Git : A brief introduction

Karim Hasnaoui

LCP Orsay, UMR8000 - Université Paris-Sud

Outline

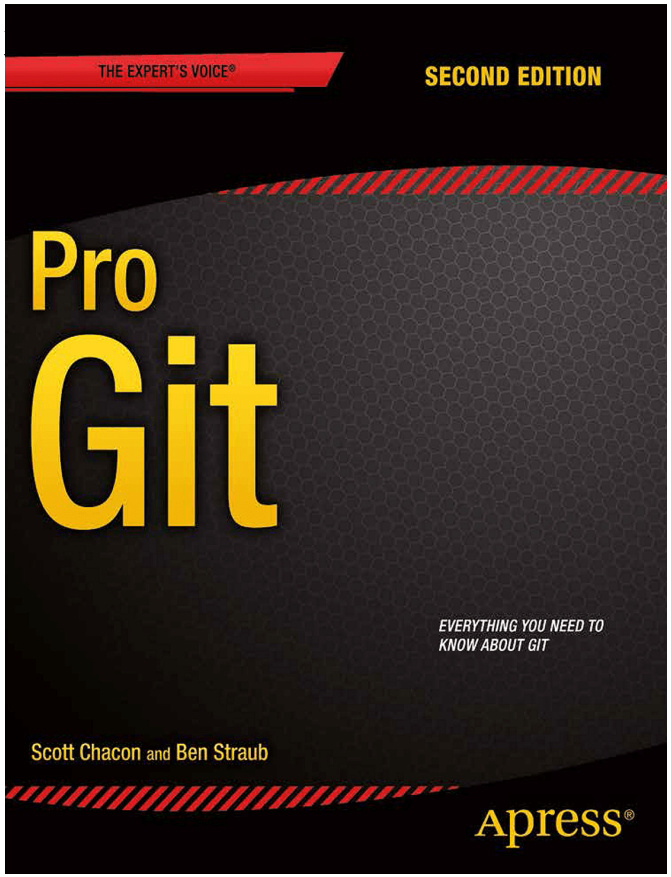
Introduction

- Git is free and open source distributed revision control system
- Originaly develloped for the Linux Kernel in 2005 by Linus Torvalds
- The main idea is to use a tool based on a decentralized server
- Still actively maintained by Junio Hamano and other developers
- Official website : https://git-scm.com

