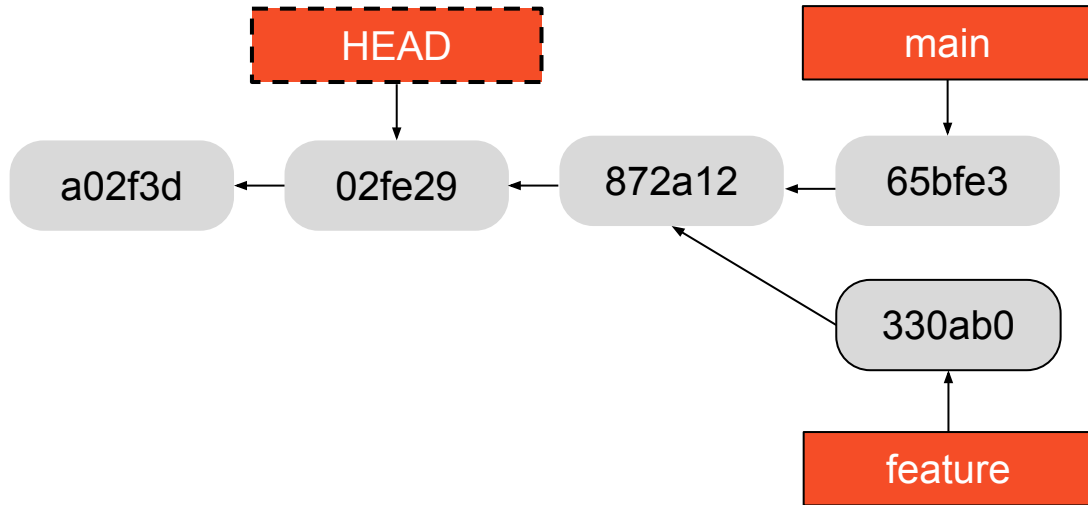
refs & commits

\$ git commit



refs & commits

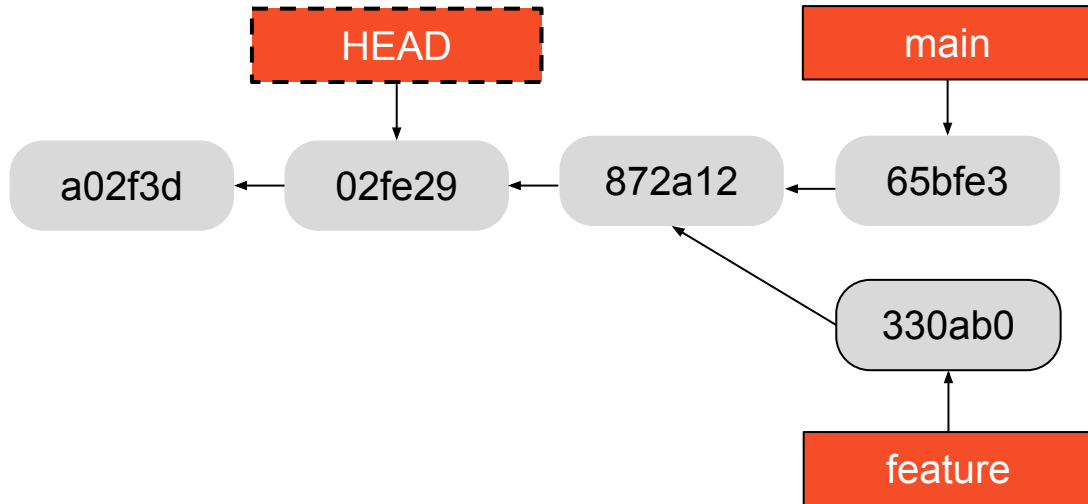
\$ git checkout 02fe29



refs & commits

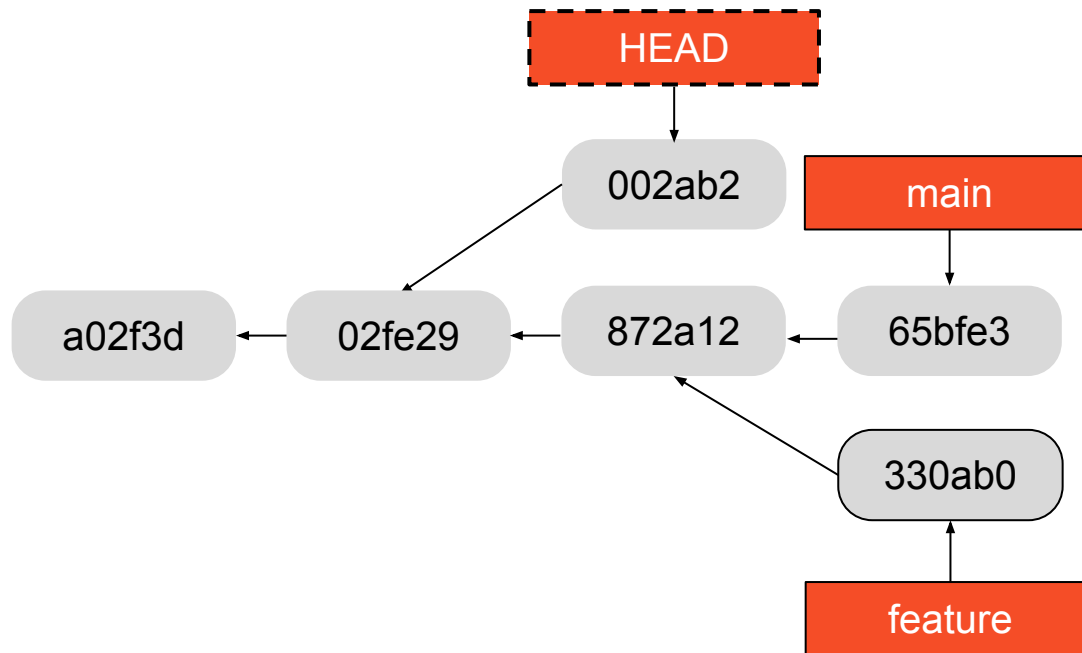
\$ git checkout 02fe29

You are in "detached HEAD" state.



