WIKIPEDIA
The Free Encyclopedia

Alternative terms for free software

Alternative terms for free software, such as open source, FOSS, and FLOSS, have been a recurring issue among free and open-source software users from the late 1990s onwards.[1] These terms share almost identical licence criteria and development practices.

In 1983 Richard Stallman launched the free software movement and founded the Free Software Foundation to promote the movement and to publish its own definition. Others have published alternative definitions of *free software*, notably the Debian Free Software Guidelines. In 1998, Bruce Perens and Eric S. Raymond began a campaign to market *open-source software* and founded the Open Source Initiative, which espoused different goals and a different philosophy from Stallman's.

Terms

Free software

In the 1950s to the 1990s software culture, the "free software" concept combined the nowadays differentiated software classes of public domain software, Freeware, Shareware and FOSS and was created in academia and by hobbyists and hackers.[2]

When the term "free software" was adopted by Richard Stallman in 1983, it was still ambiguously used to describe several kinds of software.[2] In February 1986 Richard Stallman formally defined "free software" with the publication of The Free Software Definition in the FSF's now-discontinued *GNU's Bulletin*[3] as software which can be used, studied, modified, and redistributed with little or no restriction, his four essential software freedoms.[3] Richard Stallman's Free Software Definition, adopted by the Free Software Foundation (FSF), defines free software as a matter of liberty, not price, and is inspired by the previous public domain software ecosystem.[4] The canonical source for the document is in the philosophy section[5] of the GNU Project website, where it is published in many languages.[6]

Open-source software

In 1998 the term "open-source software" (abbreviated "OSS") was coined as an alternative to "free software". There were several reasons for the proposal of a new term.[7] On the one hand a group from the free software ecosystem perceived the Free Software Foundation's attitude toward propagandizing the "free software" concept as "moralising and confrontational", which was also associated with the term.[8] In addition, the "available at no cost" ambiguity of the word "free" was seen as discouraging business adoption,[9] as also the historical ambiguous usage of the term "free software".[10] In a 1998 strategy session in California, "open-source software" was selected by Todd Anderson, Larry Augustin, Jon Hall, Sam Ockman, Christine Peterson, and Eric S. Raymond.[11] Richard Stallman had not been invited.[12] The session was arranged in reaction to Netscape's January 1998 announcement of a source code release for Navigator (as Mozilla). Those at the meeting described "open source" as a "replacement label" for free software,[13] and the Open Source Initiative was soon-after founded by Eric Raymond and Bruce

Perens to promote the term as part of "a marketing program for free software".[14] The Open Source Definition is used by the Open Source Initiative to determine whether a software license qualifies for the organization's insignia for open source software. The definition was based on the Debian Free Software Guidelines, written and adapted primarily by Bruce Perens.[15][16] Perens did not base his writing on the four freedoms of free software from the Free Software Foundation, which were only later available on the web.[17] According to the OSI, Stallman initially flirted with the idea of adopting the open source term.[18]

At the end of 1990s the term "open source" gained much traction in public media[19] and acceptance in the software industry in the context of the dotcom bubble and the open-source software driven Web 2.0. For instance, Duke University scholar Christopher M. Kelty described the Free Software movement prior to 1998 as fragmented and *"the term Open Source, by contrast, sought to encompass them all in one movement"*.[10] The term "open source" spread further as part of the open source movement, which inspired many successor movements including the Open content, Open-source hardware, and Open Knowledge movements. Around 2000, the success of "Open source" led several journalists to report that the earlier "Free software" term, movement, and its leader Stallman were becoming "forgotten".[20][21][22] In response, Stallman and his FSF objected to the term "open source software" and have since campaigned for the term "free software".[23][24] Due to the rejection of the term "open source software" by Stallman and FSF, the ecosystem is divided in its terminology. For example, a 2002 European Union survey revealed that 32.6% of FOSS developers associate themselves with OSS, 48% with free software, and only 19.4% are undecided or in between.[1] As both terms "free software" and "open-source software" have their proponents and critics in the FOSS ecosystems, unifying terms have been proposed; these include "software libre" (or libre software), "FLOSS" (free/libre and open-source software), and "FOSS" (or F/OSS, free and open-source software).

FOSS and F/OSS

The first known use of the phrase *free open-source software* (in short FOSS or seldom F/OSS) on Usenet was in a posting on March 18, 1998, just a month after the term *open source* itself was coined.[25] In February 2002, F/OSS appeared on a Usenet newsgroup dedicated to Amiga computer games.[26] In early 2002, MITRE used the term FOSS in what would later be their 2003 report *Use of Free and Open Source Software (FOSS) in the U.S. Department of Defense.*,[27] The European Union's institutions later also used the FOSS term while before using FLOSS,[28] as also scholar in publications.[29]

Software libre, Libre Software, Libreware

While probably used earlier (as early as the 1990s[30]) "Software libre" got broader public reception when in 1999[31] the European Commission had formed a "working group on libre software".[32] The word "libre", borrowed from the Spanish and French languages, means having liberty. This avoids the freedom-cost ambiguity of the English word "free".

FLOSS

FLOSS was used in 2001 as a project acronym by Rishab Aiyer Ghosh for *free/libre and open-source software*. Later that year, the European Commission (EC) used the phrase when they funded a study on the topic.[33][34]

Unlike "libre software", which aimed to solve the ambiguity problem, "FLOSS" aimed to avoid taking sides in the debate over whether it was better to say "free software" or to say "open-source software".

Proponents of the term point out that parts of the FLOSS acronym can be translated into other languages, for example the "F" representing *free* (English) or *frei* (German), and the "L" representing *libre* (Spanish or French), *livre* (Portuguese), or *libero* (Italian), etc. However, this term is not often used in official, non-English, documents, since the words in these languages for "free as in freedom" do not have the ambiguity problem of English's "free".

By the end of 2004, the FLOSS acronym had been used in official English documents issued by South Africa,[35] Spain,[36] and Brazil.[37] Other scholars and institutions use it too.[38]

Richard Stallman endorses the term FLOSS to refer to "open-source" and "free software" without necessarily choosing between the two camps, however, he asks people to consider supporting the "free/libre software" camp.[39][40] Stallman has suggested that the term "unfettered software" would be an appropriate, non-ambiguous replacement, but that he would not push for it because there was too much momentum and too much effort behind the term "free software".

The term "FLOSS" has come under some criticism for being counterproductive and sounding silly. For instance, Eric Raymond, co-founder of the Open Source Initiative, stated in 2009:

> Near as I can figure ... people think they'd be making an ideological commitment ... if they pick 'open source' or 'free software'. Well, speaking as the guy who promulgated 'open source' to abolish the colossal marketing blunders that were associated with the term 'free software', I think 'free software' is *less bad* than 'FLOSS'. Somebody, please, shoot this pitiful acronym through the head and put it out of our misery.[41]

Raymond quotes programmer Rick Moen as stating:

> "I continue to find it difficult to take seriously anyone who adopts an excruciatingly bad, haplessly obscure acronym associated with dental hygiene aids" and "neither term can be understood without first understanding both free software and open source, as prerequisite study."

Ownership and attachments

None of these terms, or the term "free software" itself, have been trademarked. Bruce Perens of OSI attempted to register "open source" as a service mark for OSI in the United States of America, but that attempt failed to meet the relevant trademark standards of specificity. OSI claims a trademark on "OSI Certified", and applied for trademark registration, but did not complete the paperwork. The United States Patent and Trademark Office labels it as "abandoned".[42]

While the term "free software" is associated with FSF's definition, and the term "open-source software" is associated with OSI's definition, the other terms have not been claimed by any group in particular. While the FSF's and OSI's definitions are worded quite differently the set of software that they cover is almost identical.[43][44]

All of the terms are used interchangeably, the choice of which to use is mostly political (wanting to support a certain group) or practical (thinking that one term is the clearest).

The primary difference between free software and open source is one of philosophy. According to the Free Software Foundation, "Nearly all open source software is free software. The two terms describe almost the same category of software, but they stand for views based on fundamentally different values."[43]

Licences

The choice of term has little or no impact on which licences are valid or used by the different camps, while recommendations might vary. At least until the release of the GPLv3,[45][46][47] the usage of the GPLv2 united the Open source and free software camp.[48][49] The vast majority of software referred to by all these terms is distributed under a small set of licences, all of which are unambiguously accepted by the various de facto and de jure guardians of each of these terms. The majority of the software is either one of few permissive software licenses (the BSD licenses, the MIT License, and the Apache License) or one of few copyleft licenses (the GNU General Public License v2, GPLv3, the GNU Lesser General Public License, or the Mozilla Public License).[50][51]

The Free Software Foundation (List of FSF approved software licences) and the Open Source Initiative (List of OSI approved software licences) each publish lists of licences that they accept as complying with their definitions of free software and open-source software respectively. The Open Source Initiative considers almost all free software licenses to also be open source and way around. These include the latest versions of the FSF's three main licenses, the GPLv3, the Lesser General Public License (LGPL), and the GNU Affero General Public License (AGPL).[52]

Apart from these two organisations, many more FOSS organizations publish recommendations and comments on licenses and licensing matters. The Debian project is seen by some to provide useful advice on whether particular licences comply with their Debian Free Software Guidelines. Debian does not publish a list of "approved" licences, but its judgments can be tracked by checking what licences are used by software they have allowed into their distribution.[53] In addition, the Fedora Project does provide a list of approved licences (for Fedora) based on approval of the Free Software Foundation (FSF), the Open Source Initiative (OSI), and consultation with Red Hat Legal.[54] Also, the copyfree movement, the various BSDs, the Apache, and the Mozilla Foundation all have their own points of views on licenses.

Public-domain software

There is also a class of software that is covered by the names discussed in this article, but which doesn't have a licence: software for which the source code is in the public domain. The use of such source code, and therefore the executable version, is not restricted by copyright and therefore does not need a free software licence to make it free software. However, not all countries have the same form of "public domain" regime and possibilities of dedicating works and the authors rights in the public domain.

Further, for distributors to be sure that software is released into the public domain, the usually need to see something written to confirm this. Thus even without a licence, a written note about lack of copyright and other exclusive rights often still exists (a waiver or anti-copyright notice), which can be seen as license substitute. There are also mixed forms between waiver and license, for instance the public domain like licenses CC0[55][56] and the Unlicense,[57][58] with an all permissive license as fallback in case of ineffectiveness of the waiver.

