Git - Fast Version Control System

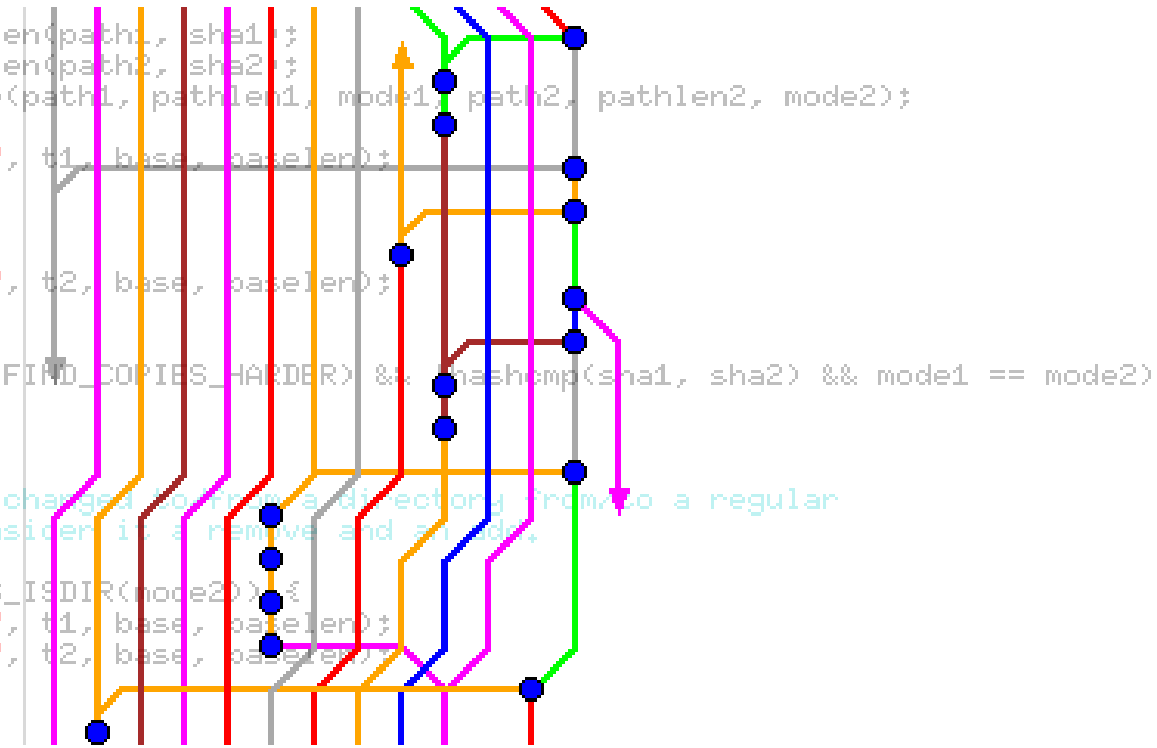
Sebastian Harl
<Sebastian.Harl@sternwarte.uni-erlangen.de>

Astronomisches Institut der Universität Erlangen-Nürnberg

17. Oktober 2008

```
»···pathlen1 = tree_entry_len(path1, sha1);
»···pathlen2 = tree_entry_len(path2, sha2);
»···cmp = base_name_compare(path1, pathlen1, mode1, path2, pathlen2, mode2);
»···if (cmp < 0) {
»···»···show_entry(opt, "-", t1, base, baselen);
»···»···return -1;
»···}
»···if (cmp > 0) {
»···»···show_entry(opt, "+", t2, base, baselen);
»···»···return 1;
»···}
»···if (!DIFF_OPT_TST(opt, FIND_COPIES_HARDER) && !hashcmp(sha1, sha2) && mode1 == mode2)
»···»···return 0;

»···/*
»··· * If the filemode has changed to/from a directory from/to a regular
»··· * file, we need to consider it a remove and an add.
»··· */
»···if (S_ISDIR(mode1) != S_ISDIR(mode2)) {
»···»···show_entry(opt, "-", t1, base, baselen);
»···»···show_entry(opt, "+", t2, base, baselen);
»···»···return 0;
»···}
```



Was ist Git?

