Git Practices

Namhyung Kim

SW Maestro project

2013-07-16 Tue

Outline

Git concept

- Distributed source code (version) management
- Content-addressable filesystem
  - SHA-1 hash (20-byte, 40-hexdigit)
- Save snapshots

Git metadata

```
$ ls -F .git/
branches/  config  description  HEAD
hooks/  info/  objects/  refs/
```

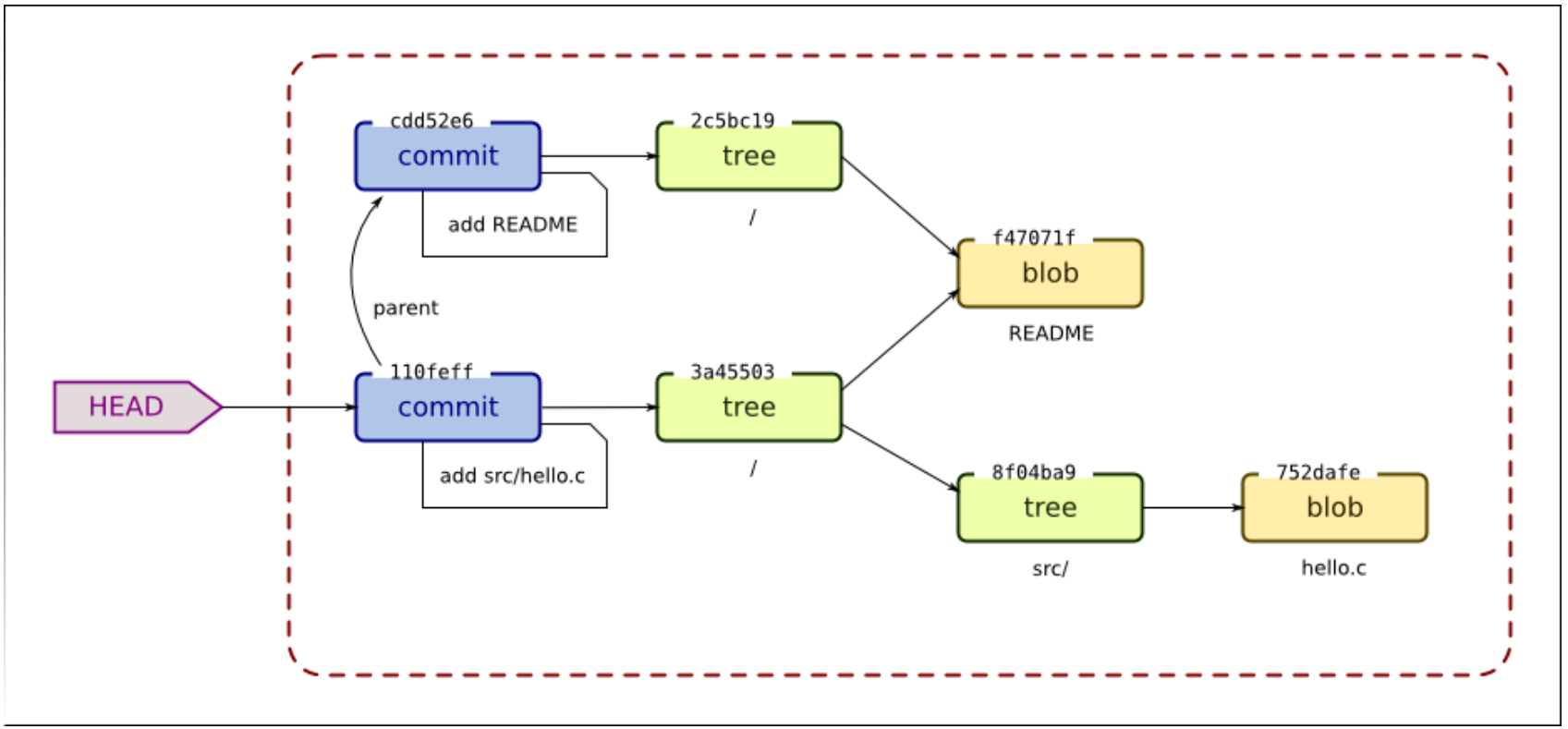
Git objects

- .git/objects/
  - 40-hexdigit = 2 (dir) + 38 (file)
- Blob
- Tree
- Commit
- Tag