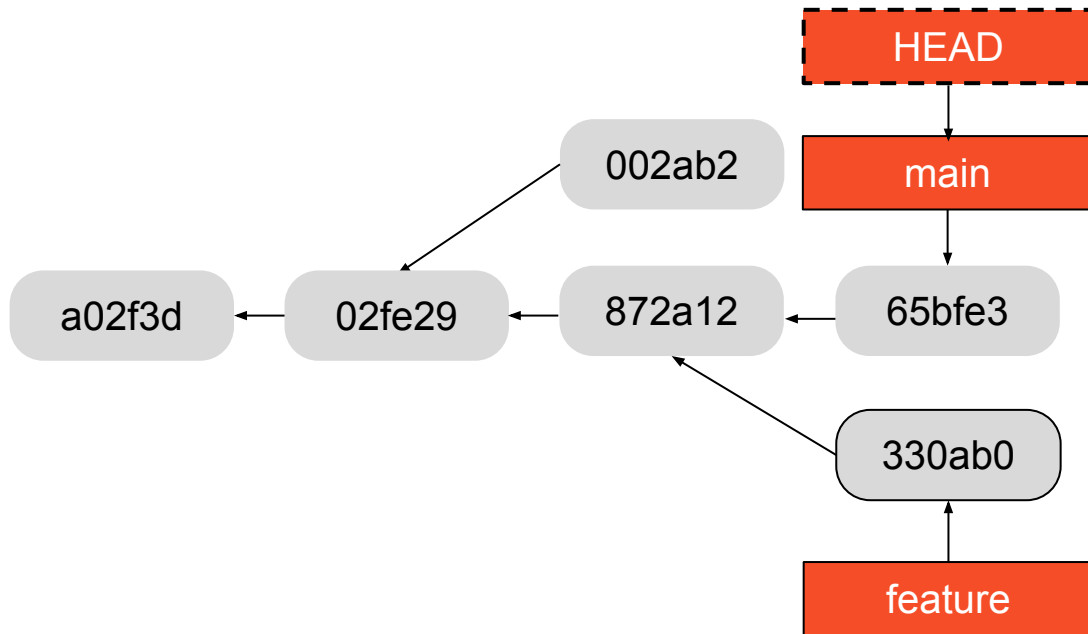
refs & commits

\$ git commit



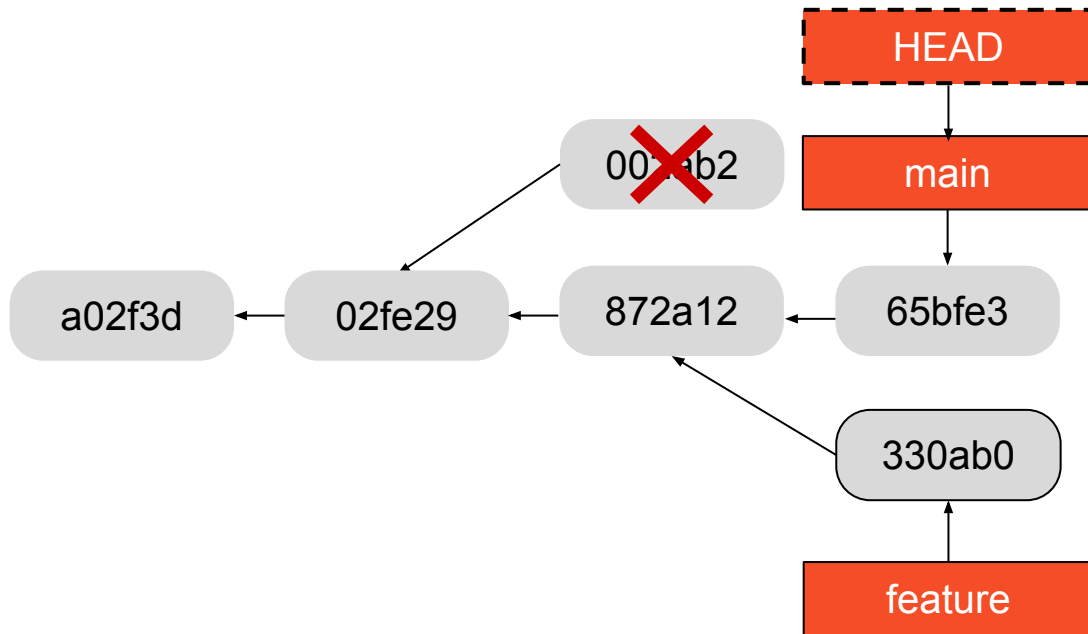
refs & commits

\$ git checkout main



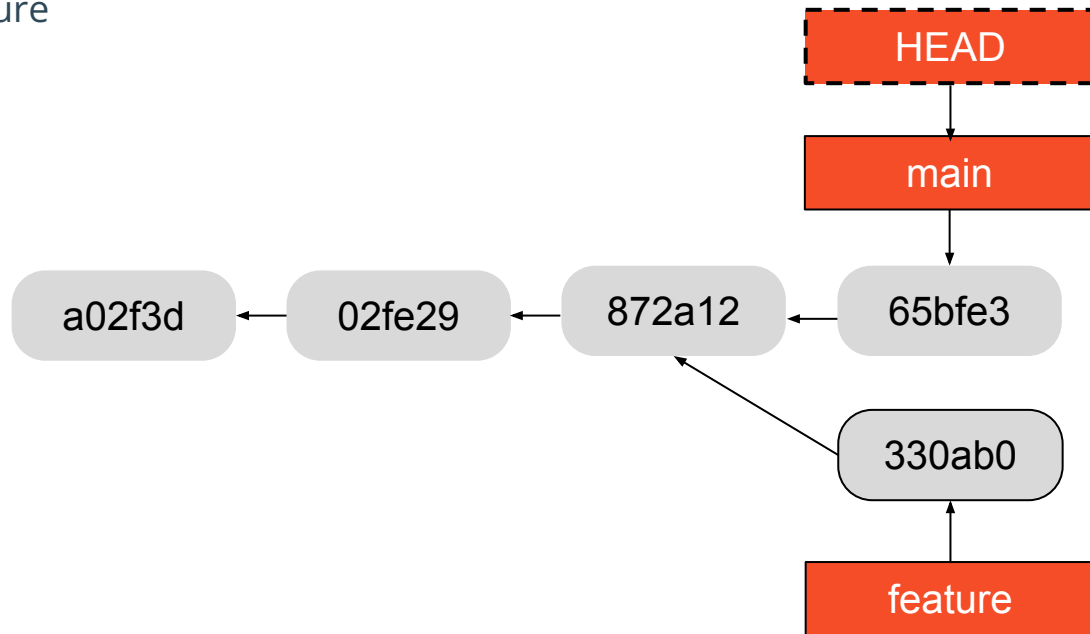
refs & commits

```
$ git gc --auto
```



