



git workshop

maximilian attems - maks@debian.org

May 19, 2007



Talk Overview

Motivation

git History

git Usage

git Implementation

Documentation + further reading

Basics: Configuration

Creating a Repository

Latest Status

Adding Files

Clone a Repository

Viewing Changelog

Exploring History

Managing Branches

Exploring History II

Cleanup

Cooperation



Motivation

Motivation

- git History
- git Usage
- git Implementation
- Documentation + further reading
- Basics:
 - Configuration
 - Creating a Repository
 - Latest Status
 - Adding Files
 - Clone a Repository
 - Viewing Changelog
 - Exploring History
 - Managing Branches
 - Exploring History II
 - Cleanup
 - Cooperation

- non-linear development
- distributed development
- scalability
- speed
- excellent performance on big projects
- cryptographic authentication of history
- protocol support: git, ssh, rsync, http, ftp



git History

Motivation

git History

git Usage

git Implementation

Documentation +
further reading

Basics:

Configuration

Creating a
Repository

Latest Status

Adding Files

Clone a Repository

Viewing Changelog

Exploring History

Managing Branches

Exploring History II

Cleanup

Cooperation

three parents: BK, cache, Monotone

timeline:

April 2005: officially used to track Linux

June 2005: Linux 2.6.12 release managed by git

December 2005: git 1.0 release

February 2007: git 1.5 release

“And then realize that nothing is perfect.
Git is just **closer** to perfect than any
other SCM out there.” -linus



git Usage

- Motivation
- git History
- git Usage**
- git Implementation
- Documentation + further reading
- Basics:
 - Configuration
 - Creating a Repository
 - Latest Status
 - Adding Files
 - Clone a Repository
 - Viewing Changelog
 - Exploring History
 - Managing Branches
 - Exploring History II
 - Cleanup
 - Cooperation

- linux-2.6
- xorg
- wine
- olpc
- alioth
- ...



git Implementation

- Motivation
- git History
- git Usage
- git Implementation**
- Documentation + further reading
- Basics:
 - Configuration
 - Creating a Repository
 - Latest Status
 - Adding Files
 - Clone a Repository
 - Viewing Changelog
 - Exploring History
 - Managing Branches
 - Exploring History II
 - Cleanup
 - Cooperation

data structure: index + blob object

index: the mutable index caches information about the working directory and the next revision to be committed

blob: append-only blob object of each file-revision
each object is identified by a SHA-1 hash of its content
a blob object is combined into packs to save space using delta compression



Documentation + further reading

Motivation
git History
git Usage
git Implementation

Documentation +
further reading

Basics:
Configuration
Creating a
Repository
Latest Status
Adding Files
Clone a Repository
Viewing Changelog
Exploring History
Managing Branches
Exploring History II
Cleanup
Cooperation

git tutorial

git wiki

everyday git with 20 commands or so

git help

git Wikipedia



Basics: Configuration

Motivation
git History
git Usage
git Implementation
Documentation +
further reading

Basics:
Configuration

Creating a
Repository
Latest Status
Adding Files
Clone a Repository
Viewing Changelog
Exploring History
Managing Branches
Exploring History II
Cleanup
Cooperation

global configuration / .gitconfig

```
$ git config --global user.name "maximilian attems"  
$ git config --global user.email maks@debian.org
```

set email per repository

```
$ git config user.email maks@debian.org
```

list config variables

```
$ git config --list
```




Creating a Repository

- Motivation
- git History
- git Usage
- git Implementation
- Documentation + further reading
- Basics:
 - Configuration
 - Creating a Repository**
 - Latest Status
 - Adding Files
 - Clone a Repository
 - Viewing Changelog
 - Exploring History
 - Managing Branches
 - Exploring History II
 - Cleanup
 - Cooperation

Either extract your favourite project

```
$ tar xzf project.tar.gz
$ cd project
$ git init
Initialized empty Git repository in .git
```

or start with a blank dir

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



Creating a Repository II

- Motivation
- git History
- git Usage
- git Implementation
- Documentation + further reading
- Basics:
- Configuration
- Creating a Repository**
- Latest Status
- Adding Files
- Clone a Repository
- Viewing Changelog
- Exploring History
- Managing Branches
- Exploring History II
- Cleanup
- Cooperation

Tell git to track any file below the current directory

```
$ git add .
```

Finally commit the current state with interactive commit message

```
$ git commit -a
```

maybe directly with commit message

```
$ git commit -a -m "Initial commit of <project>"
```