Introduction : Reference

Main reference : ProGit by Scott Chacon & Ben Straub

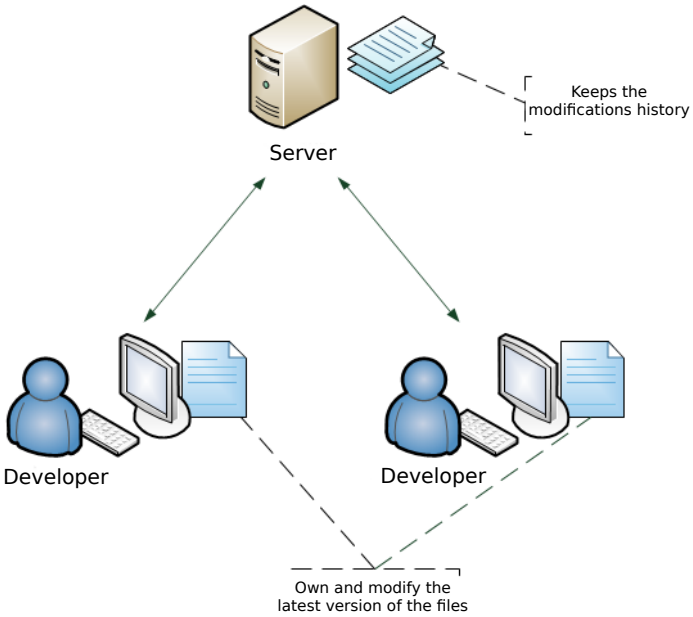
https://git-scm.com/book/



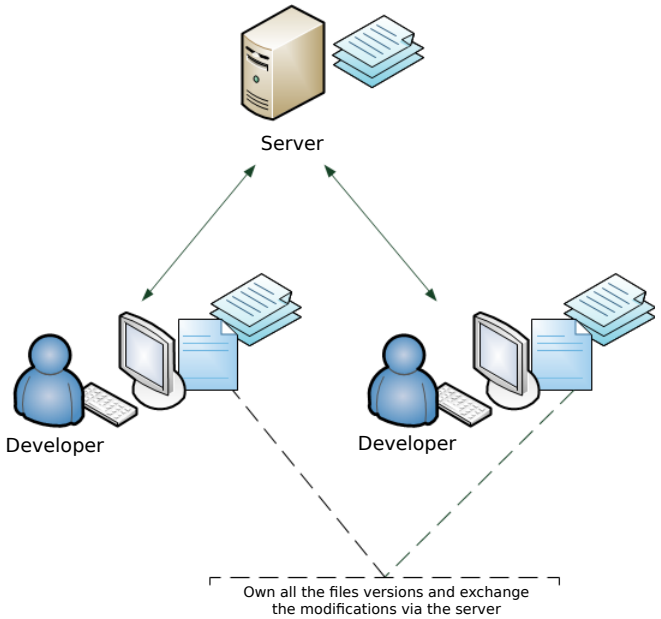
Introduction : Why Git ?

Main advantages :

- Decentralized version management (a server is not necessary needed)



CVS and SVN like



Git like

- Fast and robust
- Easy way to manage branches

But there are other alternatives : Mercurial and Bazaar

Installing Git

- Linux RedHat distribution and derivatives (Scientific Linux, CentOS, etc...)

  ```
  sudo yum install git-all
  ```

- Linux Debian distribution and derivatives (Ubuntu, Mint, etc...)

  ```
  sudo apt-get install git-all
  ```

- Windows

  https://git-scm.com/download/win

- MacOS

  https://git-scm.com/download/mac

Configuring Git

- Name or pseudo

```
git config --global user.name "name or pseudo"
```

- Email

```
git config --global user.email myemail@u-psud.fr
```

- Colors

```
git config --global color.diff auto
git config --global color.status auto
git config --global color.branch auto
```

- ~/.gitconfig can be edited

```
[user]
        name = karim
        email = karim.hasnaoui@u-psud.fr
[color]
        diff = auto
        status = auto
        branch = auto
[alias]
        ci = commit
        co = checkout
        st = status
        br = branch
```

- Aliases can be added

Initializing a repository

- How to clone an existing repository :

  ```
  git clone ssh://myname@myserveur/myrepository/git
  ```

  or

  ```
  git clone http://github.com/project/project.git
  ```

- How to creat a new repository in a directory :

  ```
  git init
  ```

  With the option `--bare` only the director ./git containing the historic will be conserved (only for the server)

Basic commands : "add" and "status"

