WIKIPEDIA

Copyleft

Copyleft symbol

Copyleft is the practice of granting the right to freely distribute and modify intellectual property with the requirement that the same rights be preserved in derivative works created from that property.[1] Copyleft in the form of licenses can be used to maintain copyright conditions for works ranging from computer software, to documents, art, scientific discoveries and even certain patents.[2]

Copyleft software licenses are considered *protective* or *reciprocal* in contrast with permissive free software licenses,[3] and require that information necessary for reproducing and modifying the work must be made available to recipients of the software program, or binaries. This information is most commonly in the form of source code files, which usually contain a copy of the license terms and acknowledge the authors of the code.

Notable copyleft licenses include the GNU General Public License (GPL), originally written by Richard Stallman, which was the first software copyleft license to see extensive use,[4] and the Creative Commons share-alike license condition,[5] which is a copyleft license intended for other forms of intellectual property such as documents and pictures.

Contents

History

An early use of the word copyleft was in Li-Chen Wang's Palo Alto Tiny BASIC's distribution notice "@COPYLEFT ALL WRONGS RESERVED" in June 1976.[6][7] Tiny BASIC was not distributed under any formal form of copyleft distribution terms, but it was presented in a context where source code was being shared and modified. In fact, Wang had earlier contributed edits to Tiny BASIC Extended before writing his own BASIC interpreter.[8] He encouraged others to adapt his source code and publish their adaptions, as with Roger Rauskolb's version of PATB published in *Interface Age*.[9]

The concept of copyleft was described in Richard Stallman's *GNU Manifesto* in 1985, where he wrote:

> GNU is not in the public domain. Everyone will be permitted to modify and redistribute GNU, but no distributor will be allowed to restrict its further redistribution. That is to say, proprietary modifications will not be allowed. I want to make sure that all versions of GNU remain free.

Stallman worked a few years earlier on a Lisp interpreter. Symbolics asked to use the Lisp interpreter, and Stallman agreed to supply them with a public domain version of his work. Symbolics extended and improved the Lisp interpreter, but when Stallman wanted access to the improvements that Symbolics had made to his interpreter, Symbolics refused. Stallman then, in 1984, proceeded to work towards eradicating this emerging behavior and culture of proprietary software, which he named *software hoarding*. This was not the first time Stallman had dealt with proprietary software, but he deemed this interaction a "turning point". He justified software sharing, protesting that when sharing, the software online can be copied without the loss of the original piece of work. The software can be used multiple times without ever being damaged or wearing out.[10][11]

As Stallman deemed it impractical in the short term to eliminate current copyright law and the wrongs he perceived it perpetuated, he decided to work within the framework of existing law; in 1985,[12] he created his own copyright license, the Emacs General Public License,[13] the first copyleft license. This later evolved into the GNU General Public License, which is now one of the most popular free-software licenses. For the first time a copyright holder had taken steps to ensure that the maximal number of rights be perpetually transferred to a program's users, no matter what subsequent revisions anyone made to the original program. This original GPL did not grant rights to the public at large, only those who had already received the program; but it was the best that could be done under existing law.

The new license was not at this time given the copyleft label.[14] Richard Stallman stated that the use of "Copyleft" comes from Don Hopkins, who mailed him a letter in 1984 or 1985, on which was written: "Copyleft – all rights reversed."[14] In the early 1970s, the self-published book *Principia Discordia* contains the notice "Ⓚ All Rites Reversed – reprint what you like" *(sic)*. In the arts, Ray Johnson had earlier coined the term independently as it pertained to his making of and distribution of his mixed media imagery in his mail art and ephemeral gifts, for which he encouraged the making of derivative works. (While the phrase appears briefly as (or on) one of his pieces in the 2002 documentary *How to Draw a Bunny*, Johnson himself is not referenced in the 2001 documentary *Revolution OS*.)

Copyleft principles

Freedom

While copyright law gives software authors control over copying, distribution and modification of their works, the goal of copyleft is to give all users/viewers of the work the freedom to carry out all of these activities. These freedoms (from the Free Software Definition) include:[10][15]

Freedom 0

the freedom to use the work

Freedom 1

the freedom to study the work

Freedom 2

the freedom to copy and share the work with others

Freedom 3

the freedom to modify the work, and the freedom to distribute modified and therefore derivative works

Similar terms are present in the Open Source Definition, a separate definition that contains similar freedoms. The vast majority of copyleft licenses satisfy both definitions, that of the Free Software Definition and Open Source Definition.[10] By guaranteeing viewers and users of a work the freedom and permission to reproduce, adapt, or distribute it, copyleft licenses are distinct from other types of copyright licenses that limit such freedoms.