Non-English terms in anglophone regions

The free software community in some parts of India sometimes uses the term "Swatantra software" since the term "Swatantra" means free in Sanskrit, which is the ancestor of all Indo-European Languages of India, including Hindi, despite English being the lingua franca.[59] In the Philippines, "malayang software" is sometimes used. The word "libre" exists in the Filipino language, and it came from the Spanish language, but has acquired the same cost/freedom ambiguity of the English word "free".[60] According to Meranau "Free" is *KANDURI*, Diccubayadan, Libre.

See also

Free and open-source software portal

- Free software community
- Free software movement
- GNU/Linux naming controversy
- History of free software
- Open source vs. closed source
- Permissive free software licences

References

1. Free/Libre and Open Source Software: Survey and Study FLOSS Deliverable D18: FINAL REPORT – Part IV: Survey of Developers (https://web.archive.org/web/20090913045502/http://www.infonomics.nl/FLOSS/report/Final4.htm) by Rishab Aiyer Ghosh et al *"According to this ongoing discussion, one would expect a sharp polarization of the whole community of developers of non-proprietary software into two very different parties, one of Open Source developers and one of Free Software developers. However, figure 39 shows that, although there is clear evidence of these parties, still a share of almost one fifth of the whole sample does not care anyway if they belong to the one or to the other party."* (2002)
2. Shea, Tom (June 23, 1983). "Free software – Free software is a junkyard of software spare parts" (https://books.google.com/books?id=yy8EAAAAMBAJ&q=us+government+public+domain+software&pg=PA31). *InfoWorld*. Retrieved October 13, 2019. "*"In contrast to commercial software is a large and growing body of free software that exists in the public domain. Public-domain software is written by microcomputer hobbyists (also known as "hackers") many of whom are professional programmers in their work life."*"
3. "GNU's Bulletin, Volume 1 Number 1, page 8" (https://www.gnu.org/bulletins/bull1.txt). GNU.org.
4. "What is free software? / The Free Software Definition" (https://www.gnu.org/philosophy/free-sw.html). *GNU.org*. September 20, 2011. Retrieved October 23, 2011.
5. "Philosophy of the GNU Project" (https://www.gnu.org/philosophy/philosophy.html). *GNU.org*. Retrieved October 13, 2019.
6. "The Free Software Definition – Translations of this page" (https://www.gnu.org/philosophy/free-sw.html#translations). GNU.org.

7. Karl Fogel (2016). "Producing Open Source Software – How to Run a Successful Free Software Project" (http://producingoss.com/en/introduction.html#free-vs-open-source). O'Reilly Media. Retrieved April 11, 2016. "*But the problem went deeper than that. The word "free" carried with it an inescapable moral connotation: if freedom was an end in itself, it didn't matter whether free software also happened to be better, or more profitable for certain businesses in certain circumstances. Those were merely pleasant side effects of a motive that was, at its root, neither technical nor mercantile, but moral. Furthermore, the "free as in freedom" position forced a glaring inconsistency on corporations who wanted to support particular free programs in one aspect of their business, but continue marketing proprietary software in others.*"
8. OSI. "History of OSI" (http://opensource.org/history). "conferees decided it was time to dump the moralizing and confrontational attitude that had been associated with "free software" in the past and sell the idea strictly on the same pragmatic, business-case grounds"
9. "Goodbye, "free software"; hello, "open source" " (http://www.catb.org/~esr/open-source.html). "The problem with it is twofold. First, ... the term "free" is very ambiguous ... Second, the term makes a lot of corporate types nervous."
10. Kelty, Christpher M. (2008). "The Cultural Significance of free Software – Two Bits" (http://twobits.net/pub/Kelty-TwoBits.pdf) (PDF). Duke University press – durham and london. p. 99. "*Prior to 1998, Free Software referred either to the Free Software Foundation (and the watchful, micromanaging eye of Stallman) or to one of thousands of different commercial, avocational, or university-research projects, processes, licenses, and ideologies that had a variety of names: sourceware, freeware, shareware, open software, public domain software, and so on. The term "open-source", by contrast, sought to encompass them all in one movement.*"
11. Michael Tiemann (September 19, 2006). "History of the OSI" (https://web.archive.org/web/20021001164015/http://www.opensource.org/docs/history.php). Archived from the original (http://www.opensource.org/history) on October 1, 2002. "The people present included Todd Anderson, Chris Peterson (of the Foresight Institute), John "maddog" Hall and Larry Augustin (both of Linux International), Sam Ockman (of the Silicon Valley Linux User's Group), Michael Tiemann, and Eric Raymond."
12. "The Saint of Free Software (page 2)" (https://web.archive.org/web/20080612105233/http://archive.salon.com/21st/feature/1998/08/cov_31feature2.html). Archived from the original (http://archive.salon.com/21st/feature/1998/08/cov_31feature2.html) on June 12, 2008. "Stallman hadn't been invited to the first such gathering of "open source" leaders, a "free software summit" held in April..."
13. Eric Raymond. "Goodbye, "free software"; hello, "open source" " (http://www.catb.org/~esr/open-source.html). "we have a problem with the term "free software" ... we came up with a replacement label we all liked: "open source"."
14. "Frequently Asked Questions" (https://web.archive.org/web/20060423094434/http://www.opensource.org/advocacy/faq.html). Open Source Initiative. Archived from the original (http://www.opensource.org/advocacy/faq.html) on April 23, 2006. "How is "open source" related to "free software"? The Open Source Initiative is a marketing program for free software."
15. "*The Open Source Definition by Bruce Perens*" (http://www.oreilly.com/catalog/opensources/book/perens.html). March 29, 1999., Open Sources: Voices from the Open Source Revolution, January 1999, ISBN 1-56592-582-3
16. "*The Open Source Definition*" (http://opensource.org/docs/osd)., The Open Source Definition according to the Open Source Initiative
17. "Slashdot.org" (http://news.slashdot.org/comments.pl?sid=1129863&cid=26875815). *News.slashdot.org*. February 16, 2009. Retrieved October 23, 2011.

18. Tiemann, Michael (September 19, 2006). "History of the OSI" (https://web.archive.org/web/20021001164015/http://www.opensource.org/docs/history.php). Open Source Initiative. Archived from the original (http://www.opensource.org/docs/history.php) on October 1, 2002. Retrieved August 23, 2008. "*We realized that the Netscape announcement had created a precious window of time within which we might finally be able to get the corporate world to listen to what we have to teach about the superiority of an open development process. We realized it was time to dump the confrontational attitude that has been associated with "free software" in the past and sell the idea strictly on the same pragmatic, business-case grounds that motivated Netscape. We brainstormed about tactics and a new label. "Open source," contributed by Chris Peterson, was the best thing we came up with. Over the next week we worked on spreading the word. Linus Torvalds gave us an all-important imprimatur :-) the following day. Bruce Perens got involved early, offering to trademark "open source" and host this web site. Phil Hughes offered us a pulpit in Linux Journal. Richard Stallman flirted with adopting the term, then changed his mind.*"
19. The Mysteries of Open Source Software: Black and White and Red All Over (https://www.computer.org/csdl/proceedings/hicss/2005/2268/07/22680196a.pdf) by Brian Fitzgerald, Pär J. Ågerfalk University of Limerick, Ireland *"Open Source software (OSS) has attracted enormous media and research attention since the term was coined in February 1998."* (2005)
20. Leander Kahney (March 5, 1999). "Linux's Forgotten Man – You have to feel for Richard Stallman" (https://web.archive.org/web/20010622041610/http://www.wired.com/news/technology/0%2C1282%2C18291%2C00.html). wired.com. Archived from the original (https://www.wired.com/news/technology/0,1282,18291,00.html) on June 22, 2001. "*Like a Russian revolutionary erased from a photograph, he is being written out of history. Stallman is the originator of the free-software movement and the GNU/Linux operating system. But you wouldn't know it from reading about LinuxWorld. Linus Torvalds got all the ink. Even the name of the operating system, to which Torvalds contributed a small but essential part, acknowledges Torvalds alone: the Stallman part – the GNU before Linux – is almost always left out. It makes Stallman mad. At a press conference during the show, one unlucky journalist thoughtlessly called it Linux and got an earful for his mistake.*"
21. "Toronto Star: Freedom's Forgotten Prophet (Richard Stallman)" (http://www.linuxtoday.com/infrastructure/2000101000421OSCY). linuxtoday.com. October 10, 2000. Retrieved March 25, 2016. "*'But if [Richard] Stallman is winning the war, he is losing the battle – for credit....Red Hat's Web site lists the major milestones in 'open source' software, beginning in the 1970s with AT&T's Unix system and jumping to Torvalds' kernel in 1991, completely bypassing Stallman. (Red Hat does, however, provide a link to the GNU Web site, but most people have no idea what it represents.)'*"
22. Nikolai Bezroukov (November 1, 2014). "Portraits of Open Source Pioneers – Part IV. Prophet" (http://www.softpanorama.org/People/Stallman/prophet.shtml#Linux_schism). Retrieved March 25, 2016. "*'And in the second part of 1998 "open source" became a standard umbrella term encompassing commercialized GPL-based software and first of all major commercial Linux distributions (Caldera, Red Hat, Slackware, Suse, etc). Still like is often is the case in religious schisms, Raymodism overtake of Stallmanism was not complete and Eric Raymond had run into his own PR problems with his unsuccessful attempt to grab an "open source" trademark, that generated a lot of resentment in the community. Later his "surprised by wealth" letter undermined his role of influential evangelist of "open source is the best economical model for the development of the software" message. He became an object of pretty nasty jokes, but that does not help RMS to restore the role of FSF.'*"
23. Stallman, Richard (1999). "Why "Free Software" is better than "Open Source" " (https://web.archive.org/web/19990224071744/https://www.gnu.org/philosophy/free-software-for-freedom.html). *gnu.org*. Archived from the original (https://www.gnu.org/philosophy/free-software-for-freedom.html) on February 24, 1999.

