

Basic Tools for System Management

Frederic.Mallet@unice.fr

Université Nice Sophia Antipolis
Membre de Université Côte d'Azur

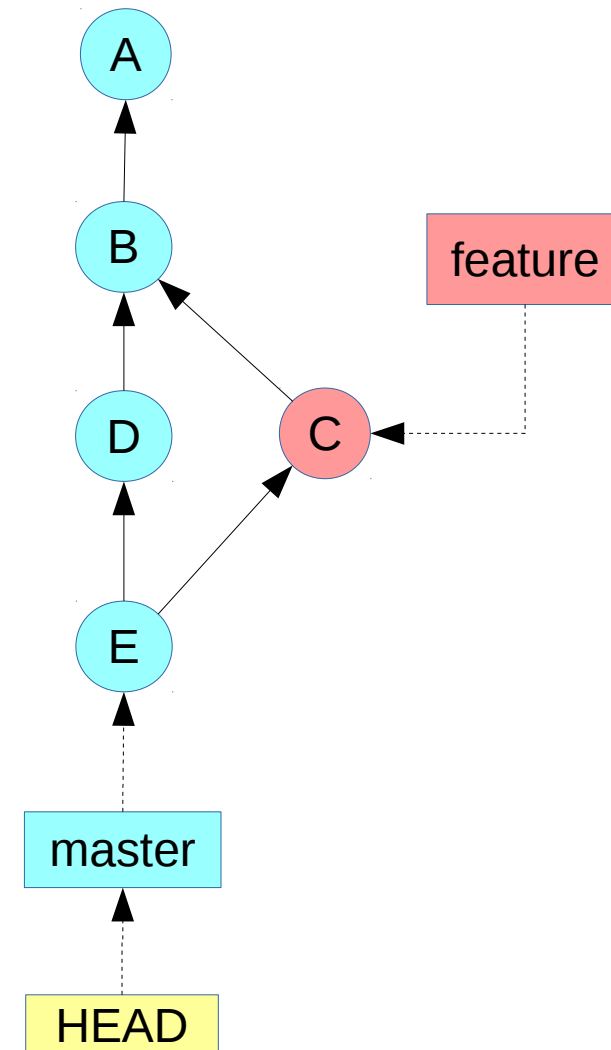
Objectives

- Learn how to manage your own computer
 - Using the shell
 - Linux / Windows
 - Version control and git
 - Installing a Virtual Machine
 - Linux Mint 18.4
 - Using bash
 - Eclipse: JDT and Modeling

Git graph

■ git log --oneline --graph

```
fmallet@chevalerios ~/git/light-ccsl
File Edit View Search Terminal Help
* 67df470 Fix bug in generating errors (add if) replace c = c + 1 by c++ add com
ment at the beginning
* ce6cf47 Fix initial State Fix problems with import due to change in package na
ming convention
* cbf4156 Add the explicit notion of state
* ebc2dd4 Fix another bug in comments Add the explicit notion of Region, Block,
Comment, Statement
* 9b8f7f9 Fix a bug in merge
* d5cf68c Merge branch 'master' of git@gitlab.inria.fr:fmallet/light-ccsl.git
\
* 1a96f7b Make all kieler specific code into a separate plugin
* | 77602a5 Fix a bug in comments
* | 9c59e71 Change Package Name Git graph
* | ad4c52f Make all kieler specific code into a separate plugin
|/
* c299baf Propose to use SCTX format
* 4d9f93a core becomes safety
* 7b97b48 Remove deprecated plugins. Has been replaced by TimeSquare safety.co
re
.ui plugin
* eb16774 Add SOS executor with SAT4J
* 5b8e523 Move dependency to TimeSquare plugin
* 480de28 Move to TimeSquare plugin
* b7cb3f1 Move dependency to TimeSquare plugin
* 6f67480 move lightccsl.core to old, replaced by fr.kairos.timesquare.safety.co
re from TimeSquare git
* 0ebdda4 fix merge
\
* d1d292e Add Comments and authors to lighccsl.core. Update the copyright 2017
,2018 removing author
* | 8dab55e Implements an interactive solver Fix bug in intersection, minus, per
iodic and delayFor
fmallet@chevalerios ~/git/light-ccsl $
```



Branching and Merging

- A merge conflict
 - It occurs when there are changes to the same part of a file in different branches
 - Merge strategies define how the merge is performed
 - Should do small frequent merges