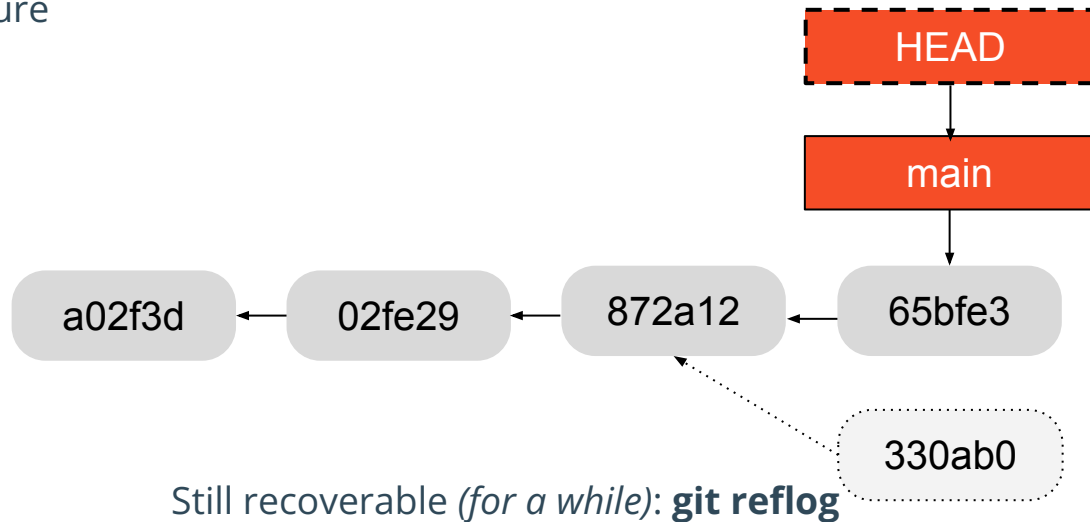
refs & commits

\$ git branch -D feature



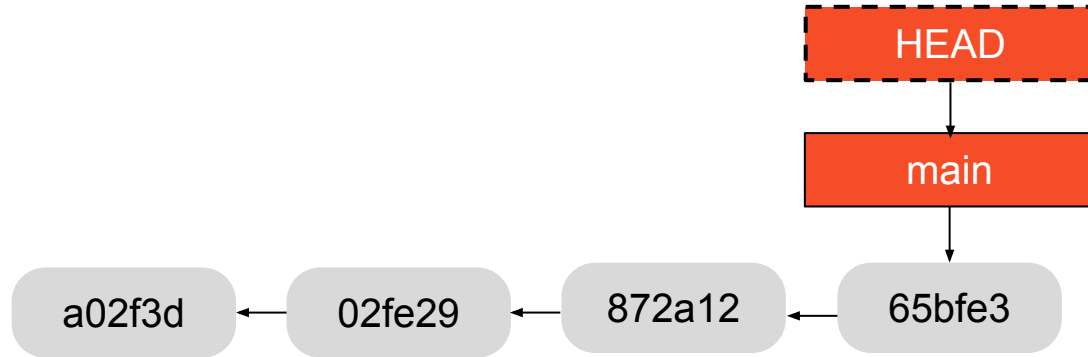
refs & commits

\$ git branch -D feature



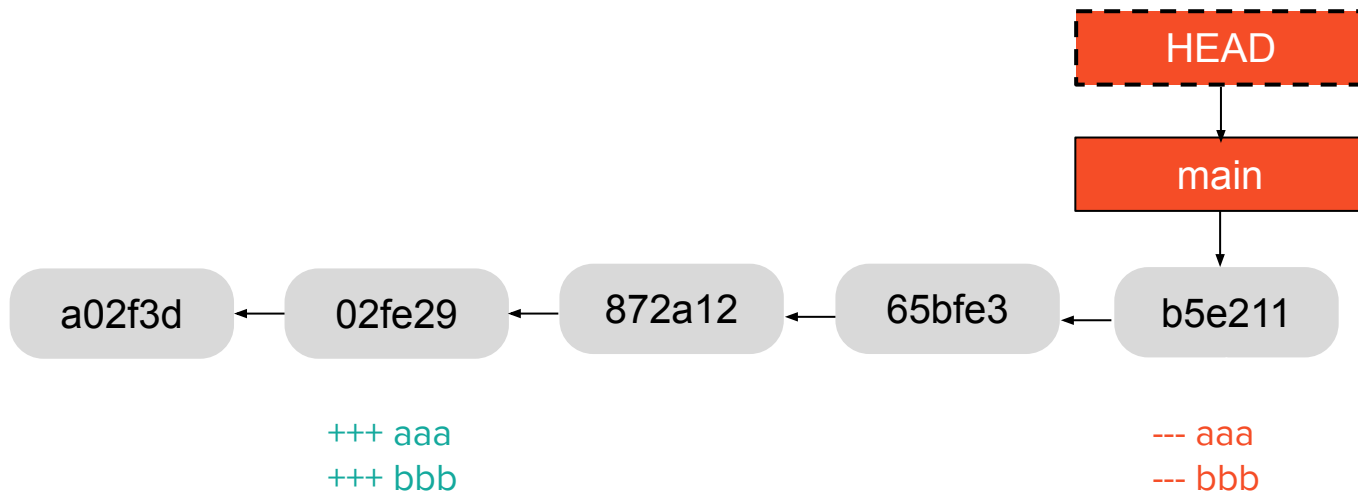
refs & commits

```
$ git revert 02fe29
```