24. Stallman, Richard. "Why "Open Source" misses the point of Free Software" (https://www.gnu.org/philosophy/open-source-misses-the-point.html). *GNU.org*. Free Software Foundation. "Nearly all open source software is free software; the two terms describe almost the same category of software."
25. "Posting re "free open source software", 18 March 1998" (https://groups.google.com/groups/search?hl=en&lr=&safe=off&num=10&q=%22free+open+source+software%22&safe=off&qt_s=Search&as_drrb=b&as_mind=1&as_minm=1&as_miny=1981&as_maxd=1&as_maxm=6&as_maxy=1998).
26. "Using m$ products is supporting them :(" (https://groups.google.com/group/comp.sys.amiga.games/msg/4a192ea899e10c55).
27. Terry Bollinger, *Use of Free and Open-Source Software (FOSS) in the U.S. Department of Defense,* Jan 2003. (https://dodcio.defense.gov/Portals/0/Documents/FOSS/dodfoss_pdf.pdf) The MITRE Corporation, 7515 Colshire Drive, McLean VA 22101-7508.
28. European Parliament resolution of 19 January 2016 on Towards a Digital Single Market Act (2015/2147(INI)) (http://www.europarl.europa.eu/sides/getDoc.do?pubRef=-//EP//TEXT+TA+P8-TA-2016-0009+0+DOC+XML+V0//EN) on www.europarl.europa.eu
29. Free and Open-Source Software (http://www.technollama.co.uk/wp-content/uploads/2009/04/EW_chap11.pdf) by ANDRÉS GUADAMUZ (2009)
30. González-Barahona, Jesús M. (September 14, 2000). "Quo vadis, libre software?" (https://web.archive.org/web/20041207003919/http://sinetgy.org/jgb/articulos/libre-software-origin/) (v0.8.1, work in progress). Archived from the original (http://sinetgy.org/jgb/articulos/libre-software-origin/) on December 7, 2004. Retrieved October 21, 2023.
31. "Programme: Free software - Behind the scenes" (https://web.archive.org/web/20000517124358/http://www.ist99.fi:80/programme/free.html). *ist99.fi (Information Society Technology Conference 1999 in Helsinki)*. Archived from the original (http://www.ist99.fi:80/programme/free.html) on May 17, 2000. Retrieved October 21, 2023.
32. "European working group on libre software" (https://web.archive.org/web/20000621172527/http://eu.conecta.it/). *eu.conecta.it*. April 26, 2000. Archived from the original (http://eu.conecta.it/) on June 21, 2000. Retrieved October 21, 2023.
33. "Free/Libre and Open Source Software: Survey and Study" (https://web.archive.org/web/20110707061843/http://www.flossproject.org/). Archived from the original (http://flossproject.org/) on July 7, 2011.
34. "UNU-MERIT »" (https://www.merit.unu.edu/about-us/profile/?staff_id=24).
35. "Free/Libre and Open Source Software and Open Standards in South Africa: A Critical Issue for Addressing the Digital Divide" (https://web.archive.org/web/20040829214202/http://www.naci.org.za/floss/index.html). National Advisory Council on Innovation. Archived from the original (http://www.naci.org.za/floss/index.html) on August 29, 2004.
36. Open Source Observatory. "Open Source Case Studies: FLOSS deployment in Extremadura, Spain" (https://web.archive.org/web/20071216171951/http://ec.europa.eu/idabc/en/document/1637/470). *IDABC*. Archived from the original (http://ec.europa.eu/idabc/en/document/1637/470) on December 16, 2007. Retrieved January 10, 2024.
37. "Relatório da ONU aponta o Software Livre (FLOSS) como melhor" (https://web.archive.org/web/20090210083211/http://www.softwarelivre.org/news/1727). Archived from the original (http://www.softwarelivre.org/news/1727) on February 10, 2009.
38. Karopka, T; Schmuhl, H; Demski, H (2014). "Free/Libre open source software in health care: a review" (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3950260). *Healthc Inform Res*. **20** (1): 11–22. doi:10.4258/hir.2014.20.1.11 (https://doi.org/10.4258%2Fhir.2014.20.1.11). PMC 3950260 (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3950260). PMID 24627814 (https://pubmed.ncbi.nlm.nih.gov/24627814).
39. "Interview with Richard Stallman, Edinburgh, 2004" (https://www.gnu.org/philosophy/rms-interview-edinburgh.html). Free Software Foundation. May 27, 2004.

40. Stallman, Richard. "FLOSS and FOSS" (https://www.gnu.org/philosophy/floss-and-foss.en.html). *GNU.org*. Free Software Foundation. Retrieved February 14, 2015.
41. Please forget to FLOSS (http://esr.ibiblio.org/?p=882) Armed and Dangerous, Eric S. Raymond, 26 March 2009
42. "Trademark Electronic Search System (TESS)" (http://tess2.uspto.gov/). "(Direct link not possible, site search required) Word Mark: OSI CERTIFIED ... Goods and Services: (ABANDONED) IC A . US A . G & S: software licensed under open-source licenses. ... Serial Number: 76020694 ... Owner: (APPLICANT) Open Source Initiative ... Live/Dead Indicator: DEAD"
43. Stallman, Richard. "Why "Open Source" misses the point of Free Software" (https://www.gnu.org/philosophy/open-source-misses-the-point.html). *GNU.org*. Free Software Foundation. Retrieved April 20, 2023. "The terms "free software" and "open source" stand for almost the same range of programs. However, they say deeply different things about those programs, based on different values."
44. "Innovation Goes Public" (https://web.archive.org/web/20081228023036/http://www.perens.com/works/speeches/InnovationGoesPublic/Speech/1.html). Archived from the original (http://perens.com/works/speeches/InnovationGoesPublic/Speech/1.html) on December 28, 2008. "(javascript slide #3) When I say "Open Source", I mean the same thing as Free Software."
45. Kerner, Sean Michael (January 8, 2008). "Torvalds Still Keen on GPLv2" (http://www.internetnews.com/dev-news/article.php/3720371/Torvalds+Still+Keen+On+GPLv2.htm). internetnews.com. Retrieved February 12, 2015. "*"In some ways, Linux was the project that really made the split clear between what the FSF is pushing which is very different from what open source and Linux has always been about, which is more of a technical superiority instead of a – this religious belief in freedom," Torvalds told Zemlin. So, the GPL Version 3 reflects the FSF's goals and the GPL Version 2 pretty closely matches what I think a license should do and so right now, Version 2 is where the kernel is."*"
46. McDougall, Paul (July 10, 2007). "Linux Creator Calls GPLv3 Authors 'Hypocrites' As Open Source Debate Turns Nasty" (https://web.archive.org/web/20080413091038/http://www.informationweek.com/blog/main/archives/2007/07/linux_creator_c.html). informationweek.com. Archived from the original (http://www.informationweek.com/blog/main/archives/2007/07/linux_creator_c.html) on April 13, 2008. Retrieved February 12, 2015. "*the latest sign of a growing schism in the open source community between business-minded developers like Torvalds and free software purists.*"
47. Landley, Rob. "CELF 2013 Toybox talk" (http://landley.net/talks/celf-2013.txt). landley.net. Retrieved August 21, 2013. "*GPLv3 broke "the" GPL into incompatible forks that can't share code. [...] FSF expected universal compliance, but hijacked lifeboat clause when boat wasn't sinking.[...]*"
48. Byfield, Bruce (November 22, 2011). "7 Reasons Why Free Software Is Losing Influence: Page 2" (http://www.datamation.com/open-source/7-reasons-why-free-software-is-losing-influence_2.html). Datamation.com. Retrieved August 23, 2013. "*At the time, the decision seemed sensible in the face of a deadlock. But now, GPLv2 is used for 42.5% of free software, and GPLv3 for less than 6.5%, according to Black Duck Software.*"

49. James E.J. Bottomley; Mauro Carvalho Chehab; Thomas Gleixner; Christoph Hellwig; Dave Jones; Greg Kroah-Hartman; Tony Luck; Andrew Morton; Trond Myklebust; David Woodhouse (September 15, 2006). "Kernel developers' position on GPLv3 – The Dangers and Problems with GPLv3" (https://lwn.net/Articles/200422/). LWN.net. Retrieved March 11, 2015. "The current version (Discussion Draft 2) of GPLv3 on first reading fails the necessity test of section 1 on the grounds that there's no substantial and identified problem with GPLv2 that it is trying to solve. However, a deeper reading reveals several other problems with the current FSF draft: 5.1 DRM Clauses [...] 5.2 Additional Restrictions Clause [...] 5.3 Patents Provisions [...]since the FSF is proposing to shift all of its projects to GPLv3 and apply pressure to every other GPL licensed project to move, we foresee the release of GPLv3 portends the Balkanisation of the entire Open Source Universe upon which we rely."
50. "Top 20 licenses" (https://web.archive.org/web/20160719043600/https://www.blackducksoftware.com/top-open-source-licenses). Black Duck Software. November 19, 2015. Archived from the original (http://www.blackducksoftware.com/resources/data/top-20-licenses) on July 19, 2016. Retrieved November 19, 2015. "*1. MIT license 24%, 2. GNU General Public License (GPL) 2.0 23%, 3. Apache License 16%, 4. GNU General Public License (GPL) 3.0 9%, 5. BSD License 2.0 (3-clause, New or Revised) License 6%, 6. GNU Lesser General Public License (LGPL) 2.1 5%, 7. Artistic License (Perl) 4%, 8. GNU Lesser General Public License (LGPL) 3.0 2%, 9. Microsoft Public License 2%, 10. Eclipse Public License (EPL) 2%*"
51. Balter, Ben (March 9, 2015). "Open source license usage on GitHub.com" (https://github.com/blog/1964-license-usage-on-github-com). github.com. Retrieved November 21, 2015. "1 MIT 44.69%, 2 Other 15.68%, 3 GPLv2 12.96%, 4 Apache 11.19%, 5 GPLv3 8.88%, 6 BSD 3-clause 4.53%, 7 Unlicense 1.87%, 8 BSD 2-clause 1.70%, 9 LGPLv3 1.30%, 10 AGPLv3 1.05%"
52. "Licenses by Name" (http://www.opensource.org/licenses/alphabetical). *Open Source License*. Open Source Initiative. Retrieved October 23, 2011.
53. "License information" (http://www.debian.org/legal/licenses/). Debian.
54. "Licensing" (https://web.archive.org/web/20100110110219/http://fedoraproject.org/wiki/Licensing). Fedora. Archived from the original (https://fedoraproject.org/wiki/Licensing/) on January 10, 2010.
55. "11/17: Lulan Artisans Textile Competition" (https://creativecommons.org/2009/06/17/1117-lulan-artisans-textile-competition/). June 17, 2009.
56. Validity of the Creative Commons Zero 1.0 Universal Public Domain Dedication and its usability for bibliographic metadata from the perspective of German Copyright Law (https://rd-alliance.org/sites/default/files/cc0-analysis-kreuzer.pdf) by Dr. Till Kreutzer, attorney-at-law in Berlin, Germany.
57. The unlicense a license for no license (http://ostatic.com/blog/the-unlicense-a-license-for-no-license) Archived (https://web.archive.org/web/20170122063132/http://ostatic.com/blog/the-unlicense-a-license-for-no-license) January 22, 2017, at the Wayback Machine on ostatic.com by Joe Brockmeier (2010)
58. The Unlicense (http://unlicense.org) Archived (https://web.archive.org/web/20160904002221/http://unlicense.org/) September 4, 2016, at the Wayback Machine on unlicense.org.
59. "FSF-India's homepage" (https://web.archive.org/web/20080705212713/http://www.fsf.org.in./). Archived from the original (http://www.fsf.org.in/) on July 5, 2008. "Think of it as swatantra software"
60. "Re: Free Software, some thoughts" (http://archives.free.net.ph/message/20020628.083546.ed5f57f6.en.html). "My suspicion is that if RMS were Filipino, he would have used Malayang Software to avoid the confusion regarding economics v. liberty."