Reciprocity

Instead of allowing a work to fall completely into the public domain, where no ownership of copyright is claimed, copyleft allows authors to impose restrictions on the use of their work. One of the main restrictions imposed by copyleft is that derived works must also be released under a compatible copyleft license.[10]

This is due to the underlying principle of copyleft: that anyone can benefit freely from the previous work of others, but that any modifications to that work should benefit everyone else as well, and thus must be released under similar terms. For this reason, copyleft licenses are also known as reciprocal licenses- any modifiers of a copyleft-licensed work are expected to reciprocate the author's action of copyleft-licensing the software by also copyleft-licensing any derivatives they might have made. Because of this requirement, copyleft licenses have also been described as "viral" due to their self-perpetuating terms.[16]

In addition to restrictions on copying, copyleft licenses address other possible impediments. They ensure that rights cannot be later revoked, and require the work and its derivatives to be provided in a form that allows further modifications to be made. In software, this means requiring that the source code of the derived work be made available together with the software itself.[10]

Economic Incentive

The economic incentives to work on copyleft content can vary. Traditional copyright law is designed to promote progress by providing economic benefits to creators. When choosing to copyleft their work, content creators may seek complementary benefits like recognition from their peers.

In the world of computer programming, copyleft-licensed computer programs are often created by programmers to fill a need they have noticed. Such programs are often published with a copyleft license simply to ensure that subsequent users can also freely use modified versions of that program. This is especially true for creators who wish to prevent "open source hijacking", or the act of reusing open sourced code and then adding extra restrictions to it, an action prevented by copyleft licensing the software. Some creators feel that preventing commercial enterprises from absorbing and selling their product is another incentive.

Furthermore, the open-source culture of programming has been described as a gift culture, where social status is determined by an individual's contributions.[17] Contributing to or creating open-source, copyleft-licensed software of high quality can lead to contributors gaining valuable experience and can lead to future career opportunities.[18]

Copyleft software has economic effects beyond individual creators. The presence of quality copyleft software can force proprietary software developers to increase the quality of their software to compete with free software.[19] This may also have the effect of preventing monopolies in areas dominated by proprietary software. However, competition with proprietary software can also be a reason to forego copyleft. The Free Software Foundation recommends that when "widespread use of the code is vital for advancing the cause of free software,"[20] allowing the code to be copied and used freely is more important than a copyleft.

Copyleft application

Common practice for using copyleft is to codify the copying terms for a work with a license. Any such license typically includes all the provisions and principles of copyleft inside the license's terms. This includes the freedom to use the work, study the work, copy and share the work with others, modify the work, and distribute modified versions of that work.

Unlike similar permissive licenses that also grant these freedoms, copyleft licenses also ensure that any modified versions of a copyleft licensed work must also grant these freedoms. Thus, copyleft licenses have restrictions: that modifications of any copyleft-licensed work must be distributed under a compatible copyleft scheme and that the distributed modified work must include a means of modifying the work. Under fair use, however, copyleft licenses may be superseded, just like regular copyrights. Therefore, any person utilizing a copyleft-licensed source for their own work is free to choose any other license provided they meet the fair use standard.[21]

Copyleft licenses necessarily make creative use of relevant rules and laws to enforce their provisions. For example, when using copyright law, those who contribute to a work under copyleft usually must gain, defer or assign copyright holder status. By submitting the copyright of their contributions under a copyleft license, they deliberately give up some of the rights that normally follow from copyright, including the right to be the unique distributor of copies of the work.

Some laws used for copyleft licenses vary from one country to another, and may also be granted in terms that vary from country to country. For example, in some countries it is acceptable to sell a software product without warranty, in standard GNU GPL style (see articles 11 and 12 of the GNU GPL version 2 (https://www.gnu.org/licenses/gpl.html)), while in most European countries it is not permitted for a software distributor to waive all warranties regarding a sold product. For this reason the extent of such warranties are specified in most European copyleft licenses. Regarding that, see the European Union Public Licence EUPL,[22] or the CeCILL license,[23] a license that allows one to use GNU GPL (see article 5 of the EUPL and article 5.3.4 of CeCILL) in combination with a limited warranty (see article 7 and 8 of the EUPL and 9 of CeCILL). For projects which will be run over a network, a variation of the GPL is provided in the Affero General Public License, which ensures that the source code is available to users of network software.