`git add`

- It adds files in the repository
- Ex : `git add file1 file2 .. fileN`

`git status`

- It tells which files have been modified or not followed, and the branch name

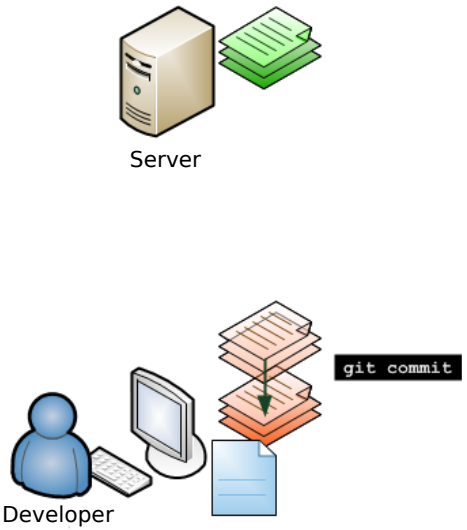
```
hasnaoui@atlas:~/Documents/omp_test_git$ git status
Sur la branche master

Validation initiale

Modifications qui seront validées :
  (utilisez "git rm --cached <fichier>..." pour désindexer)

        nouveau fichier : lect1.cpp
        nouveau fichier : lect2.cpp
        nouveau fichier : lect3.cpp
        nouveau fichier : lect4.cpp
        nouveau fichier : lect5.cpp
        nouveau fichier : lect6.cpp
        nouveau fichier : lect7.cpp
        nouveau fichier : test.cpp

Modifications qui ne seront pas validées :
  (utilisez "git add <fichier>..." pour mettre à jour ce qui sera validé)
  (utilisez "git checkout -- <fichier>..." pour annuler les modifications dans la copie de travail)

        modifié :         lect1.cpp

Fichiers non suivis:
  (utilisez "git add <fichier>..." pour inclure dans ce qui sera validé)

        lect1
        lect2
        lect3
        lect4
        lect5
        lect6
        lect7
        test
```

- Optional : `git status file_name`

Basic commands : "commit"

`git commit`

- It commits the modified files
- Ex : `git commit file1 file2 ... fileN`
- All of them : `git commit -a`



`git commit --amend`

- It allows to modify the last commit message

Basic commands : "log"

`git log`

- It gives an overview about the commits

```
hasnaoui@atlas:~/Documents/omp_test_git$ git log
commit deb8fbd74576e9521a8c74d0122460a705a9cf89
Author: karim <karim.hasnaoui@u-psud.fr>
Date:   Mon Feb 20 16:59:41 2017 +0100

    Third commit : additional comments have been added

commit 1d1b6cff3040f64f50af1ead19376a7dda7a0a09
Author: karim <karim.hasnaoui@u-psud.fr>
Date:   Mon Feb 20 15:20:46 2017 +0100

    Second commit

commit 8be474ddebcfe2f1810d6a7b52792a7f167ea6dc
Author: karim <karim.hasnaoui@u-psud.fr>
Date:   Mon Feb 20 15:17:19 2017 +0100

    First commit
hasnaoui@atlas:~/Documents/omp_test_git$
```

- Optionals :
  - Full details about modified lines : `git log -p`
  - Brief summary about modified lines : `git log -stat`

Basic commands : "reset"

`git reset HEAD`

- It cancels the commit but keeps the current changes
- Last commit : `HEAD`
- Before-last commit : `HEAD^`
- Before-before-last : `HEAD^^` or `HEAD~2`
- With the commit ID : `8be474ddebcfe2f1810d6a7b52792a7f167ea6dc`

`git reset --hard`

- It cancels the commit and also the current changes
- The options as same as before

`git reset HEAD -- file_name`

- It removes a file which had been added to be commited with `git add`

Basic commands : "rm" and "checkout"

`git rm --cached file_name`

- It removes a file from the repository without deleting it physically
- ⚠ The file is deleted physically without the option `--cached`

`git checkout`

- It gives a list of modified files since the last commit

`git checkout file_name`

- It restores a file as it was at the last commit
- Ex : `git checkout file1 file2 ... fileN`

`git checkout branch_name`

- It switches to a different branch

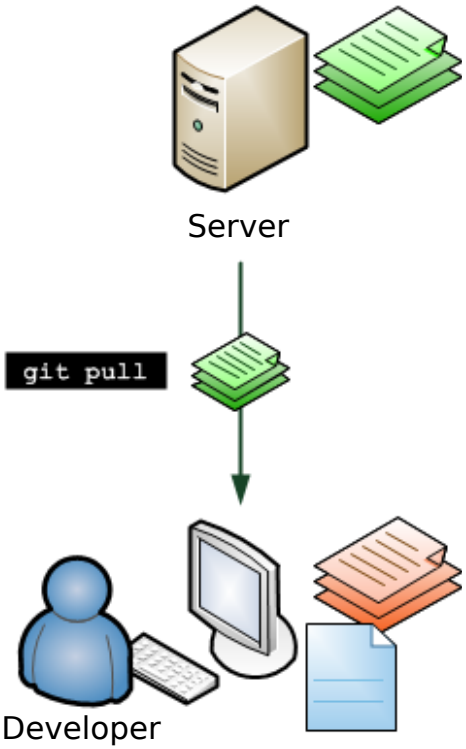
Basic commands : "pull" and "push"

