Free Software and Version Control 101

A introductory course on FOSS, git, curl and web integrations

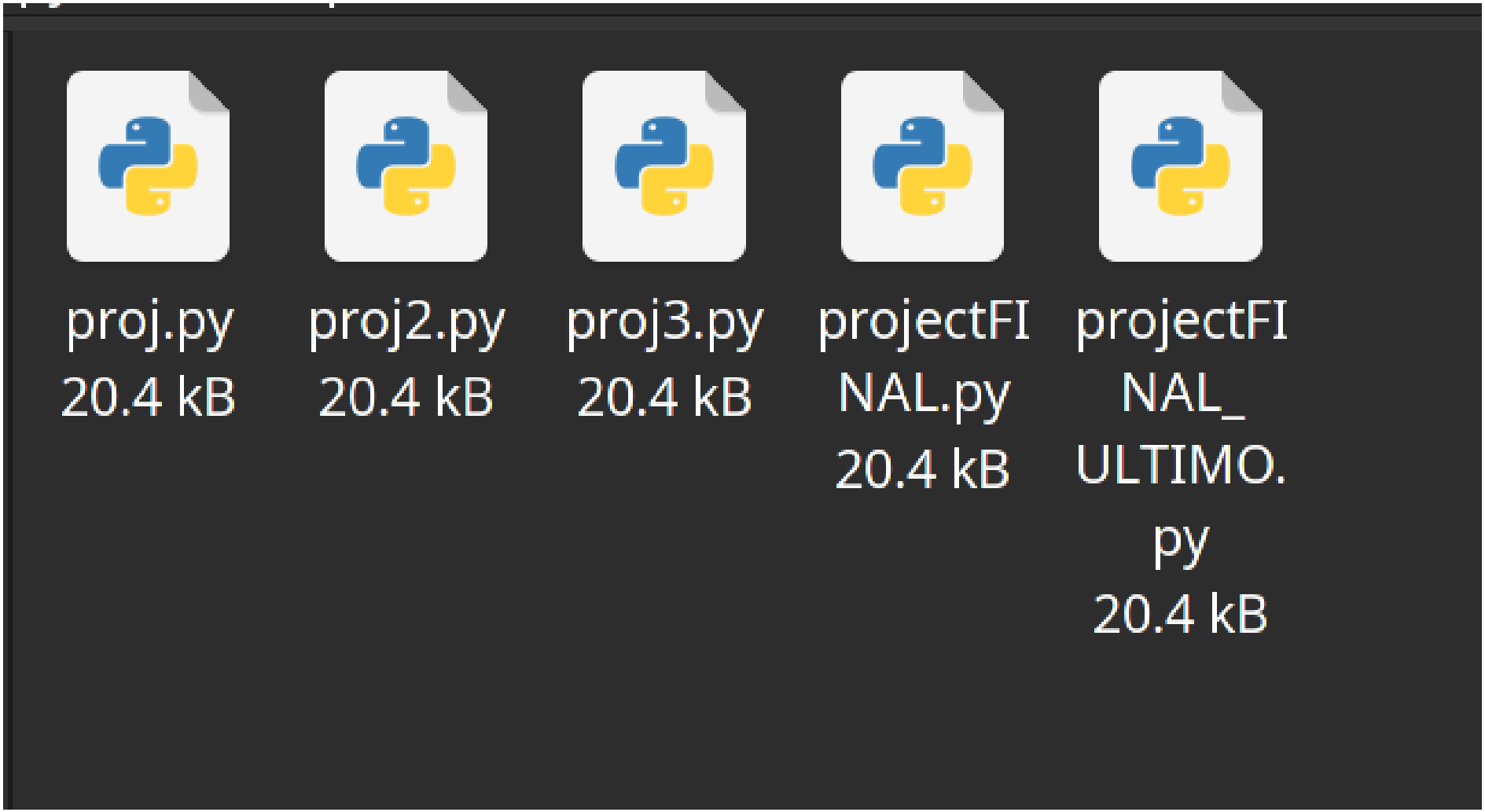
João Barreiros C. Rodrigues

HackerSchool

Section 1

Version control?