Types and relation to other licenses

	Free			**Non-free**		
	Public domain & equivalents	**Permissive license**	**Copyleft (protective license)**	**Noncommercial license**	**Proprietary license**	**Trade secret**
Description	Grants all rights	Grants use rights, including right to relicense (allows proprietization, license compatibility)	Grants use rights, forbids proprietization	Grants rights for noncommercial use only. May be combined with share-alike.	Traditional use of copyright; no rights need be granted	No information made public
For software	PD, Unlicense, CC0	BSD, MIT, Apache	GPL, AGPL	JRL, AFPL	Proprietary software, no public license	Private, internal software
For other creative works	PD, CC0	CC-BY	CC-BY-SA	CC-BY-NC	Copyright, no public license	Unpublished

Copyleft is a distinguishing feature of some free software licenses, while other free-software licenses are not copyleft licenses because they do not require the licensee to distribute derivative works under the same license. There is an ongoing debate as to which class of license provides the greater degree of freedom. This debate hinges on complex issues, such as the definition of freedom and whose freedoms are more important: the potential future recipients of a work (freedom from the creation of proprietary software) or just the initial recipient (freedom to create proprietary software). However, the current availability of both types of licenses, copyleft and permissive, allows authors to choose the type that best fits their work.

For documents, art, and other works other than software and code, the Creative Commons share-alike licensing system and GNU's Free Documentation License allows authors to apply limitations to certain sections of their work, exempting some parts of their creation from the full copyleft mechanism. In the case of the GFDL, these limitations include the use of invariant sections, which may not be altered by future editors. The initial intention of the GFDL was as a device for supporting the documentation of copylefted software. However, the result is that it can be used for any kind of document.

The Creative Commons icon for Share-Alike, a variant of the copyleft symbol

Strong and weak copyleft

The strength of the copyleft license governing a work is determined by the extent its provisions can be imposed on all kinds of derived works. Thus, the term "weak copyleft" refers to licenses where not all derived works inherit the copyleft license; whether a derived work inherits or not often depends on how it was derived.

"Weak copyleft" licenses are often used to create software libraries. This allows other software to link to the library and be redistributed without the requirement for the linking software to also be copyleft-licensed. Only changes to the weak-copyleft-licensed software itself become subject to copyleft provisions of such a license. This allows programs of any license to be compiled and linked against copylefted libraries such as glibc and then redistributed without any re-licensing required. The concrete effect of strong vs. weak copyleft has yet to be tested in court.[24] Free-software licenses that use "weak" copyleft include the GNU Lesser General Public License and the Mozilla Public License.

The most well-known free-software license using strong copyleft is the GNU General Public License. A stronger copyleft license is the AGPL, which requires the publishing of the source code for software as a service use cases ("see also" the term sometimes used, "service as a software substitute [SaaSS]"[25][26][27]) e.g. when software is deployed on servers.[28] The Sybase Open Watcom Public License is one of the strongest copyleft licenses, as this license closes the "private usage" loophole of the GPL, and requires the publishing of source code in any use case.[29] A final, notable strong copyleft license is the Design Science License, as it can apply to any work, not only software or documentation, but also literature, artworks, music, photography and video. The Design Science License was created by Michael Stutz after he took an interest in applying GNU-style copyleft to non-software works, which later came to be called open content. In the 1990s the DSL was used on music recordings, visual art, and even novels. It is now hosted on the Free Software Foundation website's license list,[30] but it is not considered compatible with the GPL by the Free Software Foundation.

Full and partial copyleft

"Full" and "partial" copyleft relate to another issue. Full copyleft exists when all parts of a work (except the license itself) may only be modified and distributed under the terms of the work's copyleft license. Partial copyleft, by contrast, exempts some parts of the work from the copyleft provisions, permitting distribution of some modifications under terms other than the copyleft license, or in some other way does not impose all the principles of copylefting on the work. An example of partial copyleft is the GPL linking exception made for some software packages.

Share-alike

The "share-alike" condition in some licenses imposes the requirement that any freedom that is granted regarding the original work must be granted on exactly the same or compatible terms in any derived work.

This implies that any copyleft license is automatically a share-alike license but not the other way around, as some share-alike licenses include further restrictions such as prohibiting commercial use. Another restriction is that not everyone wants to share their work and some share-alike agreements require that the whole body of work be shared, even if the author only wants to share a certain part. The plus side for an author of source code is that any modification to the code will not only benefit the original creator, but that the author will be recognized and hold equal claim over the changed code.[31] Some Creative Commons licenses are examples of share-alike copyleft licenses.

Permissive licenses

Permissive software licenses are those that grant users of the software the same freedoms as copyleft licenses, but do not require modified versions of that software to also include those freedoms. They have minimal restrictions on how the software can be used, modified, and redistributed, and are thus not copyleft licenses. Examples of this type of license include the X11 license, Apache license, MIT license and the BSD licenses.