External links

- Hancock, Terry. "The Jargon of Freedom: 60 Words and Phrases with Context" (https://web.archive.org/web/20100729151135/http://www.freesoftwaremagazine.com/columns/jargon_freedom_60_words_and_phrases_context) Free Software Magazine. 2010-20-24
- Berry, D M (2004). The Contestation of Code: A Preliminary Investigation into the Discourse of the Free Software and Open Software Movement, Critical Discourse Studies, Volume 1(1). (https://web.archive.org/web/20060912222653/http://opensource.mit.edu/papers/berry1.pdf)
- Differences between open-source and free software (http://www.slackbook.org/html/introduction-opensource.html) as interpreted by Slackware
- "EU study which, among other things, polled developers about terminology" (http://www.infonomics.nl/FLOSS/report/Final4.htm_Toc13908261). Retrieved January 27, 2017.
- FreeOpenSourceSoftware.org Wiki (https://freeopensourcesoftware.org/) (same as FreeLibreOpenSourceSoftware.org)
- FSF's suggested translations of free software to languages other than English (https://www.gnu.org/philosophy/fs-translations.html)
- John Stanforth, an Open Source proponent, on the differences between the Open Source Initiative and the Free Software Foundation (http://linux.omnipotent.net/article.php?article_id=12503)

Comparison of open-source and closed-source software

Free/open-source software – the source availability model used by free and open-source software (FOSS) – and closed source are two approaches to the distribution of software.

Background

Under the *closed-source* model source code is not released to the public. Closed-source software is maintained by a team who produces their product in a compiled-executable state, which is what the market is allowed access to. Microsoft, the owner and developer of Windows and Microsoft Office, along with other major software companies, have long been proponents of this business model, although in August 2010, Microsoft interoperability general manager Jean Paoli said Microsoft "loves open source" and its anti-open-source position was a mistake.[1]

The FOSS model allows for able users to view and modify a product's source code, but most of such code is not in the public domain. Common advantages cited by proponents for having such a structure are expressed in terms of trust, acceptance, teamwork and quality.[2]

A non-free license is used to limit what free software movement advocates consider to be the essential freedoms. A license, whether providing open-source code or not, that does not stipulate the "four software freedoms",[3] are not considered "free" by the free software movement. A closed source license is one that limits only the availability of the source code. By contrast a copyleft license claims to protect the "four software freedoms" by explicitly granting them and then explicitly prohibiting anyone to redistribute the package or reuse the code in it to make derivative works without including the same licensing clauses. Some licenses grant the four software freedoms but allow redistributors to remove them if they wish. Such licenses are sometimes called *permissive software licenses*.[4] An example of such a license is the FreeBSD License which allows derivative software to be distributed as non-free or closed source, as long as they give credit to the original designers.

A misconception that is often made by both proponents and detractors of FOSS is that it cannot be capitalized.[5] FOSS can and has been commercialized by companies such as Red Hat, Canonical, Mozilla, Google, IBM, Novell, Sun/Oracle, VMware and others.[6]

Commercialization

Closed-source software

The primary business model for closed-source software involves the use of constraints on what can be done with the software and the restriction of access to the original source code.[6] This can result in a form of imposed artificial scarcity on a product that is otherwise very easy to copy and redistribute. The result is that an end-user is not actually purchasing software, but purchasing the right to use the software. To this end, the source code to closed-source software is considered a trade secret by its manufacturers.

FOSS

FOSS methods, on the other hand, typically do not limit the use of software in this fashion. Instead, the revenue model is based mainly on support services. Red Hat Inc. and Canonical Ltd. are such companies that give its software away freely, but charge for support services. The source code of the software is usually given away, and pre-compiled binary software frequently accompanies it for convenience. As a result, the source code can be freely modified. However, there can be some license-based restrictions on re-distributing the software. Generally, software can be modified and re-distributed for free, as long as credit is given to the original manufacturer of the software. In addition, FOSS can generally be sold commercially, as long as the source-code is provided. There are a wide variety of free software licenses that define how a program can be used, modified, and sold commercially (see GPL, LGPL, and BSD-type licenses). FOSS may also be funded through donations.

A software philosophy that combines aspects of FOSS and proprietary software is open core software, or commercial open source software. Despite having received criticism from some proponents of FOSS,[7] it has exhibited marginal success. Examples of open core software include MySQL and VirtualBox. The MINIX operating system used to follow this business model, but came under the full terms of the BSD license after the year 2000.

Handling competition

This model has proved somewhat successful, as witnessed in the Linux community. There are numerous Linux distributions available, but a great many of them are simply modified versions of some previous version. For example, Fedora Linux, Mandriva Linux, and PCLinuxOS are all derivatives of an earlier product, Red Hat Linux. In fact, Red Hat Enterprise Linux is itself a derivative of Fedora Linux. This is an example of one vendor creating a product, allowing a third-party to modify the software, and then creating a tertiary product based on the modified version. All of the products listed above are currently produced by software service companies.

Operating systems built on the Linux kernel are available for a wider range of processor architectures than Microsoft Windows, including PowerPC and SPARC. None of these can match the sheer popularity of the x86 architecture, nevertheless they do have significant numbers of users; Windows remains unavailable for these alternative architectures, although there have been such ports of it in the past.

The most obvious complaint against FOSS revolves around the fact that making money through some traditional methods, such as the sale of the use of individual copies and patent royalty payments, is much more difficult and sometimes impractical with FOSS. Moreover, FOSS has been considered damaging to the commercial software market, evidenced in documents released as part of the Microsoft Halloween documents leak.[8][9][10]

The cost of making a copy of a software program is essentially zero, so per-use fees are perhaps unreasonable for open-source software. At one time, open-source software development was almost entirely volunteer-driven, and although this is true for many small projects, many alternative funding streams have been identified and employed for FOSS:

- Give away the program and charge for installation and support (used by many Linux distributions).
- "Commoditize complements": make a product cheaper or free so that people are more likely to purchase a related product or service you do sell.
- Cost avoidance / cost sharing: many developers need a product, so it makes sense to share development costs (this is the genesis of the X Window System and the Apache web server).
- Donations
- Crowd funding

Increasingly, FOSS is developed by commercial organizations. In 2004, Andrew Morton noted that 37,000 of the 38,000 recent patches in the Linux kernel were created by developers directly paid to develop the Linux kernel. Many projects, such as the X Window System and Apache, have had commercial development as a primary source of improvements since their inception. This trend has accelerated over time.

There are some who counter that the commercialization of FOSS is a poorly devised business model because commercial FOSS companies answer to parties with opposite agendas. On one hand commercial FOSS companies answer to volunteers developers, who are difficult to keep on a schedule, and on the other hand they answer to shareholders, who are expecting a return on their investment. Often FOSS development is not on a schedule and therefore it may have an adverse effect on a commercial FOSS company releasing software on time.[11]

Innovation

Gary Hamel counters this claim by saying that quantifying who or what is innovative is impossible.[12]

The implementation of compatible FOSS replacements for proprietary software is encouraged by the Free Software Foundation to make it possible for their users to use FOSS instead of proprietary software, for example they have listed GNU Octave, an API-compatible replacement for MATLAB, as one of their high priority projects. In the past this list contained free binary compatible Java and CLI implementations, like GNU Classpath and DotGNU. Thus even "derivative" developments are important in the opinion of many people from FOSS. However, there is no quantitative analysis, if FOSS is less innovative than proprietary software, since there are derivative/re-implementing proprietary developments, too.

Some of the largest well-known FOSS projects are either legacy code (e.g., FreeBSD or Apache) developed a long time ago independently of the free software movement, or by companies like Netscape (which open-sourced its code with the hope that they could compete better), or by companies like MySQL which use FOSS to lure customers for its more expensive licensed product. However, it is notable that most of these projects have seen major or even complete rewrites (in the case of the Mozilla and Apache 2 code, for example) and do not contain much of the original code.

Innovations have come, and continue to come, from the open-source world:

- Perl, the pioneering open-source scripting language, made popular many features, like regular expressions and associative arrays, that were unusual at the time. The newer Python language continues this innovation, with features like functional constructs and class-dictionary unification.
- dcraw is an open-source tool for decoding RAW-format images from a variety of digital cameras, which can produce better images than the closed-source tools provided by the camera vendors themselves.
- A number of laptop models are available with a particular emphasis on multimedia capabilities. While these invariably come preinstalled with a copy of Microsoft Windows, some of them[13][14] also offer an alternative "fast-boot" mode (such as Phoenix HyperSpace) based on Linux. This gets around the long time it can take to boot up Windows.
- VLC media player, Songbird, and Amarok are FOSS music players that integrate internet-based data sources to an unprecedented degree, taking song information from MusicBrainz, related track information from last.fm, album cover art from Amazon and displaying an artist's Wikipedia page within the player.
- While admittedly inspired by Mac OS X's Quartz graphics layer, Compiz Fusion has pioneered the concept of "plug in" window decorators and animation effects. Users can develop their own creative and unique effects.
- Open-source telecommunication products, such as the Asterisk PBX, have revolutionized the ICT industry.[15]
- There are substantial efforts towards the implementation of a semantic desktop in FOSS communities.
- Today's desktop environments are innovating regarding their unique idea of a Social Desktop.
- Many academic research projects release their results as FOSS.

