Using `git` for version control

Manuel Baumann, Joost van Zwieten

November 28, 2013

TU Delft
Delft
University of
Technology

Project

BaNaNa

What is Project baNaNa ?

Many (math) presentations are far too technical and only some people in the audience can follow.

The aim of project baNaNa is:

- present every-day tools to each other,

What is Project baNaNa ?

Many (math) presentations are far too technical and only some people in the audience can follow.

The aim of project baNaNa is:

- present every-day tools to each other,
- informal/interactive presentations,

What is Project baNaNa ?

Many (math) presentations are far too technical and only some people in the audience can follow.

The aim of project baNaNa is:

- present every-day tools to each other,
- informal/interactive presentations,
- focus on software, applications, ...

What is Project baNaNa ?

Many (math) presentations are far too technical and only some people in the audience can follow.

The aim of project baNaNa is:

- present every-day tools to each other,
- informal/interactive presentations,
- focus on software, applications, ...
- a series of lectures for and by fellow PhD students,

What is Project baNaNa ?

Many (math) presentations are far too technical and only some people in the audience can follow.

The aim of project baNaNa is:

- present every-day tools to each other,
- informal/interactive presentations,
- focus on software, applications, ...
- a series of lectures for and by fellow PhD students,
- there is no grading whatsoever,

What is Project baNaNa ?

Many (math) presentations are far too technical and only some people in the audience can follow.

The aim of project baNaNa is:

- present every-day tools to each other,
- informal/interactive presentations,
- focus on software, applications, ...
- a series of lectures for and by fellow PhD students,
- there is no grading whatsoever,
- eating bananas together.

What is Project baNaNa ?

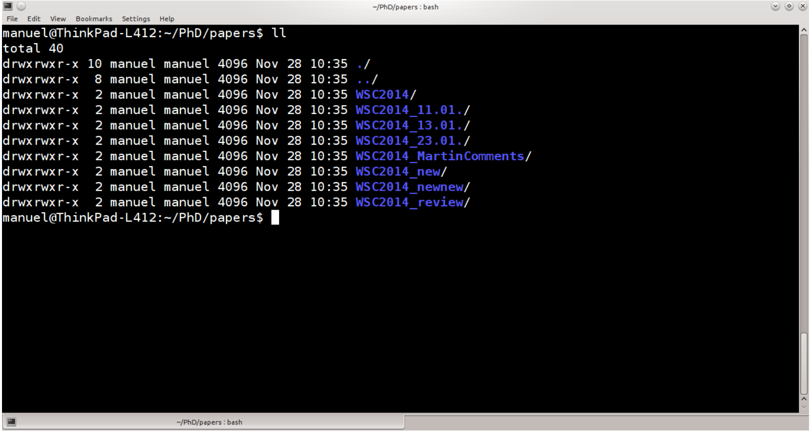
More details...

In more detail, we were thinking about

- possible topics:
  - Python
  - CUDA/OpenCL
  - Open Foam
  - MPI/OpenMP
  - latex 'best practice'
  - high end Matlab usage
- a classical lecture:
  - approx. 20 mins
  - once a month
  - everybody contributes from time to time
  - **gives many examples**

Today's lecture: git

Do you know this situation?



```
manuel@ThinkPad-L412:~/PhD/papers$ ll
total 40
drwxrwxr-x 10 manuel manuel 4096 Nov 28 10:35 ./
drwxrwxr-x  8 manuel manuel 4096 Nov 28 10:35 ../
drwxrwxr-x  2 manuel manuel 4096 Nov 28 10:35 WSC2014/
drwxrwxr-x  2 manuel manuel 4096 Nov 28 10:35 WSC2014_11.01./
drwxrwxr-x  2 manuel manuel 4096 Nov 28 10:35 WSC2014_13.01./
drwxrwxr-x  2 manuel manuel 4096 Nov 28 10:35 WSC2014_23.01./
drwxrwxr-x  2 manuel manuel 4096 Nov 28 10:35 WSC2014_MartinComments/
drwxrwxr-x  2 manuel manuel 4096 Nov 28 10:35 WSC2014_new/
drwxrwxr-x  2 manuel manuel 4096 Nov 28 10:35 WSC2014_newnew/
drwxrwxr-x  2 manuel manuel 4096 Nov 28 10:35 WSC2014_review/
manuel@ThinkPad-L412:~/PhD/papers$
```

Today's lecture: `git`

The pre-installed software package `git` can be used for version control for a (bigger) software project.

- helps to keep track of changes in your project directory
- branching, merging
- collaboration
- several sites allow you to publish your `git` repository (github, bitbucket, gitorious)

Git basics

Git stores snapshots (called *commits* in git) of files in a repository.

A commit consists of

- a set of files,
- a message,
- an author,
- a date,
- references to one or more parent commits and
- a hash (of the above).

You are in control of creating commits!

Git basics

working directory Plain directory where you can edit files using your favourite editor.

staging area Area listing the files to be commited.

repository Collection of commits.

Everything is stored locally (in the working directory):

repository	`working_directory/.git`
staging area	`working_directory/.git/index`

Basic commands

Command	Meaning
`git init`	First command, makes current folder a `git` repository
`git add <file>`	Stages a file
`git commit -m <message>`	Record changes to the repository
`git status`	Show the working tree status
`git diff`	Show changes between commits, commit and working tree, etc
`git log`	Show commit logs

↪ live demonstration...

Branches

- Branching is the git-equivalent of making a copy of your working directory.
- Branching is cheap. Use it whenever you can.
- Git is very good in merging (two or more) branches.
- NOTE: Your working directory points to only one branch.

Commands for branching and merging

Command	Meaning
`git branch`	List, create, or delete branches
`git checkout`	Checkout a branch or paths to the working tree
`git merge`	Join two or more development histories together
`git mergetool`	Run merge conflict resolution tools to resolve merge conflicts

↪ live demonstration...

Today's message

The overall message is:

Using git is optimal for collaborations and big software projects.

Moreover,

- sensitive information should not be stored in e.g. Dropbox
- `git` branches are non-linear in time
- local argument: `git` is supported (and pre-installed) at TU Delft work stations

Further reading

Many information are available online:

- https://github.com/
- http://try.github.io/levels/1/challenges/1
- http://git-scm.com/book

These slides, and much more, will be published at:

- http://projectbanana.github.io/

Project

BaNaNa