Git references

- .git/refs/
- heads/
  - For local branches
  - Updated on commit
- remotes/
  - For remote branches
  - Updated on fetch
- tags/
  - No update

Git object database



Git help

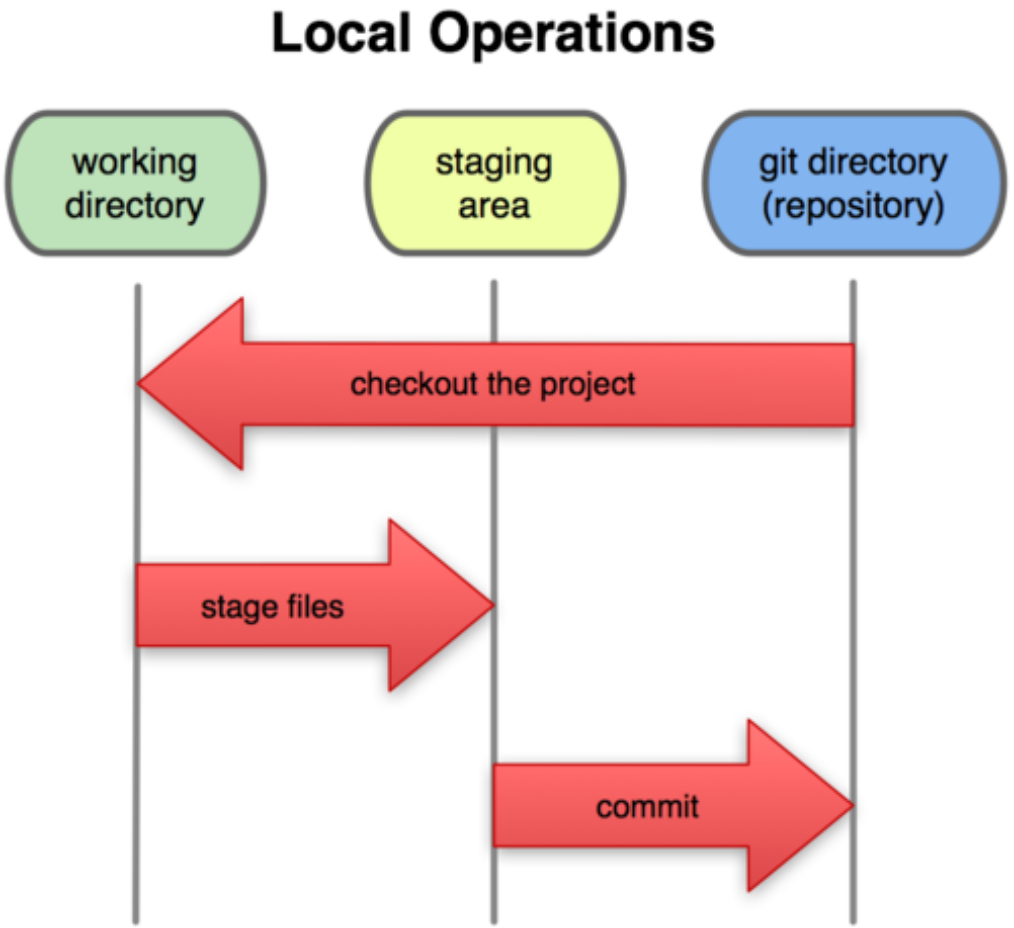
```
$ git
$ git <command> -h
$ git <command> --help
$ git help <command>
$ man git-<command>
```

Pager

- less is more
- can control scrolling
- $GIT_PAGER
- config core.pager
- setting to 'cat' disables pager