Code quality

In 2008, the Department of Management Science and Technology in the Athens University of Economics and Business published an analysis of the FreeBSD, Linux, Solaris, and Windows operating system kernels which looked for differences between code developed using open-source and proprietary processes. The study collected metrics in the areas of file organization, code structure, code style, the use of the C preprocessor, and data organization. The aggregate results indicated that they scored comparably to each other.[16] Another study conducted by Synopsys published in 2014 found open source code to be of better quality.[17]

Security

A study done on seventeen open-source and closed-source software showed that the number of vulnerabilities existing in a piece of software is not affected by the source availability model that it uses. The study used a very simple metrics of comparing the number of vulnerabilities between the open-source and closed-source software.[18] Another study was also done by a group of professors in Northern Kentucky University on fourteen open-source web applications written in PHP. The study measured the vulnerability density in the web applications and shown that some of them had increased vulnerability density, but some of them also had decreased vulnerability density.[19]

Business models

In its 2008 Annual Report, Microsoft stated that FOSS business models challenge its license-based software model and that the firms who use these business models do not bear the cost for their software development. The company also stated in the report:[20][21]

> Some of these [open source software] firms may build upon Microsoft ideas that we provide to them free or at low royalties in connection with our interoperability initiatives. To the extent open source software gains increasing market acceptance, our sales, revenue and operating margins may decline. Open source software vendors are devoting considerable efforts to developing software that mimics the features and functionality of our products, in some cases on the basis of technical specifications for Microsoft technologies that we make available. In response to competition, we are developing versions of our products with basic functionality that are sold at lower prices than the standard versions.

There are numerous business models for open source companies which can be found in the literature.[6]

See also

Free and open-source software portal

- Linux adoption
- GNU Project
- Open system
- Vendor lock-in
- Network effect

References

1. "Microsoft: 'We Love Open Source' " (https://web.archive.org/web/20170826155901/http://www.pcworld.com/article/203923/microsoft_we_love_open_source.html). Archived from the original (https://www.pcworld.com/article/203923/microsoft_we_love_open_source.html) on 26 August 2017. Retrieved 17 September 2017.

2. The GNU Manifesto – GNU Project – Free Software Foundation (FSF) (https://www.gnu.org/gnu/manifesto.html)
3. The Free Software Definition – GNU Project – Free Software Foundation (FSF) (https://www.gnu.org/philosophy/free-sw.html)
4. Various Licenses and Comments about Them – GNU Project – Free Software Foundation (FSF) (https://www.gnu.org/licenses/license-list.html)
5. Perkins, Greg (24 August 1999). "Open Source and Capitalism" (https://web.archive.org/web/20000817054044/http://slashdot.org/articles/980824/0854256.shtml). *Slashdot*. Archived from the original (http://slashdot.org/articles/980824/0854256.shtml) on 17 August 2000. Retrieved 13 February 2014.
6. Popp, Dr. Karl Michael (2015). *Best Practices for commercial use of open source software*. Norderstedt, Germany: Books on Demand. ISBN 978-3738619096.
7. Riehle, Dirk (2009). "The Commercial Open Source Business Model" (http://dirkriehle.com/publications/2009/the-commercial-open-source-business-model/). *Value Creation in e-Business Management*. Springer Verlag. pp. 18–30.
8. "[...] the documents show that while Microsoft may be dismissive of open-source software in public, it considers it a serious competitor in private." – quote from the "Documents_I_and_II" subsection of Microsoft Halloween documents leak article
9. The "Halloween VI" document (http://www.catb.org/~esr/halloween/halloween6.html) appears to give convincing evidence that Microsoft had their reasons for trying to argue against the popularity of Linux and other Free and open-source software.
10. Bill Gates, in his reply (http://www.startupgallery.org/gallery/notesViewer.php?ii=76_4&p=5) Archived (https://web.archive.org/web/20101130115904/http://www.startupgallery.org/gallery/notesViewer.php?ii=76_4&p=5) 30 November 2010 at the Wayback Machine after the public response to his own 1976 Open letter to hobbyists, said "Unfortunately, some of the companies I have talked to about microcomputer software are reluctant to have it distributed to the hobbyist, some of whom will steal it, when [...]".
11. Integrating Open Source in Commercial Solutions (http://www.packettrap.com/pdf/white_paper_open_source.pdf)
12. "Who's really innovative?" (http://opensource.com/business/11/2/whos-really-innovative).
13. "Toshiba launches multimedia Qosmio notebooks | InfoWorld | News | 2004-07-22 | By Martyn Williams, IDG News Service" (https://web.archive.org/web/20071225191034/http://www.infoworld.com/article/04/07/22/HNqosmio_1.html). Archived from the original (http://www.infoworld.com/article/04/07/22/HNqosmio_1.html) on 25 December 2007. Retrieved 23 February 2006.
14. "PC World – Acer Readies New Notebook, Tablet PC" (https://web.archive.org/web/20080829223459/http://www.pcworld.com/news/article/0,aid,112577,00.asp). Archived from the original (http://www.pcworld.com/news/article/0,aid,112577,00.asp) on 29 August 2008. Retrieved 23 February 2006.
15. "Open Source in ICT Industry" (https://web.archive.org/web/20160925081927/http://www.ictinnovations.com/content/open-source-voice-applications). Archived from the original (http://www.ictinnovations.com/content/open-source-voice-applications) on 25 September 2016. Retrieved 19 July 2011.
16. Spinellis, Diomidis (May 2008). "A Tale of Four Kernels" (http://www.dmst.aueb.gr/dds/pubs/conf/2008-ICSE-4kernel/html/Spi08b.html). *ICSE '08: Proceedings of the 30th International Conference on Software Engineering*. Leipzig, Germany: Association for Computing Machinery. pp. 381–390. doi:10.1145/1368088.1368140 (https://doi.org/10.1145%2F1368088.1368140).
17. "Coverity Scan Report Finds Open Source Software Quality Outpaces Proprietary Code for the First Time" (http://www.coverity.com/press-releases/coverity-scan-report-finds-open-source-software-quality-outpaces-proprietary-code-for-the-first-time/). Retrieved 10 August 2014.

18. Schryen, Guido (1 May 2011). "Is Open Source Security a Myth?". *Commun. ACM*. **54** (5): 130–140. doi:10.1145/1941487.1941516 (https://doi.org/10.1145%2F1941487.1941516). ISSN 0001-0782 (https://search.worldcat.org/issn/0001-0782). S2CID 16280410 (https://api.semanticscholar.org/CorpusID:16280410).
19. Walden, J.; Doyle, M.; Welch, G.A.; Whelan, M. (1 October 2009). "Security of open source web applications". *2009 3rd International Symposium on Empirical Software Engineering and Measurement*. pp. 545–553. doi:10.1109/ESEM.2009.5314215 (https://doi.org/10.1109%2FESEM.2009.5314215). ISBN 978-1-4244-4842-5. S2CID 2613367 (https://api.semanticscholar.org/CorpusID:2613367).
20. Annual Report on Form 10-K (https://www.sec.gov/Archives/edgar/data/789019/000119312508162768/d10k.htm)
21. Microsoft's annual report: Open-source mental block | The Open Road – The Business and Politics of Open Source by Matt Asay – CNET News.com (http://news.cnet.com/8301-13505_3-10005379-16.html?hhTest=1)

Wikipedia
The Free Encyclopedia

Comparison of source-code-hosting facilities

A **source-code-hosting facility** (also known as forge software) is a file archive and web hosting facility for source code of software, documentation, web pages, and other works, accessible either publicly or privately. They are often used by open-source software projects and other multi-developer projects to maintain revision and version history, or version control. Many repositories provide a bug tracking system, and offer release management, mailing lists, and wiki-based project documentation. Software authors generally retain their copyright when software is posted to a code hosting facilities.

General information

Name	Developer	Initial release	Free server?	Free client?	Associated collaborative development environment	Notes
Assembla	Assembla, Inc.	2005	No	Unknown	Unknown	
Azure DevOps Server	Microsoft	2012[1]	No	No	Azure DevOps Services Microsoft Visual Studio	Most features are free for open source projects or teams of 5 members or less[2]
Bitbucket	Atlassian	2008	No	No	Atlassian BitBucket (https://www.atlassian.com/software/bitbucket/features) Server, JIRA (https://www.atlassian.com/software/jira/features) and Confluence (https://www.atlassian.com/software/confluence/features)	Denies service to Cuba, Iran, North Korea, Sudan, Syria[3]
CloudForge	CollabNet	2012	No	Unknown	Unknown	
Codeberg	Codeberg e.V.[4]	2019[5]	Yes	Yes	Forgejo	Codeberg e.V. is a non-profit which operates a public Forgejo-based software forge and bug tracker, and related services such as Codeberg Pages, a Weblate translation server, and CI/CD features via Woodpecker CI.
Gitea	CommitGo, Inc.[6]	2016-12[7]	Yes	Yes	Gitea	Gitea is an open-source software tool funded on Open Collective that is designed for self-hosting, but also provides a free first-party instance.
GForge	The GForge Group, Inc.[8]	2006	Partial	Yes	Cloud version – free up to 5 users. On-premises version – free up to 5 users.	GForge is free for open source projects.
GitHub	GitHub, Inc. (A subsidiary of Microsoft Corporation)	2008-04	No	Yes	Unknown	Denies service to Crimea, North Korea, Sudan, Syria[9] List of government takedown requests (https://github.com/github/gov-takedowns/)
GitLab	GitLab Inc.	2011-09[10]	Partial[11]	Yes[12]	GitLab FOSS – free software GitLab Enterprise Edition (EE) – proprietary	Denies service to Crimea, Cuba, Iran, North Korea, Sudan, Syria[13]
GNU Savannah	Free Software Foundation	2001-01	Yes	Yes	Savane	For use by projects with GPL compatible licenses, subject to staff approval. Code access review.[14]
Helix TeamHub	Perforce Software	1995	No	No	Cloud version – free up to 5 users. On-premises version requires a license.	Free cloud version has no limits on projects within 5gb storage limit. On-premises version has DevOps pipeline technology and free replicas.
Launchpad	Canonical	2004	Yes	No	Launchpad	Supports Bazaar and Git for version-controlled repository hosting.[15][16]
OSDN	OSDN K.K.	2002–04	Unknown	Yes	Unknown	For open-source projects only.[17] Ad-supported.
Ourproject.org	Comunes Collective	2002	Yes	Yes	FusionForge	For free software, free culture and free content projects.
OW2	OW2	2008	No	No	GitLab	Oriented on middleware technology.
Phabricator	Phacility, Inc.	2010	Yes	Yes	Phabricator	End of life.[18]
SEUL	Unknown	1997-05	Unknown	No	Unknown	
SourceForge	Slashdot Media	1999-11	Yes[19][20]	Yes	Apache Allura	For use by open-source projects.[21] Ad-supported. Subject to American export restrictions, so denies service to Cuba, Iran, North Korea, Sudan, Syria.[22]
Name	**Manager**	**Established**	**Server side: all free software**	**Client side: all-free JS code**	**Developed or used CDE**	**Notes**