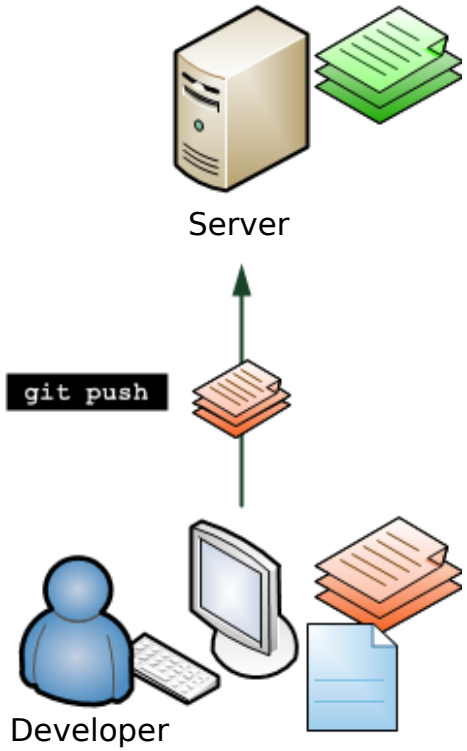
`git pull`

- It downloads the latest modifications from the server

`git push`

- It uploads the latest modifications to the server





Basic commands : "tag"

`git tag NAMETAG IDCOMMIT`

- It gives a tag to a given commit ID
- Ex : git tag v1.0 8be474ddebcfe2f1810d6a7b52792a7f167ea6dc
- Optionals :
  - The tags are only sent with a push by using : `git push --tags`
  - A tag can be deleted by using : `git tag -d NAMETAG`
  - Tags can be shown in the log by using : `git log --decorate`

Basic commands : "diff"