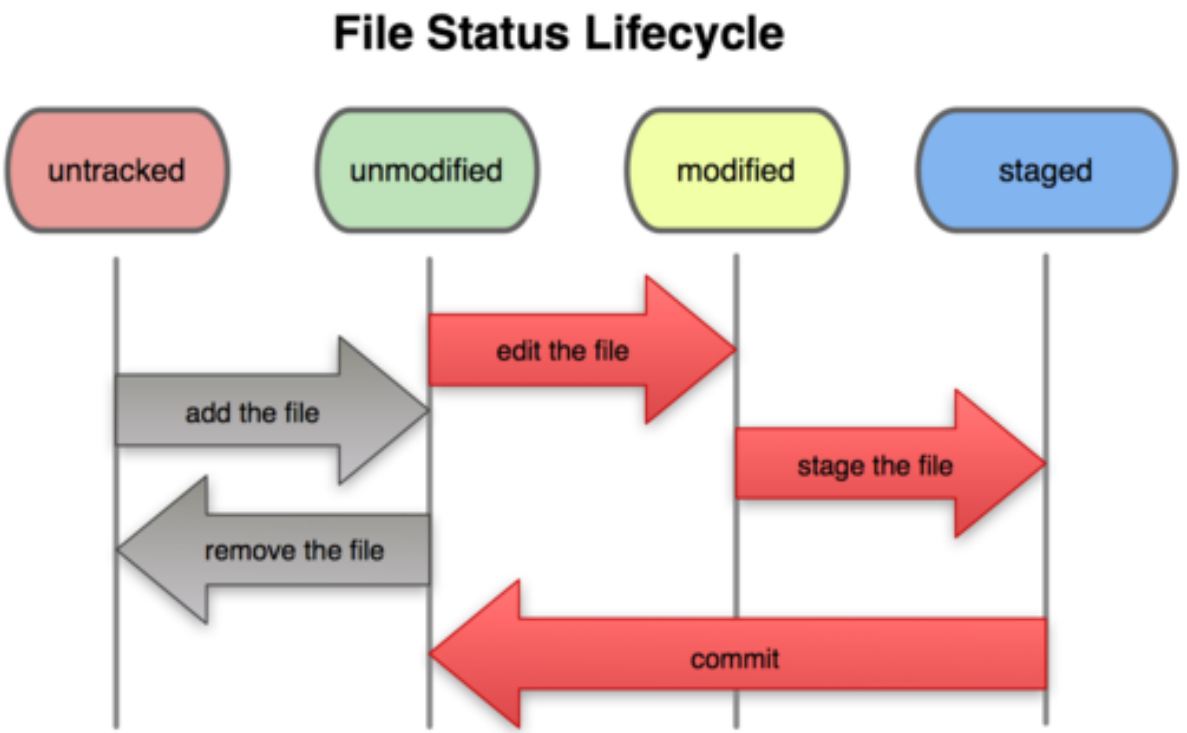
Git status

- Clean (checkout)
- Dirty (work)
- Staging/cached (add)
- Committed (commit)



