

# Git, the stupid content tracker

Jérémie Laval

UTBM - LO22

## 1 Introduction

- Glossaire
- Historique
- Caractéristiques
- Installation

## 2 Démonstrations

- Utilisation de base
- Travail à plusieurs
- Les branches

## 3 Conclusion

# Sondage

- Combien savez-vous ce qu'est un système de contrôle de révision ?

# Sondage

- Combien savez ce qu'est un système de contrôle de révision ?
- Combien utilise un système de contrôle de révision ?

# VCS, DVCS, késako ?

## Définition

VCS : Version Control System (système de gestion de révision en français). Il s'agit d'un composant logiciel qui permet de stocker, visualiser, manipuler plusieurs versions dans le temps d'une même arborescence de fichier.

## Définition

DVCS (Distributed Version Control System) porte plus précisément sur les VCS dont le workflow est distribué

# Vocabulaire des VCS

## Définition

- *Diff* désigne un ensemble de différence entre deux fichiers. Les deux fichiers peuvent être le même à deux instants différents.
- *Révision* désigne l'état de l'arborescence à un instant  $t$  donné. Une révision est composé généralement d'un identifiant numérique croissant, d'un ensemble de diff et de divers autres méta-données (date, auteur, changelog...).
- *Dépôt* désigne l'endroit où sont stocker les révisions dans un format propre au VCS.
- *Historique* désigne l'ensemble des révisions stocké dans un dépôt.

## Vocabulaire des VCS 2

### Définition

- *Copie de travail* désigne l'arborescence sur laquelle l'utilisateur travaille. Elle correspond à la révision la plus récente plus les changements que l'utilisateur n'a pas encore fait valider.
- *Commit* désigne la validation d'une copie de travail comme une révision. Lors d'un commit, un diff entre la dernière révision et la copie de travail est donc généré.

# Historique



Git (prononcé “*Guit*”), a été inventé par Linus Torvalds en 2005 suite à l’abandon du logiciel BitKeeper par le projet Linux.

Aujourd’hui, il utilisé par de nombreux gros projet libre tel que GNOME, freedesktop, GStreamer, ...



# Description de Git

- Ecrit en C. Originellement très lié au kernel Linux (utilisation brute des inodes pour des raisons de performances).
- Très grande vitesse d'exécution  $\Rightarrow$  adapté pour les projets de grande taille.
- Distribué et décentralisé. Chaque utilisateur est à la fois client et serveur.
- Concept de branches prépondérant, *"branching is cheap"*.

## Via vos paquets

- Ubuntu/Debian : `apt-get install git-core`
- Fedora/Red Hat : `yum install git-core`
- OpenSuse : `yast install git-core`
- Arch Linux : `pacman -S git`
  
- Mac : `port install git-core`
- Windows : <http://www.mingw.org>

## Via les sources

Sources dispo sur <http://kernel.org/pub/software/scm/git/>

```
tar -xzf git.tar.gz; ./configure; make; sudo make install
```

FTW

## Avant tout

Toutes les commandes qui seront présentés ont leur propre page de manuel :

```
git help commande
```

# Initialisation d'un dépôt

Commandes utilisés :

- `git init`
- `git add`
- `git diff`
- `git commit`

# Premiers changement

Commandes utilisés :

- `git diff`
- `git commit -a`
- `git reset`

# Au début

Commandes utilisés :

- `git clone`

# Avec des patches

Commandes utilisés :

- `git diff`
- `git apply`



# Avec la méthode distribuée

Commandes utilisés :

- `git pull`
- `git push`
- `git daemon`

⇒ Exemple de workflow : le kernel Linux.

# Pourquoi

- Changements isolés.
- Plus de souplesse pour les fusions.
- Partage facilité.
- Les branches sous git c'est facile, faut en profiter !

# B-aba

Commandes utilisés :

- `git checkout -b`
- `git branch`
- `gitk`

# Divergence

Commandes utilisés :

- `git commit`
- `git checkout`
- `gitk`

# Fusion

Commandes utilisés :

- `git diff`
- `git merge`
- `gitk`

## Conseils et liens

- `man` et `git help` sont vos amis.
- Ressources ouaib intéressantes :
  - <http://kernel.org/pub/software/scm/git/docs/gittutorial.html>
  - <http://git-scm.com/documentation>
- Hébergement gratuit
  - Chez les tchèques - <http://repo.or.cz>
  - Version kikoolol - <http://github.com>
  - Sur votre machine - Git over ssh