

Git

a fast distributed revision control system

Eric Müller

UHEI

FACETS CodeJam #2

2008-05-06



Features

- distributed development model
 - gives each developer a local copy of the entire history
 - ⇒ disconnected operation
- good support for non-linear development
 - lightweight & fast branching
 - easy merging
 - private branches
- cryptographic authentication of history (`git-tag -s -m "Message"`)

Features

- distributed development model
 - gives each developer a local copy of the entire history
 - ⇒ disconnected operation
- good support for non-linear development
 - lightweight & fast branching
 - easy merging
 - private branches
- cryptographic authentication of history (`git-tag -s -m "Message"`)
- easy access to svn/svk upstream repositories (`git-svn`)
- cvsserver emulation

Features

- distributed development model
 - gives each developer a local copy of the entire history
 - ⇒ disconnected operation
- good support for non-linear development
 - lightweight & fast branching
 - easy merging
 - private branches
- cryptographic authentication of history (`git-tag -s -m "Message"`)
- easy access to svn/svk upstream repositories (`git-svn`)
- cvsserver emulation

Now

internal structure of git

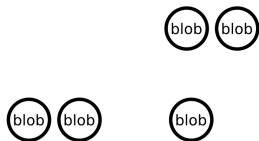
- Git manages a set of objects (revisions, files, directories, ...)
- Each object is identified by its SHA1 sum (e.g. 4cdeed861b5f797b3fa661eb331a6bd6ad669c6a)
- Objects can point to each other

Git – A directed acyclic graph

- Blob

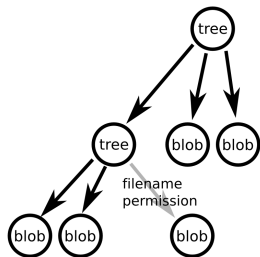
- chunk of binary data, i.e. contents of files, symlinks, ...
- identified by SHA1 (160 bits) *checksum*

⇒ one blob per file revision



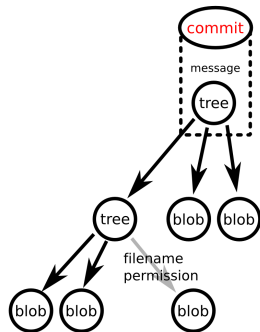
Git – A directed acyclic graph

- Tree
 - collection of blobs with filenames and (fs-) permissions
 - state of a directory hierarchy at a given revision/time



Git – A directed acyclic graph

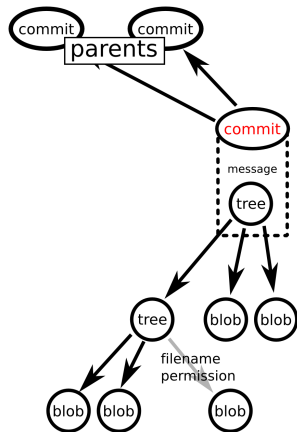
- Commit
 - contains a message and a tree object



Git – A directed acyclic graph

- Commit

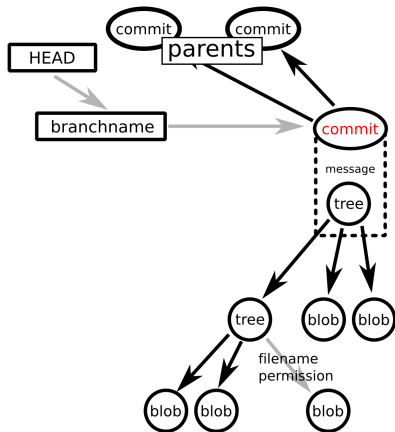
- contains a message and a tree object
- pointers to ≥ 1 *parent* commits



