Copyleft

From Wikipedia, the free encyclopedia

The "reversed 'c' in a full circle" is the copyleft symbol. It is the copyright symbol mirrored. Unlike the copyright symbol, it has no legal meaning.

Copyleft is a play on the word copyright to describe the practice of using copyright law to remove restrictions on distributing copies and modified versions of a work for others and requiring that the same freedoms be preserved in modified versions.

Copyleft is a form of licensing and can be used to modify copyrights for works such as computer software, documents, music and art. In general, copyright law allows an author to prohibit others from reproducing, adapting, or distributing copies of the author's work. In contrast, an author may, through a copyleft licensing scheme, give every person who receives a copy of a work permission to reproduce, adapt or distribute the work as long as any resulting copies or adaptations are also bound by the same copyleft licensing scheme. A widely used and originating copyleft license is the GNU General Public License. Creative Commons provides a similar license called ShareAlike.

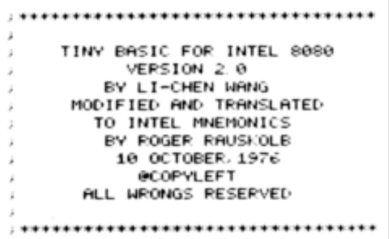
Copyleft can also be characterized as a copyright licensing scheme in which an author surrenders some but not all rights under copyright law. Instead of allowing a work to fall completely into the public domain (where no copyright restrictions are imposed), copyleft allows an author to impose some, but not all, copyright restrictions on those who want to engage in activities that would otherwise be considered copyright infringement. Under copyleft, copyright infringement can be avoided if the would-be infringer perpetuates the same copyleft scheme. For this reason copyleft licenses are also commonly known as "reciprocal" or "viral" licenses.[1]

While copyright law protects the rights of the creator by providing control of distribution and modification, the idea of copyleft is to grant subjective *libre* freedom to end users. Copyleft licenses specify clauses which explicitly remove those restrictions the creator considers to not provide libre freedom to the end user. In software, open source copyleft licenses place the primary restriction that information helpful in supporting modification of software (e.g. source code) must be made available to a user with a copy of the licensed software and allows the original author to be acknowledged.

Contents

History

An early example of copyleft was the Tiny BASIC project started in the newsletter of the People's Computer Company in 1975. Dennis Allison wrote a specification for a simple version of the BASIC programming language.[2] This design did not support text strings and only used integer arithmetic. The goal was for the program to fit in 2 to 3 kilobytes of memory.



```
;*********************************
;
;    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
;
;*********************************
```

The use of "Copyleft; All Wrongs Reserved" in 1976

The Tiny BASIC contents of the newsletter soon became *Dr. Dobb's Journal of Tiny BASIC* with a subtitle of "Calisthenics & Orthodontia, Running Light Without Overbyte." Hobbyists began writing BASIC language interpreters for their microprocessor based home computers and sending the source code to *Dr. Dobb's Journal* and other magazines to be published. By the middle of 1976, Tiny BASIC interpreters were available for the Intel 8080, the Motorola

6800 and MOS Technology 6502 processors. This was a shared source project before the internet allowed easy transfer of files. Computer hobbyists would exchange paper tapes, cassettes or even retype the files from the printed listings.[3]

Jim Warren, editor of *Dr. Dobb's Journal*, wrote in the July 1976 ACM Programming Language newsletter about the motivations and methods of this successful project. He started with this: "There is a viable alternative to the problems raised by Bill Gates in his irate letter to computer hobbyists concerning 'ripping off' software. When software is free, or so inexpensive that it's easier to pay for it than to duplicate it, then it won't be 'stolen'." The method was to have an experienced professional do the overall design and then outline an implementation strategy. Knowledgeable amateurs would implement the design for a variety of computer systems. Warren predicted this strategy would be continued and expanded.[3]

The May 1976 issue of *Dr. Dobbs Journal* had Li-Chen Wang's Palo Alto Tiny BASIC for the Intel 8080 microprocessor. The listing began with the usual title, author's name and date but it also had "@COPYLEFT ALL WRONGS RESERVED".[4] A fellow Homebrew Computer Club member, Roger Rauskolb, modified and improved Li-Chen Wang's program and this was published in the December 1976 issue of *Interface Age* magazine.[5] Roger added his name and preserved the COPYLEFT Notice.

A later instance of copyleft arose when Richard Stallman was working 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*.[6]

As Stallman deemed it impractical in the short term to eliminate current copyright law and the wrongs he perceived it perpetuating, he decided to work within the framework of existing law; he created his own copyright license, the Emacs General Public License,[7] 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.[8]

Richard Stallman stated that it comes from Don Hopkins, whom he calls a very imaginative fellow, who mailed him a letter in 1984 or 1985 on which was written: "Copyleft—all rights reversed."[8] The term "kopyleft" with the notation "All Rites Reversed" was also in use in the early 1970s within the Principia Discordia, which may have inspired Hopkins or influenced other usage. And 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 itself 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*.)

