Git - Fast Version Control System

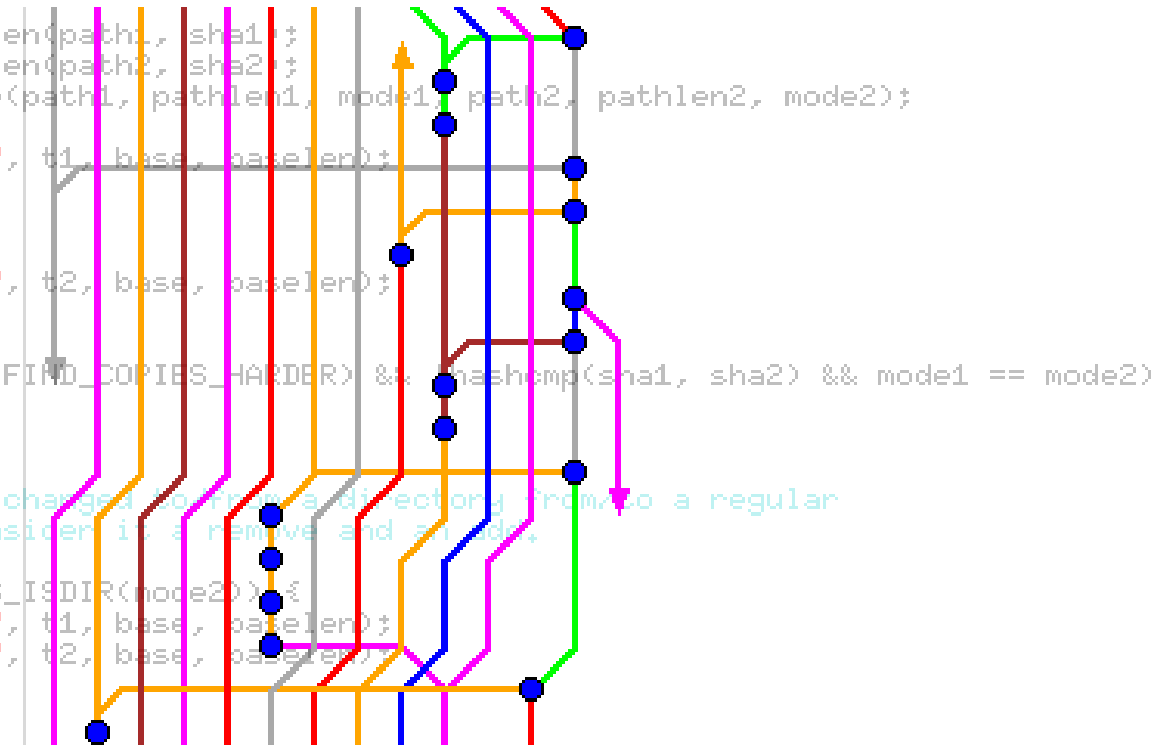
Sebastian Harl
<tokkee@lusc.de>

LUSC Workshop Weekend 2008

04. 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`
- 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>`

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/
```

```
$ 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`

```
$ git commit
```

Editieren, ändern, ...

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

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

```
$ git commit
```

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

- `reset`, `revert`, `checkout`

Änderungen und Historie betrachten

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

Änderungen und Historie betrachten

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

```
$ git log
$ tig
```

Änderungen und Historie betrachten

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

```
$ git log
$ tig
```

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

Änderungen und Historie betrachten

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

```
$ git log
$ tig
```

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

```
$ git tag
```

Branching und Merging

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

Branching und Merging

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

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

Branching und Merging

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

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

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

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`

Fragen?

History:

- 2006/10/01: LUSC Workshop Weekend 2006
- 2008/10/04: LUSC Workshop Weekend 2008