Features

Name	Code review	Bug tracking	Web hosting	Wiki	Translation system	Shell server	Mailing list	Forum	Personal repository	Private repository	Announce	Build system	Team
Assembla	Yes[23]	Yes	Yes	Yes	Yes	No	No	No	Yes	Yes[24]	Yes	Yes	Yes
Azure DevOps Server	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Bitbucket	Yes[25]	Yes[a]	Yes[26]	Yes	No	No	No	No	Yes	Yes[b]	No	Yes[27]	Yes
Buddy	Yes	Yes	No	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes[d]	Yes
CloudForge	Unknown	Yes	Yes	Yes	No	No	No	No	Unknown	Unknown	Unknown	Unknown	Unknown
Codeberg	Yes	Yes	Yes	Yes	Yes	No	No	No	Yes	Yes[e][29]	Unknown	Yes[30]	Yes
GForge	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Gitea	Yes	Yes	No	Yes	No	No	No	No	Yes	Yes	Unknown	Yes[31]	Yes
GitHub	Yes[32]	Yes[33][f]	Yes[34]	Yes	No	No	No	No	Yes	Yes	Yes	Yes[35]	Yes
GitLab	Yes[36]	Yes	Yes[37]	Yes	No	No	No	No	Yes	Yes	Yes	Yes[38]	Yes
GNU Savannah	Yes[40]	Yes	Yes	No	No	Yes	Yes	No[41]	No	No	Yes	No	Yes
Helix TeamHub	Yes[42]	Yes	No	Yes	No	No	Yes	Yes	Yes	Yes	No	Yes, with hooks. Jenkins, TeamCity, etc.	No
Kallithea	Yes	No	Yes	No	No	Unknown	No	No	Yes	Yes	No	No	Yes
Launchpad	Yes	Yes	No	No	Yes	No	Yes	No	Yes	Yes[h]	Yes	Yes[i]	Yes
OSDN	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	No	Yes	No	Yes
Ourproject.org	Unknown	Yes	Yes	Yes	No	Unknown	Yes	Yes	Unknown	Unknown	Unknown	Unknown	Unknown
Phabricator	Yes	Yes	Yes	Yes	Unknown	Yes	Unknown	Yes	Unknown	Unknown	Unknown	Unknown	Unknown
RhodeCode	Yes	No	Yes	No	No	Unknown	No	No	Yes	Yes	Yes	No	Yes
SourceForge	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes[j]	Yes	No	Yes
Name	**Code review**	**Bug tracking**	**Web hosting**	**Wiki**	**Translation system**	**Shell server**	**Mailing list**	**Forum**	**Personal repository**	**Private repository**	**Announce**	**Build system**	**Team**

Version control systems

Name	CVS	Git	Hg	SVN	BZR	TFVC	Arch	Perforce	Fossil
Assembla	No	Yes	No	Yes	No	No	No	Yes	No
Azure DevOps Server	No	Yes	No	No	No	Yes	No	No	No
Bitbucket	No	Yes	Until Feb 2020[c]	No	No	No	No	No	No
Buddy	No	Yes	No	No	No	No	No	No	No
CloudForge	No	Yes	No	Yes	No	No	No	No	No
Codeberg	No	Yes	No	No	No	No	No	No	No
GForge	Yes	Yes	No	Yes	No	No	No	No	No
Gitea	No	Yes	No	No	No	No	No	No	No
GitHub	No	Yes	No	Partial, until Jan 2024[43][44]	No	No	No	No	No
GitLab	No	Yes	No	No	No	No	No	No	No
GNU Savannah	Yes	Yes	Yes	Yes	Yes[45]	No	Yes	No	No
Kallithea	No	Yes	Yes	No	No	No	No	No	No
Launchpad	Import only	Yes[16][46]	Import only[47]	Import only	Yes	No	No	No	Unknown
OSDN	Yes	Yes	Yes	Yes	Yes	No	No	Unknown	Unknown
Ourproject.org	Yes	No	No	Yes	No	No	No	Unknown	Unknown
OW2	Dropped[48]	Yes	No	Dropped[48]	No	No	No	No	No
Helix TeamHub	No	Yes	Yes	Yes	No	No	No	Yes	No
Phabricator	No	Yes	Yes	Yes	No	No	No	No	No
RhodeCode	No	Yes	Yes	Yes	No	No	No	No	No
SEUL.org	Yes	No	No	Yes	No	No	No	Unknown	Unknown
SourceForge	Dropped[49]	Yes	Yes	Yes	Dropped[50]	No	No	Unknown	No[51]
Name	CVS	Git	Hg	SVN	BZR	TFVC	Arch	Perforce	Fossil

Popularity

Name	Users	Projects
Assembla	Unknown	526,581+[52]
Bitbucket	5,000,000[53]	Unknown
Buddy	Unknown	Unknown
CloudForge	Unknown	Unknown
Codeberg	110,000[54]	138,000[54]
Gitea	Unknown	Unknown
GitHub	94,000,000[55]	330,000,000[55]
GitLab	31,190,000[56]	546,000[57][k]
GNU Savannah	93,346[58]	3,848[58]
Launchpad	3,965,288[59]	40,881[60]
OSDN	54,826[61]	6,294[61]
Ourproject.org	6,353[62]	1,846[62]
OW2	Unknown	Unknown
SEUL	Unknown	Unknown
SourceForge	3,700,000[63]	500,000[63]
Name	Users	Projects

Discontinued: CodePlex, Gna!, Google Code.

Specialized hosting facilities

The following are open-source software hosting facilities that only serve a specific narrowly focused community or technology.

Name	Ad-free	CVS	Git	SVN	Arch	Notes
Drupal	Yes	No	Yes	No	No	Only for Drupal related projects.
freedesktop.org	Yes	No	Yes	No	No	Only for interoperability and shared base technology for free software desktop environments on Linux and other Unix-like operating systems, including the X Window System (X11) and cairo (graphics).
mozdev.org	Yes	Yes	Unknown	No	No	Only for Mozilla-related projects. Defunct as of July 2020.
Name	**Ad-free**	**CVS**	**Git**	**SVN**	**Arch**	**Notes**

Former hosting facilities

- Alioth (Debian) – In 2018, Alioth has been replaced by a GitLab based solution hosted on salsa.debian.org. Alioth has been finally switched off in June 2018.
- BerliOS – abandoned in April 2014[64]
- Betavine – abandoned somewhere in 2015.
- CodeHaus – shut down in May 2015[65]
- CodePlex – shut down in December 2017.
- Fedora Hosted – closed in March 2017[66]
- Gitorious – shut down in June 2015.
- Gna! – shut down in 2017.
- Google Code – closed in January 2016, all projects archived. See http://code.google.com/archive/.
- java.net – Java.net and kenai.com hosting closed April 2017.
- Phabricator – wound down operations 1 June 2021, all projects continued to be hosted with very limited support after 31 August 2021.[18]
- Tigris.org – shut down in July 2020.[67]
- Mozdev.org - shut down in July 2020.

See also

Free and open-source software portal

- Comparison of version-control software
- Distributed version control
- Forge (software)
- List of free software project directories
- List of version-control software
- Source code escrow for closed-source software
- Version control (source-code-management systems)

Notes

a. Anyone can submit Bug Reports without logging in.
b. Limited to 5 users on free plan *(see Pricing – bitbucket.org (https://bitbucket.org/product/pricing))*
c. Self hosted version is known as BitBucket Server (https://bitbucket.org/product/server) and only supports Git repositories
d. Builds are run in Docker containers
e. Codeberg is only for public open-source code, private repositories exist but are not officially permitted except as needed to support FLOSS projects
f. Requires one to log in to report a Bug.
g. Has an open source FOSS edition and commercial Enterprise Edition
h. Currently only available for security vulnerability updates
i. Ubuntu
j. Private repositories can be used to set up a project before going live. However, SourceForge requires that the project remains open source. See SourceForge Support (https://sourceforge.net/p/forge/site-support/7229/#e95c).
k. GitLab is not fundamentally organized by projects, so the count is somewhat difficult.

References

1. Somasegar, S. (31 October 2012). "Team Foundation Service is Released" (http://blogs.msdn.microsoft.com/somasegar/2012/10/31/team-foundation-service-is-released/). *blogs.MSDN.Microsoft.com*.
2. "Pricing for Azure DevOps Services" (https://azure.microsoft.com/en-us/pricing/details/devops/azure-devops-services/). *Microsoft Azure*. Retrieved 11 October 2019.
3. "Export Restrictions" (https://www.atlassian.com/legal/cloud-terms-of-service). Retrieved 19 January 2020.}}
4. "Imprint" (https://codeberg.org/Codeberg/org/src/branch/main/Imprint.md). Archived (https://web.archive.org/web/20230603071244/https://codeberg.org/Codeberg/org/src/branch/main/Imprint.md) from the original on 3 June 2023. Retrieved 30 May 2023.
5. "Codeberg.org launched" (https://blog.codeberg.org/codebergorg-launched.html). 1 January 2019. Archived (https://web.archive.org/web/20230103115652/https://blog.codeberg.org/codebergorg-launched.html) from the original on 3 January 2023. Retrieved 3 January 2023.
6. "Gitea Official Website" (https://about.gitea.com).
7. "Announcement blog post" (https://blog.gitea.io/2016/12/welcome-to-gitea/). *Gitea Blog*. 8 December 2016. Retrieved 9 May 2022.
8. "Comprehensive, Elegant, Scalable Teamwork" (https://gforge.com). GForge. Retrieved 5 April 2022.
9. "GitHub and Trade Controls" (https://help.github.com/en/github/site-policy/github-and-trade-controls). Retrieved 19 January 2020.
10. "About" (https://about.gitlab.com). *GitLab.com*. Retrieved 21 March 2019.

