



Introduction to open-Source Software (OSS)

Concepts, strategies, and methodologies related to open-source software development

Week 02 – Lecture 02

- ## ○ History of OSS and Licenses



Jamil Hussain
jamil@sejong.ac.kr
010-6252-8807

Office: 421, Innovation Center
Sejong University

Recap

- Closed Source Software
 - Proprietary source code.
 - Professional support.
 - Licensing fees may apply.
 - Limited customization and transparency
- Open-Source Software (OSS)
 - Source code is publicly available.
 - Community-driven development.
 - Free to use and modify.
 - Transparency and customization.

Introduction to OSS

- Open-Source Software (OSS) is software whose source code is freely available to use, modify, and distribute.
- **Beyond Code:** OSS is not just about free access—it's about **community, transparency, collaboration, and philosophy.**
- **Comparison with Closed Source:**
 - **Closed Source:** proprietary, licensing fees, limited transparency, controlled innovation.
 - **OSS:** free use, modification, and distribution, rapid innovation, community-driven.

Philosophy of OSS

- Promotes **collaboration**: Anyone can contribute.
- Encourages **transparency**: Code is open to peer review.
- Supports **community-driven innovation**: Bugs and features evolve faster.



Key Characteristics

- Open access to source code.
- Community-based development model.
- Rapid innovation cycles.
- Lower entry barrier for students, researchers, and startups.



Discussion Point

Proprietary software companies argue that OSS reduces profits and discourages innovation. Do you agree or disagree?



Short Answer



Discussion Point

If you had to choose between using proprietary software that costs money and open-source software that is free but requires more learning effort, which would you choose? Why?



Short Answer



세종대학교
SEJONG UNIVERSITY

Today, Agenda

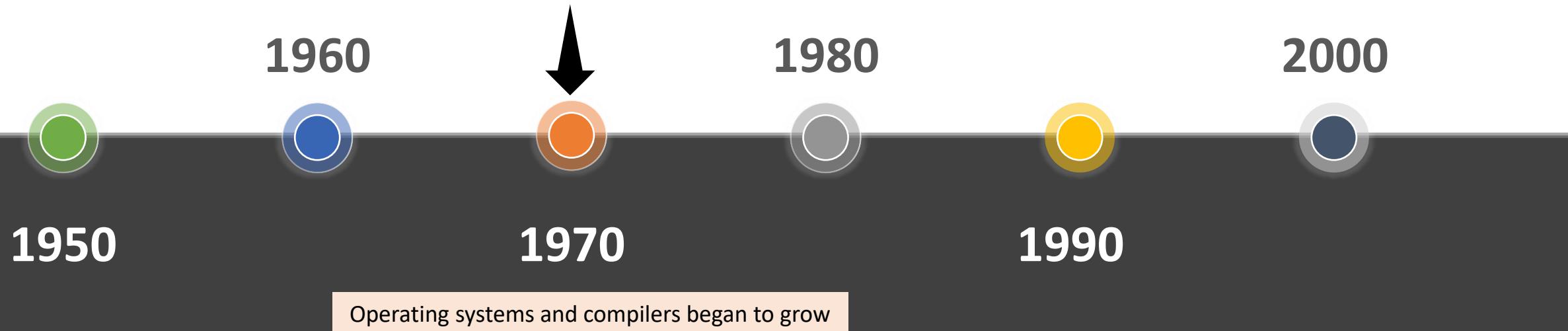


- History of Free and Open Source
- GNU and the Free Software Foundation (FSF)
- Open-Source Initiative
- Overview of Copyright and Licenses
- Understanding the legal implications of open-source
- Licenses in a Nutshell
- Software license categories

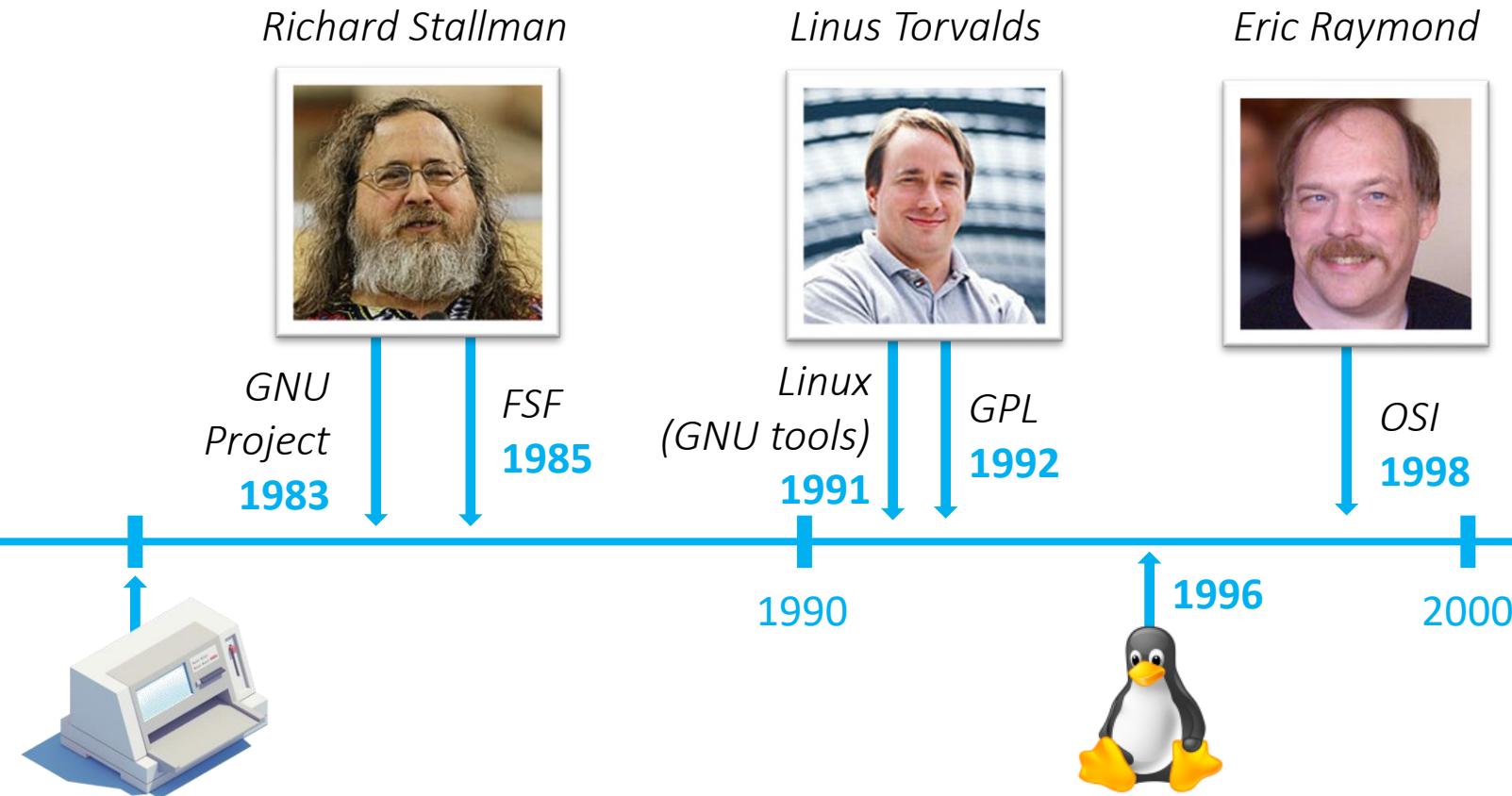
History of Open-Source (OSS) Software

The Origins of Open Source

- The story of open-source development began before the Free Software movement.
- In the 50s and 60s, research institutes primarily produced software.



Some Historical Facts



Xerox 9700 printer - AI lab at MIT
No source code for drivers

Linus visits aquarium, gets bitten by a penguin and chooses it as Linux mascot (Tux)



1950



1960



1970



1980



1990



2000

History of Open-Source (OSS) Software

1950s and Earlier

- Software sharing has a long history alongside the software itself.
- Early computer manufacturers primarily emphasized hardware innovation.
- Customers, often scientists or technicians, modified and extended the provided software.
- They shared their patches not only with the manufacturer but also with other machine owners.
- Manufacturers viewed software improvements as enhancing hardware appeal to potential customers.



세종대학교
SEJONG UNIVERSITY



History of Open-Source (OSS) Software



1950



1960



1970



1980



1990



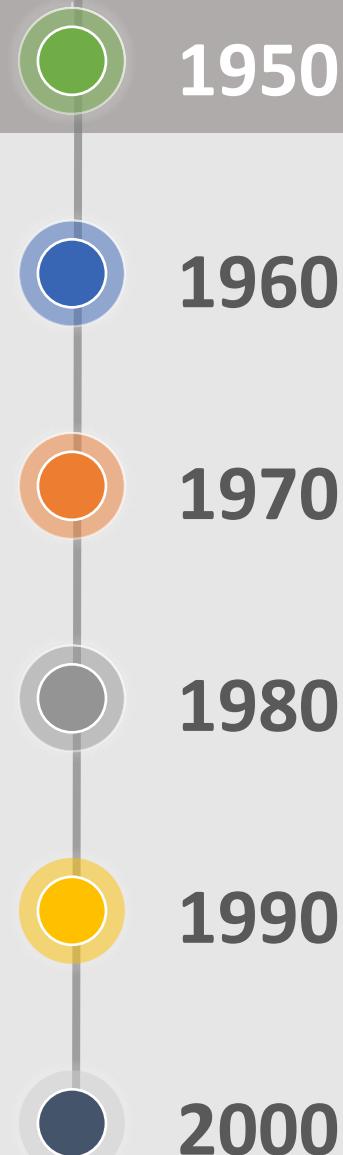
2000

1950s and Earlier (cont'd)

- The earliest commercial computers were shipped with their own *operating code* so that users could customize the code for their needs.
 - There were no manuals, no instructions, no help from vendors for the shipped software - **users just had to figure it all out on their own.**
 - Code-sharing was common: It was a **culture of sharing**.
- Typical hardware business model:
 - Buyers paid for hardware, the software did not matter much because it was not standardized - it worked only on the purchased hardware.
 - **Software had NO monetary value.**



History of Open-Source (OSS) Software



1950s and Earlier (Cont'd)

- Software arise from researcher, both academic and corporate
- Distributed openly and cooperatively
- Source always distributed, binaries less often
- Software is not seen as a separate commodity; instead bundled for free with hardware
- They didn't care about Licensing



History of Open-Source (OSS) Software



1950



1960



1970



1980



1990



2000

1950s and Earlier (Cont'd)

- The early period resembled today's free software culture but with two critical differences.
- Little hardware standardization;
 - diverse and incompatible computing architectures.
 - Expertise was architecture-specific, unlike today's language-oriented expertise.
 - Manufacturers promoted the spread of machine-specific code and knowledge.
- Lack of widespread Internet;
 - fewer legal restrictions but greater technical hurdles.
 - Inconvenient data transmission methods
 - limited sharing possibilities.
 - Overcoming barriers through local networks, mail, and university collaborations.