Git add

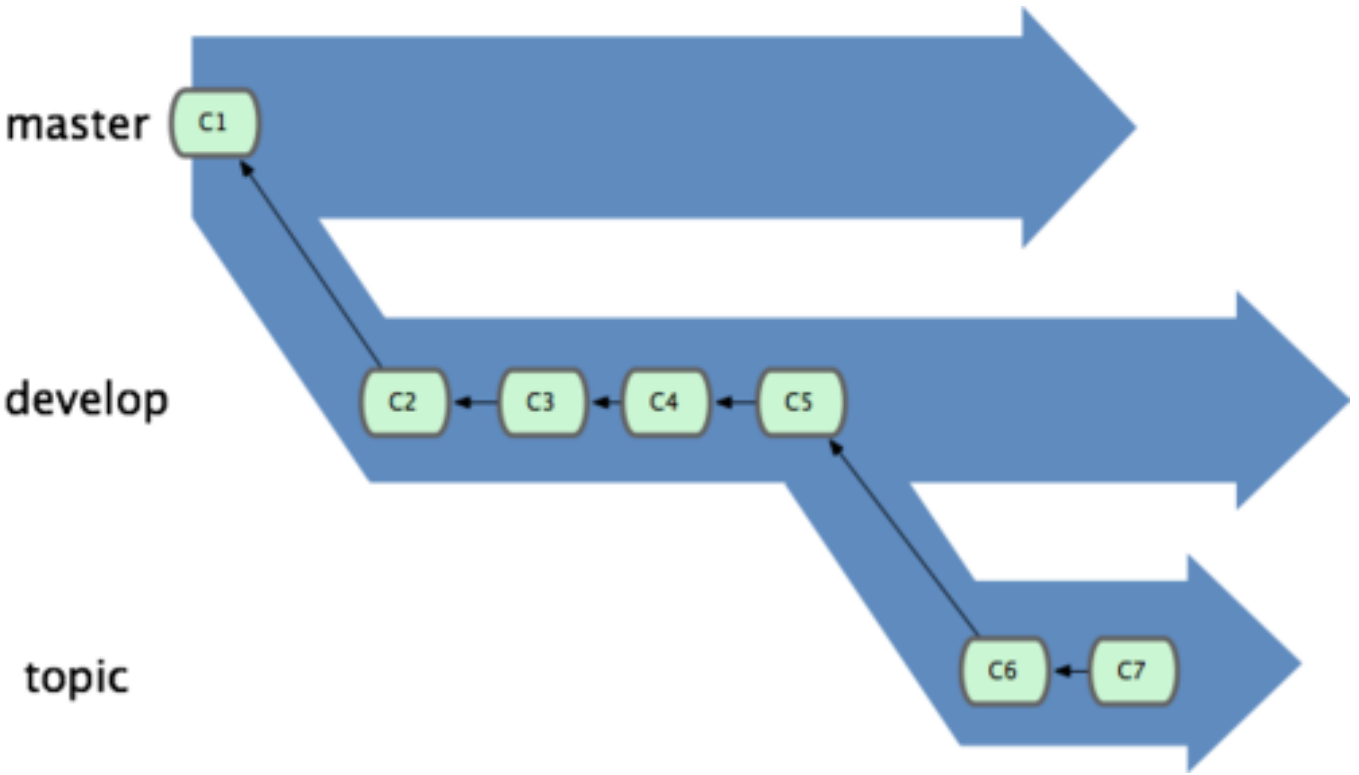
- Add current snapshot to the index
  - Add untracked file
  - Add modified file



Git branch

- Manage a history of development
- branch is very cheap operation on git!

