Free Software and Version Control 101

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

J. Rodrigues F. Carvalho

HackerSchool

Section 1

Version control?

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.

Section 2

Tools

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 3

How does git instance work?

Repositories and actions

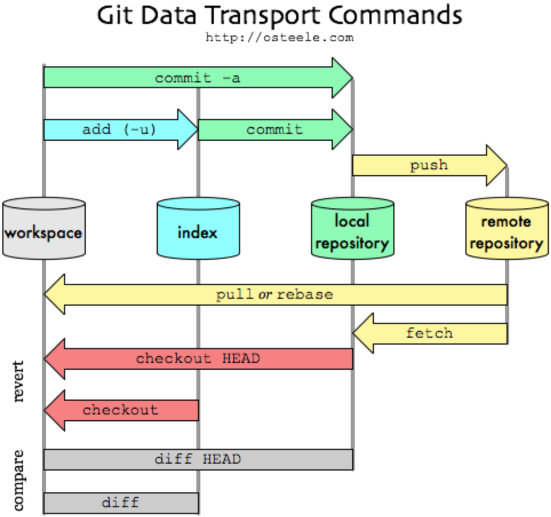


Figure 1: Actions and interactions between repositories

A pratical example: hackerschool.io

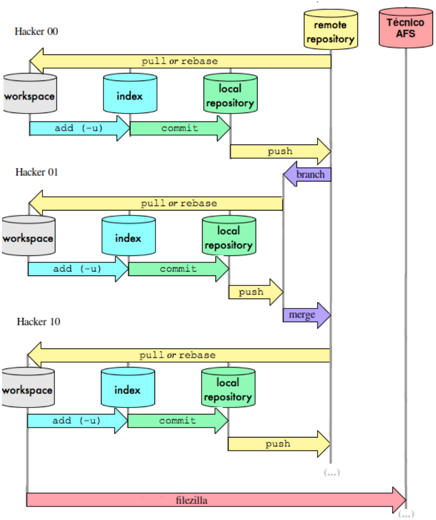


Figure 2: Git/GitHub flow of the hackerschool.io repository

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!

Add

Commit

Push

Section 6

Working together!

Cloning

Pulling

Forking with the web

Merging with the web

Other Actions!

Section 7

Tying our work with freedom

On FOSS and CC

The first written document that described FOSS as a trend among hackers, programmers, engineers, etc. was the GNU Manifesto. This file also gave the philosophical foundations for the Free Software Foundation Network, and FOSS worldwide.

In the context of HackerSchool FOSS is a core principle. Initiated in the administration of 22-23, HackerSchool has embraced free alternatives such as *GNU/Linux distros, FreeCAD, Jitsi, Signal*, allowing hackers to walk a path that is flexible, secure, and overall hacker-y.

Since we benefit from this communal effort it is only fair that we also contribute to the greater good, therefore all HackerSchool code and documents are non-proprietary.

The two software architectures

- **The Cathedral**

  We can apply this architectural decision to both:

  - **Free Software**

    The source code is centralized in an organizational environment, however it is released with any main binary. You will rarely look to the code in development.

  - **Proprietary Software**

    The source code is never released, it is kept confined within the Corporations walls. The binary is spewed out of it, but you will have to reverse it to understand what it does!

The two software architectures

- **The Bazaar**

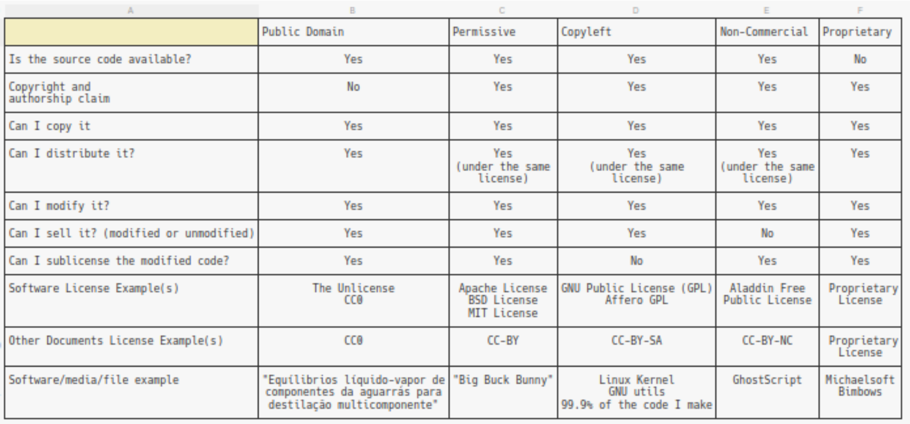
  The Bazaar is characteristic of free software.

  In a Bazaar architecture the source code is visible at all times, being normally communal projects, made of differently written modules. Its all nodes that compose a de-centralized tree!

  This architecture as the advantage of going according to Linus's Law:

  "given enough eyeballs, all bugs are shallow"

Licenses



	Public Domain	Permissive	Copyleft	Non-Commercial	Proprietary
Is the source code available?	Yes	Yes	Yes	Yes	No
Copyright and authorship claim	No	Yes	Yes	Yes	Yes
Can I copy it	Yes	Yes	Yes	Yes	Yes
Can I distribute it?	Yes	Yes (under the same license)	Yes (under the same license)	Yes (under the same license)	Yes
Can I modify it?	Yes	Yes	Yes	Yes	Yes
Can I sell it? (modified or unmodified)	Yes	Yes	Yes	No	Yes
Can I sublicense the modified code?	Yes	Yes	No	Yes	Yes
Software License Example(s)	The Unlicense CC0	Apache License BSD License MIT License	GNU Public License (GPL) Affero GPL	Aladdin Free Public License	Proprietary License
Other Documents License Example(s)	CC0	CC-BY	CC-BY-SA	CC-BY-NC	Proprietary License
Software/media/file example	"Equilibrios líquido-vapor de componentes da aguarrás para destilação multicomponente"	"Big Buck Bunny"	Linux Kernel GNU utils 99.9% of the code I make	GhostScript	Michaelsoft Bimbows

Figure 3: Licenses and properties