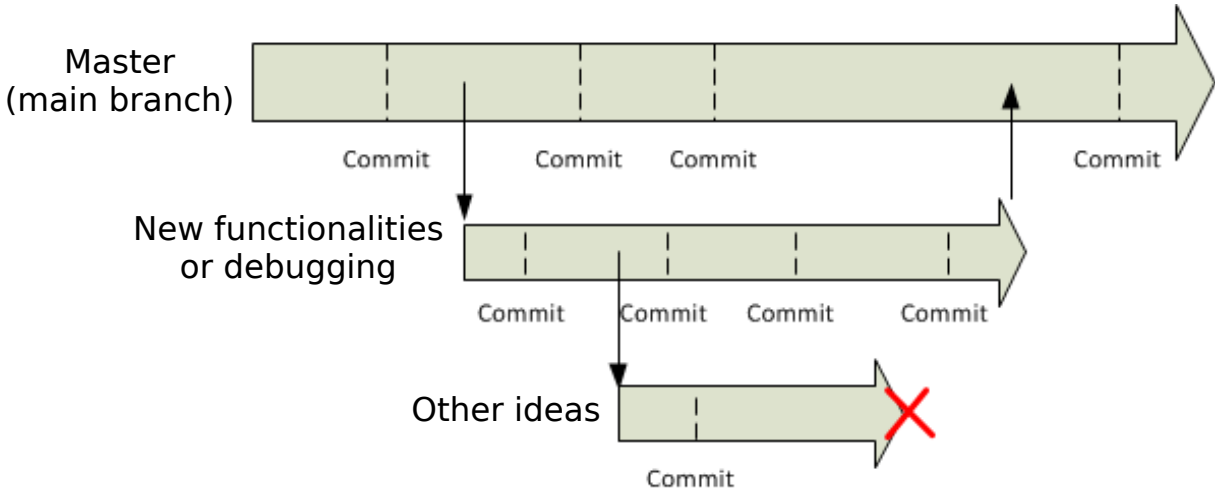
`git diff`

- It show all the modifications since the last commit

```
hasnaoui@atlas:~/Documents/omp_test_git$ git diff
diff --git a/lect6.cpp b/lect6.cpp
index 83c35ef..a9e4229 100644
--- a/lect6.cpp
+++ b/lect6.cpp
@@ -2,6 +2,8 @@
 #include <math.h>
 #include <omp.h>

+// toto titi tata
+
 // OpenMP lecture 6: runtime librairies

 double f(double x);
diff --git a/lect7.cpp b/lect7.cpp
index 2c70787..c8597c9 100644
--- a/lect7.cpp
+++ b/lect7.cpp
@@ -2,6 +2,8 @@
 #include <math.h>
 #include <omp.h>

+// tutu titi toto
+
 // OpenMP lecture 7: shared and private variables

 int main()
hasnaoui@atlas:~/Documents/omp_test_git$
```

- Optionals :
  - For a given file : `git diff file_name`
  - Comparison with a given commit : `git diff IDCOMMIT`

Branches : Introduction



Why branches ?

- To implement, try or test new functionalities
- To debug some parts of codes
- To split tasks between co workers

Branches : How to create a branch or to switch between branches ?

`git branch`

- It shows all the branches

```
hasnaoui@atlas:~/Documents/omp_test_git$ git branch
  debug
* master
  new_functionalities
hasnaoui@atlas:~/Documents/omp_test_git$
```

- The active branch is shown with a *

`git branch branch_name`

- It creats a new branch
- The new branch is creatd from the active branch

`git checkout branch_name`

- It switches to a different branch
- ⚠ By switching to another branch, it is strongly suggested to commit or to stash the modifications ! ! !

Branches : How to save modifications without doing any commit ?

Problematic : By switching to another branch without making any commit, all the modifications done will be seen by the other branches
$\Longrightarrow$ In order to avoid this problem, the changes can be stashed

`git stash`

- It saves/hides the modifications done since the last commit
- It is only effective on the branch where the command is used
- After using this command, the files look like as they were at the last commit

`git stash apply`

- It restores the modifications which have been saved/hidden on the branch

Branches : How to merge and delete a branch ?

`git merge branch_name`

- It merges one branch to another one
- Must be executed on the branch where the modifications will be merged
- Ex : the "debug" branch has to be merged to the "master" branch
  1. `git checkout master`
  2. `git merge debug`

`git branch -d branch_name`

- It deletes a branch which has been already merged
- If the branch has not yet been merged, it won't be deleted

`git branch -D branch_name`

- It forces to delete a branch which has not yet been merged

Branches : Link with server

`git branch -r`

- It shows the branches followed by the server
- Server branches always follows the syntax : `origin/server_branch`

`git branch --track localbranch origin/server_branch`

- It creates a local branch already existing on the server
- It is necessary to download the news by using the command : `git pull`
- ⚠ Before doing a `pull`, it is strongly suggested to check localy from which branch it is done ! ! !