Some have suggested that copyleft became a divisive issue in the ideological strife between the Open Source Initiative and the free software movement.[9]. 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.[10][11]
- The OSI's original Legal Counsel Lawrence Rosen has written a copyleft license, the Open Software License.
- The OSI's licensing howto recognises the GPL as a "best practice" license.[12]
- Some of the software programs of Project GNU are published under non-copyleft licenses,[13]
- Stallman himself has endorsed the use of non-copyleft licenses in certain circumstances, most recently in the case of the Ogg Vorbis license change.[14]

Applying copyleft

Common practice for using copyleft is to codify the copying terms for a work with a license. Any such license typically gives each person possessing a copy of the work the same freedoms as the author, including (from the Free Software Definition):

0. the freedom to use the work,
1. the freedom to study the work,
2. the freedom to copy and share the work with others,
3. the freedom to modify the work, and the freedom to distribute modified and therefore derivative works.

(Note that the list begins from 0 due to a hacker tradition — first array element in C is numbered as 0.)

These freedoms do not ensure that a derivative work will be distributed under the same liberal terms. In order for the work to be truly copyleft, the license has to ensure that the author of a derived work can only distribute such works under the same or equivalent license.

In addition to restrictions on copying, copyleft licenses address other possible impediments. These include ensuring the rights cannot be later revoked and requiring the work and its derivatives are provided in a form that facilitates modification. In software, this requires that the source code of the derived work is made available together with the software itself.

Copyleft licenses necessarily make creative use of relevant rules and laws. 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 (http://www.fsf.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 CeCILL license (http://www.inria.fr/valorisation/logiciels/Licence.CeCILL-V1.US.pdf) , a license that allows one to use GNU GPL (see article 5.3.4 of CeCILL) in combination with a limited warranty (see article 9 of CeCILL).

Types of copyleft and relation to other licenses

See also: Free software licences (Freedom preserving restrictions)

Copyleft is a distinguishing feature of some free software licenses. Many 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, or whether to maximize the freedom of all potential future recipients of a work (*freedom from* the creation of proprietary software). Non-copyleft free software licenses maximize the freedom of the initial recipient (*freedom to* create proprietary software).

In common with the Creative Commons share-alike licensing system, 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.

Strong and weak copyleft

The copyleft governing a work is considered to be "stronger", to the extent that the copyleft provisions can be efficiently imposed on all kinds of derived works. "Weak copyleft" refers to licenses where not all derived works inherit the copyleft license; whether a derived work inherits or not often depends on the manner in which it was derived.

"Weak copyleft" licenses are generally used for the creation of software libraries, to allow other software to link to the library, and then be redistributed without the legal requirement for the work to be distributed under the library's copyleft license. Only changes to the weak copylefted software itself become subject to the copyleft provisions of such a license, not changes to the software that links to it. This allows programs of any license to be compiled and linked against copylefted libraries such as glibc (the GNU project's implementation of the C standard library), and then redistributed without any re-licensing required.

The most well known free software license that uses strong copyleft is the GNU General Public License. Free software licenses that use "weak" copyleft include the GNU Lesser General Public License and the Mozilla Public License. Examples of non-copyleft free software licenses include the X11 license, Apache license and the BSD licenses.

The Design Science License is a strong copyleft license that can apply to any work that is not software or documentation, such as art, music, sports photography, and video. It is hosted on the Free Software Foundation website's license list (http://www.gnu.org/licenses/license-list.html) , 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 exempts some parts of the work from the copyleft provisions, thus 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. For example, the GPL linking exception made for some software packages (see below).

Share-alike

Share-alike 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, for instance prohibiting commercial use.[15] Some permutations of the Creative Commons licenses are examples of share-alike.

Viral licensing

Copyleft licenses are sometimes referred to as **viral copyright licenses**, because any works derived from a copyleft work must themselves be copyleft when distributed. The term "General Public Virus", or "GNU Public Virus" (GPV), has a long history on the Internet, dating back to shortly after the GPL was first conceived.[16][17][18] Many BSD License advocates used the term derisively[19][20][21] 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. Microsoft has used language with this term,[22] even calling it "a cancer that attaches itself in an intellectual property sense to everything it touches".[23] The term "viral" is used as an analogy of computer viruses. According to FSF compliance engineer David Turner, it creates a misunderstanding and a fear of using copylefted free software.[24]

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.[25] 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, 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.

One should also note that on one hand this “viral” effect is a normal property of any conventional license on derived works of non-copyleft free material, on the other hand the intended effect when using BSD-licensed works as part of proprietary software. The GNU project using BSD code is in this respect not different from Microsoft using BSD code.