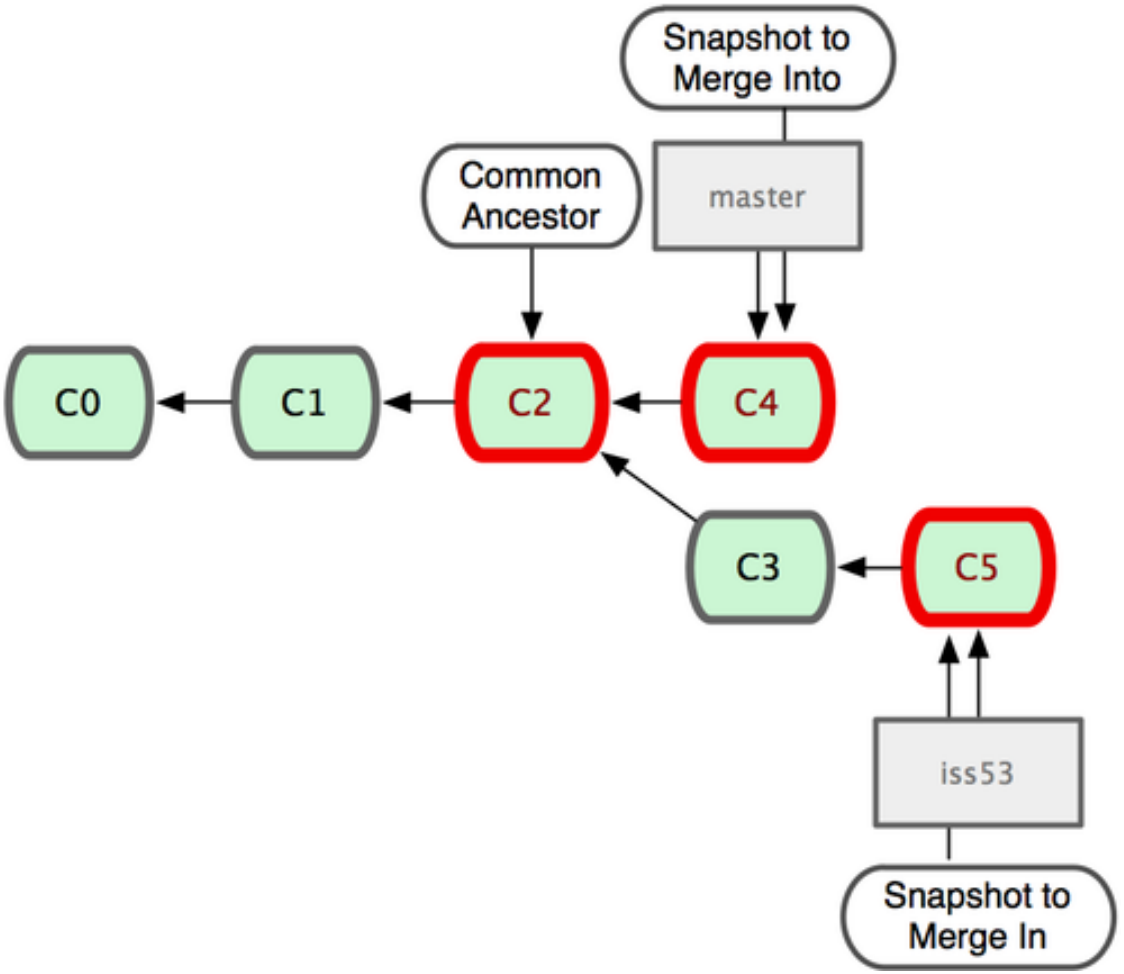
#



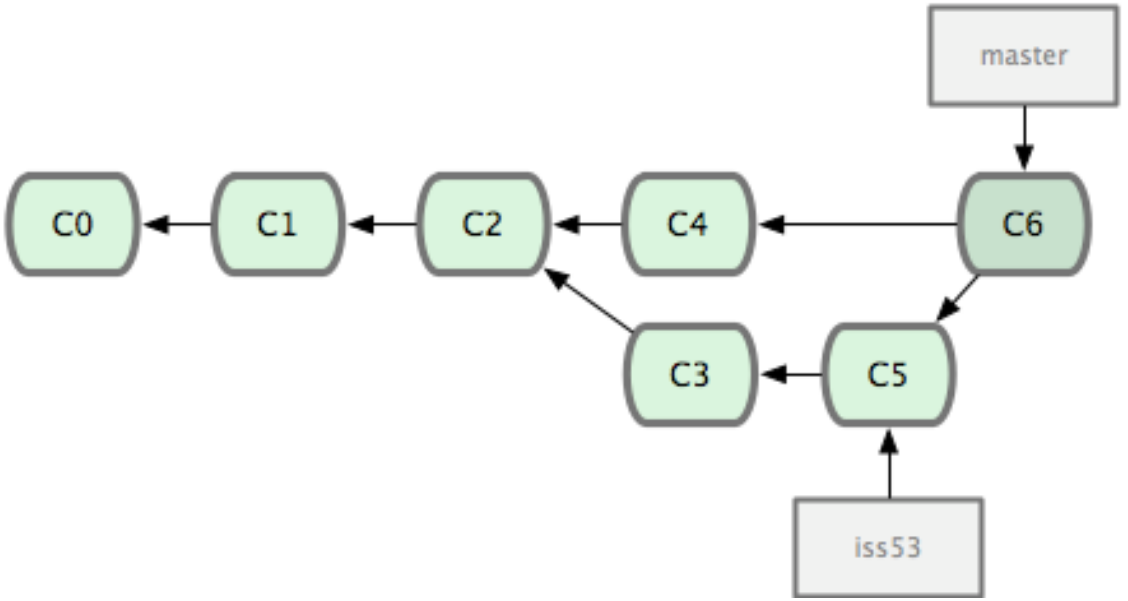
Git checkout

- Update working directory
- Change current branch
  - Update HEAD symbolic ref