Debate and controversy

It has been suggested that copyleft became a divisive issue in the ideological strife between the Open Source Initiative and the free-software movement.[32] However, there is evidence that copyleft is both accepted and proposed by both parties:

- Both the OSI and the FSF have copyleft and non-copyleft licenses in their respective lists of accepted licenses.[33][30]
- The OSI's original Legal Counsel Lawrence Rosen has written a copyleft license, the Open Software License.
- The OSI's licensing how-to recognises the GPL as a "best practice" license.[34]
- Some of the software programs of the GNU Project are published under non-copyleft licenses.[35]
- Stallman himself has endorsed the use of non-copyleft licenses in certain circumstances, most recently in the case of the Ogg Vorbis license change.[36]

Viral licensing

Viral license is a pejorative name for copyleft licenses.[37][38][39][40][41] It originates from the terms 'General Public Virus' or 'GNU Public Virus' (GPV), which dates back to 1990, a year after the GPLv1 was released.[42][43][44] The name "viral licenses" refers to the fact that any works derived from a copyleft work must preserve the copyleft permissions when distributed.

Some BSD License advocates used the term derisively in regards to the GPL's tendency to absorb BSD licensed code without allowing the original BSD work to benefit from it, while at the same time promoting itself as "freer" than other licenses.[45][46][47] Microsoft vice-president Craig Mundie remarked, "This viral aspect of the GPL poses a threat to the intellectual property of any organization making use of it."[48] In another context, Steve Ballmer declared that code released under GPL is useless to the commercial sector, since it can only be used if the resulting surrounding code is licensed under a GPL compatible license, and described it thus as "a cancer that attaches itself in an intellectual property sense to everything it touches".[49]

In response to Microsoft's attacks on the GPL, several prominent free-software developers and advocates released a joint statement supporting the license.[50] According to FSF compliance engineer David Turner, the term "viral license" creates a misunderstanding and a fear of using copylefted free software.[51] While a person can catch a virus without active action, license conditions take effect upon effective usage or adoption.[52] David McGowan has also written that there is no reason to believe the GPL could force proprietary software to become free software, but could "try to enjoin the firm from distributing commercially a program that combined with the GPL'd code to form a derivative work, and to recover damages for infringement." If the firm "actually copied code from a GPL'd program, such a suit would be a perfectly ordinary assertion of copyright, which most private firms would defend if the shoe were on the other foot."[53] Richard Stallman has described this view with an analogy, saying, "The GPL's domain does not spread by proximity or contact, only by deliberate inclusion of GPL-covered code in your program. It spreads like a spider plant, not like a virus."[54]

Popular copyleft licenses, such as the GPL, have a clause allowing components to interact with non-copyleft components as long as the communication is abstract, such as executing a command-line tool with a set of switches or interacting with a Web server.[55] As a consequence, even if one module of an otherwise non-copyleft product is placed under the GPL, it may still be legal for other components to communicate with it normally. This allowed communication may or may not include reusing libraries or routines via dynamic linking – some commentators say it does,[56] the FSF asserts it does not and explicitly adds an exception allowing it in the license for the GNU Classpath re-implementation of the Java library. This ambiguity is an important difference between the GPL and the LGPL, in that the LGPL specifically allows linking or compiling with the covered work.[57]

Symbol

🄯 🄯	
Copyleft symbol	
In Unicode	U+1F12F 🄯 COPYLEFT SYMBOL (HTML &#127279;) Alternative symbol: (ɔ)
Different from	
Different from	U+00A9 © COPYRIGHT (HTML &#169; · &copy;, &COPY;)

The copyleft symbol is a mirror image of the copyright symbol, ©: a reversed C in a circle. It has no legal status.[58] A 2016 proposal[59] to add the symbol to a future version of Unicode was accepted by the Unicode Technical Committee.[60] The code point U+1F12F 🄯 COPYLEFT SYMBOL was added in Unicode 11.[60][61]

As of 2018, it is largely unimplemented in fonts, but can be approximated with character U+2184 ↄ LATIN SMALL LETTER REVERSED C or the more widely available character U+0254 ɔ LATIN SMALL LETTER OPEN O between parenthesis (ɔ) or, if supported by the application or web browser, by combining a reversed c with the character U+20DD ⃝ COMBINING ENCLOSING CIRCLE: ↄ⃝.[62]

For a list of fonts that include this glyph, see Unicode fonts#List of SMP Unicode fonts and then row "Enclosed Alphanumeric Supplement (173: 1F100–1F1FF)" (This list is not guaranteed to be current).