History of Open-Source (OSS) Software

1950

1960

1970

1980

1990

2000

1960s

- Major aspects of computer science and software rapidly developed both in academia (MIT, UC Berkeley) and industrial research labs (Bell Labs, Xerox)
- 1968: ARPANET emerges
- 1969: UNIX given birth at Bell Labs (AT&T); IBM forced to break software and hardware apart and sell/distribute separately



History of Open-Source (OSS) Software



1950



1960



1970



1980



1990



2000

1970s - Free Software's Origins

- During the 1970's software became a commodity: companies started selling it for profit.
- Code became proprietary and closed.
- The era of personal computers (PCs) arrived. Almost all are shipped with proprietary, closed code.
- 1976: emaces released
- 1978: First version of TeX released (still in widespread use, usually in the LaTeX version)

Discussion Point

Imagine you're a 1970s researcher. How would you feel if free software suddenly required fees?



Short Answer



세종대학교
SEJONG UNIVERSITY

History of Open-Source (OSS) Software

1950

1960

1970

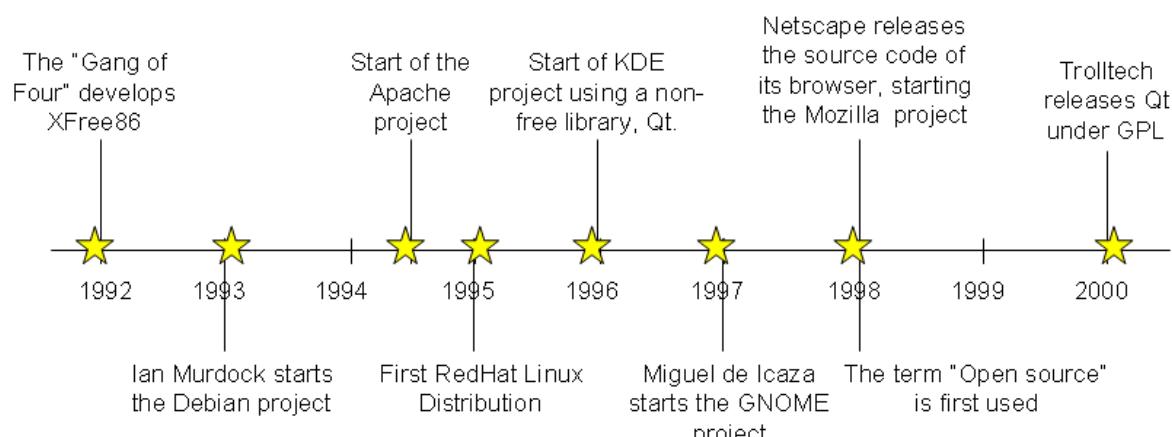
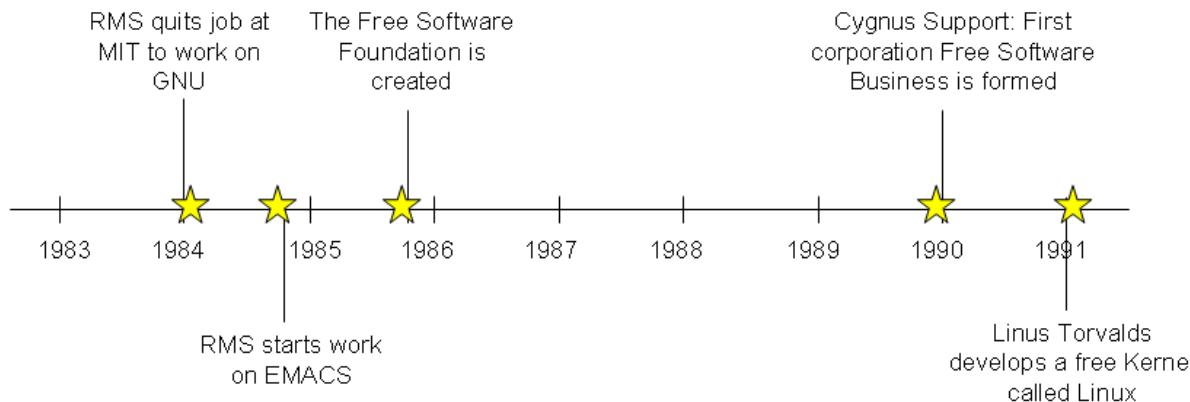
1980

1990

2000

1980s - Evolution of Open-Source development

- The evolution of open source development from the creation in 1986 of the Free Software Foundation by Richard Stallman
- After this foundation was established, several major open source projects were initiated as shown





History of Open-Source (OSS) Software



1950



1960



1970



1980



1990



2000

1980s

- **1980:** Usenet begins as the ancestor of user forums and the World Wide Web
- **1982:** GNU project announced
- **1984:** X Window System released (X11 protocol released in 1987)
- **1985:** Free Software Foundation (FSF) founded
- **1987:** **GCC** (now known as GNU C Compiler) and Perl released



History of Open-Source (OSS) Software



1950



1960



1970



1980



1990



2000

1990s

- **1991:** Linux was begun by Linus Torvalds
- **1992:** Python and 386BSD released; Samba developed
- **1993:** Debian (still the largest non-commercial Linux distribution), NetBSD, FreeBSD, and Wine released; Red Hat is founded
- **1994:** MySQL development begins (first release in 1995)

The Rise of Linux

- Linus Torvalds built Linux as a personal project → became the foundation of global infrastructure.
- **Why Linux succeeded:**
 - Open collaboration (thousands of contributors).
 - Flexible licensing (GPL ensured contributions remained open).
 - Industry adoption (IBM, Google, Red Hat).
- **Impact:**
 - Linux now powers >90% of cloud servers.
 - Android built on Linux kernel.

OSS thrives when community and industry collaborate.





History of Open-Source (OSS) Software

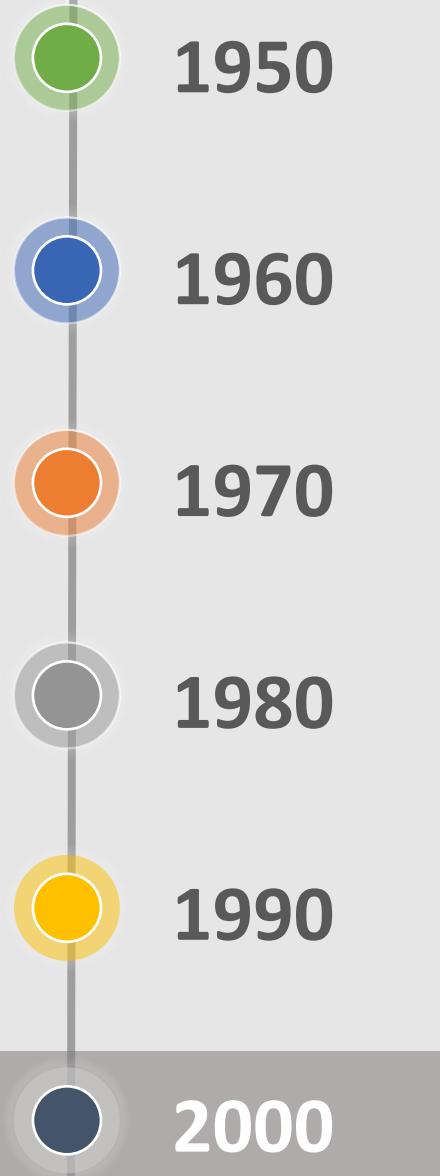


1990s (Cont.)

- **1995:** PHP, GIMP, and Ruby released
- **1996:** Apache web server and KDE released
- **1997:** GNOME released
- **1998:** Netscape open sources its browser (later becomes Firefox)
- **1999:** OpenOffice released (eventually forks into LibreOffice)



History of Open-Source (OSS) Software



2000s

- **2000:** LLVM compiler project begun
- **2002:** Blender released as an open-source project
- **2003:** Firefox released
- **2004:** Ubuntu released
- **2005:** Git released by Linus Torvalds
- **2007:** Android released, based on Linux kernel; first devices on the market in 2008
- **2008:** Chromium released by Google; the basis of Google Chrome

Discussion Point

Why is Linux everywhere (servers, IoT) but not dominant on personal desktops?



Short Answer



Discussion Point

What lessons can students/startups learn from Linux?



Short Answer



세종대학교
SEJONG UNIVERSITY

Clarifying "Open Source" vs. "Free Software"

- **Open Source vs. Free Software Debate**

- Richard Stallman and similar thinkers criticize "open source" for not emphasizing user freedom.
- Stallman's perspective: "Free software" = Freedom, not just price.
- Confusion: "Free software" is often misunderstood as no-cost software.

- **Introducing FLOSS: Free, Libre, Open-Source Software**

- Three terms explain the concept:
 - **Free software**: Focuses on liberty, not price; can be used, studied, and modified without restrictions.
 - **Libre Software**: Emphasizes freedom as a fundamental right, not zero-cost
 - **Open Source**: Emphasizes freely available source code.

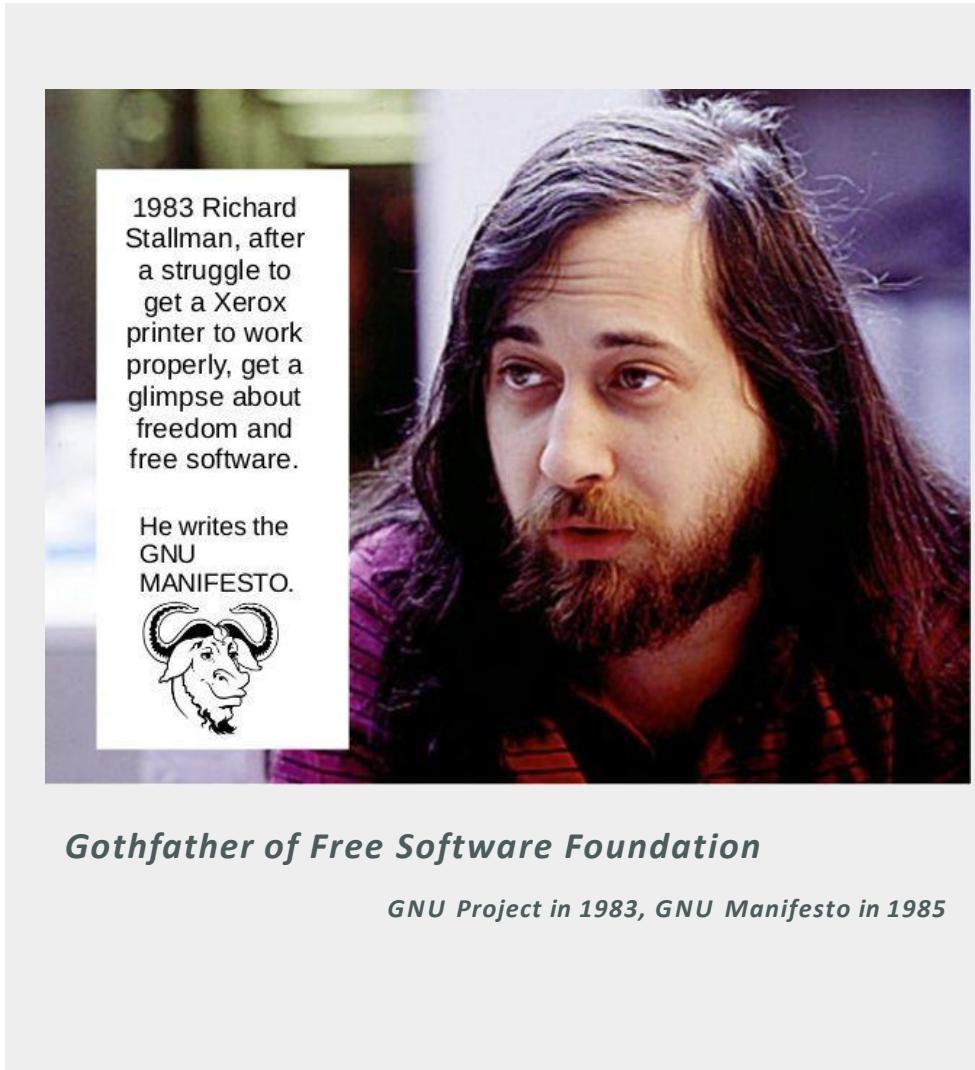
Advantages and disadvantages of open source