- Update specified file contents

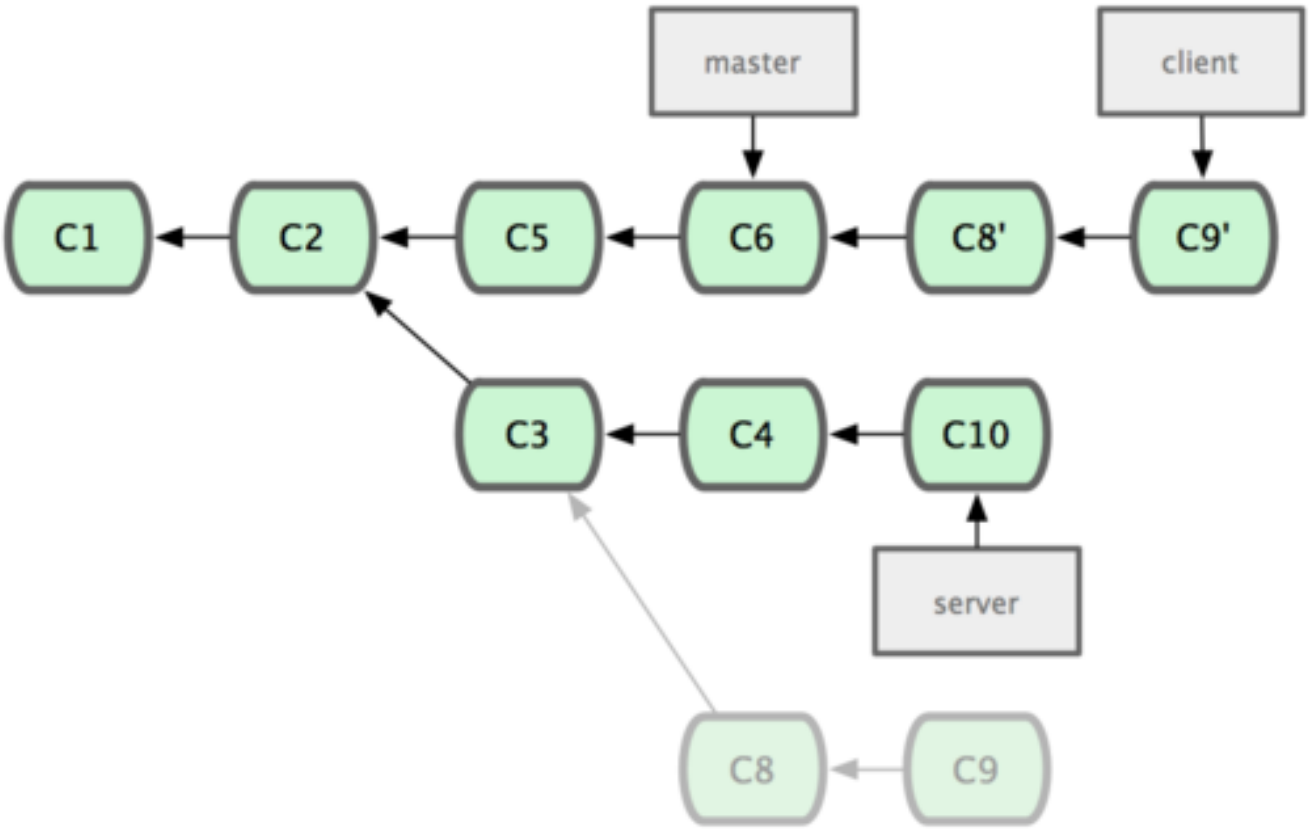
Git merge


Snapshot to
Merge Into
Common
Ancestor
master
C0
C1
C2
C4
C3
C5
iss53
Snapshot to
Merge In