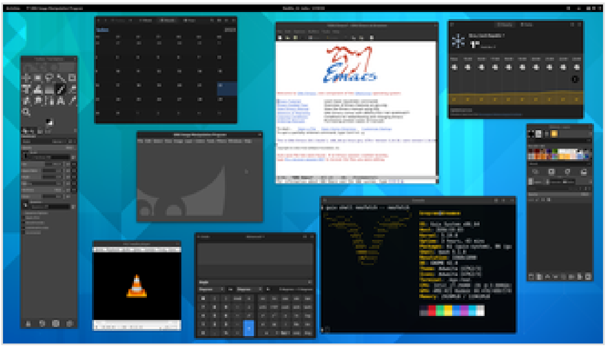
11. "GitLab FOSS – free software" (https://gitlab.com/gitlab-org/gitlab-foss). *GitLab.com*.
12. Gerwitz, Mike (20 May 2015). "GitLab, Gitorious, and Free Software" (https://about.gitlab.com/2015/05/20/gitlab-gitorious-free-software/). *GitLab.com*. GitLab. Retrieved 19 March 2016.
13. "GCP migration and Areas where google is blocked" (https://gitlab.com/gitlab-com/migration/issues/649).
14. Hosting requirements [Savannah] (http://savannah.gnu.org/register/requirements.php). Savannah.gnu.org. Retrieved 2015-04-01.
15. "Code/Git" (https://help.launchpad.net/Code/Git).
16. "Launchpad Blog" (http://blog.launchpad.net/general/git-code-hosting-beta). *Blog.launchpad.net*. 1 May 2015. Retrieved 20 May 2015.
17. "About OSDN" (https://osdn.net/projects/docs-en/wiki/About_OSDN). *OSDN*. Retrieved 22 May 2017.
18. "Phacility is Winding Down Operations" (https://admin.phacility.com/phame/post/view/11/phacility_is_winding_down_operations/). *Phacility*. Retrieved 13 July 2021.
19. "About Allura" (https://web.archive.org/web/20130820133754/http://sourceforge.net/p/allura/wiki/Allura%20Wiki/). *SourceForge*. Archived from the original (https://sourceforge.net/p/allura/wiki/Allura%20Wiki/) on 20 August 2013. Retrieved 25 August 2013.
20. "The Next SourceForge" (https://sourceforge.net/create/). *SourceForge*. Retrieved 25 August 2013.
21. "About (SourceForge)" (https://sourceforge.net/about). *SourceForge*. Retrieved 25 August 2013.
22. "Terms of Use" (https://slashdotmedia.com/terms-of-use/). *slashdotmedia.com*. SlashdotMedia. 18 February 2016. 8. Registration; Use of Secure Areas and Passwords.
23. Andy Singleton (27 March 2012). "Announcing Advanced Merge Requests for Git" (https://web.archive.org/web/20150521054411/http://blog.assembla.com/AssemblaBlog/tabid/12618/bid/80158/Announcing-Advanced-Merge-Requests-for-Git.aspx). *Blog.assembla.com*. Archived from the original (http://blog.assembla.com/assemblablog/tabid/12618/bid/80158/Announcing-Advanced-Merge-Requests-for-Git.aspx) on 21 May 2015. Retrieved 20 May 2015.
24. "Get Started for Free in 60 Seconds | Assembla Plans" (https://web.archive.org/web/20160713183349/https://www.assembla.com/plans). *Assembla.com*. Archived from the original (https://www.assembla.com/plans) on 13 July 2016. Retrieved 20 May 2015.
25. – Using Mercurial Queues And Bitbucket.org (http://ches.nausicaamedia.com/articles/technogeekery/using-mercurial-queues-and-bitbucket-org) Archived (https://web.archive.org/web/20091228083131/http://ches.nausicaamedia.com/articles/technogeekery/using-mercurial-queues-and-bitbucket-org) 28 December 2009 at the Wayback Machine
26. Publishing a Website on Bitbucket – Bitbucket – Atlassian Documentation (https://confluence.atlassian.com/display/BITBUCKET/Publishing+a+Website+on+Bitbucket) Archived (https://web.archive.org/web/20130923042859/https://confluence.atlassian.com/display/BITBUCKET/Publishing+a+Website+on+Bitbucket) 23 September 2013 at the Wayback Machine. Confluence.atlassian.com. Retrieved 2013-09-21.
27. Bitbucket Pipelines (https://bitbucket.org/product/features/pipelines)
28. Issue #11404 – Bitbucket equivalent of GitHub Releases? (BB-13572) (https://bitbucket.org/site/master/issues/11404/bitbucket-equivalent-of-github-releases-bb)
29. "Can I use private repositories for my project? | Frequently Asked Questions" (https://docs.codeberg.org/getting-started/faq/#can-i-use-private-repositories-for-my-project%3F). *Codeberg Docs*. Retrieved 8 June 2024.
30. "Using Codeberg's instance of Woodpecker CI | Frequently Asked Questions" (https://docs.codeberg.org/ci/#using-codeberg's-instance-of-woodpecker-ci). *Codeberg Docs*. Retrieved 8 June 2024.
31. "Gitea compared to other Git hosting options – Docs" (https://docs.gitea.io/en-us/comparison/).
32. "Pull Requests 2.0 · GitHub" (https://github.com/blog/712-pull-requests-2-0). *Github.com*. 31 August 2010. Retrieved 20 May 2015.
33. no file attachments, but images can be embedded GitHub Issue Tracker – GitHub (https://github.com/blog/1347-issue-attachments)
34. "GitHub Pages" (https://pages.github.com/). *GitHub*.
35. "Features • GitHub Actions" (https://github.com/features/actions). *GitHub*. Retrieved 15 May 2021.
36. "Features" (https://about.gitlab.com/features/). *GitLab*. Retrieved 14 June 2018.
37. "GitLab Pages" (https://web.archive.org/web/20160707175901/http://doc.gitlab.com/ee/pages/README.html). *GitLab*. Archived from the original (http://doc.gitlab.com/ee/pages/README.html) on 7 July 2016. Retrieved 7 March 2016.
38. "Continuous Integration" (https://web.archive.org/web/20181024103506/https://about.gitlab.com/features/gitlab-ci-cd/). *GitLab*. Archived from the original (https://about.gitlab.com/features/gitlab-ci-cd/) on 24 October 2018. Retrieved 20 May 2017.
39. "GitLab 8.2 released" (https://web.archive.org/web/20170118213054/https://about.gitlab.com/2015/11/22/gitlab-8-2-released/). *GitLab*. 22 November 2015. Archived from the original (https://about.gitlab.com/2015/11/22/gitlab-8-2-released/) on 18 January 2017. Retrieved 28 June 2017.
40. "Savannah's Maintenance Docs: How To Get Your Project Approved Quickly" (http://savannah.gnu.org/maintenance/HowToGetYourProjectApprovedQuickly). "The review we do can be lengthy and difficult for both the submitter and the reviewer. Be sure to follow these steps; if your project doesn't comply with our requirements, we will ask you to make changes to your project or register again. This ensures a level of quality for projects hosted at Savannah, and even more important, raises awareness of these legal and philosophical issues related to free software."
41. "Savannah Administration – In Depth Guide [Savannah]" (https://web.archive.org/web/20180419120709/http://savannah.nongnu.org/userguide/?file=ch03s02.html#id2469729). *Savannah.nongnu.org*. Archived from the original (http://savannah.nongnu.org/userguide/?file=ch03s02.html#id2469729) on 19 April 2018. Retrieved 20 May 2015.
42. "Code Repository Tools for Seamless Collaboration" (https://www.perforce.com/products/helix-teamhub).
43. Collaborating on GitHub with Subversion (https://github.com/blog/1178-collaborating-on-github-with-subversion). Github.com (26 June 2012). Retrieved 2015-04-01.
44. Cooper, Matt (20 January 2023). "Sunsetting Subversion support" (https://github.blog/2023-01-20-sunsetting-subversion-support/). *GitHub*. Retrieved 1 October 2023.
45. Savannah Support Request, sr #106417 (24 October 2008), *GNU Bazaar on Savannah* (http://savannah.gnu.org/support/?106417), retrieved 10 December 2008
46. "Launchpad Blog" (http://blog.launchpad.net/bazaar/git-imports). *Blog.launchpad.net*. 8 July 2009. Retrieved 20 May 2015.
47. "Launchpad Blog" (http://blog.launchpad.net/bazaar/accessing-git-subversion-and-mercurial-from-bazaar). *Blog.launchpad.net*. 29 October 2009. Retrieved 20 May 2015.
48. "Gforge decommission" (https://tc.ow2.org/view/wiki/Gforge_decommission). *OW2 Technology Council*. Retrieved 5 May 2022.
49. "SourceForge Support / Documentation / CVS" (https://sourceforge.net/p/forge/documentation/CVS/).
50. SourceForge docs for bazaar (http://sourceforge.net/p/forge/documentation/Bazaar/), Bazaar is no longer available for new projects, they only offer limited support for Bazaar for projects previously using it on the Classic SourceForge system (1 July 2013).
51. Feature Request: Fossil Repositories (https://sourceforge.net/p/forge/feature-requests/507/)
52. "Assembla Keeps Code, Tasks, and Teams Happily Together" (http://assembla.com/home). *Assembla.com*. Retrieved 6 December 2015.
53. "Bitbucket Cloud: 5 million developers and 900,000 teams" (https://blog.bitbucket.org/2016/09/07/bitbucket-cloud-5-million-developers-900000-teams/). *Bitbucket.com*. 7 September 2016. Retrieved 25 March 2017.
54. "Codeberg.org" (https://codeberg.org/about). *Codeberg.com*. Retrieved 8 June 2024.
55. "About" (https://github.com/about). *Github.com*. Retrieved 19 December 2022.
56. "Is it any good?" (https://about.gitlab.com/is-it-any-good/#gitlab-is-the-most-used-devops-tool). *GitLab*. Retrieved 7 July 2021.
57. Luke Babb (11 February 2016). "2015 was a great year at GitLab!" (https://web.archive.org/web/20160629171615/https://about.gitlab.com/2016/02/11/gitlab-retrospective/). *about.gitlab.com*. GitLab Inc. Archived from the original (https://about.gitlab.com/2016/02/11/gitlab-retrospective/) on 29 June 2016. Retrieved 28 July 2016. "564k January 2016"
58. "Statistics [Savannah]" (http://savannah.gnu.org/stats/). *Savannah.gnu.org*. Retrieved 25 December 2018.
59. People and teams in Launchpad (https://launchpad.net/people). *launchpad.net*. Retrieved 2017-10-18.