- **Advantages of Open Source Software**
 - **Cost-Free Access:** Open source software is freely available to everyone.
 - **Transparency:** The source code is open and can be viewed by anyone.
 - **Collaboration:** A global community of developers can contribute and improve the software.
 - **Customizability:** Users can modify the software to meet their specific needs.
 - **Security:** The collective effort often results in quicker identification and resolution of vulnerabilities.
- **Challenges of Nonprofit Community Development**
 - **Lack of Central Authority:** The absence of a single entity can lead to coordination challenges.
 - **Funding Issues:** Relying on donations and volunteers may limit resources for development.
 - **Quality Control:** Maintaining software quality can be a concern without strict oversight.
 - **Fragmentation:** Diverse contributions can result in varying versions and compatibility issues.
 - **Support and Documentation:** This may be less comprehensive compared to commercial software.
 - **Desktop Adoption Challenges:** Open source software has primarily focused on servers, leading to slower desktop adoption.

Open Source Trends and Perspectives

- **Increasing Adoption:**
 - Companies favor open source for development and testing.
 - Open source gaining ground in production.
- **LAMP Stack Dominance:**
 - LAMP (Linux, Apache, MySQL, PHP/Perl/Python) is widely used for web applications.
- **Sustained Projects:**
 - Many '80s-era open source projects thriving.
 - Example: Eclipse IDE.
- **Open Source-Centric Companies:**
 - Businesses offer integrated solutions based on open source.
 - Reflects open source community strength.

GNU and the Free Software Foundation (FSF)

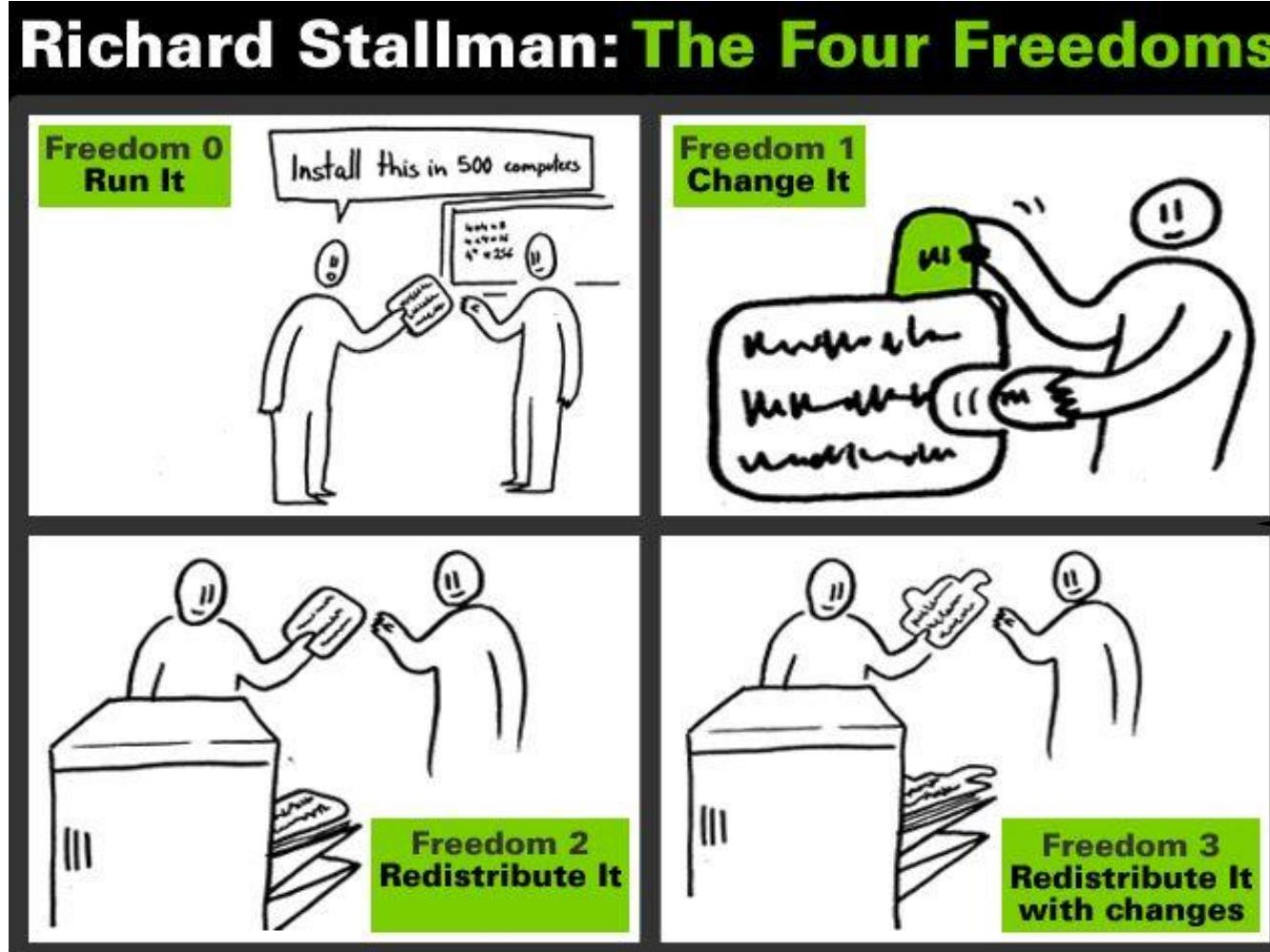


Starting this Thanksgiving

"I am going to write a complete Unix-compatible software system called GNU (for Gnu's Not Unix), and give it away free to everyone who can use it"

He defined the Four Freedoms as the core tenets of the Free Software Movement (purposely starting with number zero)

Free Software Foundation - Freedoms explained



Any software that does not guarantee these freedoms to its users cannot be considered “free” because it limits the users’ rights in some way.

Software Licenses and the GPL

- Richard and the FSF invented a class of software licenses that turned copyright on its head, and they named them copyleft licenses.
- Copyleft software licenses ensure that software can never violate the Four Freedoms.
- The invention of these licenses is probably his most significant contribution to software; it really smooth the way for the open-source movement that followed.

The Open Source Movement's Origins

- On January 22, 1998, Netscape released its browser Netscape Navigator together with its source code, drawing attention of businesses.[2]
- Although interested in the concept, businesses were not keen on the politics of the FSF and the Free Software movement.
- In February 1998, Christine Peterson suggested "open source" to replace "free" to make the idea more appealing to businesses.
- Later that month the Open Source Initiative (OSI) was created by a group of advocates (Bruce Perens and Eric S. Raymond), with the mission of explaining and protecting the "open source label".

The Open Source Definition

- OSI created a definition of open source software that was more detailed than Stallman's Free Software Definition, and called it the Open Source Definition (OSD).
- The OSD defines valid software licenses.
- The software license is what determines whether software is open source!



The two families

Free Software Foundation

"In 1983, Richard Stallman published the GNU Manifesto and launched the GNU Project to write a complete operating system free from constraints on use of its source code."

Open Source Initiative

"The organization was founded in February 1998 by Bruce Perens and Eric S. Raymond, part of a group inspired by the Netscape Communications Corporation..."



Free Software vs Open Source

Free Software Foundation

“In 1983, Richard Stallman published the GNU Manifesto and launched the GNU Project to write a complete operating system free from constraints on use of its source code.”



Open Source Initiative

“The organization was founded in February 1998 by Bruce Perens and Eric S. Raymond, part of a group inspired by the Netscape Communications Corporation...”



The two families and their own definition of open

Free Software Foundation

1. *The freedom to run the program as you wish, for any purpose .*
2. *The freedom to study how the program works, and change it so it does your computing as you wish.*
3. *The freedom to redistribute copies so you can help your neighbor.*
4. *The freedom to distribute copies of your modified versions to others.*

By doing this you can give the whole community a chance to benefit from your changes. Access to the source code is a precondition for this.



Open Source Initiative

1. *Free Redistribution*
2. *Source Code Availability*
3. *Derived Works*
4. *Integrity of Author's Source Code*
5. *No Discrimination Against Persons or Groups*
6. *No Discrimination against Fields of Endeavor*
7. *Same Distribution License*
8. *License not Specific to a Product*
9. *License Must Not Restrict Other Software*
10. *License Must be Technology Neutral*



Free Software vs Open Source

Free Software Foundation

Free Software Foundation (FSF)

Founded in 1985 by
Richard Stallman



Non-profit organization

Defend the rights of all software users

This is a social movement. They are "software activists".

Free software is software that ensures the user's freedoms

Run, Study, Share, Modify

Open Source Initiative

Open Source Initiative (OSI)

Founded in 1998 by
Eric S. Raymond and Bruce Perens



Non-profit organization

Educes about and defends open source

Promotes this model of collaboration for companies.

Open Source is what complies with the OSD

Similar benefits but less restrictions



Free Software vs Open Source

Free Software Foundation

Free Software Foundation (FSF)

Founder: *Richard Stallman*



Created GNU Project

Activist, software hippie

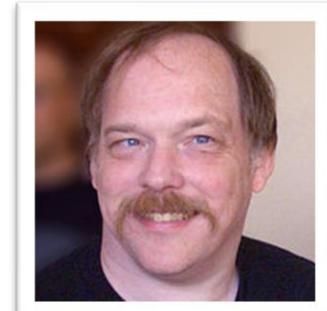
"A proprietary program puts its developer or owner in a position of power over its users. This power is in itself an injustice."