Branches : Link with server

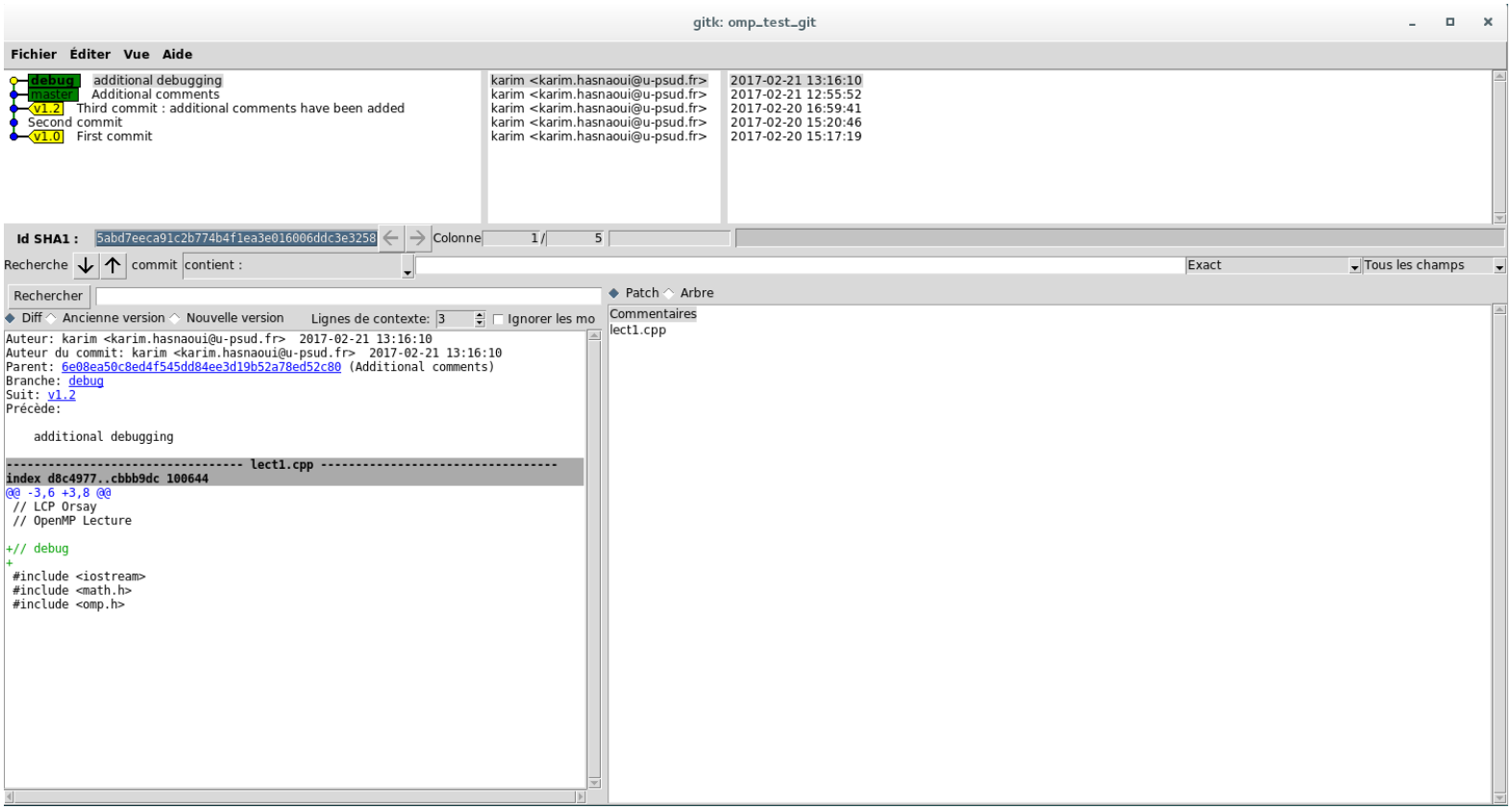
`git push origin :heads/name_branch_to_delete`

- It delete a branch on the server
- Local branches must be deleted by using the command : `git branch -r -d origin/name_branch_to_delete`