Git merge



Git rebase



Git reset

- Move current branch pointer (tip)
  - Optionally update the index (–mixed)
  - Optionally update working dir (–hard)
- Or update specified file contents

NOTE

- 'checkout' updates HEAD pointer
- 'revert' adds a new commit

Git clone

- Initialize project (init)
- Add remote 'origin' repository (remote)
- Download changes (fetch)
- Checkout current branch (checkout)

Git remote

- Remote (tracking) branch
  - Save snapshot of remote repository
  - Considered as read-only
- Local tracking branch
  - Checked-out version of remote branch
  - Track and show remote changes
  - git pull knows what you want

Git pull

- git pull [remote] [branch]
  - remote and branch can be omitted on tracking branch
  - Download objects from remote (fetch)
  - Update remote branches
  - Merge into current branch (merge)
  - Or rebase current branch (pull -r)

Git push

- Upload a local branch to a remote repository
- git push [remote] [refspec]
  - use 'origin' as remote, ':' as refspec by default
  - the default refspec part can be changed via config variable
  - special refspec ':' means any matching branches
  - blank src on refspec means deletion
  - fail if non-fast-forward merging

refspec

[+]src:dst

Writing better changelog

- Selected changelogs from linux kernel (links)
  - description with warning message
  - explaining race condition
  - showing performance improvement
  - adding example code
  - describing logic
  - proof by disasm

Git config

- system, global or local

```
$ cat ~/.gitconfig
[user]
        name = Namhyung Kim
        email = namhyung@kernel.org
[alias]
        ci = commit -s
[color]
        ui = auto
[push]
        default = simple
```

Git alias

```
$ git config.alias co checkout
$ git config alias.l1 "log -1 --oneline"
$ git config alias.l1
log -1 --oneline
$ git l1
d1c3ed669a2d Linux 3.8-rc2
$ git config alias.rc "!git add . && \
git rebase --continue"
```

Git log

- Show development history
- Can limit output by
  - number (-n 10)
  - author (–author=Namhyung)
  - changelog (–grep=<regex>)
  - change (-S<string> or -G<regex>)
  - rev-list (see next slide)
  - path (– <pathname>)

Specifying revisions

- refs (branch, tag, HEAD)
- parents of refs (HEAD^ , master~3)
- rev-list
  - –since 2012-7-1 –until “two weeks ago”
  - master..topic
  - ^master topic
  - –not master topic
  - master...topic

Git reflog

- Change log for each 'ref' itself
- Saved on local repository (not shared)
- Useful if a ref is screwed-up
  - don't be panic :)

Git bisect

- Find offending commit using binary search
- git bisect start [bad] [good]
- git bisect bad
- git bisect good
- git bisect run
  - Run a script (exit code 0 means good)

Git stash

- Save uncommitted files temporarily
- Make working dir clean (to checkout another branch)
- git stash [save/pop]

Git blame

- See changelog of selected lines with Vim
  - press 'v' to block region, and '\ + g' to git blame

.vimrc

```
vmap <Leader>g :<C-U>!git blame \
  -L <C-R>=line("'<")<CR>,<C-R>=line("'>")<CR> \
  <C-R>=expand("%:p")<CR> <CR>
```

Git hooks

```
$ ls .git/hooks
applypatch-msg.sample   commit-msg.sample
post-update.sample      pre-applypatch.sample
pre-commit.sample       prepare-commit-msg.sample
pre-rebase.sample       update.sample
```