Latest Status

- Motivation
- git History
- git Usage
- git Implementation
- Documentation + further reading
- Basics:
 - Configuration
 - Creating a Repository
 - Latest Status**
 - Adding Files
 - Clone a Repository
 - Viewing Changelog
 - Exploring History
 - Managing Branches
 - Exploring History II
 - Cleanup
 - Cooperation

check if anything changed

```
$ git status
```

really no diff!?

```
$ git diff
```

hmm ok we are lazy, show latest commit

```
$ git show
```



Adding Files

- Motivation
- git History
- git Usage
- git Implementation
- Documentation + further reading
- Basics:
 - Configuration
 - Creating a Repository
 - Latest Status
 - Adding Files**
 - Clone a Repository
 - Viewing Changelog
 - Exploring History
 - Managing Branches
 - Exploring History II
 - Cleanup
 - Cooperation

create file

```
$ > test1.c
```

tell git which files you want to commit

```
$ git add test1.c  
$ git commit
```

Alternatively add all new files + changes

```
$ git commit -a
```



Clone a Repository

- Motivation
- git History
- git Usage
- git Implementation
- Documentation + further reading
- Basics:
 - Configuration
 - Creating a Repository
 - Latest Status
 - Adding Files
 - Clone a Repository**
 - Viewing Changelog
 - Exploring History
 - Managing Branches
 - Exploring History II
 - Cleanup
 - Cooperation

on the example of initramfs-tools

```
$ git clone \  
git://git.debian.org/git/kernel/initramfs-tools.git
```

developer access for pushing

```
$ git clone \  
git+ssh://maks@git.debian.org/git/kernel/initramfs-tools.git
```



Viewing Changelog

- Motivation
- git History
- git Usage
- git Implementation
- Documentation + further reading
- Basics:
- Configuration
- Creating a Repository
- Latest Status
- Adding Files
- Clone a Repository
- Viewing Changelog**
- Exploring History
- Managing Branches
- Exploring History II
- Cleanup
- Cooperation

At any point you can checkout the history of changes

```
$ git log
```

Show the diff of each step too

```
$ git log -p
```

summary overview

```
$ git log --stat --summary
```



Exploring History

- Motivation
- git History
- git Usage
- git Implementation
- Documentation + further reading
- Basics:
 - Configuration
 - Creating a Repository
 - Latest Status
 - Adding Files
 - Clone a Repository
 - Viewing Changelog
 - Exploring History**
 - Managing Branches
 - Exploring History II
 - Cleanup
 - Cooperation

see the great grandparent of HEAD

```
$ git show HEAD~4
```

see a specific commit

```
$ git show <object>  
$ git show b71721f02b6b46fddfc624888f61aafbc2399129
```

use short notation of the object

```
$ git show b71721f02b6
```

(first few chars are enough to uniquely define that commit)



Managing Branches

- Motivation
- git History
- git Usage
- git Implementation
- Documentation + further reading
- Basics:
 - Configuration
 - Creating a Repository
 - Latest Status
 - Adding Files
 - Clone a Repository
 - Viewing Changelog
 - Exploring History
 - Managing Branches**
 - Exploring History II
 - Cleanup
 - Cooperation

list current available branches
(the asterisk marks the branch you are on):

```
$ git branch
```

switch to specific branch

```
$ git checkout <branchname>  
$ git checkout david
```

merge specific branch

```
$ git merge <branchname>
```

create new branch from current state on

```
$ git checkout -b <branchname>
```




Exploring History II

- Motivation
- git History
- git Usage
- git Implementation
- Documentation + further reading
- Basics:
- Configuration
- Creating a Repository
- Latest Status
- Adding Files
- Clone a Repository
- Viewing Changelog
- Exploring History
- Managing Branches
- Exploring History II**
- Cleanup
- Cooperation

see log between two versions

```
$ git log 0.86..0.87
```

list commit that where made on the david branch but not on maks branch

```
$ git log maks..david
```

see what changed in the boot scripts the last six weeks

```
$ git log --since="6 weeks ago" scripts
```

crazy search

```
$ git log --since="June 5, 2005" \  
    --grep="regex" --author="guy@domain.org" \  
    --pretty=oneline some-branch -- path
```



Cleanup

- Motivation
- git History
- git Usage
- git Implementation
- Documentation + further reading
- Basics:
- Configuration
- Creating a Repository
- Latest Status
- Adding Files
- Clone a Repository
- Viewing Changelog
- Exploring History
- Managing Branches
- Exploring History II
- Cleanup**
- Cooperation

reset the last commit

```
$ git reset  
$ git reset HEAD~1
```

reset the last commit and disregard any change

```
$ git reset --hard
```

revert changes to a file

```
$ git checkout file
```



Cooperation

- Motivation
- git History
- git Usage
- git Implementation
- Documentation + further reading
- Basics:
- Configuration
- Creating a Repository
- Latest Status
- Adding Files
- Clone a Repository
- Viewing Changelog
- Exploring History
- Managing Branches
- Exploring History II
- Cleanup
- Cooperation**

define remote repository

```
$ git remote add bob /home/bob/myrepo
```

fetching remote repository

```
$ git fetch bob
```

after some checks ;) merging this repository into master

```
$ git merge bob/master
```