`git diff local_branch origin/server_branch`

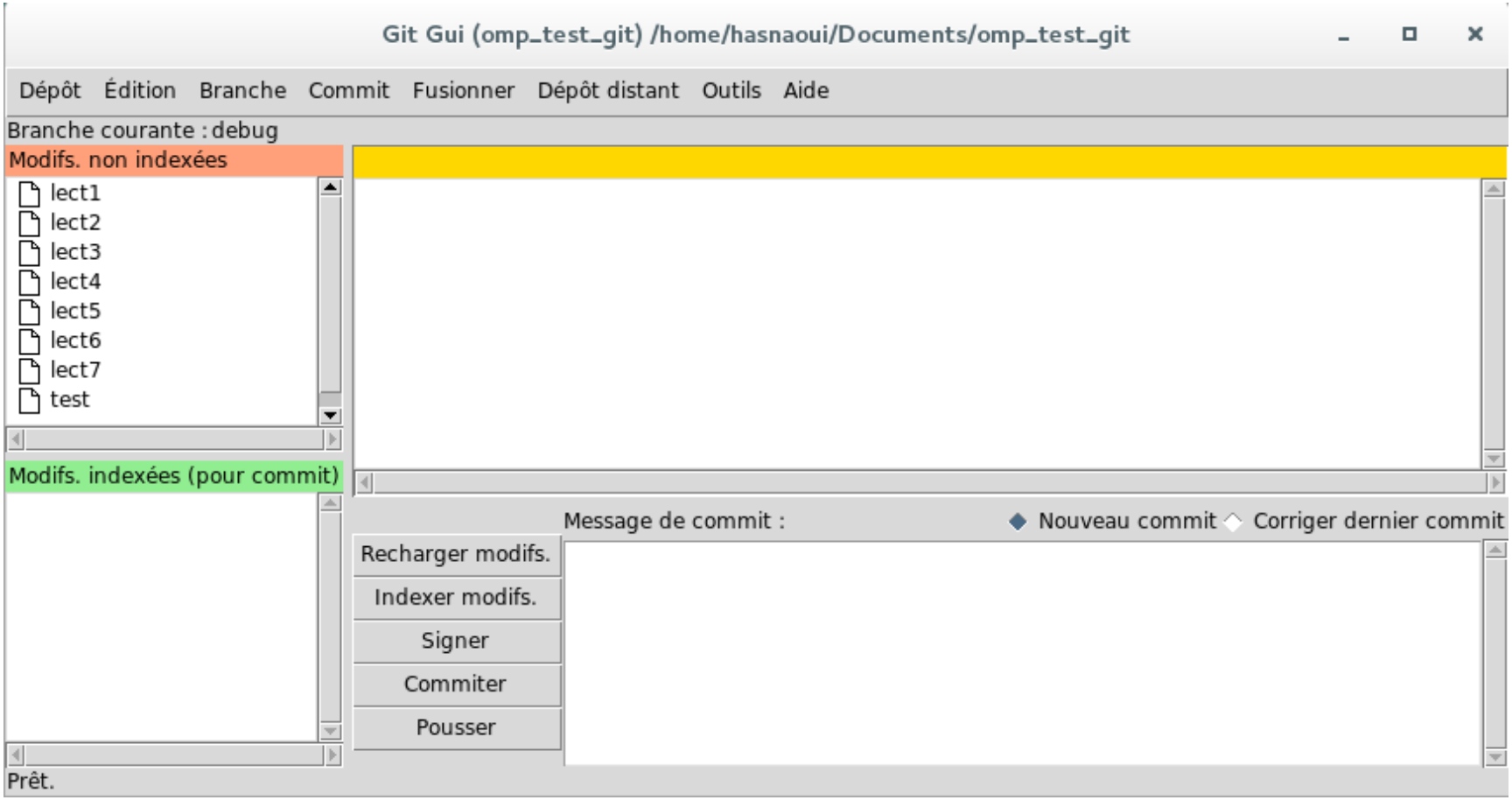
- It compares a local branch and a given branch on server
- Ex : `git diff debug origin/master`

Other tools : gitk



- It shows the logs, branches and files modifications
- It can also loads **git-gui**

Other tools : git-gui



- For the users that don't like to use commands