See also

- List of copyleft software licenses
- All rights reversed
- Anti-copyright notice
- Commercial use of copyleft works
- Comparison of open source and closed source
- Copyright
- Creative Commons licences
- Free content
- Free Culture movement
- Free-software movement
- GNU General Public License
- HESSLA – a license which prohibits uses that violate human rights or add spyware
- History of free and open-source software
- Kopimi
- Open content
- Opposition to copyright
- Patentleft
- Permissive free-software licence
- Public copyright license
- Public domain
- Share-alike

Notes and references

1. "What is Copyleft?" (https://www.gnu.org/copyleft/). GNU Project.
2. Newman, John (29 December 2011). "Copyright and Open Access at the Bedside". *NEJM*. **365** (26): 2447–2449. doi:10.1056/NEJMp1110652 (https://doi.org/10.1056%2FNEJMp1110652). PMID 22204721 (https://pubmed.ncbi.nlm.nih.gov/22204721).
3. Open Source from a Proprietary Perspective (https://web.archive.org/web/*/https://www.redhat.com/f/summitfiles/presentation/May31/Open%20Source%20Dynamics/Troan_OpenSourceProprietyPersp.pdf) at the Wayback Machine (archive index)
4. Stallman, Richard (29 June 2007). "GNU General Public License" (https://www.gnu.org/copyleft/gpl.html). *GNU Project*. Retrieved 1 May 2017.
5. "Attribution-ShareAlike 4.0 International (CC BY-SA 4.0)" (https://creativecommons.org/licenses/by-sa/4.0/). *Creative Commons*. Creative Commons. Retrieved 14 August 2015.
6. Wang, Li-Chen (May 1976). "Palo Alto Tiny BASIC". *Dr. Dobb's Journal of Computer Calisthenics & Orthodontia, Running Light Without Overbyte*. **1** (5): 12–25. (NB. Source code begins with the following six lines. "TINY BASIC FOR INTEL 8080; VERSION 1.0; BY LI-CHEN WANG; 10 JUNE, 1976; @COPYLEFT; ALL WRONGS RESERVED". The June date in the May issue is correct. The magazine was behind schedule, the June and July issues were combined to catch up.)
7. Rauskolb, Roger (December 1976). "Dr. Wang's Palo Alto Tiny BASIC". *Interface Age*. **2** (1): 92–108. (NB. The source code begins with the following nine lines: "TINY BASIC FOR INTEL 8080; VERSION 2.0; BY LI-CHEN WANG; MODIFIED AND TRANSLATED TO INTEL MNEMONICS; BY ROGER RAUSKOLB; 10 OCTOBER, 1976 ; @COPYLEFT; ALL WRONGS RESERVED")
8. "Tiny BASIC Extended". *Dr. Dobb's Journal of Computer Calisthenics & Orthodontia, Running Light Without Overbyte*. **1** (2). February 1976.
9. Rauskolb, Roger (December 1976). "Dr. Wang's Palo Alto Tiny BASIC". *Interface Age*. **2** (1): 92–108. (NB. The source code begins with the following nine lines: "TINY BASIC FOR INTEL 8080; VERSION 2.0; BY LI-CHEN WANG; MODIFIED AND TRANSLATED TO INTEL MNEMONICS; BY ROGER RAUSKOLB; 10 OCTOBER, 1976 ; @COPYLEFT; ALL WRONGS RESERVED")
10. Carver, Brian W. "Share and Share Alike: Understanding and Enforcing Open Source and Free Software Licenses" (https://scholarship.law.berkeley.edu/btlj/vol20/iss1/46/). Berkeley Technology Law Journal. SSRN 1586574 (https://ssrn.com/abstract=1586574). Retrieved 6 February 2012.
11. Williams, Sam (March 2002). "7" (http://www.oreilly.com/openbook/freedom/ch07.html). *Free as in Freedom – Richard Stallman's Crusade for Free Software* (https://archive.org/details/freeasinfreedomr00will). O'Reilly Media. ISBN 978-0-596-00287-9.
12. Moody, Glyn (2002). *Rebel Code*. p. 26.
13. "Emacs General Public License" (http://www.free-soft.org/gpl_history/emacs_gpl.html). 5 July 2001. Retrieved 23 August 2008.
14. Stallman, Richard (21 January 2008). "About the GNU Project" (https://www.gnu.org/gnu/thegnuproject.html). Free Software Foundation. Retrieved 1 May 2017.
15. "What is free software?" (https://www.gnu.org/philosophy/free-sw.html). *gnu.org*. 30 July 2019. Retrieved 22 July 2020.
16. Mundie, Craig (3 May 2001). "Prepared Text of Remarks by Craig Mundie, Microsoft Senior Vice President – The Commercial Software Model" (https://web.archive.org/web/20050621082004/http://www.microsoft.com/presspass/exec/craig/05-03sharedsource.mspx). New York University Stern School of Business. Archived from the original (http://www.microsoft.com/presspass/exec/craig/05-03sharedsource.mspx) on 21 June 2005. Retrieved 1 October 2009.
17. Maher, Marcus (2000). "Open Source Software: The Success of an Alternative Intellectual Property Incentive Paradigm" (https://web.archive.org/web/20140502033143/http://iplj.net/blog/wp-content/uploads/2009/09/Article-OPEN-SOURCE-SOFTWARE-THE-SUCCESS-OF-AN-ALTERNATIVE-INTELLECTUAL-PROPERTY-INCENTIVE-PARADIGM.pdf) (PDF). Fordham Intellectual Property, Media & Entertainment Law Journal. Archived from the original (http://iplj.net/blog/wp-content/uploads/2009/09/Article-OPEN-SOURCE-SOFTWARE-THE-SUCCESS-OF-AN-ALTERNATIVE-INTELLECTUAL-PROPERTY-INCENTIVE-PARADIGM.pdf) (PDF) on 2 May 2014. Retrieved 1 May 2014.