FSF website

Open Source Initiative

Open Source Initiative (OSI)

Co-founder: *Eric S. Raymond*



Released Netscape's source code

He also defends proprietary software

"I think that if a programmer wants to write a program and sell it, it's neither my business nor anyone else's but his customer's..."

Essay in 2008

10 criteria for OSS



1. Free redistribution

- No restriction on redistribution (free or paid) of the software
- The software can be redistributed alone or as a component of an aggregate software distribution

2. Source Code

- The software must include the source code, or it must be easily obtainable through well-published means
- The source code should not be deliberately obfuscated, and intermediate forms (preprocessor/translator output) are not allowed.

3. Derived works

- The license must allow modification and derived works
- Derived works must be allowed to be distributed under the same terms as the original



10 criteria for OSS



4. Integrity of the author's source code

- The license may only restrict source-code from being distributed in modified form if it allows patch-files that can modify it at build time
- The license might require derived works to use a different name and/or version number

5. No discrimination against persons or groups

6. No discrimination against fields/domain

7. Distribution of license

- License for software must also apply to those it is redistributed to

10 criteria for OSS



8. License for software must not depend on it being part of a software distribution
 - The same license must apply if the software is extracted from a distribution and distributed separately
9. License must not restrict other software
 - The license must not place restrictions on other software that is distributed alongside the licensed software
10. License must be technology-neutral
 - No provision of the license may be predicated on any individual technology or style of interface

10 criteria for OSS

- If the license does not meet the definition, it is not open source!
- If there is No license, it is not open source!
 - (It is code available, not open source)



Free vs Open Source

- There was and still is much controversy about **Free versus Open-Source software.**
- The difference is mostly philosophical:
 - The Four Freedoms and Free software express an ideology and a moral point.
 - The OSI OSD expresses a practical system for supporting the expansion and growth of open source, acceptable to business interests.
- Many people use the term **Free and Open-Source Software (FOSS)** to avoid the contention. We do the same.



Discussion Point

Do you think ‘free software’ and ‘open-source software’ mean the same thing?
Why might the terminology matter?



Short Answer



Discussion Point

Proprietary companies argue OSS discourages innovation. Do you agree?



Short Answer



Discussion Point

FSF emphasizes *freedom*, OSI emphasizes *practicality*. Which approach is more appealing?



Short Answer

Discussion Point

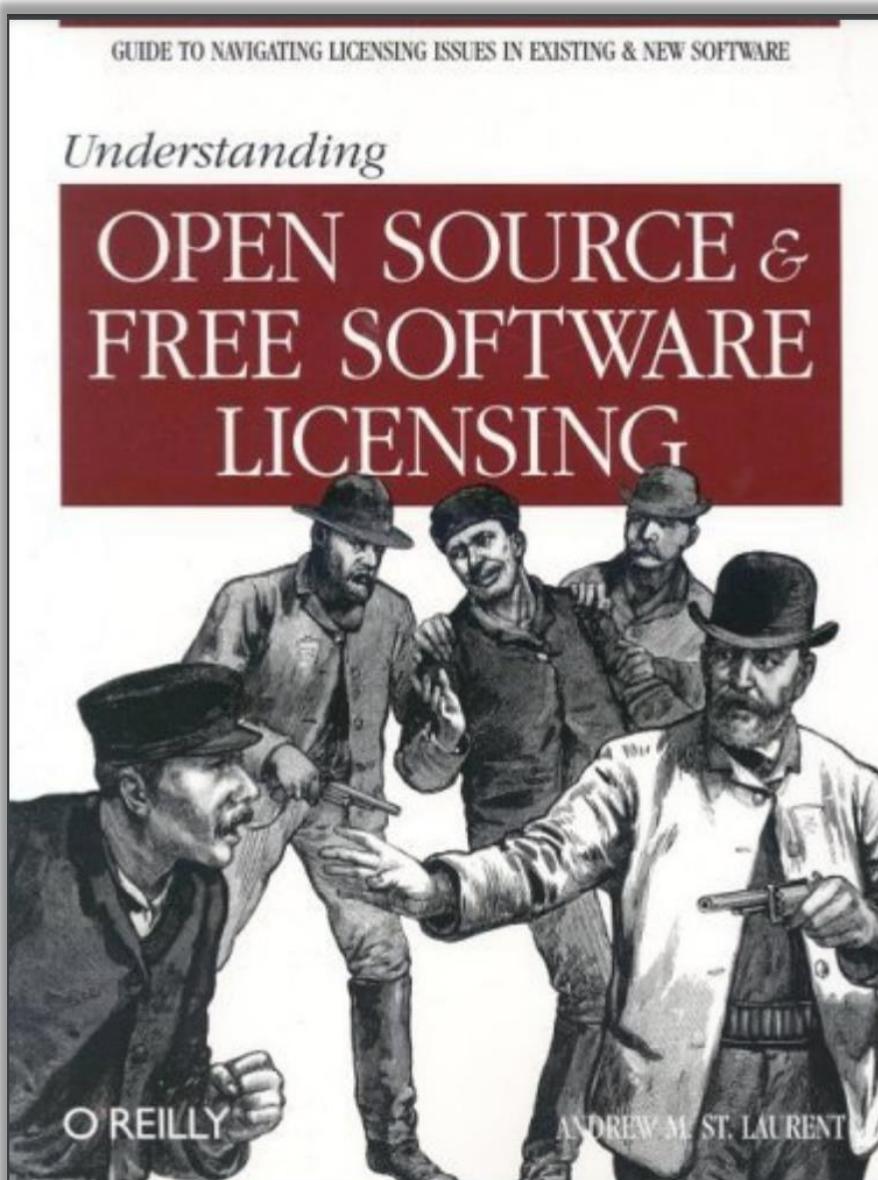
Why did Microsoft (once calling Linux a threat) later become a top OSS contributor?



Short Answer



Additional recommend book



Understanding Open Source
and Free Software Licensing

Andrew M. St. Laurent -

Intellectual property, copyright and licensing

- Intellectual Property (IP) is legally protected rights over new ideas or creations.
- Common types of intellectual property include
 - Copyrights,
 - Trademarks,
 - Patents
 - Industrial design rights and trade secrets.

Overview of Copyright and Licenses

- Whether software is FOSS depends on its license.
- A license can only be applied by the copyright holder.
- So ...
 - **What is copyright?**
 - **What is a license?**
- The next few slides answer these questions.

Copyright in a Nutshell

- Copyright gives the **creator** of the **original work exclusive rights** in terms of **usage, distribution**, and **customization** of the work.
- Roughly, copyright is the **legal right granted** to the creator of a creative work to the exclusive **publication, production, sale, or distribution** of that work.
- By default, the creator has **All Rights Reserved**, meaning no one has the right to do anything with the work except to use it as is **allowed**.
- When all rights are reserved by the **copyright owner**, others cannot **copy, reuse, redistribute**, or do **anything** other than **use the work**

Copyright definition

Legal right that grants the creator of an **original work**



literary works



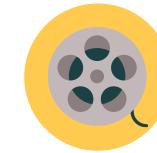
music



photography



paintings



motion pictures



software

Copyright definition



reproduce



perform



modify



distribute

The copyright owner controls what can be done.

Rights of copyright holders

- **Copyright owners can:**
 - Reproduce the work
 - Adapt or derive other works from the original work
 - Distribute copies of the work
 - Display the work publicly
 - Perform the work publicly



Rights of copyright holders (cont'd)

- Some of the privileges copyright provides to the author of the **software** include:
 - The right to **produce** and **sell copies** of the work
 - The right to **create derivative works**
 - The right to **sell**, **transfer**, or **reassign** any of the rights granted by copyright to others

