LICENSING in FOSS

Short Notes on Licensing

What Is a License?

1. A license is a legal agreement granting permission to use, modify, distribute, or access intellectual property.

2. It defines the scope and limitations of usage for software, media, or other creations.

3. Licenses ensure the creator retains control while allowing specified rights to others.

How to Create Your Own Licenses

1. Identify the rights you want to grant (e.g., usage, modification, distribution).

2. Define limitations (e.g., non-commercial use, no modifications).

3. Use existing templates (e.g., Creative Commons, MIT) to ensure legal compliance.

4. Consult a legal expert to draft and validate your license terms.

S.No	Feature	GPL (General Public License)	LGPL (Lesser General Public License)	Apache License	BSD License	MIT License
1	Copyleft Enforcement	Yes (Strong Copyleft)	Partial (Weak Copyleft)	No	No	No
2	Allows Use in Proprietary Software	No	Yes (when linking)	Yes	Yes	Yes
3	Requires Attribution	Yes	Yes	Yes	Yes	Yes
4	Patent Protection	No explicit clause	No explicit clause	Yes	No explicit clause	No explicit clause
5	Redistribution Terms	Must distribute under GPL	Library must remain LGPL	Attribution and license copy required	Attribution and license copy required	Attribution and license copy required
6	Flexibility for Commercial Use	Limited to GPL compliance	More flexible than GPL	Highly flexible	Highly flexible	Highly flexible

LICENSING in FOSS

Important FOSS Licenses

Apache License

1. Permissive license allowing use, modification, and distribution with attribution.

2. Includes patent grants to protect users from patent claims.

BSD License

1. Permissive license with minimal restrictions.

2. Allows use, modification, and redistribution, often used in academic and open-source projects.

Public License (PL)

1. Broad term covering licenses for public use (e.g., GNU GPL).

2. Focuses on software freedom and accessibility.

Lesser General Public License (LGPL)

1. A more permissive version of the GPL allowing linking with non-GPL libraries.

2. Balances open-source benefits with proprietary software integration.

Copyrights and Copylefts

Copyrights

1. Protects the creator's rights to their intellectual property (e.g., software, art, text).

2. Grants exclusive rights to reproduce, distribute, and modify works.

Copylefts

1. A licensing method ensuring derived works remain free and open-source.

2. Requires redistribution under the same terms as the original license.

Patent

1. Protects inventions and grants exclusive rights to the inventor for a fixed period.

2. Encourages innovation by securing ownership of new ideas and technologies.

3. Software patents are controversial due to potential limitations on innovation.

LICENSING in FOSS

The **Apache License** is a type of open-source software license. Here's a simple explanation:

1. **Freedom to Use**: Anyone can use the software for personal, commercial, or educational purposes without restrictions.

2. **Freedom to Modify**: You can change or improve the software as you like.

3. **Freedom to Distribute**: You can share the original software or your modified version with others.

4. **Requirement for Attribution**: If you distribute the software (original or modified), you must credit the original creators. This is done by including a copy of the license and a notice stating what changes you made.

5. **Patent Protection**: The license includes a **patent clause**, which means the creators also give you a license to use any patents related to the software. This protects you from being sued for patent infringement while using or modifying the software.

6. **No Warranty**: The software comes "as is," meaning the creators aren't responsible if it doesn't work as expected or causes issues.

In short, the Apache License gives you a lot of freedom with the software while ensuring the original creators are acknowledged and users are protected from patent claims.

The **BSD License** is another type of open-source license, designed to be simple and flexible. Here's a straightforward explanation:

1. **Freedom to Use**: You can use the software for any purpose, including personal, educational, or commercial use.

2. **Freedom to Modify**: You're allowed to modify the software as much as you like.

3. **Freedom to Distribute**: You can share the original software or your modified version with others.

4. **Minimal Requirements**: If you distribute the software:

    - You must include the original copyright notice and license terms.

    - You cannot use the names of the original authors to promote your version without permission.

5. **No Warranty**: The software is provided "as is," meaning the authors are not responsible for any problems it might cause.

Key Features:

- **Permissive License**: It allows you to incorporate the software into proprietary (closed-source) projects without needing to share your changes.
- **Lightweight**: The license has very few restrictions, making it attractive for both open-source and commercial use.

In essence, the BSD License offers great flexibility, requiring only basic credit to the original creators while imposing no strict conditions on how the software is used or shared.

