APPS@UCU

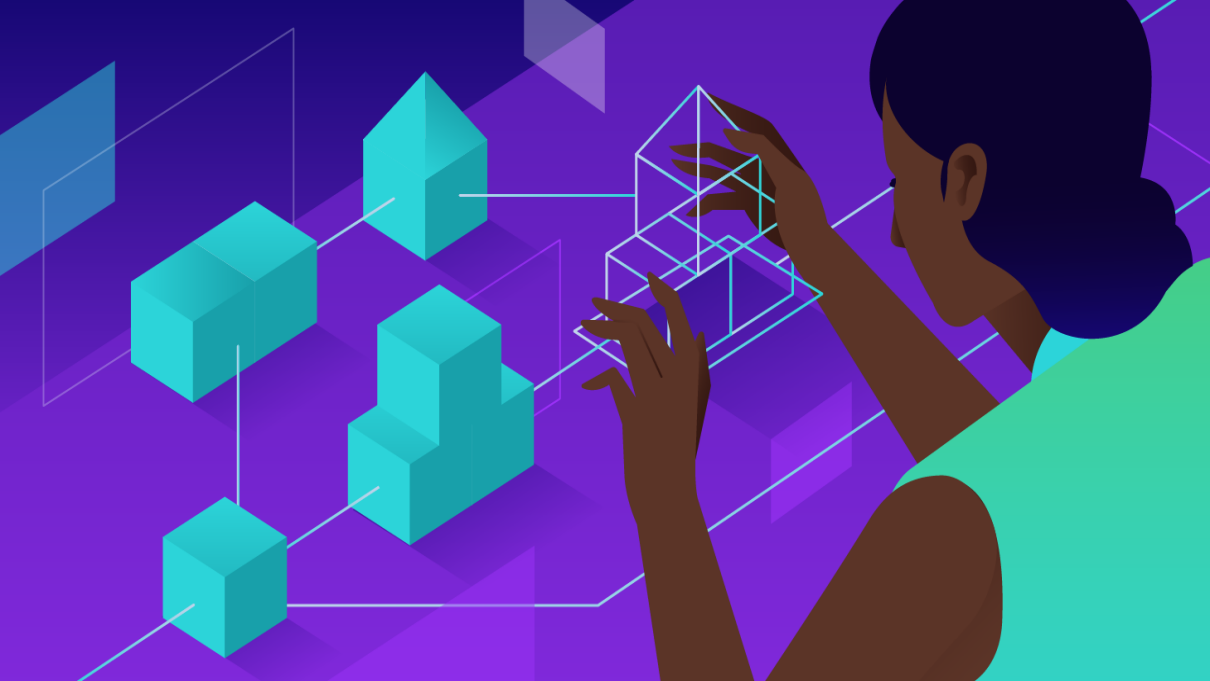
Linux course

Version control systems

Morhunenko Mykola

APPLIED SCIENCES FACULTY

Contents



Version control systems

- Sooner or later, during the development process, it is necessary to check, what was before, how it became broken
- Maybe it's easier to use Ctrl+z , but it's impossible to check what was three weeks ago with any keybinding
- So in 1972 people started to think about version control systems
- Firstly, it had been just a tool for saving a history of binary files, but in 1977 the first source code control system was introduced
- The main idea behind - to save the program source code on some checkpoints (commits) , add features, develop them leaving trunk untouchable (branches) , merge new features with a trunk, and release some tags
- Since than, the concept itself was developed, and a lot of version control systems have apeared

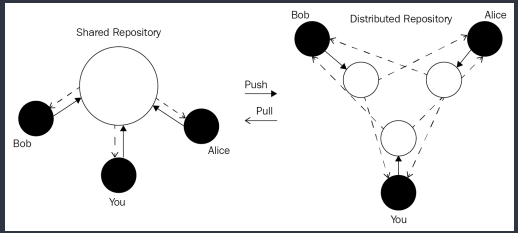
git

Linus again...