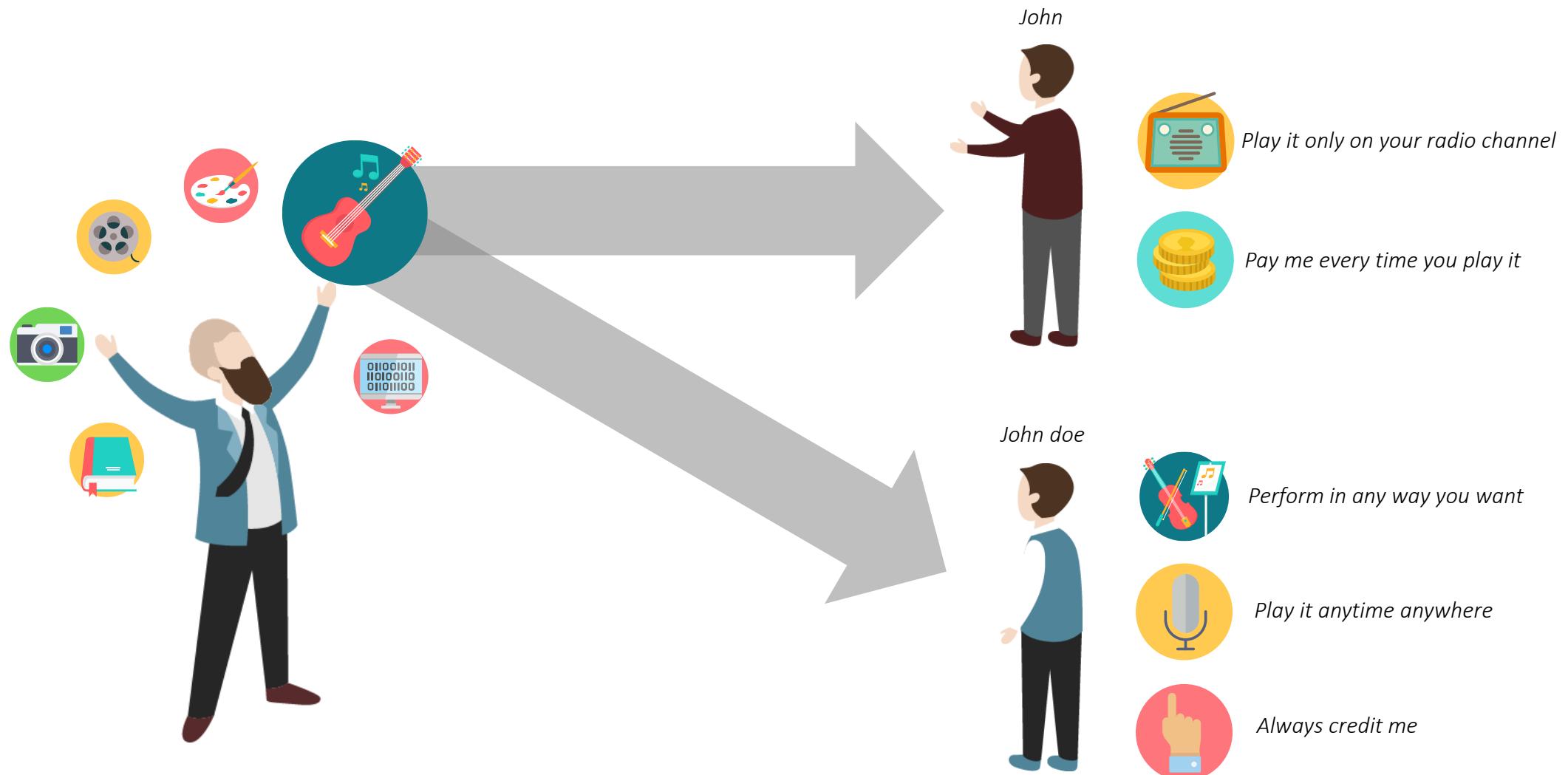


Copyright limitations

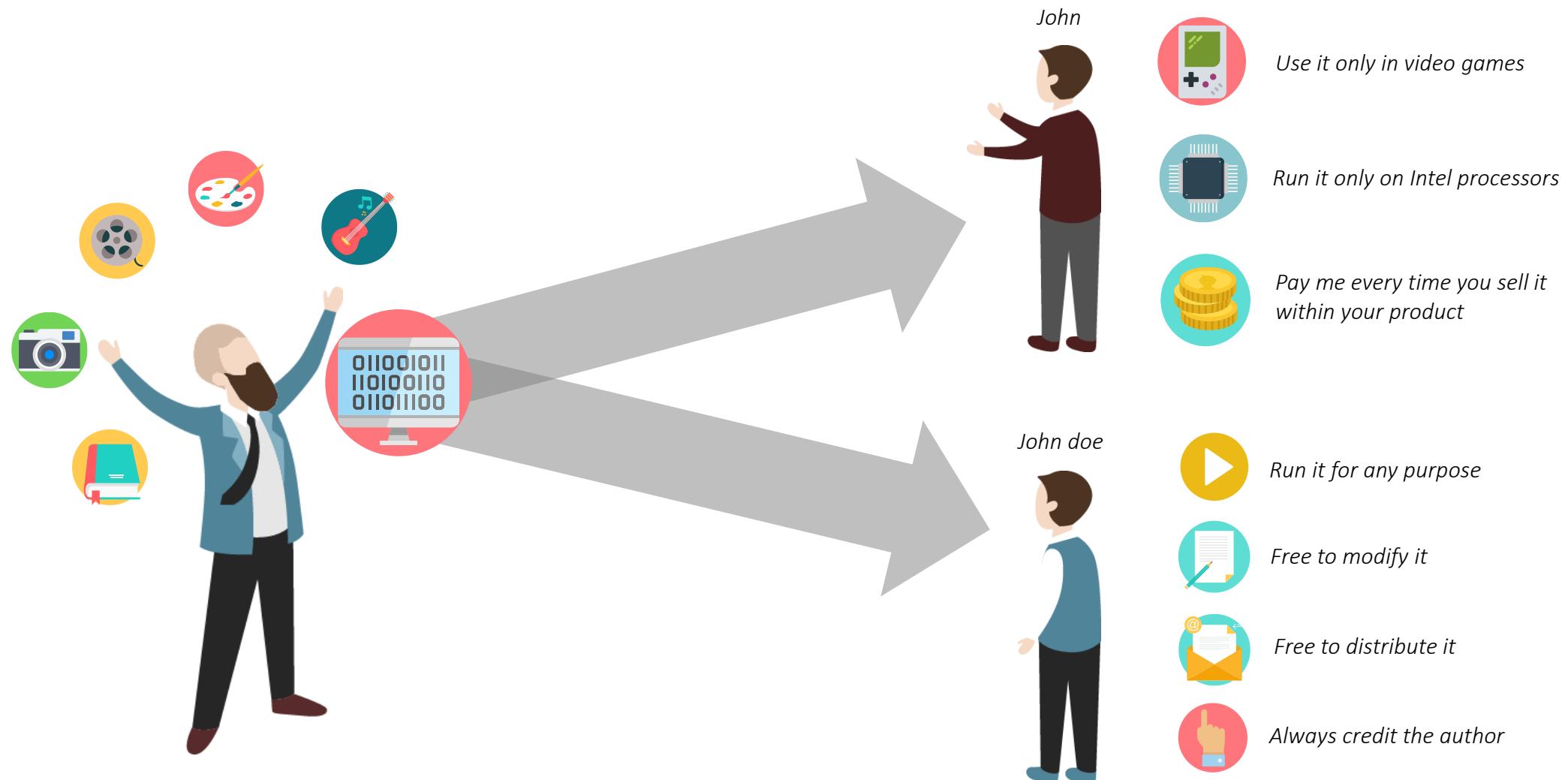
- Copyright is time-bound - normally a number of **years (70)** after the death of the author
- Two general limitations:
 - **Fair use** - limited use of copyrighted work is allowed, for commenting, news reporting, research, teaching etc.
 - **First sale** - copyright holders who have sold copies of a work cannot interfere with subsequent sales of those copies



Exclusive rights exercise – example 01 (Music)



Exclusive rights exercise – example 02 (Software)



Derivative work

Expressive creation that includes major copyright-protected elements of an original previously created first work (underlying work)

Leonardo da Vinci

1519

Mona Lisa



Marcel Duchamp

1919

Additions to the Mona Lisa:

Moustache

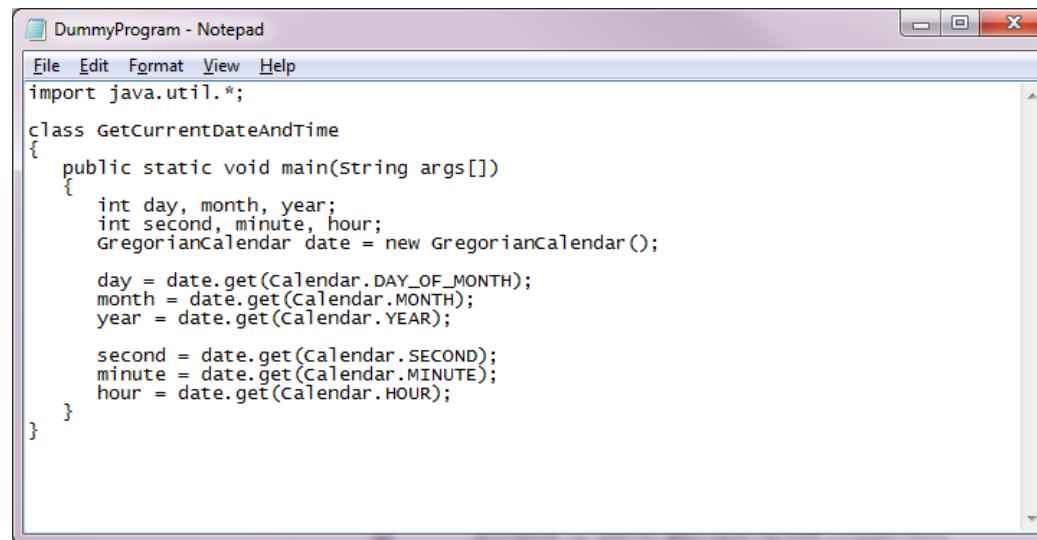
L.H.O.O.Q.



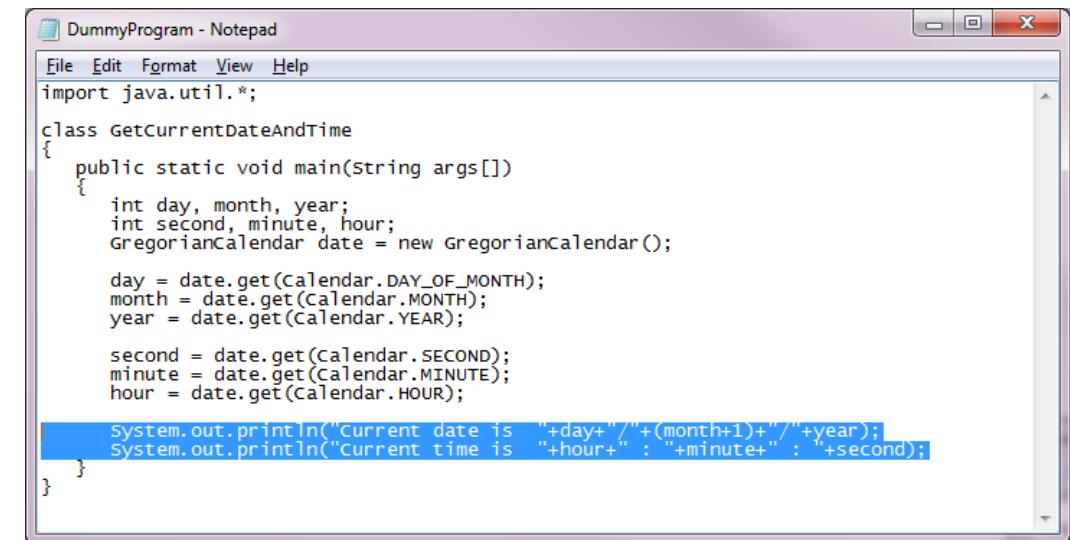
*Copyright protection = Authors life + 70 years (depends on)

Derivative work

Expressive creation that includes major copyright-protected elements of an original previously created first work (underlying work)



```
import java.util.*;  
  
class GetCurrentDateAndTime  
{  
    public static void main(String args[])  
    {  
        int day, month, year;  
        int second, minute, hour;  
        GregorianCalendar date = new GregorianCalendar();  
  
        day = date.get(Calendar.DAY_OF_MONTH);  
        month = date.get(Calendar.MONTH);  
        year = date.get(Calendar.YEAR);  
  
        second = date.get(Calendar.SECOND);  
        minute = date.get(Calendar.MINUTE);  
        hour = date.get(Calendar.HOUR);  
    }  
}
```



```
import java.util.*;  
  
class GetCurrentDateAndTime  
{  
    public static void main(String args[])  
    {  
        int day, month, year;  
        int second, minute, hour;  
        GregorianCalendar date = new GregorianCalendar();  
  
        day = date.get(Calendar.DAY_OF_MONTH);  
        month = date.get(Calendar.MONTH);  
        year = date.get(Calendar.YEAR);  
  
        second = date.get(Calendar.SECOND);  
        minute = date.get(Calendar.MINUTE);  
        hour = date.get(Calendar.HOUR);  
  
        System.out.println("Current date is "+day+"/"+(month+1)+"/"+year);  
        System.out.println("Current time is "+hour+":"+minute+":"+second);  
    }  
}
```

Q3: Can you publish the copyright content under your name?



- A** Yes
- B** No



Multiple Choice



세종대학교
SEJONG UNIVERSITY

Copyleft

A form of licensing that was initiated by the Free Software movement.