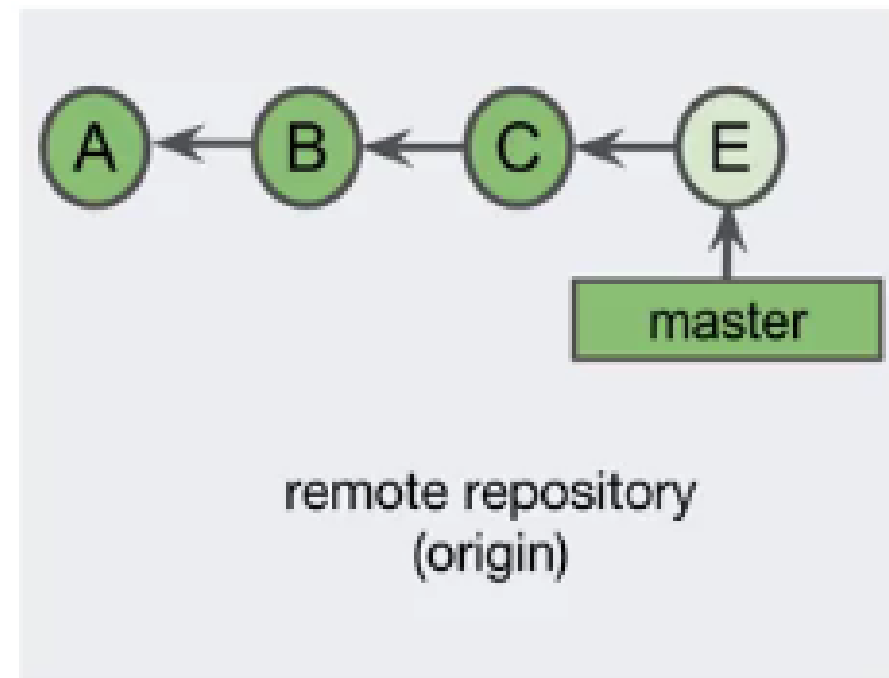
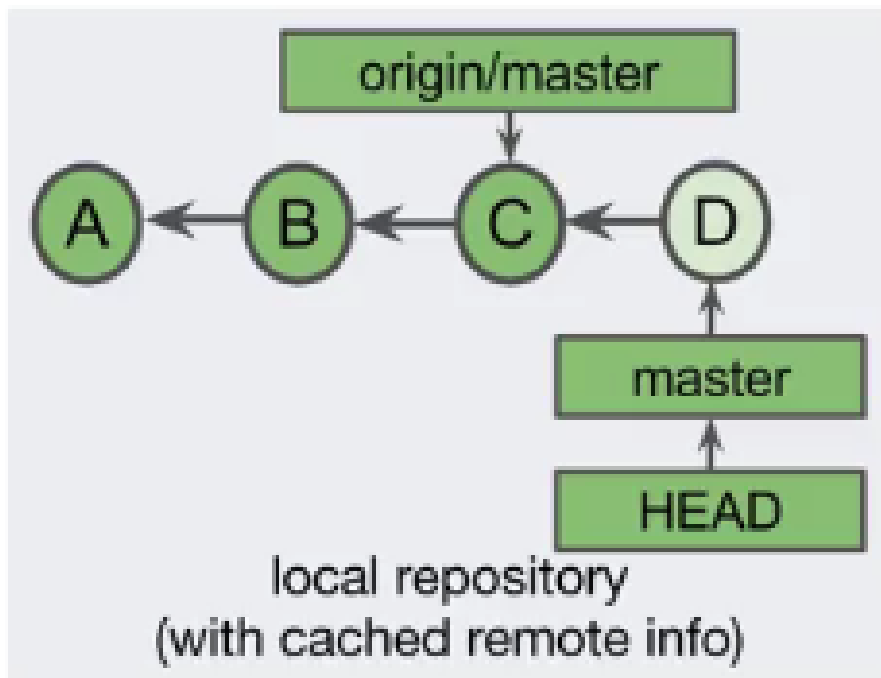
Resolving Merge Conflicts

- Several steps
 - Checkout master (base branch)
 - Merge feature (topic branch)
 - CONFLICT
 - Fix CONFLICT by hand or abort
 - Stage file in conflict
 - Commit the merge
 - Delete the feature branch label

```
fmallet@chevalerios ~/git/test $ git log --oneline --graph
* 2d412bb fix conflict
|
* 00a206d add feature2
|
* 4b8ff02 adding feature 3
|
* c983508 initial commit
```

Tracking Branches

- Local branch that represents a remote branch
 - <remote>/<branch>



Viewing Tracking Branch

- `git branch --all`
 - Includes both local and tracking branches
 - Default tracking remotes/origin/HEAD
- Change the default tracking branch
 - `git remote set-head <remote> <branch>`
- Viewing tracking branch status
 - `git status`
 - Use only the cached information from last fetch
 - `git log --all --oneline --graph`
 - It gives information about remote branches