18. Sarmah, Harshajit (23 September 2019). "5 Reasons Why Contributing To Open Source Projects Helps In Landing A Job" (https://analyticsindiamag.com/5-reasons-why-contributing-to-open-source-projects-helps-in-landing-a-job/). *Analytics India Magazine*. Retrieved 22 July 2020.
19. Mustonen, Mikko. "Copyleft - The Economics of Linux and Other Open Source Software" (https://web.archive.org/web/20140502032254/http://www.cyberinet05.inet-tr.org.tr/telekom/MI_mustonen.pdf) (PDF). Information Economics and Policy. Archived from the original (http://www.cyberinet05.inet-tr.org.tr/telekom/MI_mustonen.pdf) (PDF) on 2 May 2014. Retrieved 1 May 2014.
20. "How to choose a license for your own work" (https://www.gnu.org/licenses/license-recommendations.html#libraries). Free Software Foundation's Licensing and Compliance Lab. Retrieved 1 May 2017.
21. Kirk St.Amant & Brian Still (2008). "Examining Open Source Software Licenses through the Creative Commons Licensing Model". *Handbook of Research on Open Source Software: Technological, Economic, and Social Perspectives*. Information Science Reference. pp. 382 of 728. ISBN 978-1-59140-999-1.
22. "The EUPL – European Union Public Licence" (http://www.osor.eu/eupl/european-union-public-licence-eupl-v.1.1). European Commission. Retrieved 9 January 2007.
23. "Free Software Licensing Agreement CeCILL" (https://web.archive.org/web/20100808040542/http://www.inria.fr/valorisation/logiciels/Licence.CeCILL-V1.US.pdf) (PDF). INRIA. Archived from the original (http://www.inria.fr/valorisation/logiciels/Licence.CeCILL-V1.US.pdf) (PDF) on 8 August 2010. Retrieved 24 August 2010.
24. Patrice-Emmanuel Schmitz (3 April 2015). "The meaning of "Copyleft" in EUPL" (https://joinup.ec.europa.eu/community/eupl/news/meaning-%E2%80%9Ccopyleft%E2%80%9D-eupl). *Joinup*.
25. For some of the philosophy behind the term "SaaSS", see the "Server Software" section of "How to choose a license for your own work" (https://www.gnu.org/licenses/license-recommendations.html). GNU Project. 20 December 2015. Retrieved 1 May 2017.
26. Stallman, Richard (18 November 2016). "Who does that server really serve?" (https://www.gnu.org/philosophy/who-does-that-server-really-serve.html). GNU Project. Retrieved 1 May 2017.
27. "Why the Affero GPL" (https://www.gnu.org/licenses/why-affero-gpl.html). GNU Project. 10 May 2015. Retrieved 1 May 2017.
28. List of free-software licences on the GNU website (https://www.gnu.org/philosophy/license-list.html#AGPL): "*We recommend that developers consider using the GNU AGPL for any software which will commonly be run over a network*".
29. "Various Licenses and Comments about Them - Sybase Open Watcom Public License version 1.0 (#Watcom)" (https://www.gnu.org/licenses/license-list.html#Watcom). GNU.org. Retrieved 23 December 2015. "This is not a free software license. It requires you to publish the source code publicly whenever you "Deploy" the covered software, and "Deploy" is defined to include many kinds of private use."
30. "Various Licenses and Comments about Them" (https://www.gnu.org/licenses/license-list.html). GNU Project.
31. Engelfriet, Arnoud (2010). "Choosing an Open Source License". *IEEE Software*. Athens University of Economics and Business. **27**: 48–49. doi:10.1109/MS.2010.5 (https://doi.org/10.1109%2FMS.2010.5). S2CID 36712745 (https://api.semanticscholar.org/CorpusID:36712745).
32. Biancuzzi, Federico (30 June 2005). "ESR: "We Don't Need the GPL Anymore" " (http://www.onlamp.com/pub/a/onlamp/2005/06/30/esr_interview.html). ONLamp.com. Retrieved 23 August 2008.
33. Tiemann, Michael (18 September 2006). "Licenses by Name" (http://www.opensource.org/licenses/alphabetical). Retrieved 23 August 2008.
34. Raymond, Eric Steven (9 November 2002). "Licensing HOWTO" (http://www.catb.org/~esr/Licensing-HOWTO.html#choosing). Retrieved 23 August 2008.
35. "What the GPLv3 Means for MS-Novell Agreement" (http://yro.slashdot.org/comments.pl?sid=225606&cid=18272276). Yro.slashdot.org. 8 March 2007. Retrieved 14 May 2014.
36. Stallman, Richard (26 February 2001). "LWN.net: RMS on the Ogg Vorbis license" (https://lwn.net/2001/0301/a/rms-ov-license.php3). Retrieved 23 August 2008. "[M]y agreement with the idea of a lax [Ogg/Vorbis] license in this special case is just as pragmatic as my preference for the GPL in most cases. In both cases it is a matter of how we can attain freedom."
37. "Microsoft license spurns open source" (http://news.cnet.com/2100-1001-268889.html). *CNET*. CBS Interactive.

