

Free and Open Source Software

James C. Craven

April 17, 2025

Table of Contents

1 Introduction

2 History

3 Licenses

4 Conclusion

Why This Matters

Why This Matters

- 1 Portfolio building
 - Personal projects
 - Contributing to open source

Why This Matters

- ➊ Portfolio building
 - Personal projects
 - Contributing to open source
- ➋ Using open source in professional settings
 - Legality of using others' projects in proprietary code
 - Working in corporate-owned FOSS

Why This Matters

- 1 Portfolio building
 - Personal projects
 - Contributing to open source
- 2 Using open source in professional settings
 - Legality of using others' projects in proprietary code
 - Working in corporate-owned FOSS
- 3 Understanding your rights

Definitions

Free and Open Source Software (FOSS):

Software where the source code is made publicly available, allowing anyone to freely use, study, modify, and distribute the software.



Definitions



Free Software

- Four Core Freedoms:
 - 1 Use/run program for any purpose
 - 2 Study and modify source code
 - 3 Redistribute original source code
 - 4 Redistribute modified source code
- Mostly Ideological:
*"Free as in freedom of speech,
not free as in beer"*

Definitions



Free Software

- Four Core Freedoms:
 - 1 Use/run program for any purpose
 - 2 Study and modify source code
 - 3 Redistribute original source code
 - 4 Redistribute modified source code
- Mostly Ideological:
*"Free as in freedom of speech,
not free as in beer"*

Open Source

- Rebranding of Free Software
Movement to be more appealing
- Focuses on marketing utility of FOSS

The Early Days and Unix

- In the early 70s, Commercial software was distributed as source code.
- Users frequently modified and/or distributed modifications to software, and it was legal to do so.
- By the late 70s, UNIX had thousands of users.
- With the rise of general purpose operating systems, desire to monetise software increased in the mid 1980s.



Figure 1: Ken Thompson (seated) and Dennis Ritchie in Bell Labs, 1972

Free Software Movement

- AT&T started commercialising UNIX more, and required NDAs to receive the software's source code.
- Outraged with this, Richard Stallman started the GNU project, which intended to build a UNIX-like operating system "for free, forever."
- Stallman conceived the Four Core Freedoms, and founded the Free Software Foundation. His ideas became very influential for the next decade to come.



Figure 2: Richard Stallman, founder of the GNU Project.

Open Source Initiative

- By the early 1990s, the GNU Utilities were complete, but the operating system was missing the kernel.
- Linus Torvalds, began writing the Linux Kernel, causing another surge in interest in free software.
- Despite public interest, corporations were hesitant to interact with something "free," and thus not monetisable.
- In 1998, free software was rebranded as open source software.

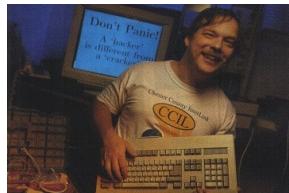


Figure 3: Eric Raymond, founder of OSI (above) and Linus Torvalds, designer of the Linux Kernel (below)

Permissive Vs Copyleft

The ideological divide gave rise to similar but significantly different classes of FOSS licenses: **Permissive** and **Copyleft** licenses.

| Purpose | Permissive | Copyleft |
|-------------------|------------|----------|
| Movement | OSI | FSF |
| Use (Priv./Comm.) | ✓ | ✓ |
| Modify | ✓ | ✓ |
| Distribute | ✓ | ✓ |
| Different License | ✓ | × |
| Liability | × | × |
| Warranty | × | × |

Popular Licenses

Popular Licenses

- **GNU Public License (GPL v3):**

Well-known copyleft license created by Stallman and the GNU License.

Popular Licenses

- **GNU Public License (GPL v3):**
Well-known copyleft license created by Stallman and the GNU License.
- **MIT License:**
Most popular permissive license. Essentially allows end-user to do anything.

Popular Licenses

- **GNU Public License (GPL v3):**
Well-known copyleft license created by Stallman and the GNU License.
- **MIT License:**
Most popular permissive license. Essentially allows end-user to do anything.
- **BSD License:**
Permissive license originally created to accompany Berkley Software Distribution's fork of UNIX. Exists in 2, 3, and 4 clause forms with varying levels of strictness

Popular Licenses

- **GNU Public License (GPL v3):**
Well-known copyleft license created by Stallman and the GNU License.
- **MIT License:**
Most popular permissive license. Essentially allows end-user to do anything.
- **BSD License:**
Permissive license originally created to accompany Berkley Software Distribution's fork of UNIX. Exists in 2, 3, and 4 clause forms with varying levels of strictness
- **Apache License:**
Open Source License created by the Apache Software Foundation. Unique in its explicit denial of trademark rights to the product. Affords additional protections of GPL without being copyleft

Licenses Summary

| License | Popularity | Copyleft | Disclose Mod. | Patent |
|---------|------------|----------|---------------|--------|
| MIT | 1 | x | x | x |
| Apache | 2 | x | ✓ | ✓ |
| GPLv3 | 3 | ✓ | ✓ | ✓ |
| BSD 3 | 4-5 | x | x | x |
| BSD 2 | 10-12 | x | x | x |

<https://innovationgraph.github.com/global-metrics/licenses>

How to Use Licenses

Manually:

- 1 Create a file in root directory of your project called LICENSE.
- 2 Copy and paste the contents of a suitable license into the file.

OR, using GitHub:

- 1 Select add file in your repository
- 2 Name the file LICENSE, and select the prompt for a License template when prompted
- 3 Find a suitable license, and select "review and submit."
- 4 Commit the changes when prompted.

Conclusion

- Understanding the software licensing is essential not only for your personal projects, but also any other project you contribute to publicly or privately.
- Copyleft and permissive licenses are similar, but have unique aspects that must be considered before usage, especially as a component of a larger project.
- Licensing your projects is not only important, but it's really easy to do and can potentially save you future headaches.

Questions

Thank You! Questions?