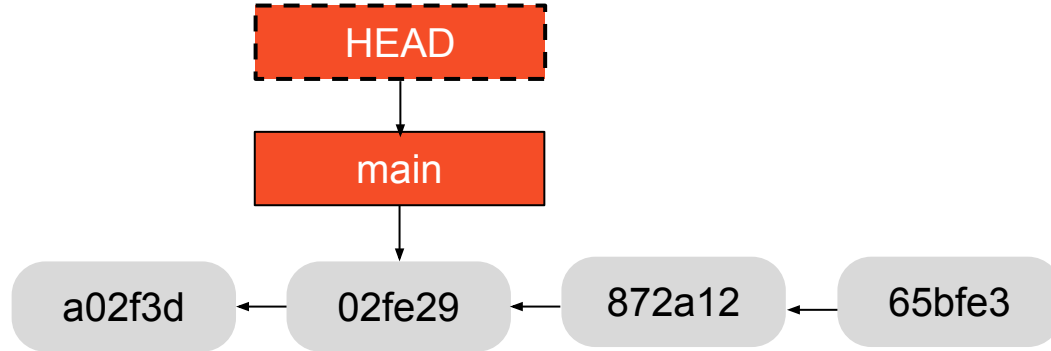
refs & commits

```
$ git revert 02fe29
```



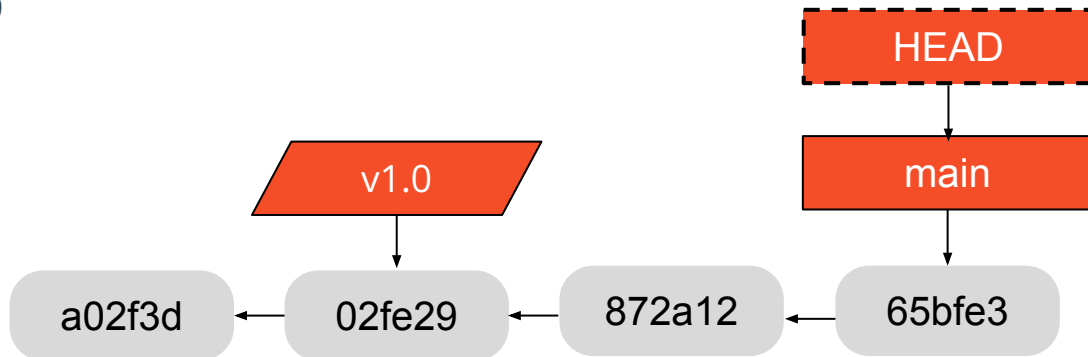
refs & commits

```
$ git reset 02fe29
```



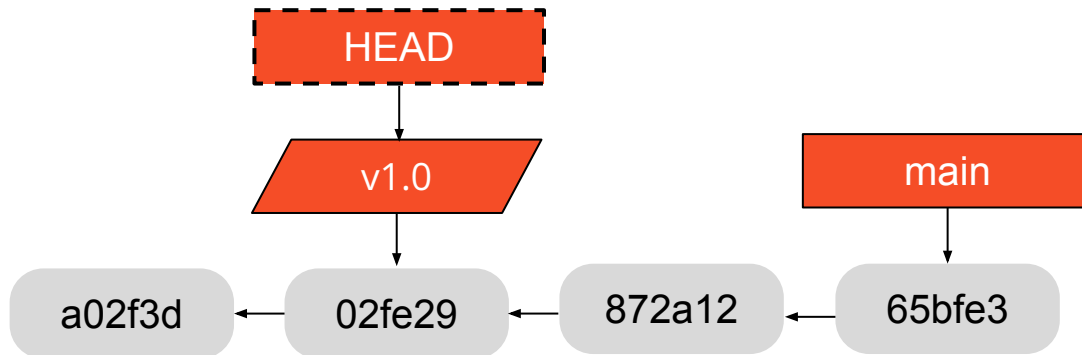
refs & commits

```
$ git tag v1.0 02fe29
```