It has happened to all of us!



Time to ditch this...

For something waaaay better

```
* commit eebc2e011fa88bfae93e8105e665fa5873499569 (HEAD -> master, origin/master)
| Author: Francisco Carvalho <franciscojcarvalho@tecnico.ulisboa.pt>
| Date:   Thu Oct 13 16:42:39 2022 +0100
|
|     Database changes (to be implemented in backend)
|
* commit bf2162b4080243bb4458f916319a53ebfa3d9253
| Author: Francisco Carvalho <franciscojcarvalho@tecnico.ulisboa.pt>
| Date:   Sat Oct 8 19:36:46 2022 +0100
|
|     Login system bypass as it's not needed. LF endings
|
* commit bf5c4c51b9fba1ffdc18b44abf24491d7a1bf6c4
| Author: Francisco Carvalho <franciscojcarvalho@tecnico.ulisboa.pt>
| Date:   Fri Oct 7 14:32:27 2022 +0100
|
|     .desktop file and autorun script
|
* commit 0b9bfc255efc9fc897189b46885acc7d51251e2c
| Author: Francisco Carvalho <franciscojcarvalho@tecnico.ulisboa.pt>
| Date:   Fri Sep 30 21:01:59 2022 +0100
|
|     Code cleanup
|
* commit 764c31a6c26b3ef19e8921895cbeb50cc271920f
| Author: Francisco Carvalho <franciscojcarvalho@tecnico.ulisboa.pt>
| Date:   Fri Sep 30 20:51:04 2022 +0100
|
|     Bug: week number stuck at 1
|
```

What is version control?

Version control is the practice of managing and documenting *data* (code, schematics, etc.) iterations.

It is particularly important in our context of Free and Open Source software, as a careful documentation of alterations between versions and the ability to inspect older or deprecated sources can make issue resolution and feature integration much more agile.

git?

Git is a version control software created by Linus Torvalds (which also created the Linux Kernel). Its free software under the GPL v2.0.

Git allows cloning, pulling, pushing, etc. of data stored in git instances.

curl?

Curl (short for client-url), is the command line tool that makes use of libcurl, a data transfer library that supports a array of network protocols such as FTP, HTTP, etc.

We will use curl to communicate with the GitHub API in the next sections.

Curl and libcurl are FOSS licensed under the curl license, based on the MIT License, and compatible with the GPL v3.0

Section 2

How does git instance work?