Public License (PL): Simplified

A **Public License (PL)** is a broad term for licenses that make software freely available for use, modification, and distribution. It ensures that:

1. Users have the freedom to access, use, and modify the software.
2. Distribution of the software may require compliance with specific conditions, such as attribution or sharing modifications.
3. Public licenses aim to promote openness and collaboration, ensuring the software remains widely accessible.

Examples include **Creative Commons Public Licenses** and **General Public Licenses (GPL)**.

General Public License (GPL): Simplified

The **GPL** is one of the most widely used open-source licenses. Here's what it means:

1. **Freedom to Use**: You can use the software for any purpose, including personal, educational, or commercial use.
2. **Freedom to Modify**: You're allowed to modify the software to fit your needs.
3. **Freedom to Share**: You can distribute the software, whether it's the original version or your modified version.
4. **Copyleft Requirement**:
    - If you share the software or your modified version, you **must** distribute it under the same GPL license.
    - This ensures that all versions of the software (including modifications) remain open-source and free for others.

5. **No Warranty**: Like most open-source licenses, the software comes "as is," with no guarantees about its functionality.

Key Differences Between PL and GPL:

- **PL**: Refers broadly to licenses that make software publicly accessible. It might not always enforce open-source principles like copyleft.
- **GPL**: A specific public license that ensures software and any derived works remain free and open-source, thanks to its copyleft clause.

In short, the GPL is a specific type of Public License with strict rules to keep software open-source forever, ensuring a cycle of collaboration and sharing.

Lesser General Public License (LGPL): Simplified

The **Lesser General Public License (LGPL)** is a variant of the **GPL**, designed to strike a balance between open-source and proprietary software. Here's what it means:

1. **Freedom to Use**:
    - You can use the software in your personal, educational, or commercial projects.
2. **Freedom to Modify**:
    - You can change the software to meet your needs.
3. **Freedom to Distribute**:
    - You can share the software, whether it's the original or a modified version.
4. **Key Difference from GPL (Copyleft Relaxed)**:
    - With LGPL, you can link the open-source software to **proprietary (closed-source)** applications.
    - Only the open-source part (the LGPL-covered library or code) must remain open-source.
    - You **don't need to open-source your entire proprietary program**, as long as it only links to the LGPL software and does not modify it.
5. **Copyleft Still Applies**:

- If you modify the LGPL-licensed library itself, you **must share those modifications** under the same LGPL license.

6. **No Warranty**:
   - Like other open-source licenses, it comes "as is," with no guarantees about its performance or suitability.

When to Use LGPL:

- **For Libraries**: LGPL is commonly used for libraries that developers want to share freely but still allow integration into proprietary software.
- **Greater Flexibility**: It provides more flexibility than GPL, making it suitable for mixed environments (open-source + proprietary).

Summary:

The LGPL is less strict than the GPL. It allows proprietary software to use open-source libraries without requiring the entire proprietary code to be open-sourced, but it ensures that the open-source parts remain freely available and modifiable.

BSD License in FOSS (Free and Open-Source Software): Simplified

The **BSD License** is a highly permissive open-source license that originated from the Berkeley Software Distribution (BSD) operating system. Here's how it fits into the world of Free and Open-Source Software (FOSS):

Key Features of the BSD License in FOSS:

1. **Permissive Nature**:
   - The BSD License imposes minimal restrictions on how the software can be used, modified, or redistributed.
   - It allows integration with both open-source and proprietary projects, making it very flexible.
2. **Encourages Collaboration**:
   - Developers can freely use and modify BSD-licensed software, which promotes sharing and collaboration in the open-source community.

3. **Attribution Requirement**:
    - Users must include the original copyright notice and license text when redistributing the software or derivative works.
    - This ensures proper credit is given to the original authors.
4. **No Copyleft**:
    - Unlike licenses like GPL, the BSD License does **not require derivative works to be open-sourced**.
    - This means that companies can incorporate BSD-licensed code into their proprietary software without any obligation to share the source code.
5. **No Warranty**:
    - The license explicitly states that the software is provided "as is," with no guarantees of functionality or liability for issues.

Why BSD Is Popular in FOSS:

- **Flexibility**: Developers and companies can use BSD-licensed software in almost any project, including proprietary ones.
- **Wide Adoption**: Many widely used open-source projects, like FreeBSD, NetBSD, and OpenBSD, use the BSD License.
- **Compatibility**: It works well with other licenses and encourages adoption without forcing strict conditions.