Copyright

Legal right that grants the creator of an original work exclusive rights to its use and distribution



Copyleft

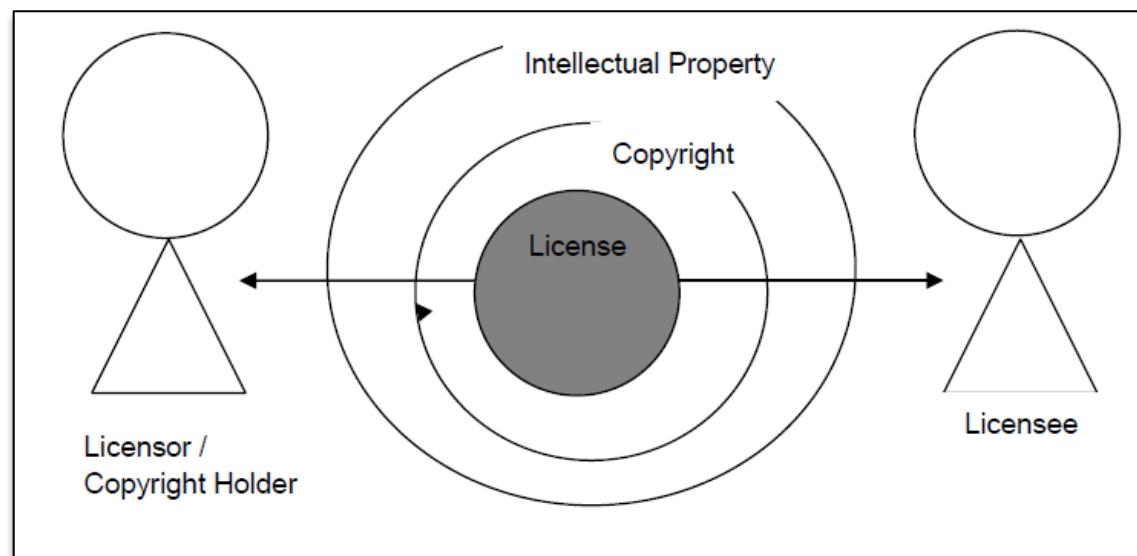
Offering people the right to freely distribute copies and modified versions of a work

Exists within the legal structure of copyright



Copyleft (cont'd)

- The transfer of rights by the author partly or wholly on his own terms is what we refer to as **licensing**.
- The term **license** means **permission**. The copyright holder, or licensor, grants another person, known as the licensee, specific permissions to use the work.



Relationship between licensor, licensee, and license

Understanding the legal implications of open source

Sharing your creative work with the world can be an exciting and rewarding experience. It can also mean a bunch of **legal things** you didn't know you had to worry about. Thankfully, you don't have to **start from scratch**.

Open-source projects are made available and contributed to under licenses that include terms that, for the protection of contributors, make clear that the projects are offered “**as-is**”, without **warranty**, and **deny responsibility for damages** resulting **from using the projects**.

Why do people care so much about the legal side of open source?

When you make a creative work (such as writing, graphics, or code), that work is under exclusive copyright by default. That is, the law assumes that as the author of your work, you have a say in what others can do with it.



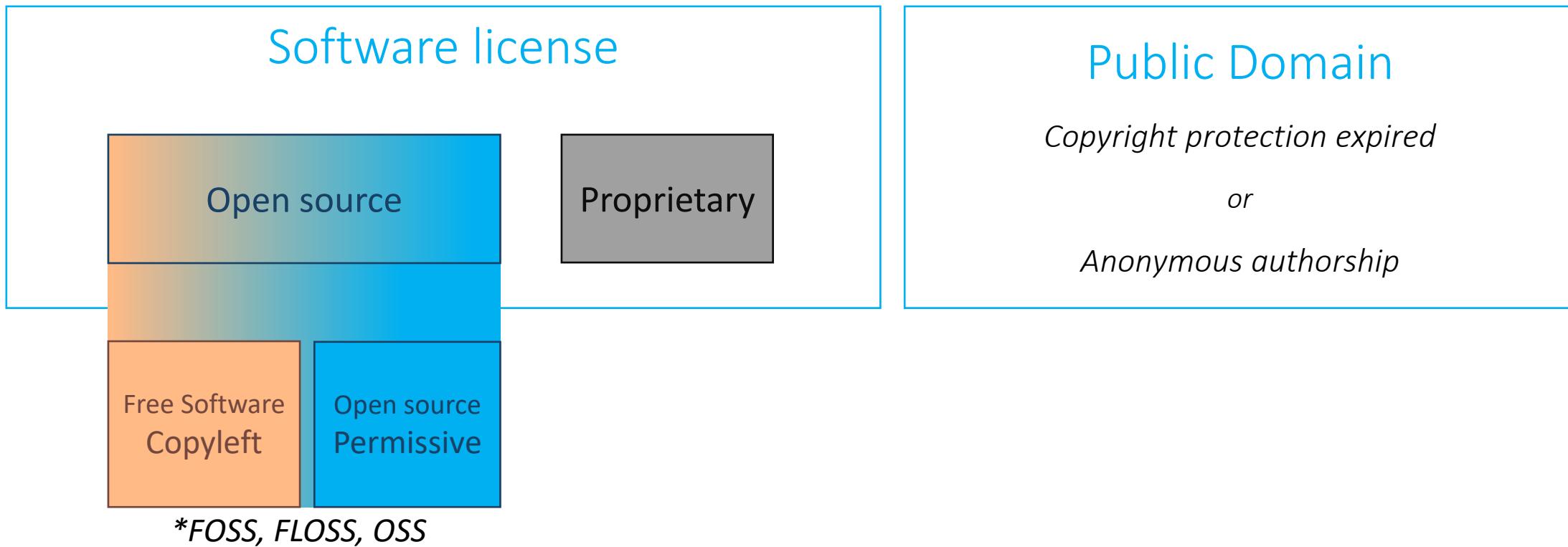
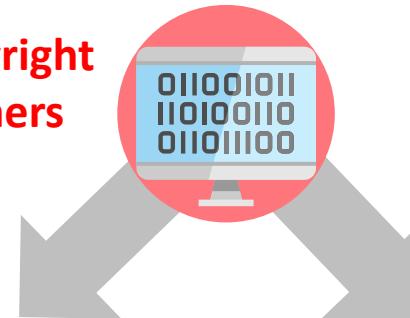
Licenses in a Nutshell

- If you want to give others the **permission** to do certain things with your work, you do that by issuing a license.
- A license is a legal document that gives people **permission** to do **specific things** with the **copyrighted work**.



Software license categories

- Software is created by an author and is subject to **copyright**
- A license **determines** how software can be **used by others**



Copyleft vs. Permissive

Both copyleft and permissive licenses require that anyone who uses works licensed under them must have permission to view, modify, and share the source.

Copyleft

- Restrictive (copyleft) licenses require source code to be distributed along with binary code
 - aim to keep software free in the future

Permissive

- Permissive licenses allow distribution of source code, but only require attribution.

This is a simplification. It is more complicated than this; some licenses are more permissive than others and there are other complications.

Copyleft vs. Permissive

Free Software - Copyleft

Governed by the
Free Software Foundation (FSF)



They keep a list of accepted licenses

GPL, LGPL, AGPL, MPL, EPL,...

They are referred to as:

Copyleft, Restrictive, protective, reciprocal...

Open Source - Permissive

Governed by the
Open Source Initiative (OSI)



They keep a list of accepted licenses

BSD, MIT, Apache License, ...

They are referred to as:

Permissive, Non-copyleft



License and Project Examples

Free Software - Copyleft

License Examples:

GPLv2, GPLv3 - *GNU General Public License*

LGPLv2.1, LGPLv3 - *GNU Lesser General Public License*

MPL - *Mozilla Public License*

EPL - *Eclipse Public License*

Project Examples:



Linux (GPL v2) – *Most deployed OS*

GCC (GPL v3) – *Most popular compiler*



Firefox (MPL) – *Top 5 most used browser*

Eclipse (EPL) – *Most used IDE*

Open Source - Permissive

License Examples:

BSD License - *Berkeley Software Distribution*

MIT License - *Massachusetts Institute of Technology*

Apache License

Project Examples:



Android (Apache) – *Most used mobile OS*

Apache (Apache) – *Most used web server*

PHP (PHP) – *Popular server scripting language*

Python (Python) – *Popular high-level language*



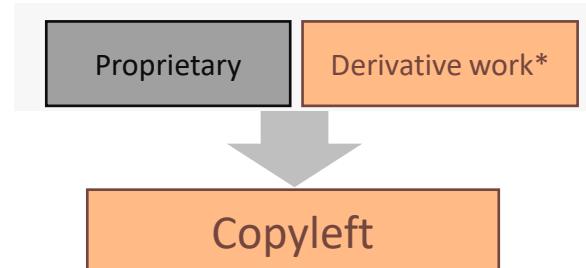
Usage Examples

Free Software - Copyleft

- Derivative works (modifications) :



- Combined works (linking) :



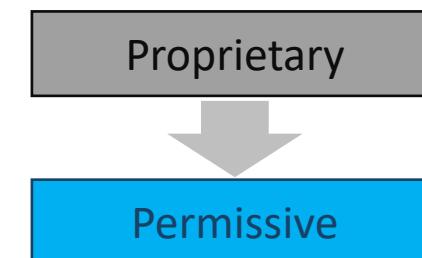
*License propagation, Copyleft effect

Open Source - Permissive

- Derivative works (modifications) :



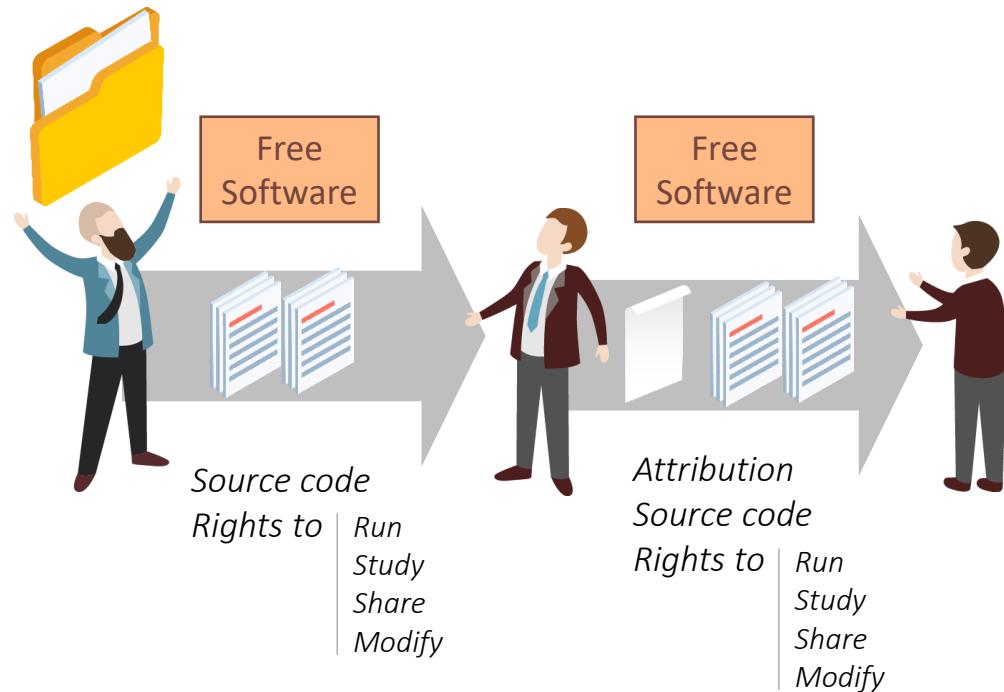
- Combined works (linking) :