- Linux kernel is a huge project, and it is important to have some source-control management system (SCMs) to maintain it
- From 2002 to 2005 BitKeeper, a proprietary SCMs was used to maintain the project
- At some point (3 April 2005), Linus Torvalds realized, that existing tools are not suitable for Linux development, so in three days he anounced a project and become a self-hosting of Git on the next day
- It was totally different SCMs. Linus maintained it for half a year, and Junio Hamano has been the core maintainer since then
- It was open-source, free software, with a very strong safeguards against corruption, either accidental or malicious
- Torvalds sarcastically quipped about the name Git , means unpleasant person in British English
- He said: *"I'm an egotistical bastard, and I name all my projects after myself. First 'Linux', now 'git'."* =)

What makes Git so good

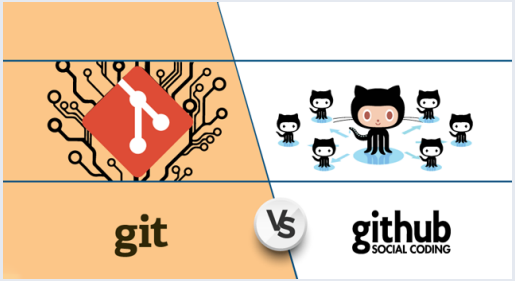
- Strong support for non-linear development
- Distributed development
- Efficient handling of large projects
- Toolkit-based design
- Pluggable merge strategies
- And more other features
- It's hard to find any statistics, but that is clear - Git is the most popular SCMs of ourdays



github
SOCIAL CODING

Git is not GitHub

- GitHub, Inc - provifer of internet hosting for software development and version control using git
- It offers all the functionality of Git + it's own features
- Since 2018 - subsidiary of Microsoft
- Not an Open Source project , but there is a forum for feature requests...
- As of January 2020, GitHub reports having over 40 million users
- More about it's features after Git usage part



```
$ git push
```

Git is not GitHub

- First difference - GitLab was created by Ukrainian people, in Ukraine =)
- It has deffinitely more features, than GitHub , but there is no critical difference
- It is Open Source , unlike github
- It is possible to have the same repositoru on both servers, and I do it sometimes
- So as for 2021 it’s just a question of taste

GIT USAGE


IN CASE OF FIRE
Git Commit
Git Push
Git Out

Creating a repo

- A repository contains all of your project's files and each file's revision history. You can discuss and manage your project's work within the repository.
- Repository is NOT a project folder. Repository is s a *data structure that stores metadata for a set of files or directory structure*
- Command to inicialize a repo in your current folder

```
git init
```

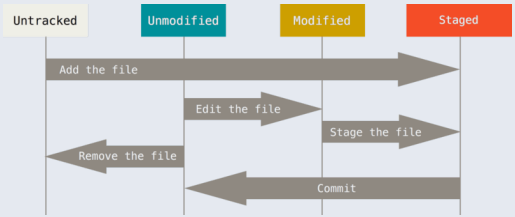
- use git add command to cpecify files you want to track, followed by git commit - add a cpecifick message to your commit

```
git add *.sh
git add .gitignore
git commit -m "add gitignore file; add scripts for some task"
```

- How to write correct commit messages is another art, but remember to write meaningdull messages
- So at this point, we have a Git repository in our project directory with tracked files and an initial commit

Changes to the repo

- At this point you have a git repo with scripts and some files
- Each file in your working directory can be in one of two states: tracked or untracked
- Tracked files are files that were in the last snapshot, as well as any newly staged files; they can be unmodified, modified, or staged
- In short, tracked files are files that Git knows about
- Untracked files are everything else
- Use git status to check the status of each file in current directory
- Files in a .gitignore are ignored by git repo



Manipulations with repo

- As far as git is a decenralized system, you already have your repo with all version control features
- But now about the most powerful git feature and why we use it - remote repo
- You can either clone existing repo, or add a remote to local one
-

Sources

Sources

- Version control systems comparison
- Why Git is Better than X
- Git Wiki
- GitHub Wiki
- GitLab history
- GitHub documentation
- Git documentation