See also

- All rights reversed
- Anti-copyright
- Commercial use of copyleft works
- Copyright
- Creative Commons licenses
- Free content
- Free Culture movement
- Free Software Foundation
- GNU General Public License
- Open content
- Patentleft
- Permissive free software licence
- Public domain
- Share-alike
- WTFPL - a "free" licence that is absolutely non-copyleft
- HESSLA - a copyleft license which prohibits uses that violate human rights or add spyware
- CrimethInc. N©! license - a public domain license that restricts contents use by governments and corporations

Notes and references

1. ^ Mundie, Craig (May 3, 2001). "Prepared Text of Remarks by Craig Mundie, Microsoft Senior Vice President - The Commercial Software Model (http://www.microsoft.com/presspass/exec/craig/05-03sharedsource.mspx) ". New York University Stern School of Business.

http://www.microsoft.com/presspass/exec/craig/05-03sharedsource.mspx. Retrieved October 1, 2009.

2. ^ Allison, Dennis (July 1976). "Design notes for TINY BASIC". *SIGPLAN Notices* (ACM) **11** (7): pp. 25–33. doi:10.1145/987491.987494 (http://dx.doi.org/10.1145%2F987491.987494) . The ACM Special Interest Group on Programming Languages (SIGPLAN) reprinted the Tiny Basic design notes from the January 1976 Tiny BASIC Journal.
3. ^ ***a b*** Warren, Jim C. (July 1976). "Correspondence". *SIGPLAN Notices* (ACM) **11** (7): pp. 1–2. ISSN 0362-1340 (http://worldcat.org/issn/0362-1340) .
4. ^ 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. 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.
5. ^ Rauskolb, Roger (December 1976). "Dr. Wang's Palo Alto Tiny BASIC". *Interface Age* **2** (1): 92–108. 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
6. ^ Williams, Sam (March 2002). "7 (http://www.oreilly.com/openbook/freedom/ch07.html) ". *Free as in Freedom - Richard Stallman's Crusade for Free Software (http://www.oreilly.com/openbook/freedom)* . O'Reilly Media. ISBN 0-596-00287-4. http://www.oreilly.com/openbook/freedom/ch07.html.
7. ^ "Emacs General Public License (http://www.free-soft.org/gpl_history/emacs_gpl.html) ". 2001-07-05. http://www.free-soft.org/gpl_history/emacs_gpl.html. Retrieved 2008-08-23.
8. ^ ***a b*** Stallman, Richard (2008-01-21). "About the GNU Project (http://www.gnu.org/gnu/thegnuproject.html) ". Free Software Foundation. http://www.gnu.org/gnu/thegnuproject.html. Retrieved 2008-08-23.
9. ^ Biancuzzi, Federico (2005-06-30). "ESR: "We Don't Need the GPL Anymore" (http://www.onlamp.com/pub/a/onlamp/2005/06/30/esr_interview.html) ". ONLamp.com. http://www.onlamp.com/pub/a/onlamp/2005/06/30/esr_interview.html. Retrieved 2008-08-23.
10. ^ Tiemann, Michael (2006-09-18). "Licenses by Name (http://www.opensource.org/licenses/alphabetical) ". http://www.opensource.org/licenses/alphabetical. Retrieved 2008-08-23.
11. ^ "Licenses (http://www.fsf.org/licensing/licenses) ". 2008-06-05. http://www.fsf.org/licensing/licenses. Retrieved 2008-08-23.
12. ^ Raymond, Eric Steven (2002-11-09). "Licensing HOWTO (http://www.catb.org/~esr/Licensing-HOWTO.html#choosing) ". http://www.catb.org/~esr/Licensing-HOWTO.html#choosing. Retrieved 2008-08-23.
13. ^ What the GPLv3 Means for MS-Novell Agreement (http://yro.slashdot.org/comments.pl?sid=225606&cid=18272276)
14. ^ Stallman, Richard (2001-02-26). "LWN.net: RMS on the Ogg Vorbis license (http://lwn.net/2001/0301/a/rms-ov-license.php3) ". http://lwn.net/2001/0301/a/rms-ov-license.php3. Retrieved 2008-08-23. "[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."
15. ^ Rob Myers, NonCommercial Sharealike is not Copyleft (http://www.robmyers.org/weblog/2008/02/24/noncommercial-sharealike-is-not-copyleft/)
16. ^ Vixie, Paul (2006-03-06). "Re: Section 5.2 (IPR encumberance) in TAK rollover requirement draft (http://psg.com/lists/namedroppers/namedroppers.2006/msg00246.html) ". IETF Namedroppers mailing list. http://psg.com/lists/namedroppers/namedroppers.2006/msg00246.html. Retrieved 2007-04-29.
17. ^ "General Public Virus (http://catb.org/esr/jargon/oldversions/jarg221.txt) ". *Jargon File 2.2.1*. 1990-12-15. http://catb.org/esr/jargon/oldversions/jarg221.txt. Retrieved 2007-04-29.
18. ^ Hackvän, Stig (September 1999). "Reverse-engineering the GNU Public Virus — Is copyleft too much of a good thing? (http://devlinux.org/lw-gnu-published.html) ". *Linux Journal*. http://devlinux.org/lw-gnu-published.html. Retrieved 2007-04-29.
19. ^ Stewart, Bill (1998-10-08). "Re: propose: `cypherpunks license' (Re: Wanted: Twofish source code) (http://cypherpunks.venona.com/date/1998/10/msg00429.html) ". Cypherpunks mailing list. http://cypherpunks.venona.com/date/1998/10/msg00429.html. Retrieved 2007-04-29.
20. ^ Buck, Joe (2000-10-10). "Re: Using of parse tree externally (http://gcc.gnu.org/ml/gcc/2000-10/msg00198.html) ". GCC mailing list. http://gcc.gnu.org/ml/gcc/2000-10/msg00198.html. Retrieved 2007-04-29.
21. ^ Griffis, L. Adrian (2000-07-15). "The GNU Public Virus (http://themes.freshmeat.net/articles/view/172#comment-5548) ". http://themes.freshmeat.net/articles/view/172#comment-5548. Retrieved 2007-04-29.
22. ^ Mundie, Craig (2001-05-03). "Speech Transcript - Craig Mundie (http://www.microsoft.com/presspass/exec/craig/05-03sharedsource.mspx) ". New York University Stern School of Business. http://www.microsoft.com/presspass/exec/craig/05-03sharedsource.mspx. Retrieved 2008-08-23.
23. ^ Newbart, Dave (2001-06-01). "Microsoft CEO takes launch break with the Sun-Times (http://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 2001-06-15. http://web.archive.org/web/20010615205548/http://suntimes.com/output/tech/cst-fin-micro01.html.(Internet archive link)
24. ^ Byfield, Bruce (2006-08-29). "IT Manager's Journal: 10 Common Misunderstandings About the GPL (http://www.linuxtoday.com/developer/2006082902126OSHLLL) ". http://www.linuxtoday.com/developer/2006082902126OSHLLL. Retrieved 2008-08-23.
25. ^ "Frequently Asked Questions about the GNU Licenses (http://www.gnu.org/licenses/gpl-faq.html#MereAggregation) ". Free Software Foundation. 2008-06-24. http://www.gnu.org/licenses/gpl-faq.html#MereAggregation. Retrieved 2008-08-23.