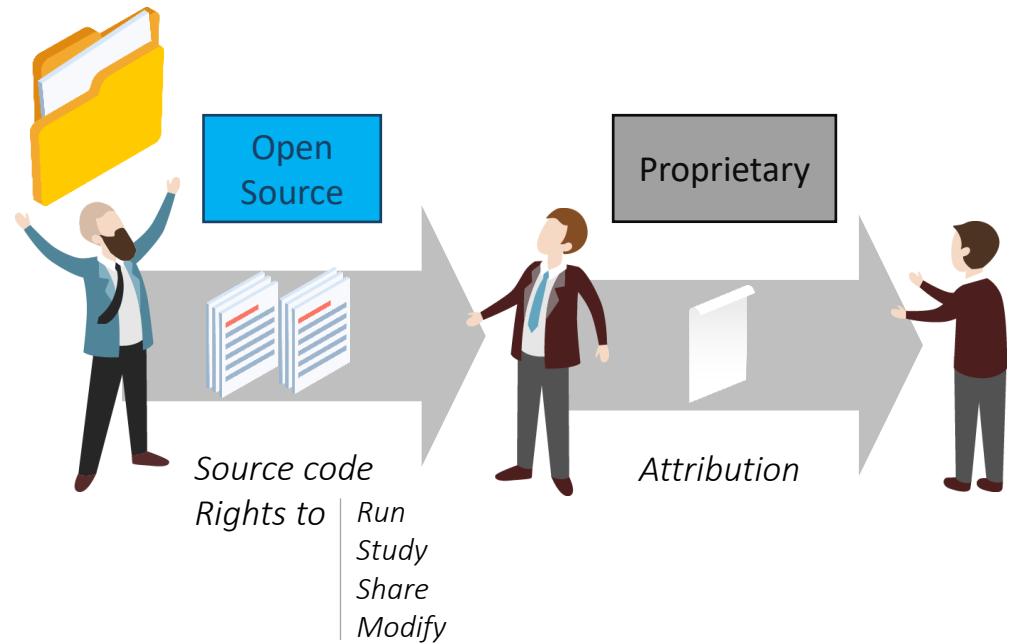
No license propagation

Distribution Examples

Copyleft



Permissive



Business friendly



세종대학교
SEJONG UNIVERSITY

Community and Software Licenses

- When a software work has been **written by many people**, each person **retains copyright** over the portions that they contributed to the whole work.
- It would be **hard for decisions to be made** about what can be done with that software if **each copyright owner** had a different set of **rights**.
- Putting a project under a **free and open source software license** can simplify who can do what.

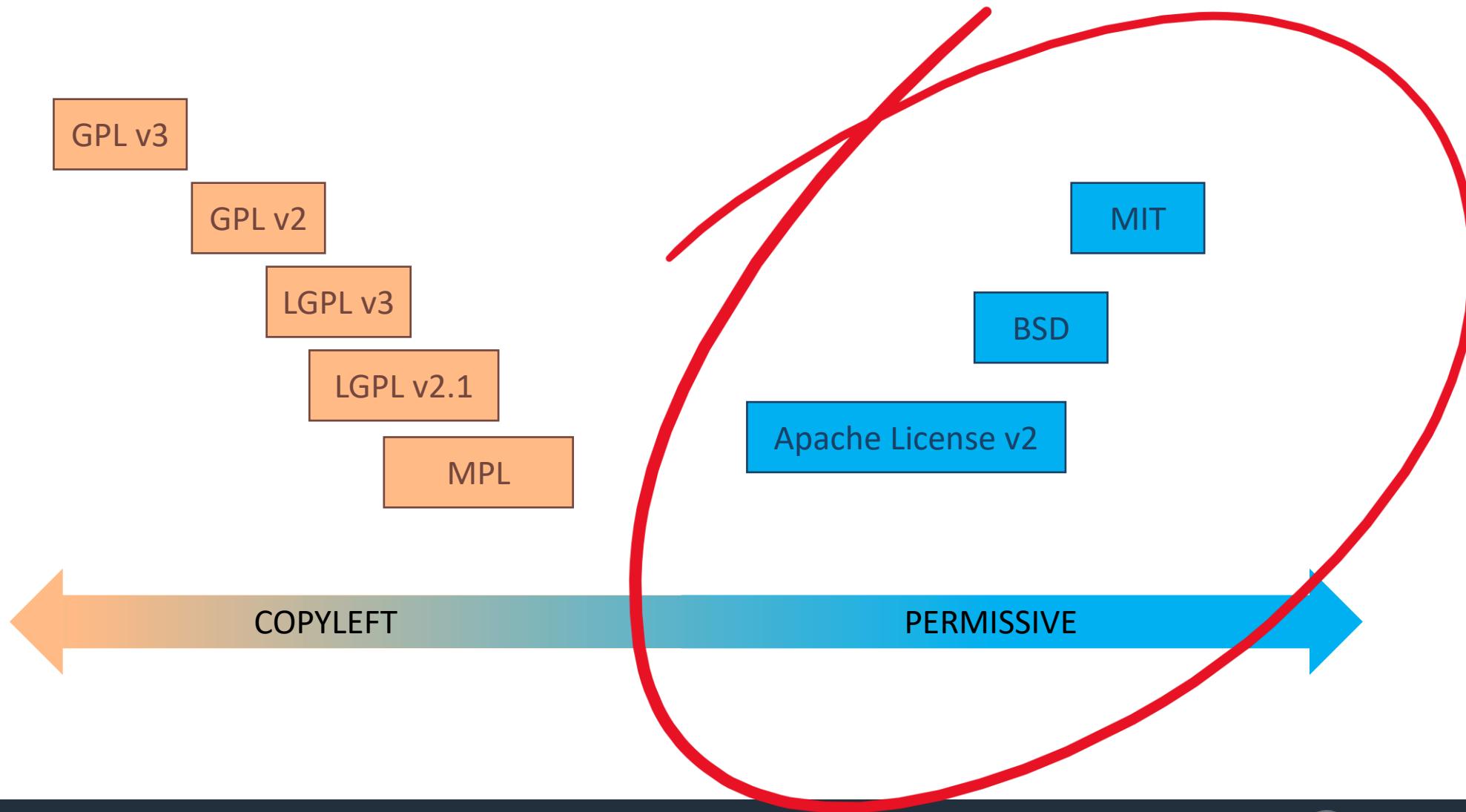
Community and Software Licenses

- For example, most FOSS licenses require contributions to a project to be contributed and released under the same license as the original work.
 - If you contribute to a project with an OSI approved license, you still keep the copyright over your work, but your contribution is **covered by the same license** as the project and is open source.
 - This means your code is shareable, not secret.
 - It can be part of **your portfolio**.
 - You can show it to a **prospective employer**.

FOSS versus Not FOSS

- If you write code **for an employer**, whether as employee, intern, or even sometimes as an independent contractor, it is work for hire, unless you have a contract that states otherwise.
- **Work for hire** is any created work that can be copyrighted like songs, stories, essays, sculptures, paintings, graphic designs, or computer programs.
- Beware:
 - You do not **own the copyright**.
 - You cannot share your code with anyone or **put it in your portfolio**.
 - You cannot show it to a **prospective employer**.

License Spectrum



Common Open-Source Licenses

- Some common permissive licenses:
 - The Apache License
 - The BSD (Berkeley Software Distribution) License
 - The MIT License
- Some common copyleft licenses:
 - The GNU Public License
 - The GNU Affero Public License
 - GNU Lesser General Public License
- Hybrids:
 - The Mozilla Public License (weakly copyleft)

Common permissive licenses

- The MIT and BSD Licenses were two of the **earliest open source licenses**.
- Because these licenses are relatively **straightforward** and illustrate some of the basic principles of open source licensing, they are described here first.
- The MIT (or X), BSD, and Apache Licenses are classic open source licensing software licenses and are used in **many open source projects**.



Common permissive licenses (Cont.)

- It allows that **code to be used in proprietary software** and does not require that open-source versions of the code be distributed.
- Code created under these licenses, or derived from such code, may go “**closed**” and developments can be made under that **proprietary license**, which are **lost to the open source community**.
- For the same reason, however, these licenses are very flexible and compatible with almost **every form of open-source license**.



The MIT (or X) License



Copyright (c) <year> <copyright holders>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT (or X) License



Copyright (c) <year> <copyright holders>

Distributed under the terms of the MIT License

The first of these two conditions is almost universal in open source licensing and serves the straightforward and necessary purpose of alerting future users of the work of the restrictions on it. (Copyright laws used to require that copyrighted works carry an explicit notice in published forms to receive copyright protection.) The second of the two conditions provides the warranty disclaimer

MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The BSD License



The **BSD** License, which is only slightly more restrictive than the **MIT** License, exists in a number of substantially similar forms.

Copyright (c) <YEAR>, <OWNER>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

All advertising materials mentioning features or use of this software must display the following acknowledgement: This product includes software developed by the University of California, Lawrence Berkeley Laboratory.

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of the <ORGANIZATION> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The BSD License



The BSD License is only slightly restrictive than the MIT License, exists in a number of substantially similar forms.

Copyright (c) <YEAR>, <OWNER>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

All advertising materials mentioning features or use of this software must display

This provision specifically claims any express warranty, as well as the warranties of merchantability and fitness for a particular purpose, and it operates much like the disclaimer in the MIT License

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The Apache License, v1.1 and v2.0



- The Apache License is very similar to the BSD and MIT Licenses already described.
- The Apache License, Version 1.1, follows substantially the same pattern as the BSD License in premising distribution and modification upon compliance with relatively unrestrictive terms.
- Version 2.0, a top-down rewriting of the license, was first published in 2004.

The Apache License v1.1



It is slightly longer than the licenses discussed earlier, but it operates in much the same way

Copyright (c) 2000 The Apache Software Foundation.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>)."
4. The names "Apache" and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org.
5. Products derived from this software may not be called "Apache" nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The Apache License v2.0



- Released in January, 2004, the Apache License, v2.0, is a thorough revision of the Apache License.
- While the Apache License, v1.1, operates much like a BSD or MIT License with a non-endorsement provision barring the use of the Apache name without permission.
- v2.0 is a fuller and more complex license, laying out in more specific

The Apache License v2.0



Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation

of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

<https://www.apache.org/licenses/LICENSE-2.0>

How to identify the applicable license?

