

VERSIONS- KONTROLLE MIT GIT

WER BIN ICH?

- Senior Software R&D Engineer @ Baylibre
- PWM subsystem maintainer
- Debian Entwickler
- Flug Mailingliste + Webserver
- ukraine on libera and OFTC
- PGP: E2DCDD9132669BD6

WOZU VERSIONSKONTROLLE?

- Releasemanagement
- Entwicklerteam
- Was hat sich wann und warum wie geändert?

GESCHICHTE

commit e83c5163316f89bfbde7d9ab23ca2e25604af290

Author: Linus Torvalds <torvalds@linux-foundation.org>

Date: Thu Apr 7 15:13:13 2005 -0700

Initial revision of "git", the information manager from hell

GESCHICHTE II

```
commit 1da177e4c3f41524e886b7f1b8a0c1fc7321cac2 (tag: v2.6.12-  
Author: Linus Torvalds <torvalds@ppc970.osdl.org>  
Date: Sat Apr 16 15:20:36 2005 -0700
```

```
Linux-2.6.12-rc2
```

```
Initial git repository build. I'm not bothering with the full  
history, even though we have it. We can create a separate  
"historical" git archive of that later if we want to, and in  
the meantime it's about 3.2GB when imported into git - space  
that would just make the early git days unnecessarily  
complicated, when we don't have a lot of good infrastructure  
for it.
```

```
Let it rinl
```

EIGENSCHAFTEN

- Verteiltes VCS (=> keine Revisionsnummern, sha1-Hashes)
- flott auch für große Repositories
- Marktanteil: 87.6 % (laut <https://6sense.com/tech/version-control/git-market-share>)
- freie Lizenz
- Entwicklung wie Linux über Mailingliste

GIT EINRICHTUNG

```
sudo apt install git  
git config --global user.name "Uwe Kleine-König"  
git config --global user.email "uwe@kleine-koenig.org"
```

NEUES PROJEKT

```
$ mkdir myproject  
$ cd myproject  
$ git init
```

COMMIT ERSTELLEN

```
$ vim somefile  
$ git add somefile  
$ git commit -m 'somefile: description'
```

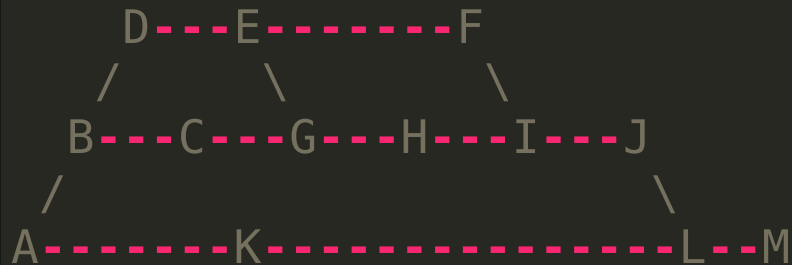
COMMIT INHALT

- Metadaten: Autor, Committer, Datum
- Commit Message
- Tree
- Parent (0, 1, oder >1)
- optional: OpenPGP Signatur

BRANCHING

- Standard Branch: main
- Neuer Branch

```
git switch -c fix-issue42
```



MERGEN

```
$ git merge bugfix
```

LOG

```
$ git log --graph --oneline
* 15130f84c70a (HEAD -> main, tag: v2023.09.0) Release microco
*   0cb882af7fdb Merge branch 'negotiate-binary-transmission'
|\
| * 2fdc58e5ef19 (ukl/negotiate-binary-transmission) telnet: D
* |   c41d017f22ae Merge branch 'dont-send-do-comport' of http
|\ \
| | /
| / |
| * ac3dd0836945 (ukl/dont-send-do-comport) telnet: Don't send
* | 679049a610f7 telnet: Don't quote IAC for DONT and WONT rep
| /
*   5a3bc9d127fd Merge branch 'quote-IAC-on-send' of https://g
```

LOG II

- gitk
- tig

TAGS

- `git tag -s -m "Release $myproject 1.0" v1.0`

DEMO

VERTEILTE REPOSITORIES

```
git clone https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux
cd linux
git remote add spi git://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux
git remote update
```

BEFÜLLEN VON ENTFERNTEN REPOS

```
git push spi fix-issue42
```

MERGEN VON ENTFERNTEN REPOS

Pull request:

<https://lore.kernel.org/lkml/8c0c7f276b6ea8988bd8b824>

```
git pull https://git.kernel.org/pub/scm/linux/kernel/git/broon
```

FORGES

- bitbucket
- codeberg.org
- github
- gitlab
- sr.ht
- ...

SELBST HOSTEN

- ssh
- gitolite
- gitlab
- cgit

EMAIL-WORKFLOW

```
git format-patch -1 --base @~  
git send-email --to linux-pwm@vger.kernel.org --cc ... <patchf  
git am <mailbox>
```

PUBLIC-INBOX + B4

Patchserie auf lore.k.o

```
$ b4 am 20250520-pwm-axi-pwmgen-add-external-clock-v1-0-6cd63c
Grabbing thread from lore.kernel.org/all/20250520-pwm-axi-pwmg
Analyzing 18 messages in the thread
Looking for additional code-review trailers on lore.kernel.org
Analyzing 0 code-review messages
Checking attestation on all messages, may take a moment...
---
✓ [PATCH 1/3] dt-bindings: pwm: adi,axi-pwmgen: update docum
✓ [PATCH 2/3] dt-bindings: pwm: adi,axi-pwmgen: add external
✓ [PATCH 3/3] pwm: axi-pwmgen: add support for external cloc
---
✓ Signed: openpgp/dlechner@baylibre.com
✓ Signed: DKIM/baylibre-com.20230601.gappssmtp.com (From: dl
---
Total patches: 3
```

THE END

HILFREICHE ALIASES

[alias]

```
lg = log --pretty=oneline --abbrev-commit  
one = show -s --pretty='format:%h (%s)'  
oneq = show -s --pretty='format:%h ("%s")'  
lgg = log --pretty=oneline --abbrev-commit --decorate --bo  
please = push --force-with-lease
```

BACKEND

- Objekte: blob, tree, commit, tag
- Identifiziert per SHA-1
- Binäre Datenbank