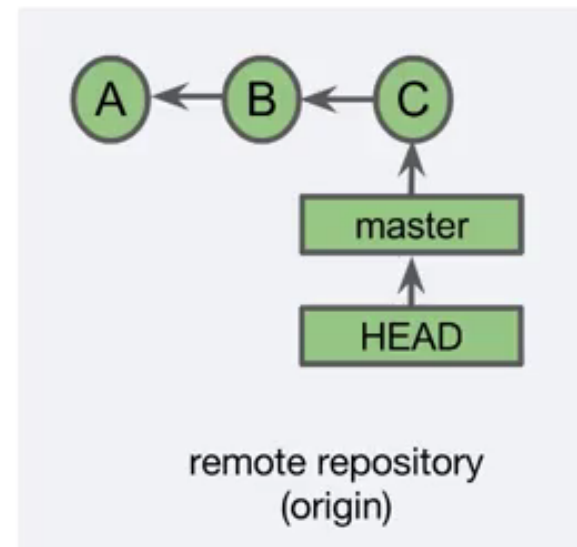
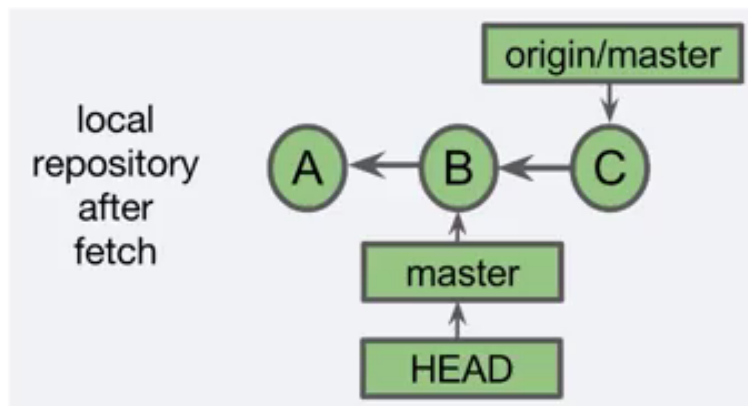
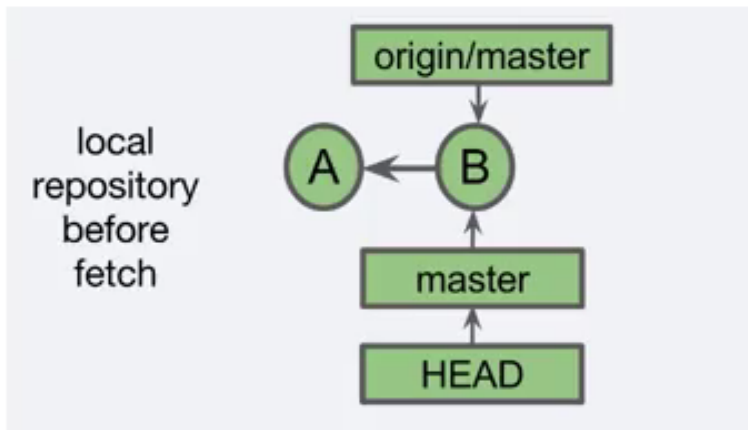
Fetch, Pull and Push

■ Network Commands

- Clone: copies remote repository
- Fetch: Retrieves new objects/refs from the remote repository
- Pull: Combines a fetch and a merge
- Push: Adds new objects/refs to the remote repository

Fetch

- Git fetch <repository>
 - Retrieves new objects
 - Updates tracking branches



git pull [<repository>] [<branch>]

■ Combines

- git fetch
- git merge FETCH_HEAD
- FETCH_HEAD is an alias for the tip of the tracking branch

■ Options

- --ff : fast-forward if possible, merge commit otherwise
- --no-ff : merge commit
- --ff-only : fails if fast-forward is not possible
- --rebase

git push [-u] [<repository>] [<branch>]

- Specify repository and branch only the first time
 - It sets up a local tracking branch
 - Should Fetch or Pull before you push

Rebasing

- Rebase rewrite the commit history
 - Do not rewrite if already shared
- Rebase
 - Move commits to a new parent (base)
 - May prevent merge
 - It reapplies the “diff” of the commits to the tip of the base
 - May result in conflicts

Git rebase <upstream> [<branch>]

- It checkouts the branch and changes its parent to <upstream>
git checkout feature
git rebase master
- Is equivalent to
git rebase master feature
- If conflicts, resolve them and then
 - **git rebase --continue**
- Can also decide to abort the rebase
 - **git rebase --abort**

Rewriting History

■ Amending a commit

- You can change the most recent commit
 - Change the message and/or the files
 - `git commit --amend -m "new message"`

■ Interactive rebase

- `git rebase -i <after-this-commit>`
- Can change the commit message (`--amend`)
- Delete/reorder/squash/fixup a Commit
- `git rebase --continue`

Squash merge

- Merge the tip of the feature branch onto the tip of the base branch
 - There can be conflicts
 - `git merge --squash featureX`