- VCS (Version Control System)
- dezentral
- schnell und effizient
- kryptographisch gesichert
- „Toolkit design“
- OpenSource (GPLv2)

Geschichte

- ursprünglich von Linus Torvalds geschrieben (2005)
- aktuell von Junio C. Hamano gepflegt
- große Entwickler-Gemeinde
- 14. Februar 2007: Git 1.5.0
- mittlerweile weite Verbreitung (Linux Kernel, Ruby on Rails, WINE, X.org, Debian, collectd ;-))

Community

- Webseite: `http://git.or.cz/`
- Mailing-Liste: `git@vger.kernel.org` (auch über Gmane)
- IRC: `#git` auf FreeNode

Inhalt

Die Git Objekt-Datenbank

It's show time! ;-)

Arbeiten mit Git

Übersicht

- >> 100 einzelne Befehle
- „Porcelains“ und „Plumbing“
- Dokumentation als Manpages - `git(7)`
- `git help`, `git <command> -h`
- sehr nette Tab-Completion in der zsh
- Benutzer Handbuch:
  `http://www.kernel.org/pub/software/scm/git/docs/user-manual.html`

Sich Git vorstellen

- `git config --global user.name <Dein Name>`
- `git config --global user.email <du@deine-domain.tld>`

Bund und in Farbe

- `git config --global color.branch auto`
- `git config --global color.diff auto`
- `git config --global color.status auto`

Repositories erstellen

```
$ mkdir project
$ cd project
$ git init
Initialized empty Git
repository in .../.git/
```

Repositories erstellen

```
$ mkdir project
$ cd project
$ git init
Initialized empty Git
repository in .../.git/
```

```
$ mkdir project-repo
$ svnadmin create project-repo
$ svn checkout \
    file:///‘pwd‘/project-repo \
    project
Checked out revision 0.
```

Repositories erstellen

```
$ mkdir project
$ cd project
$ git init
Initialized empty Git
repository in .../.git/
```

```
$ mkdir project-repo
$ svnadmin create project-repo
$ svn checkout \
    file:///‘pwd‘/project-repo \
    project
Checked out revision 0.
```

```
$ git clone <rep>
...
```

Repositories erstellen

```
$ mkdir project
$ cd project
$ git init
Initialized empty Git
repository in .../.git/
```

```
$ mkdir project-repo
$ svnadmin create project-repo
$ svn checkout \
    file:///‘pwd‘/project-repo \
    project
Checked out revision 0.
```

```
$ git clone <rep>
...
```

-

Editieren, ändern, ...

```
$ vim foo bar
$ git add foo bar
```

- `add`, `rm`, `mv`

Editieren, ändern, ...

```
$ vim foo bar
$ git add foo bar
```

- add, rm, mv

```
$ vim foo bar
$ svn add foo bar
```

- add, delete, move

Editieren, ändern, ...

```
$ vim foo bar
$ git add foo bar
```

- add, rm, mv

```
$ vim foo bar
$ svn add foo bar
```

- add, delete, move

```
$ git commit
```

Editieren, ändern, ...

```
$ vim foo bar
$ git add foo bar
```

- add, rm, mv

```
$ git commit
```

```
$ vim foo bar
$ svn add foo bar
```

- add, delete, move

```
$ svn commit
```

Editieren, ändern, ...

```
$ vim foo bar
$ git add foo bar
```

- add, rm, mv

```
$ git commit
```

```
$ git reset --hard HEAD^
```

- reset, revert, checkout

```
$ vim foo bar
$ svn add foo bar
```

- add, delete, move

```
$ svn commit
```

Editieren, ändern, ...

```
$ vim foo bar
$ git add foo bar
```

- add, rm, mv

```
$ git commit
```

```
$ git reset --hard HEAD^
```

- reset, revert, checkout

```
$ vim foo bar
$ svn add foo bar
```

- add, delete, move

```
$ svn commit
```

```
$ svn merge -r 124:123 .
```

- revert, merge

Commit Meldungen schreiben

- Einzeilige, kurze (< 80, optimal < 50 Zeichen) Zusammenfassung
- Leerzeile
- Detaillierte Beschreibung
- nicht vorgeschrieben, aber „common practice“ und von vielen Tools erwartet

```
git-remote: do not use user input in a printf format string

'git remote show' substituted the remote name into a string that
was later used as a printf format string. If a remote name contains
a printf format specifier like this:

    $ git remote add foo%sbar .

then the command

    $ git remote show foo%sbar

would print garbage (if you are lucky) or crash. This fixes it.
```