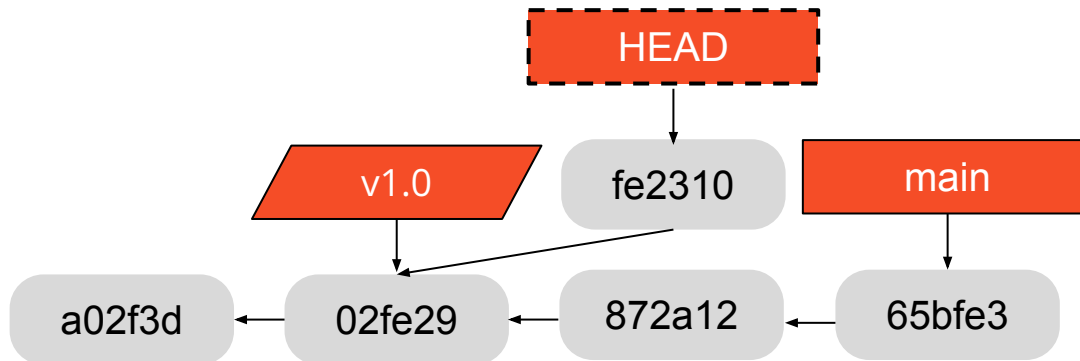
refs & commits

\$ git checkout v1.0



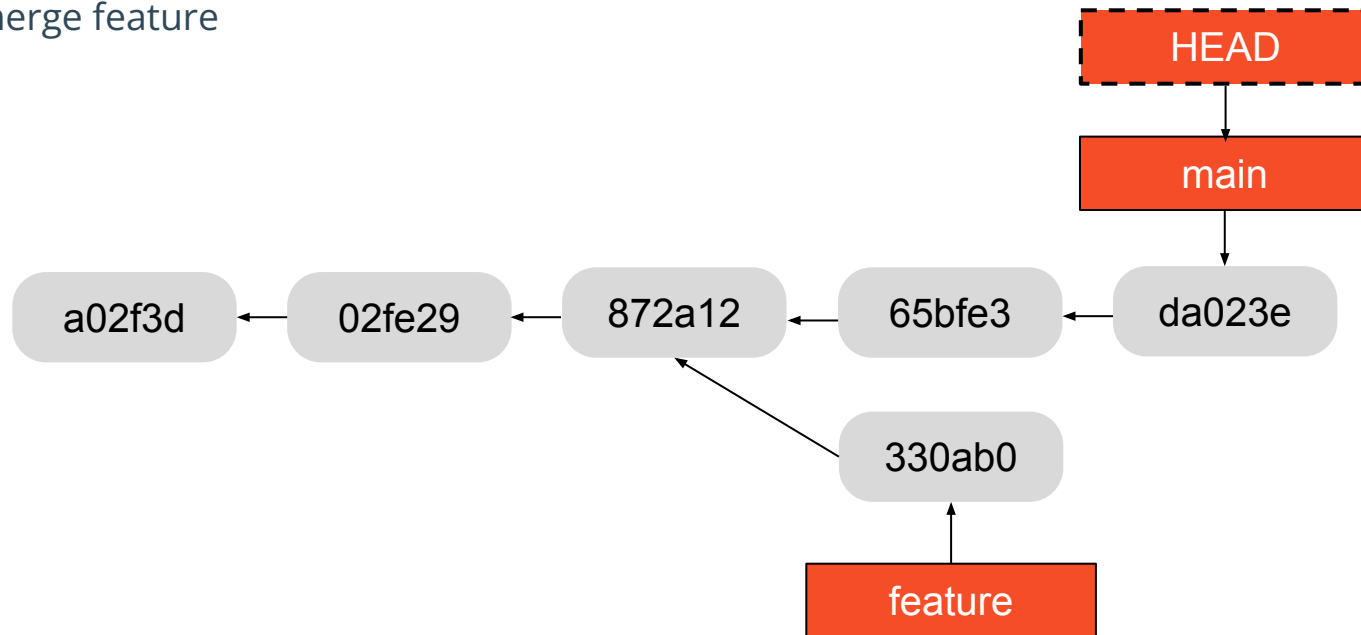
refs & commits

\$ git commit



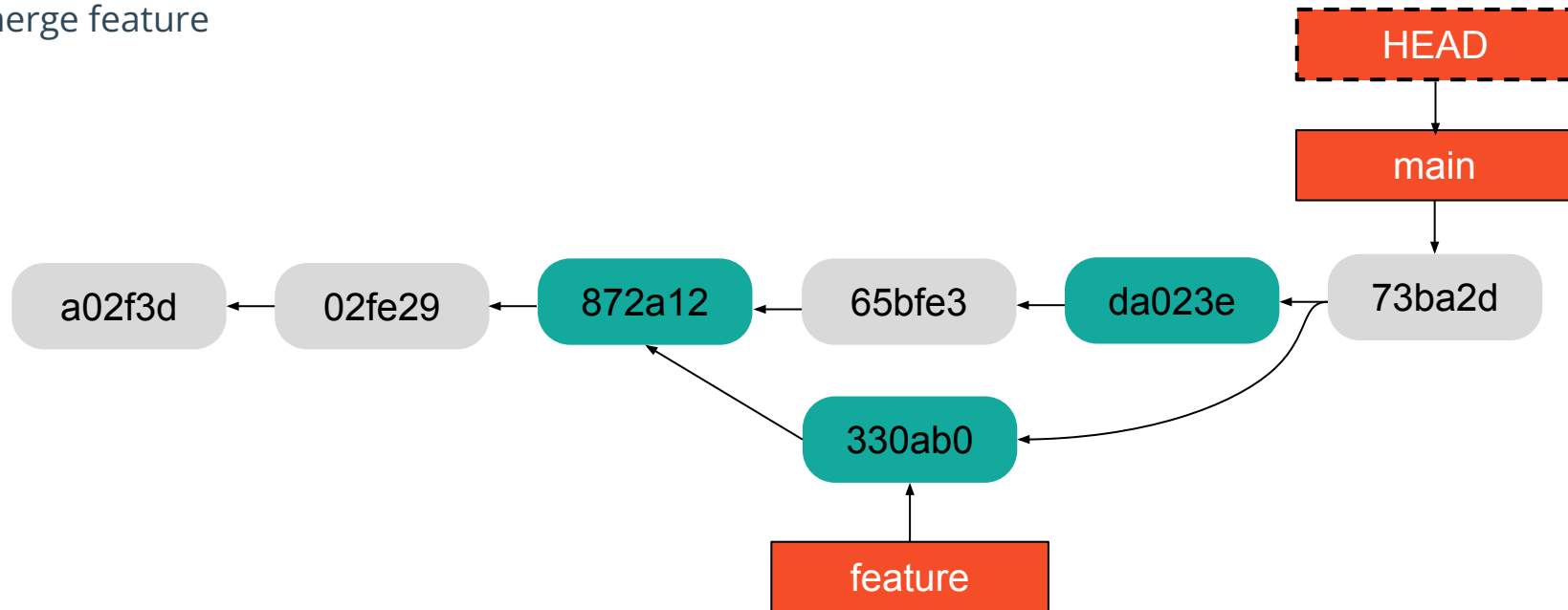
refs & commits

\$ git merge feature



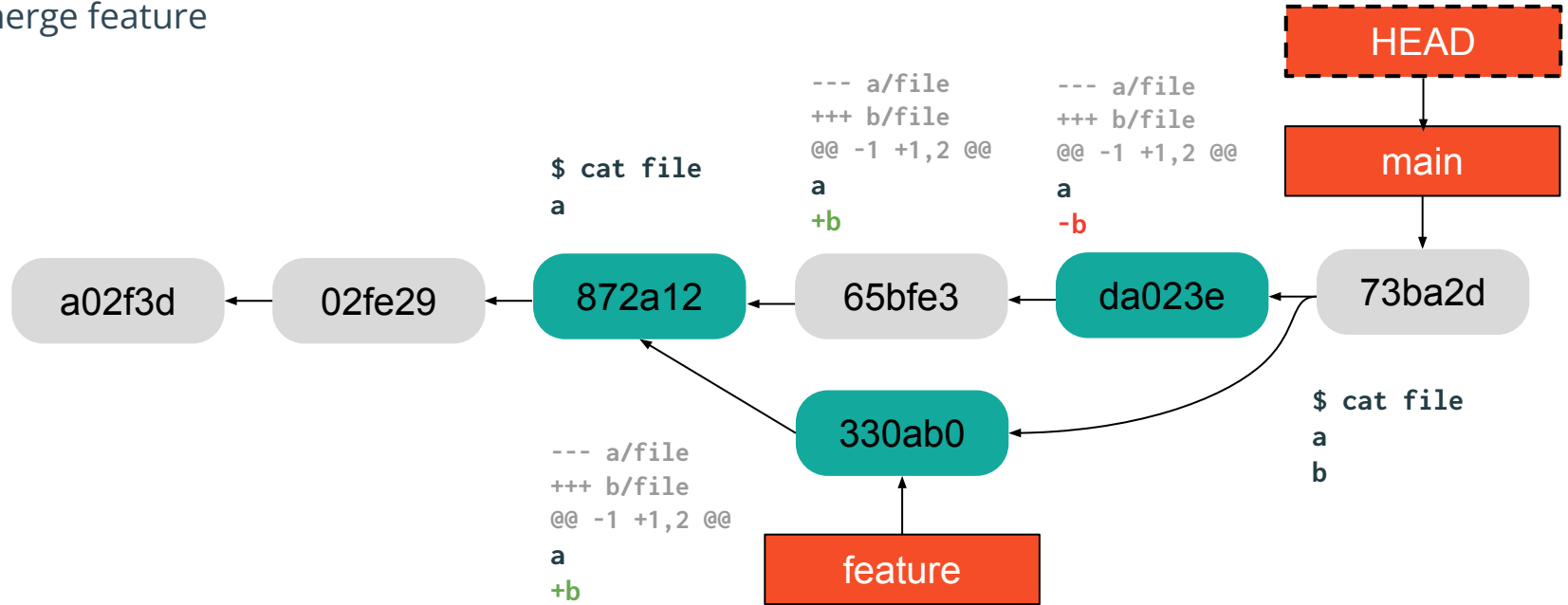
refs & commits

\$ git merge feature



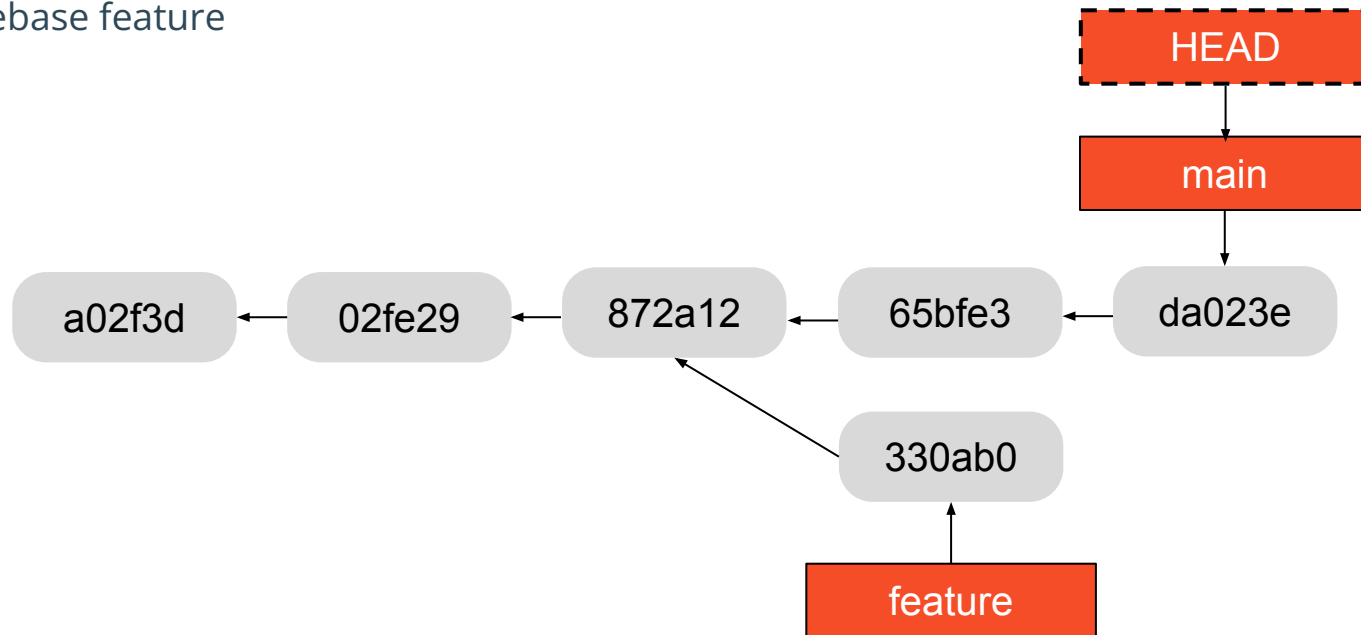
refs & commits

\$ git merge feature



refs & commits

\$ git rebase feature



Part II: git for navigation & debugging



THE LINUX FOUNDATION
OPEN SOURCE SUMMIT

NORTH AMERICA



Embedded Linux
Conference



pathspecs

- **Wildcards**
 - *.js
 - **/src/Makefile
 - test/**/*.*
- **Bracket expressions**
 - Logger.[ch]
- **Magic signatures**
 - :/README
 - :^*.md
 - :(icase)*.jpg
- **TIP:** '...' avoids shell expansion.
- More pathspec info [here](#).

grep

- Like GNU grep
 - Recursive by default
 - Honors *.gitignore* and *.git*
- Also **searches commits/trees**

grep

```
$ git grep show_submodule
builtin/ls-files.c:static void show_submodule(struct repository *superproj
builtin/ls-files.c:        show_submodule(repo, dir, ce->name);