Git svn

- git can import or commit to svn server

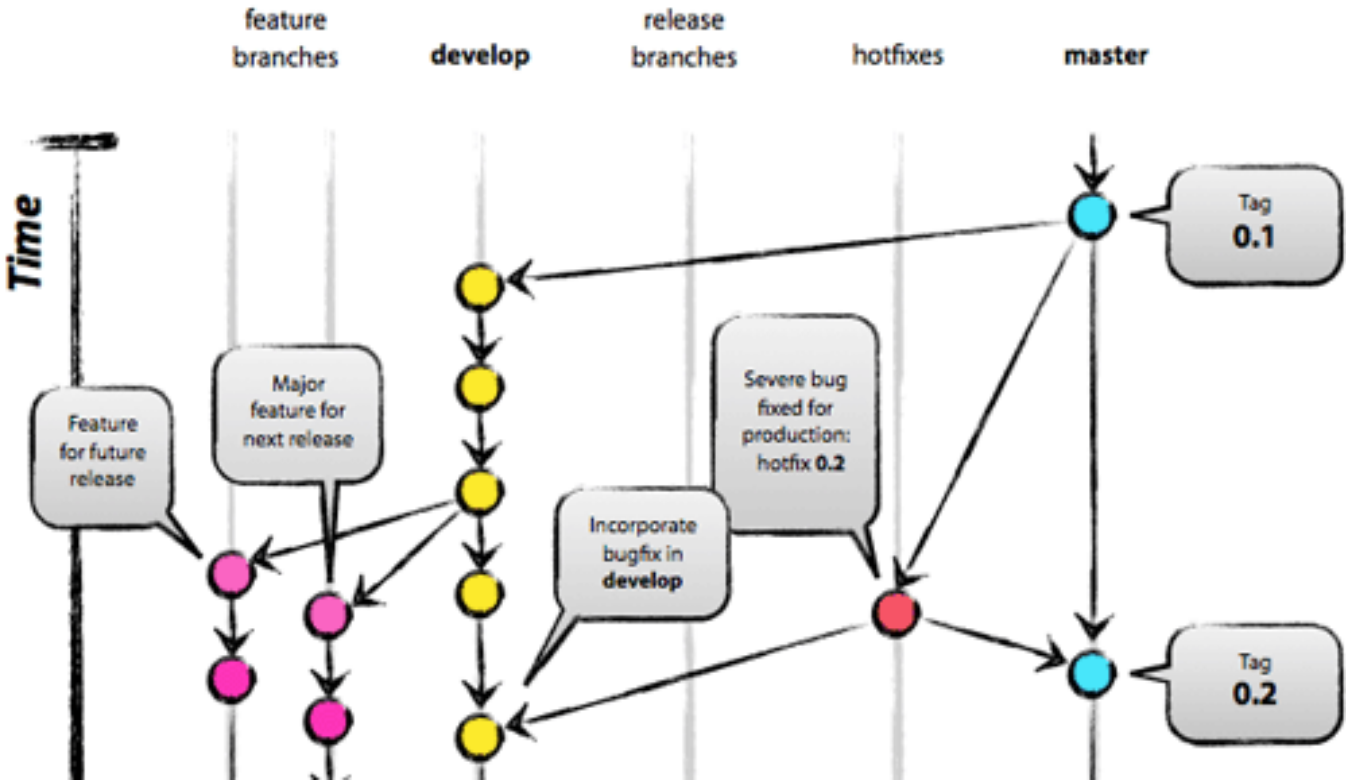
```
$ git svn clone -s http://gperftools.googlecode.com/svn g
```

Repo

- Collection of python scripts
- Manage multiple git repositories
- Require a manifest project
- Run git command on selected/all projects
- Integrated with Gerrit review system
- Used/Developed for the Android

Git flow

- A successful git branching model
  - by Vincent Driessen



Interactive commands

- git add -p (or -i)
- git rebase -i

References

- ProGit
- git help <command>
- http://dogfeet.github.com
- http://github.com/nvie/gitflow
- http://namhyung.github.io/git-user-manual-ko/
- http://robots.thoughtbot.com/post/4747482956/streamline-your-git-workflow-with-aliases