Änderungen und Historie betrachten

```
$ git status
$ git diff
```

Änderungen und Historie betrachten

```
$ git status
$ git diff
```

```
$ svn status
$ svn diff | less
```

Änderungen und Historie betrachten

```
$ git status
$ git diff
```

```
$ svn status
$ svn diff | less
```

```
$ git log
$ tig
```

Änderungen und Historie betrachten

```
$ git status
$ git diff
```

```
$ svn status
$ svn diff | less
```

```
$ git log
$ tig
```

```
$ svn log | less
```

Änderungen und Historie betrachten

```
$ git status
$ git diff
```

```
$ svn status
$ svn diff | less
```

```
$ git log
$ tig
```

```
$ svn log | less
```

```
$ git show
$ git show HEAD:foo
```

Änderungen und Historie betrachten

```
$ git status
$ git diff
```

```
$ svn status
$ svn diff | less
```

```
$ git log
$ tig
```

```
$ svn log | less
```

```
$ git show
$ git show HEAD:foo
```

```
$ svn diff -c 123
$ svn cat -r 123 foo
```

Änderungen und Historie betrachten

```
$ git status
$ git diff
```

```
$ svn status
$ svn diff | less
```

```
$ git log
$ tig
```

```
$ svn log | less
```

```
$ git show
$ git show HEAD:foo
```

```
$ svn diff -c 123
$ svn cat -r 123 foo
```

```
$ git tag
```

Änderungen und Historie betrachten

```
$ git status
$ git diff
```

```
$ svn status
$ svn diff | less
```

```
$ git log
$ tig
```

```
$ svn log | less
```

```
$ git show
$ git show HEAD:foo
```

```
$ svn diff -c 123
$ svn cat -r 123 foo
```

```
$ git tag
```

```
$ svn copy trunk tags/v123
```

Branching und Merging

```
$ git checkout -b new-branch
```

Branching und Merging

```
$ git checkout -b new-branch
```

```
$ svn copy trunk \
    branches/new-branch
```

Branching und Merging

```
$ git checkout -b new-branch
```

```
$ svn copy trunk \
    branches/new-branch
```

```
$ git branch
  master
* new-branch
```

Branching und Merging

```
$ git checkout -b new-branch
```

```
$ svn copy trunk \
    branches/new-branch
```

```
$ git branch
  master
* new-branch
```

```
$ svn list <repo>/branches
new-branch
```

Branching und Merging

```
$ git checkout -b new-branch
```

```
$ svn copy trunk \
    branches/new-branch
```

```
$ git branch
  master
* new-branch
```

```
$ svn list <repo>/branches
new-branch
```

```
$ git merge master
$ git rebase master
```

Branching und Merging

```
$ git checkout -b new-branch
```

```
$ svn copy trunk \
    branches/new-branch
```

```
$ git branch
  master
* new-branch
```

```
$ svn list <repo>/branches
new-branch
```

```
$ git merge master
$ git rebase master
```

```
$ svn merge <some magic>
-
```

Arbeiten mit anderen Repositories

```
$ git clone <rep>
```

Arbeiten mit anderen Repositories

```
$ git clone <rep>
```

```
$ git pull
$ git push
```

- `git format-patch`, `git send-mail`

Arbeiten mit anderen Repositories

```
$ git clone <rep>
```

```
$ git pull
$ git push
```

- `git format-patch`, `git send-mail`

```
$ git remote add foo <rep>
```

Unterstützte Protokolle

- lokal: `/path/to/repository/`
- http: `http://domain.tld/repository.git`
- git: `git://domain.tld/repository.git`
- ssh: `domain.tld:path/to/repository/`

Repository-Pflege

```
$ git gc
```

Frontends

- `tig`
- `gitk`
- `git gui`

Git Goodies

`git reflog`

`git rebase -i`

`git commit --amend`

`git add -p`

`git stash`

git bisect

`git cherry`

Fragen?

History:

- 2006/10/01: LUSC Workshop Weekend 2006
- 2008/10/04: LUSC Workshop Weekend 2008
- 2008/10/17: fpipe Team Schulung