diff.c:        show_submodule_diff_summary(o, one->path ? one->path : two-
diff.c:        show_submodule_inline_diff(o, one->path ? one->path : two-
submodule.c:static void show_submodule_header(struct diff_options *o,
submodule.c:void show_submodule_diff_summary(struct diff_options *o, const
submodule.c:    show_submodule_header(o, path, one, two, dirty_submodule,
submodule.c:void show_submodule_inline_diff(struct diff_options *o, const
submodule.c:    show submodule header(o, path, one, two, dirty submodule,
```

```
$ git grep show_submodule 752c0c24
752c0c24:diff.c:        show_submodule_summary(o->file, one ? one->pa
752c0c24:submodule.c:void show_submodule_summary(FILE *f, const char *path,
752c0c24:submodule.h:void show_submodule_summary(FILE *f, const char *path,
```

grep

```
$ git grep -n --heading --break show_submodule
builtin/ls-files.c
221:static void show_submodule(struct repository *superproject,
313:        show_submodule(repo, dir, ce->name);

diff.c
3473:        show_submodule_diff_summary(o, one->path ? one->path : two->path,
3481:        show_submodule_inline_diff(o, one->path ? one->path : two->path,

submodule.c
546:static void show_submodule_header(struct diff_options *o,
617:void show_submodule_diff_summary(struct diff_options *o, const char *path,
627:    show_submodule_header(o, path, one, two, dirty_submodule,
657:void show_submodule_inline_diff(struct diff_options *o, const char *path,
669:    show_submodule_header(o, path, one, two, dirty_submodule,
```

grep

```
$ git grep -W show_submodule
builtin/ls-files.c:221:static void show_submodule(struct repository *superproject,
builtin/ls-files.c-222-                struct dir_struct *dir, const char *path)
builtin/ls-files.c-223-{
builtin/ls-files.c-224- struct repository subrepo;
builtin/ls-files.c-225-
builtin/ls-files.c-226- if (repo_submodule_init(&subrepo, superproject, path, null_oid()))
builtin/ls-files.c-227-     return;
builtin/ls-files.c-228-
builtin/ls-files.c-229- if (repo_read_index(&subrepo) < 0)
builtin/ls-files.c-230-     die("index file corrupt");
builtin/ls-files.c-231-
builtin/ls-files.c-232- show_files(&subrepo, dir);
builtin/ls-files.c-233-
builtin/ls-files.c-234- repo_clear(&subrepo);
builtin/ls-files.c-235-}
--
```