38. "Some rights reserved: the alternatives to copyright" (https://www.wired.co.uk/news/archive/2011-12/16/alternatives-to-copyright). *Wired UK*.
39. "Glossary" (https://web.archive.org/web/20090426091132/http://a2knetwork.org/glossary). *a2knetwork.org*. Archived from the original (http://a2knetwork.org/glossary) on 26 April 2009.
40. Jason Greenberg; Daniel Glazer (March 2013). "Inoculating Your Purchase – Contractual Protection from Viral Licenses in M&A Transactions" (http://www.friedfrank.com/siteFiles/Publications/ACC-GNY_Newsletter_OpenSourceInMATransactions.pdf) (PDF). *Fried Frank*. Association of Corporate Counsel. Retrieved 10 June 2016.
41. Philip Wood; Amy Ryburn; Allan Yeoman; Andrew Matangi; Steve Nightingale (1 July 2013). "Legal update on Information and Communication Technology – July 2013" (https://web.archive.org/web/20130921110256/http://www.buddlefindlay.com/article/2013/07/01/legal-update-on-information-and-communication-technology-%E2%80%93-july-2013). Archived from the original (http://www.buddlefindlay.com/article/2013/07/01/legal-update-on-information-and-communication-technology-%E2%80%93-july-2013) on 21 September 2013. Retrieved 10 June 2016.
42. Vixie, Paul (6 March 2006). "Re: Section 5.2 (IPR encumberance) in TAK rollover requirement draft" (https://web.archive.org/web/20070927175628/http://psg.com/lists/namedroppers/namedroppers.2006/msg00246.html). IETF Namedroppers mailing list. Archived from the original (http://psg.com/lists/namedroppers/namedroppers.2006/msg00246.html) on 27 September 2007. Retrieved 29 April 2007.
43. "General Public Virus" (http://catb.org/esr/jargon/oldversions/jarg221.txt). *Jargon File* *2.2.1*. 15 December 1990. Retrieved 29 April 2007.
44. Hackvän, Stig (September 1999). "Reverse-engineering the GNU Public Virus – Is copyleft too much of a good thing?" (https://web.archive.org/web/20110718132324/http://devlinux.org/lw-gnu-published.html). *Linux Journal*. Archived from the original (http://devlinux.org/lw-gnu-published.html) on 18 July 2011. Retrieved 29 April 2007.
45. Stewart, Bill (8 October 1998). "Re: propose: 'cypherpunks license' (Re: Wanted: Twofish source code)" (https://web.archive.org/web/20070529202454/http://cypherpunks.venona.com/date/1998/10/msg00429.html). Cypherpunks mailing list. Archived from the original (http://cypherpunks.venona.com/date/1998/10/msg00429.html) on 29 May 2007. Retrieved 29 April 2007.
46. Buck, Joe (10 October 2000). "Re: Using of parse tree externally" (https://gcc.gnu.org/ml/gcc/2000-10/msg00198.html). GCC mailing list. Retrieved 29 April 2007.
47. Griffis, L. Adrian (15 July 2000). "The GNU Public Virus" (http://themes.freshmeat.net/articles/view/172#comment-5548). Retrieved 29 April 2007.
48. Mundie, Craig (3 May 2001). "Speech Transcript – Craig Mundie" (https://web.archive.org/web/20050621082004/http://www.microsoft.com/presspass/exec/craig/05-03sharedsource.mspx). New York University Stern School of Business. Archived from the original (http://www.microsoft.com/presspass/exec/craig/05-03sharedsource.mspx) on 21 June 2005. Retrieved 23 August 2008.
49. Newbart, Dave (1 June 2001). "Microsoft CEO takes launch break with the Sun-Times" (https://web.archive.org/web/20010615205548/http://suntimes.com/output/tech/cst-fin-micro01.html). Chicago Sun-Times. Archived from the original (http://suntimes.com/output/tech/cst-fin-micro01.html) on 15 June 2001.(Internet archive link)
50. *Free Software Leaders Stand Together* (https://en.wikisource.org/wiki/Free_Software_Leaders_Stand_Together) – via Wikisource.
51. Byfield, Bruce (29 August 2006). "IT Manager's Journal: 10 Common Misunderstandings About the GPL" (http://www.linuxtoday.com/developer/2006082902126OSHLLL). Retrieved 23 August 2008.
52. "The license term smorgasbord: copyleft, share-alike, reciprocal, viral, or hereditary? – Luis Villa: Open Law and Strategy" (https://lu.is/blog/2012/02/02/the-license-term-smorgasbord-copyleft-share-alike-reciprocal-viral-or-hereditary/). *lu.is*. Retrieved 22 August 2018.
53. David McGowan (2005). "Legal Aspects of Free and Open Source Software". In Joseph Feller; Brian Fitzgerald; Scott A. Hissam; Karim R. Lakahani (eds.). *Perspectives on Free and Open Source Software* (https://archive.org/details/perspectivesonfr00fell). MIT Press. p. 382 (https://archive.org/details/perspectivesonfr00fell/page/n395). ISBN 978-0-262-06246-6.
54. Poynder, Richard (21 March 2006). "The Basement Interviews: Freeing the Code" (https://archive.org/stream/The_Basement_Interviews/Richard_Stallman_Interview_djvu.txt). Retrieved 5 February 2010.
55. "Frequently Asked Questions about the GNU Licenses" (https://www.gnu.org/licenses/gpl-faq.html#MereAggregation). Free Software Foundation. 24 June 2008. Retrieved 1 May 2017.