Git – A directed acyclic graph

- Commit

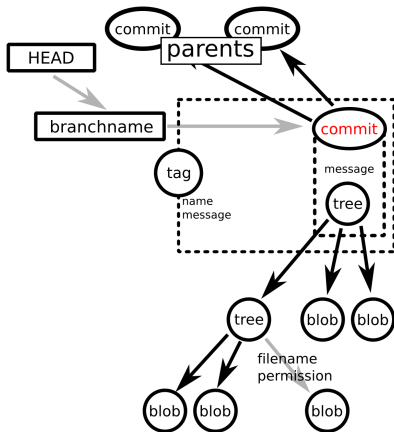
- contains a message and a tree object
- pointers to ≥ 1 *parent* commits
- branchname gets updated (HEAD too)



Git – A directed acyclic graph

- Tag

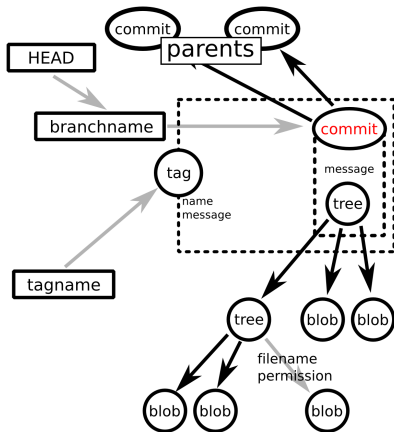
- points to a commit
- contains an (optional cryptographically signed) message



Git – A directed acyclic graph

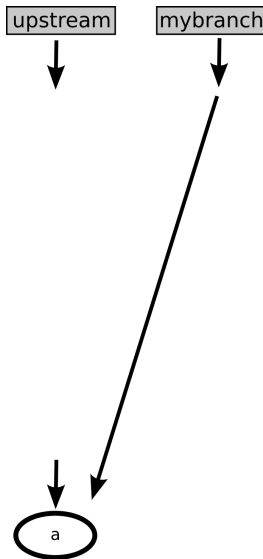
- Tag

- points to a commit
- contains an (optional cryptographically signed) message
- tagname



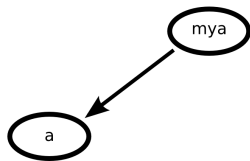
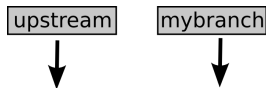
Git – History

- 1 remote repository @ revision a



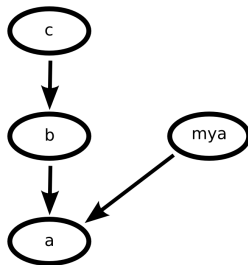
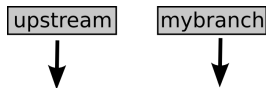
Git – History

- 1 remote repository @ revision a
- 2 adding a local commit mya into local *mybranch* based on a



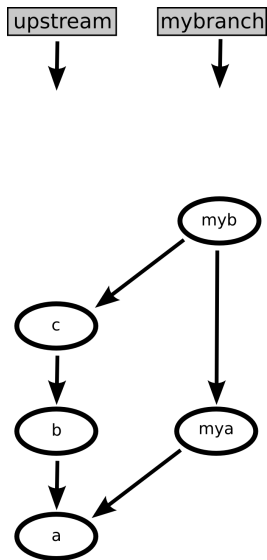
Git – History

- 1 remote repository @ revision a
- 2 adding a local commit `mya` into local `mybranch` based on a
- 3 some remote commits



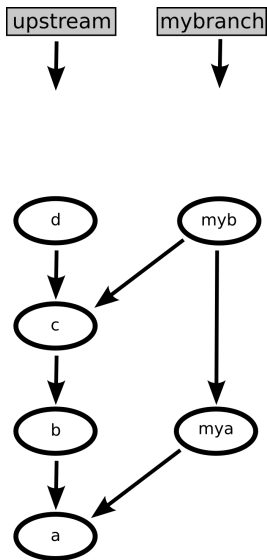
Git – History

- 1 remote repository @ revision a
- 2 adding a local commit `mya` into local `mybranch` based on a
- 3 some remote commits
- 4 merging remote commits into `mybranch`