blame

- **Who last changed this line?**
- **tig blame** for recursive/iterative blaming
- **Limitations:** last change is not always interesting
 - E.g. variable renamed

tig blame example

log

- **pathspec**
- **--since / --after** date filtering
- **--grep** searches in the commit message
- **hashA...hashB** limits between two commits
- **--graph** show graph
- **-p** show diff
- **--format** change output

log: pickaxe

- **log -G:** diff contains string
- **log -S:** diff changes #string occurrences

```
$ git log -S show_submodule --oneline --reverse
188dce131f ls-files: use repository object
fd47ae6a5b diff: teach diff to display submodule difference
8e6df65015 submodule: refactor show_submodule_summary with
752c0c2492 Add the --submodule option to the diff option fa
```

data recovery

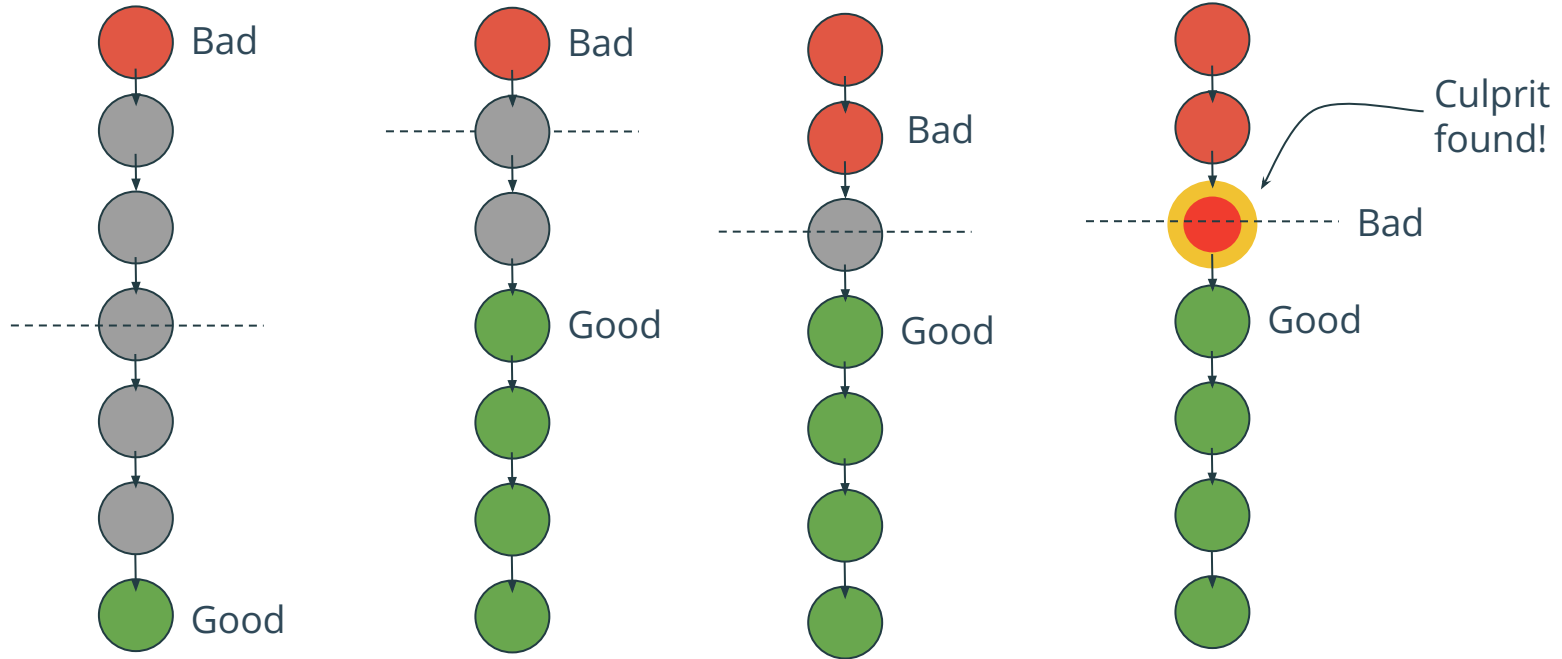
- **reflog**: logging branches, HEAD, etc.

```
$ git reflog HEAD
b1485644f9 (HEAD -> master, origin/master, origin/main, origin/HEAD) HEAD@{0}: pull:
7c2ef319c5 (mailmap) HEAD@{1}: checkout: moving from mailmap to master
7c2ef319c5 (mailmap) HEAD@{2}: rebase (finish): returning to refs/heads/mailmap
7c2ef319c5 (mailmap) HEAD@{3}: pull: Fast-forward
bd5df96b79 HEAD@{4}: checkout: moving from master to mailmap
bd5df96b79 HEAD@{5}: pull: Fast-forward
79f2338b37 HEAD@{6}: checkout: moving from 87d8ec31e8cb14c4076abc93bf73f070f3951834 t
```

- **git fsck --lost-found**: seek and save unreachable objects

bisect

Which change introduced a bug, or **changed a behavior**



References

1. **Git Archeology:**

<https://bmuskalla.github.io/blog/2021-01-12-git-archeology/>

2. **Git as debugging tool:**

<https://lucasoshiro.github.io/posts-en/2023-02-13-git-debug/>

3. **Pro Git book**

<https://git-scm.com/book/en/v2>

4. **How to Write a Git Commit Message:**

<https://chris.beams.io/posts/git-commit/>

Thank you!

<https://matheustavares.dev/>