Three common places to find the applicable license of a project

- 1 Inside the source code
- 2 In the root directory of a project
- 3 In the project's website

Let's look at some examples



EXAMPLE_PROJECT

project_sample_mix

android-bluetooth

BluetoothActivityEnergyInfo.java

BluetoothAudioConfig.java

BluetoothAvrcp.java

BluetoothAvrcpPlayerSettings.java

BluetoothDevicePicker.java

BluetoothGattCallback.java

BluetoothGattCallbackWrapper.java

BluetoothGattDescriptor.java

BluetoothGattIncludedService.java

BluetoothGattServerCallback.java

duplicated

openfastpath

prop_files_with_snippets

whitelisting

Local file



BluetoothActivityEnergyInfo.java

```
1  /*
2  * Copyright (C) 2014 The Android Open Source Project
3  *
4  * Licensed under the Apache License, Version 2.0 (the "License");
5  * you may not use this file except in compliance with the License.
6  * You may obtain a copy of the License at
7  *
8  *     http://www.apache.org/licenses/LICENSE-2.0
9  *
10 * Unless required by applicable law or agreed to in writing, software
11 * distributed under the License is distributed on an "AS IS" BASIS,
12 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
13 * See the License for the specific language governing permissions and
14 * limitations under the License.
15 */
16
17 package android.bluetooth;
18
19 import android.os.Parcel;
20 import android.os.Parcelable;
21
22 /**
23 * Record of energy and activity information from controller and
24 * underlying bt stack state. Timestamp the record with system
25 * time
26 * @hide
27 */
28 public final class BluetoothActivityEnergyInfo implements Parcelable {
29     private final long mTimestamp;
30     private final int mBluetoothStackState;
31     private final long mControllerTxTimeMs;
32     private final long mControllerRxTimeMs;
33     private final long mControllerIdleTimeMs;
```



Quick View



- > EXAMPLE_PROJECT
- ▽ project_sample_mix
 - > android-bluetooth
 - > duplicated
 - ▽ openfastpath
 - Doxyfile
 - INSTALL
 - LICENSE
 - Makefile.am
 - README
 - TODO
 - configure.ac
 - > include
 - > scripts
 - > src
- > prop_files_with_snippets
- > whitelisting

Local file



LICENSE

```
1 Copyright (c) 2014, OpenFP
2 All rights reserved.
3
4 Redistribution and use in source and binary forms, with or without
5 modification, are permitted provided that the following conditions are met:
6
7 * Redistributions of source code must retain the above copyright notice, this
8   list of conditions and the following disclaimer.
9
10 * Redistributions in binary form must reproduce the above copyright notice,
11   this list of conditions and the following disclaimer in the documentation
12   and/or other materials provided with the distribution.
13
14 * Neither the name of the {organization} nor the names of its
15   contributors may be used to endorse or promote products derived from
16   this software without specific prior written permission.
17
18 THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
19 AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
20 IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
21 DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
22 FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
23 DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
24 SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
25 CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
26 OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
```

End of file



세종대학교
SEJONG UNIVERSITY



License

```
/* zlib.h -- interface of the 'zlib' general purpose compression library
version 1.2.11, January 15th, 2017
```

Copyright (C) 1995-2017 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly Mark Adler
jloup@gzip.org madler@alumni.caltech.edu

*/

Click here to return to the [zlib Home Page](#).

Last modified 15 January 2017

Send all questions about zlib or its license to zlib@gzip.org

Web page copyright © 1996-2017 Greg Roelofs, [Jean-loup Gailly](#) and [Mark Adler](#).
zlib software copyright © 1995-2017 [Jean-loup Gailly](#) and [Mark Adler](#).

zlib.org domain name donated by Andrew Green.



세종대학교
SEJONG UNIVERSITY

```
1 Apache License
2
3 Version 2.0, January 2004
4 http://www.apache.org/licenses/
5
6 TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
7
8 1. Definitions.
9
10 "License" shall mean the terms and conditions for use, reproduction,
11 and distribution as defined by Sections 1 through 9 of this document.
12
13 "Licensor" shall mean the copyright owner or entity authorized by
14 the copyright owner that is granting the License.
15
16 "Legal Entity" shall mean the union of the acting entity and all
17 other entities that control, are controlled by, or are under common
18 control with that entity. For the purposes of this definition,
19 "control" means (i) the power, direct or indirect, to cause the
20 direction or management of such entity, whether by contract or
21 otherwise, or (ii) ownership of fifty percent (50%) or more of the
22 outstanding shares, or (iii) beneficial ownership of such entity.
23
24 "You" (or "Your") shall mean an individual or Legal Entity
25 exercising permissions granted by this License.
26
27 "Source" form shall mean the preferred form for making modifications,
28 including but not limited to software source code, documentation
source, and configuration files.
```

But keep in mind...

Publicly available code is not always open source

If not clearly stated, assume it's proprietary

Projects sometimes include other licenses inside

You need to be compliant with underlying licenses too

Copy pasted code also carries license obligations

*Stackoverflow user contributions CC-BY-SA 4.0**

License text should be reviewed/scanned

Sometimes there are modifications or additions

*Creative Commons Attribution-Share Alike <https://stackoverflow.com/help/licensing>



History for [SkywarsPe](#) / LICENSE

Commits on Nov 28, 2014

[Update LICENSE](#)

 Wantedkillerss committed on Nov 28, 2014



[7608970](#)



Commits on Sep 10, 2014

[Update LICENSE](#)

 Wantedkillerss committed on Sep 10, 2014



[7b8fe2a](#)



Commits on Aug 18, 2014

[Update LICENSE](#) ...

 Wantedkillers authored and Wantedkillers committed on Aug 18, 2014



[1c804e5](#)



[Update LICENSE](#)

 Wantedkillers authored and Wantedkillers committed on Aug 18, 2014



[2b76ef8](#)



Commits on Aug 16, 2014

[Create LICENSE](#)

 Wantedkillerss committed on Aug 17, 2014



[2340456](#)



<https://github.com/PocketWarriors/SkywarsPe/commits/master/LICENSE>



세종대학교
SEJONG UNIVERSITY

8	LICENSE		View file
		@@ -86,20 +86,22 @@ Apache License	
86	86	granted to You under this License for that Work shall terminate	
87	87	as of the date such litigation is filed.	
88	88		
89	-	4. Redistribution. You may reproduce and distribute copies of the	
89	+	4. Redistribution. You <ins>can not</ins> reproduce and distribute copies of the	
90	90	Work or Derivative Works thereof in any medium, with or without	
91	-	modifications, and in Source or Object form, provided that You	
91	+	modifications, and in Source or Object form, <ins>unless</ins> You	
92	92	meet the following conditions:	
93	93		
94	94	(a) You must give any other recipients of the Work or	
95	95	Derivative Works a copy of this License; and	
96	96		
97	97	(b) You must cause any modified files to carry prominent notices	
98	-	stating that You changed the files; and	
98	+	stating that You changed the files with holding a difference of 85%	
99	+	if u lied you can be sewd in the court of law; and	
99	100		
100	101	(c) You must retain, in the Source form of any Derivative Works	
101	102	that You distribute, all copyright, patent, trademark, and	
102	103	attribution notices from the Source form of the Work,	
104	+	with permission of the legal owner of these files	
102	105		

 Showing 1 changed file with 8 additions and 1 deletion.

Unified Split

9	LICENSE	View file	▼
		@@ -174,7 +174,14 @@ Apache License	
174	174	defend, and hold each Contributor harmless for any liability	
175	175	incurred by, or claims asserted against, such Contributor by reason	
176	176	of your accepting any such warranty or additional liability.	
177	-		
177	+		
178	+		
179	+ 10. Usage of these files are allowed by the legal owner under these terms		
180	+		
181	+ (a): You do not edit them under any circumstances		
182	+		
183	+ (b): You only use these files for "pocket-mine API " website: pocketmine.net		
184	+		
178	185 END OF TERMS AND CONDITIONS		
179	186		
180	187 APPENDIX: How to apply the Apache License to your work.		



Aspects of Licenses

- Although there are many different free software licenses available, in the important respects they all say the same things:
 - that anyone can modify the code,
 - that anyone can redistribute it both in original and modified form, and
 - that the copyright holders
- and authors provide no warranties whatsoever (avoiding liability is especially important given that people might run modified versions without even knowing it).
- The differences between licenses boil down to a few oft-recurring issues

The GPL and License Compatibility

- The sharpest dividing line in licensing is that between proprietary-incompatible and proprietary-compatible licenses, that is,
 - between the copyleft licenses and everything else.
- The canonical example of a copyleft license is the GNU General Public License and one of the most important considerations in choosing the GPL or AGPL is the extent to which it is compatible with other licenses
- Because the primary goal of the GPL's authors is the promotion of free software, they deliberately crafted the license to make it impossible to mix GPLed code into proprietary programs.
- Specifically, among the GPL's requirements are these two:

The GPL and License Compatibility

1. Any derivative work — that is, any work containing a nontrivial amount of GPLv3 code — must itself be distributed under the GPLv3.
2. No additional restrictions may be placed on the redistribution of either the original work or a derivative work. (The exact language is: "You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License.")

With these conditions, the GPL succeeds in making freedom contagious

Reading Materials

- Book
 - **Producing Open-Source Software How to Run a Successful Free Software Project** → Chapter 9
 - **Getting started with open-source development** → Chapter 3 – Licensing
 - **Understanding Open Source and Free Software Licensing** → Chapter 1: Open Source Licensing, Contract, and Copyright Law
- <https://opensource.guide/legal/>

Reading Materials

- **Books:**
 1. VM Brasseur, *Forge Your Future with Open Source*, The Pragmatic Programmers, LLC. 2018.
 2. Karl Fogel, *Producing Open Source Software: How to Run a Successful Free Software Project*, O'Reilly Media, 2009.
- <https://www.gnu.org/philosophy/free-sw.html>
- <https://blog.lizardwrangler.com/2008/01/22/january-22-1998-the-beginning-of-mozilla>
- <https://opensource.org/licenses>
- <https://www.coredna.com/blogs/comparing-open-closed-source-software>
- <https://linkedretail.com/open-vs-closed-source/>
- <https://www.assignmentprime.com/blog/open-vs-closed-source-software>
- <https://www.gnu.org/philosophy/free-sw.en.html>
- <https://opensource.org/osd>
- <https://pavanganeshbutha1998.blogspot.com/2019/10/free-software-philosophy.html>
- https://www.youtube.com/watch?v=Ag1AKII_2GM&t=89s

Attribution

- Some Slides are copied from **Prof. Stewart Weiss Lectures :**
http://www.compsci.hunter.cuny.edu/~sweiss/course_materials/csci395.86/slides/introduction.html#1

Thanks

Office Time: Monday-Friday (1000 - 1800)

You can send me an email for meeting, or any sort of discussion related to class matters.

jamil@sejong.ac.kr