56. Raymond, Eric Steven (9 November 2002). "Licensing HOWTO" (http://www.catb.org/~esr/Licensing-HOWTO.html#id2789302). Retrieved 21 March 2010.
57. Stallman, Richard. Why you shouldn't use the Lesser GPL for your next library (https://www.gnu.org/licenses/why-not-lgpl.html). GNU Project
58. Hall, G. Brent (2008). *Open Source Approaches in Spatial Data Handling* (https://books.google.com/books?id=JZNuu8XODQMC&pg=PA29). Springer. p. 29. Bibcode:2008osas.book.....H (https://ui.adsabs.harvard.edu/abs/2008osas.book.....H). ISBN 9783540748311.
59. "Proposal to add the Copyleft Symbol to Unicode" (https://www.unicode.org/L2/L2016/16059-copyleft.pdf) (PDF).
60. "Proposed New Characters: Pipeline Table" (http://unicode.org/alloc/Pipeline.html). *Unicode Character Proposals*. Unicode Consortium. Retrieved 18 April 2017.
61. "Announcing The Unicode® Standard, Version 11.0" (http://blog.unicode.org/2018/06/announcing-unicode-standard-version-110.html). *blog.unicode.org*. Retrieved 6 June 2018.
62. "Unicode copyleft inquiry" (http://unicode.org/mail-arch/unicode-ml/Archives-Old/UML022/0846.html). 6 May 2000.

External links

- What is copyleft? (https://www.gnu.org/licenses/copyleft.html) – by Richard Stallman
- Copyleft: Pragmatic Idealism (https://www.gnu.org/philosophy/pragmatic.html) – by Richard Stallman

Retrieved from "https://en.wikipedia.org/w/index.php?title=Copyleft&oldid=1035328491"

This page was last edited on 25 July 2021, at 01:14 (UTC).