Repositories and actions

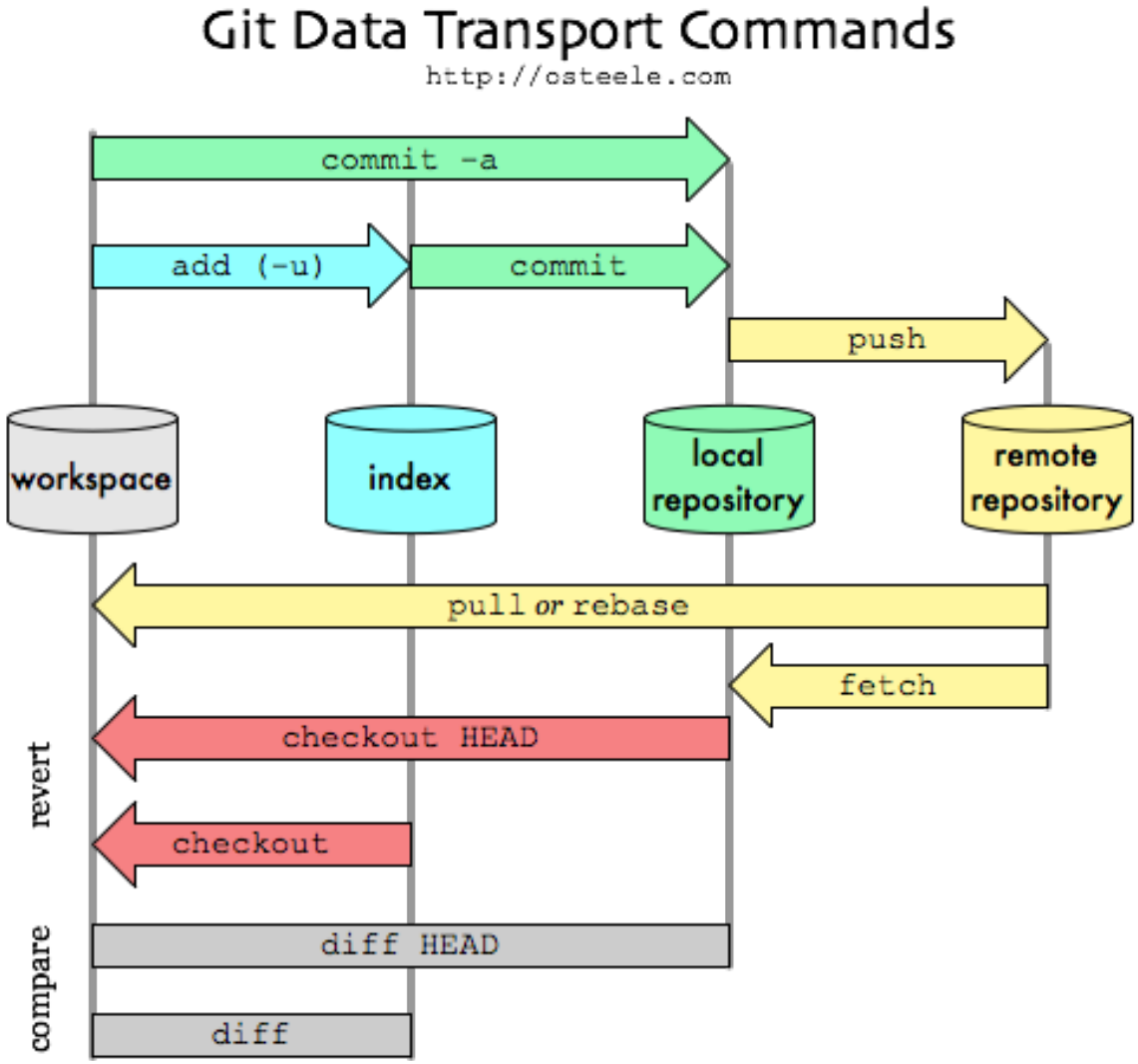


Figure 2: Actions and interactions between repositories

A pratical example: hackerschool.io

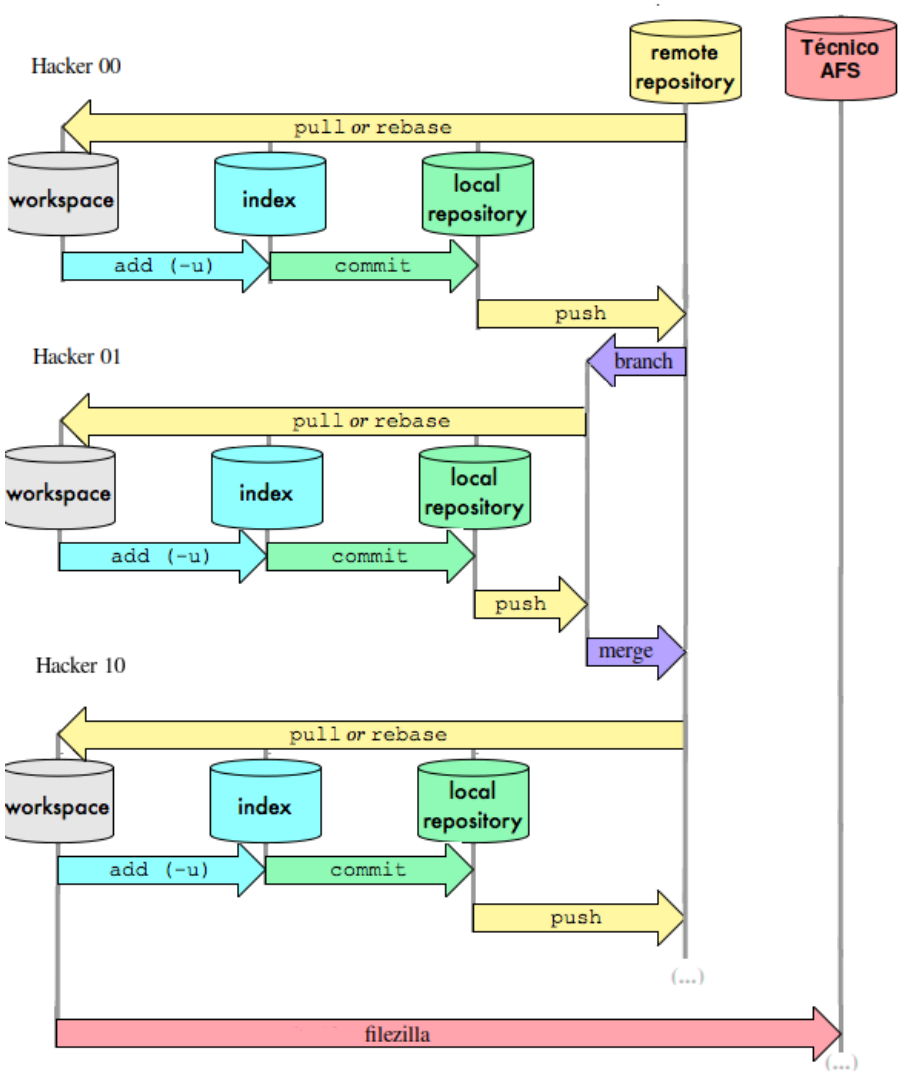


Figure 3: Git/GitHub flow of the hackerschool.io repository

Section 3

Tying our work with freedom

On FOSS

Licenses

Section 4

Let's Start!

Setting up your git/GitHub environment

- Create a GitHub account (using your institutional e-mail is often valuable).
- Get the git and curl packages.
- Generate a OAuth key, ssh key, or any mean of remote authentication.
- Save it somewhere safe (encrypt it with gpg, or use a password manager (for example: keepass).

Let's waddle back to the terminal

```
git config --global user.name "@user.name"

git config --global user.email @user.email
```

Setting up a GitHub repository with curl and git

Let's start with creating a remote and local repository

```
curl -u @user https://api.github.com/user/repos -d \
'{"name":"@string","private":false}'
```

```
mkdir @string && cd @string
```

Now we can initialize our local repo and link it to the remote one

```
git init
```

```
git remote add origin https://github.com/@name/@string.git
```

Section 5

Your first commit!

Git Status

This command alows you to view the state of your project (repo)

```
francisco@archboxSigma:~/repos/studyTracker$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   main.py
        modified:   templates/dash.html

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        README.md
        __pycache__/
        app/
        config.ini.old
        diagramaEA.png
        ects_weeks.py
        etcs.db
        populate.sql
        static/favicon.svg
        templates/timer.html
        test.sql

no changes added to commit (use "git add" and/or "git commit -a")
```

Add

- When we want the SCM to start track a specific file we use `git add <file_path>`
- We also use git add to stage files changes for commiting

```
francisco@archboxSigma:~/repos/studyTracker$ git add main.py README.md
francisco@archboxSigma:~/repos/studyTracker$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   README.md
        modified:   main.py

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   templates/dash.html
```

Figure 4: Git Add

Commit

- To commit (record the selected changes to the history) we use the command `git commit -m "commit message"`

Push

Section 6

Working together!

Cloning

Pulling

Forking with the web

Merging with the web

Other Actions!