External links

- What is copyleft? (http://www.gnu.org/copyleft/copyleft.html) – by Richard Stallman
- GNU's Bulletin, vol. 1 no. 4 (http://www.gnu.org/bulletins/bull4.html) – First appearance of article on *What is copyleft?*

- Freedom or Power (http://www.gnu.org/philosophy/freedom-or-power.html) by Richard Stallman and Bradley Kuhn
- Copyleft: Pragmatic Idealism (http://www.gnu.org/philosophy/pragmatic.html) – by Richard Stallman
- October 2004 interview (http://seattletimes.nwsource.com/html/businesstechnology/2002059632_linus11.html) with Linus Torvalds on commercial potential of Linux software
- MEHGLOBAL (http://www.mehglobal.com/nix) – Linux and Communism
- Eye Magazine – Copyleft and Copyright article (http://www.eyemagazine.com/opinion.php?id=117&oid=290)
- Two explanations of copyleft and its history by Richard Stallman: one in April 2006 (http://fsfeurope.org/projects/gplv3/fisl-rms-transcript.en.html#what-is-copyleft) , and one in June 2006 (http://fsfeurope.org/projects/gplv3/barcelona-rms-transcript.en.html#copyleft)
- Article: Working Without Copyleft by Bjørn Reese and Daniel Stenberg, 12/19/2001 (http://www.oreillynet.com/pub/a/policy/2001/12/12/transition.html)
- David M. Berry (2008). *Copy, Rip, Burn: The Politics of Copyleft and Open Source (http://www.amazon.com/Copy-Rip-Burn-Politics-Source/dp/0745324142)* . London:Pluto Press. http://www.amazon.com/Copy-Rip-Burn-Politics-Source/dp/0745324142.
- Steven Weber (2005). *The Success of Open Source (http://www.amazon.com/Success-Open-Source-Steven-Weber/dp/0674018583/ref=ed_oe_p)* . Harvard: Harvard University Press. http://www.amazon.com/Success-Open-Source-Steven-Weber/dp/0674018583/ref=ed_oe_p.

Retrieved from "http://en.wikipedia.org/wiki/Copyleft"
Categories: Copyright law | Copyleft licensing | Free software culture and documents | Intellectual property activism

- This page was last modified on 1 October 2009 at 18:54.
-