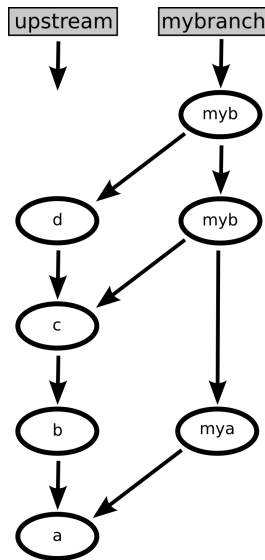
Git – History

- 1 remote repository @ revision a
- 2 adding a local commit `mya` into local `mybranch` based on a
- 3 some remote commits
- 4 merging remote commits into `mybranch`
- 5 another remote commit



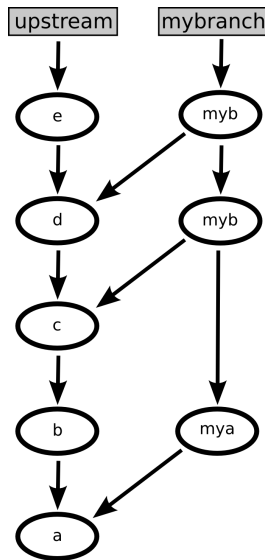
Git – History

- 1 remote repository @ revision a
- 2 adding a local commit `mya` into local `mybranch` based on a
- 3 some remote commits
- 4 merging remote commits into `mybranch`
- 5 another remote commit
- 6 another merge from remote into local branch



Git – History

- 1 remote repository @ revision a
- 2 adding a local commit `mya` into local `mybranch` based on a
- 3 some remote commits
- 4 merging remote commits into `mybranch`
- 5 another remote commit
- 6 another merge from remote into local branch
- 7 another remote commit

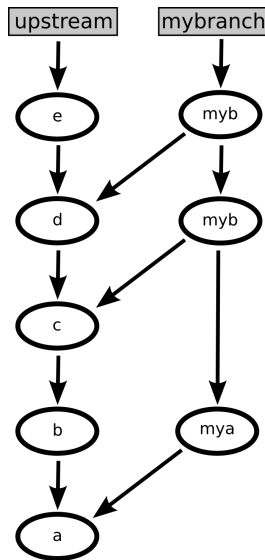


Git – History

- 1 remote repository @ revision a
- 2 adding a local commit `mya` into local `mybranch` based on a
- 3 some remote commits
- 4 merging remote commits into `mybranch`
- 5 another remote commit
- 6 another merge from remote into local branch
- 7 another remote commit

more features

- $\#parents \geq 2$ and successive merging is possible
- instead of merging, rewriting history (*rebasing*) is also possible



Integration

- *VIM: vcscommand*
<http://code.google.com/p/vcscommand/>
- *Trac: GitPlugin*
<http://trac-hacks.org/wiki/GitPlugin>
- *Emacs: git.el (DVC is better?)*
<http://www.emacswiki.org/cgi-bin/wiki/Git>
- *Eclipse: J/EGIT*
<http://git.or.cz/gitwiki/EclipsePlugin>

Integration

- *VIM: vcscommand*
<http://code.google.com/p/vcscommand/>
- *Trac: GitPlugin*
<http://trac-hacks.org/wiki/GitPlugin>
- *Emacs: git.el (DVC is better?)*
<http://www.emacswiki.org/cgi-bin/wiki/Git>
- *Eclipse: J/EGIT*
<http://git.or.cz/gitwiki/EclipsePlugin>

Information

- *Git – SVN Crash Course*
<http://git.or.cz/course/svn.html>
- *Git cheat sheet, extended edition*
<http://jan-krueger.net/development/git-cheat-sheet-extended-edition>

- good documentation (manpages, user- & developer tutorials)
- *The guts of git*
<http://lwn.net/Articles/131657/>
- *Git for Computer Scientists*
<http://eagain.net/articles/git-for-computer-scientists/>