



Berner Fachhochschule
Haute école spécialisée bernoise
Bern University of Applied Sciences

Introduction to Git

Embedded Linux and Android (CAS EBX)

Wednesday, 10th June 2015 16:30 - 18:00

BFH - Room 102 (Wankdorffeldstrasse, Bern)

Daniel Baumann <daniel.baumann@bfh.ch>

IT System Engineer, Team Infrastructure Services

Overview

Part 1

- ▶ Introduction
- ▶ Git Basics
- ▶ Git Core Concepts

Part 2

- ▶ Advanced Git Usage
- ▶ Question and Answers

Part 1

Introduction

What is Git?

- ▶ Distributed Version Control System
- ▶ Stupid Content Tracker (KISS)
- ▶ defacto industry standard

What makes Git special?

- ▶ fast
- ▶ cloning with full history
- ▶ it is quite hard to loose data
- ▶ relatively straight forward yet powerful
- ▶ multiple ways to get the same thing

What people say about Git who did not get it yet

- ▶ I don't need the „distributed“
- ▶ I don't need the „complexity“

Git Basic Usage (1/2)

Download a repository from the server to your computer

- ▶ `git clone git@git.example.org:[...]/${repository}.git`

Update an existing repository with changes from the server to your computer

- ▶ `git pull`

Check what has changed in the repository on your computer

- ▶ `git status`

Check how it has changed in the repository on your computer

- ▶ `git diff`

Git Basic Usage (2/2)

Save your changes as a commit

- ▶ `git commit -a -m "My commit message."`

Push your commit(s) from your computer to the server

- ▶ `git push`

Show the list of your commits

- ▶ `git log`

Git Core Concepts

- ▶ **staging area**
- ▶ **commit and hashsums:** you cannot change commits, you can create new ones (and garbage-collect the old ones)
- ▶ **fastforward / non-fastforward**
- ▶ **known solutions for problem A can be used for problem B by first making problem B into A**
- ▶ **branches:** master
- ▶ **references:** explicit and relative; branch, tags and HEAD
- ▶ **remotes and origin**
- ▶ **merge and rebase:** (see part 2)

Part 2

Advanced Git Usage (1): Finding broken stuff

Find the person to blame for

- ▶ `git blame`

Find the commit to blame for

- ▶ `git bisect`

Revert the broken commit

- ▶ `git revert`

Advanced Git Usage (2): Changing the Past

Editing Commits

- ▶ `git amend`
- ▶ `git rebase` and `edit/reword` commits
- ▶ tip: create an initial empty commit

- ▶ `git filter branch`

Reordering Commits

- ▶ `git rebase` and `squash/remove` etc. commits

Cutting Commits

- ▶ `graft points`

Advanced Git Usage (3): Branches and Remotes

Branches

- ▶ branches are cheap
- ▶ remote and local
- ▶ origin
- ▶ adding remote branches

- ▶ fetching branches
- ▶ push other remote branches as local branches

Fetch commits

- ▶ git cherry-pick

Advanced Git Usage (4): Shortcuts

commit

- ▶ no need for precision: `head~10`
- ▶ no need to repeat commit message: `-C HEAD`

partial commits

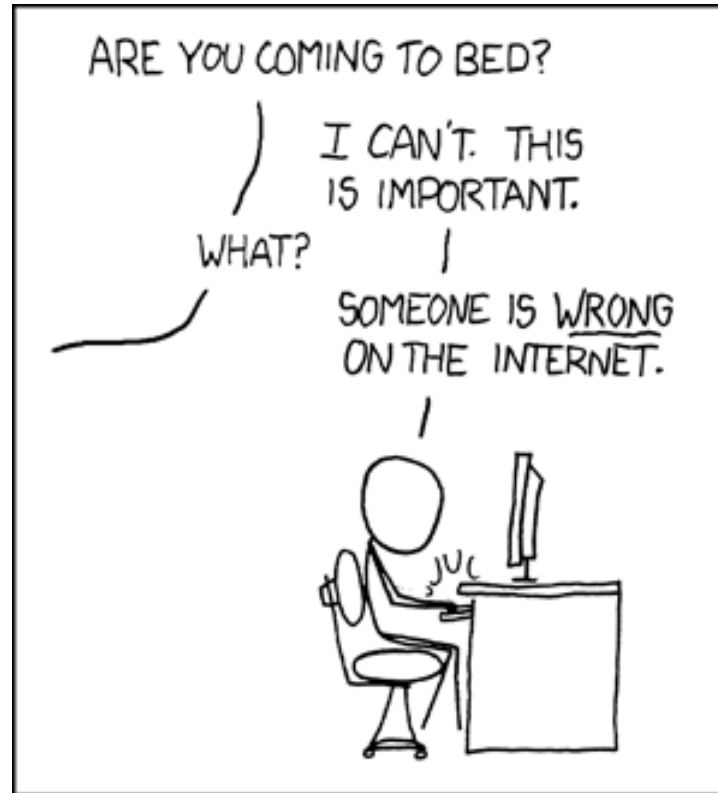
- ▶ `git commit -p`

reopen commits

- ▶ `git reset HEAD~`

Duty Calls

<https://xkcd.com/386/>



Tooltip: *What do you want me to do? LEAVE? Then they'll keep being wrong!*



Advanced Git Usage (5): Rebase versus Merge

- ▶ don't listen to the Internet, people are wrong
(or have very, very, veeeery specific workflows like kernel.org
and are in now way representative or good role models)
- ▶ if you don't, you're creating a mess for you and everyone else.
- ▶ always do `git pull -rebase`
- ▶ always do fast-forward merges (aka rebase), never non-fastforward merges (aka merges)

Questions and Answers

Further Information

Installation, Configuration, Usage, Ressources...

- ▶ <https://linux.bfh.ch/software/git>

BFH Git Services

- ▶ <https://linux.bfh.ch/services/git>

Public Mailinglist for Help

- ▶ bfh-linux-users@lists.bfh.ch

Thank You for Your Attention.

♥ Source Code is freely available

```
git clone git://git.bfh.ch/git/staff/bad9/other/talks.git
```