60. Projects registered in Launchpad (https://launchpad.net/projects/). *launchpad.net*. Retrieved 2017-10-18
61. "OSDN Site top" (https://osdn.net/). *OSDN*. Retrieved 18 October 2017.
62. "Welcome" (https://web.archive.org/web/20110226005437/http://ourproject.org/). *ourproject.org*. Archived from the original (http://ourproject.org) on 26 February 2011. Retrieved 18 October 2017.
63. "About" (http://sourceforge.net/about).
64. "BerliOS Developer: New berliOS portal launched" (https://web.archive.org/web/20140407013438/http://developer.berlios.de/forum/forum.php?forum_id=39220). Archived from the original (http://developer.berlios.de/forum/forum.php?forum_id=39220) on 7 April 2014.
65. "Codehaus: The once great house of code has fallen" (https://www.javaworld.com/article/2892227/codehaus-the-once-great-house-of-code-has-fallen.html). 2 March 2015. Retrieved 29 December 2019.
66. "Infrastructure/Fedorahosted-retirement – FedoraProject" (https://fedoraproject.org/wiki/Infrastructure/Fedorahosted-retirement). *fedoraproject.org*.
67. "Tigris.org: Shut down on 1-July-2020" (https://web.archive.org/web/20200701011330/http://www.tigris.org/). Archived from the original (http://www.tigris.org/) on 1 July 2020.

External links

Wikipedia
The Free Encyclopedia

Free software



GNU Guix. An example of a GNU FSDG complying free-software operating system running some representative applications. Shown are the GNOME desktop environment the GNU Emacs text editor, the GIMP image editor, and the VLC media player.

Free software, **libre software**, **libreware**[1][2] sometimes known as **freedom-respecting software** is computer software distributed under terms that allow users to run the software for any purpose as well as to study, change, distribute it and any adapted versions.[3][4][5][6] Free software is a matter of liberty, not price; all users are legally free to do what they want with their copies of a free software (including profiting from them) regardless of how much is paid to obtain the program.[7][2] Computer programs are deemed "free" if they give end-users (not just the developer) ultimate control over the software and, subsequently, over their devices.[5][8]

The right to study and modify a computer program entails that the source code—the preferred format for making changes—be made available to users of that program. While this is often called "access to source code" or "public availability", the Free Software Foundation (FSF) recommends against thinking in those terms,[9] because it might give the impression that users have an obligation (as opposed to a right) to give non-users a copy of the program.

Although the term "free software" had already been used loosely in the past and other permissive software like the Berkeley Software Distribution released in 1978 existed,[10] Richard Stallman is credited with tying it to the sense under discussion and starting the free software movement in 1983, when he launched the GNU Project: a collaborative effort to create a freedom-respecting operating system, and to revive the spirit of cooperation once prevalent among hackers during the early days of computing.[11][12]

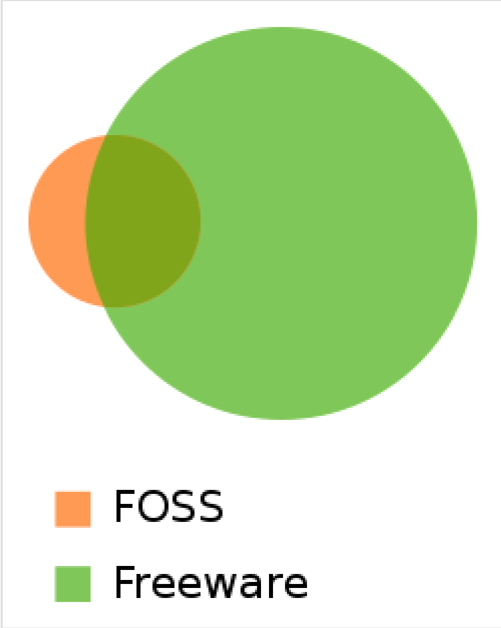
Context

Free software differs from:

- proprietary software, such as Microsoft Office, Windows, Adobe Photoshop, Facebook or FaceTime. Users cannot study, change, and share their source code.
- freeware or gratis[14] software, which is a category of proprietary software that does not require payment for basic use.

For software under the purview of copyright to be free, it must carry a software license whereby the author grants users the aforementioned rights. Software that is not covered by copyright law, such as software in the public domain, is free as long as the source code is also in the public domain, or otherwise available without restrictions.

Proprietary software uses restrictive software licences or EULAs and usually does not provide users with the source code. Users are thus legally or technically prevented from changing the software, and this results in reliance on the publisher to provide updates, help, and support. (*See also vendor lock-in and abandonware*). Users often may not reverse engineer, modify, or redistribute proprietary software.[15][16] Beyond copyright law, contracts and a lack of source code, there can exist additional obstacles keeping users from exercising freedom over a piece of software, such as software patents and digital rights management (more specifically, tivoization).[17]



This Euler diagram describes the typical relationship between freeware and free and open-source software (FOSS): According to David Rosen from Wolfire Games in 2010, open source / free software (orange) is most often gratis but not always. Freeware (green) seldom expose their source code.[13]

Free software can be a for-profit, commercial activity or not. Some free software is developed by volunteer computer programmers while other is developed by corporations; or even by both.[18][7]

Naming and differences with open source

Although both definitions refer to almost equivalent corpora of programs, the Free Software Foundation recommends using the term "free software" rather than "open-source software" (an alternative, yet similar, concept coined in 1998), because the goals and messaging are quite dissimilar. According to the Free Software Foundation, "Open source" and its associated campaign mostly focus on the technicalities of the public development model and marketing free software to businesses, while taking the ethical issue of user rights very lightly or even antagonistically.[19] Stallman has also stated that considering the practical advantages of free software is like considering the practical advantages of not being handcuffed, in that it is not necessary for an individual to consider practical reasons in order to realize that being handcuffed is undesirable in itself.[20]

The FSF also notes that "Open Source" has exactly one specific meaning in common English, namely that "you can look at the source code." It states that while the term "Free Software" can lead to two different interpretations, at least one of them is consistent with the intended meaning unlike the term "Open Source".[a] The loan adjective "libre" is often used to avoid the ambiguity of the word "free" in the English language, and the ambiguity with the older usage of "free software" as public-domain software.[10] (*See Gratis versus libre*.)

Definition and the Four Essential Freedoms of Free Software

The first formal definition of free software was published by FSF in February 1986.[21] That definition, written by Richard Stallman, is still maintained today and states that software is free software if people who receive a copy of the software have the following four freedoms.[22][23] The numbering begins with zero, not only as a spoof on the common usage of zero-based numbering in programming languages, but also because "Freedom 0" was not initially included in the list, but later added first in the list as it was considered very important.

- Freedom 0: The freedom to use the program for any purpose.
- Freedom 1: The freedom to study how the program works, and change it to make it do what you wish.
- Freedom 2: The freedom to redistribute and make copies so you can help your neighbor.
- Freedom 3: The freedom to improve the program, and release your improvements (and modified versions in general) to the public, so that the whole community benefits.

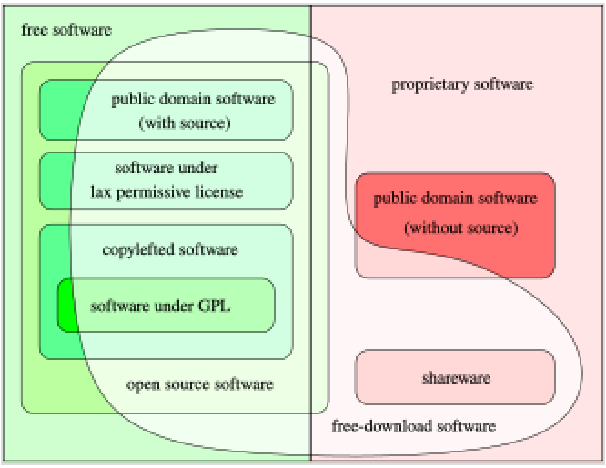


Diagram of free and nonfree software, as defined by the Free Software Foundation. Left: free software, right: proprietary software, encircled: gratis software

Freedoms 1 and 3 require source code to be available because studying and modifying software without its source code can range from highly impractical to nearly impossible.

Thus, free software means that computer users have the freedom to cooperate with whom they choose, and to control the software they use. To summarize this into a remark distinguishing *libre* (freedom) software from *gratis* (zero price) software, the Free Software Foundation says: "Free software is a matter of liberty, not price. To understand the concept, you should think of 'free' as in 'free speech', not as in 'free beer' ".[22] (*See Gratis versus libre.*)

In the late 1990s, other groups published their own definitions that describe an almost identical set of software. The most notable are *Debian Free Software Guidelines* published in 1997,[24] and *The Open Source Definition*, published in 1998.

The BSD-based operating systems, such as FreeBSD, OpenBSD, and NetBSD, do not have their own formal definitions of free software. Users of these systems generally find the same set of software to be acceptable, but sometimes see copyleft as restrictive. They generally advocate permissive free software licenses, which allow others to use the software as they wish, without being legally *forced* to provide the source code. Their view is that this permissive approach is more free. The Kerberos, X11, and Apache software licenses are substantially similar in intent and implementation.

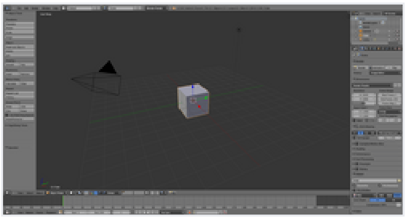
Examples

There are thousands of free applications and many operating systems available on the Internet. Users can easily download and install those applications via a package manager that comes included with most Linux distributions.

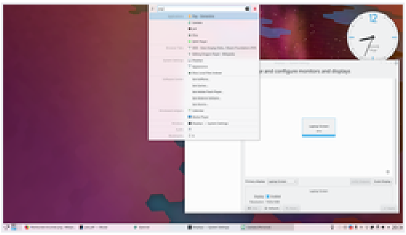
The Free Software Directory maintains a large database of free-software packages. Some of the best-known examples include Linux-libre, Linux-based operating systems, the GNU Compiler Collection and C library; the MySQL relational database; the Apache web server; and the Sendmail mail transport agent.

Other influential examples include the Emacs text editor; the GIMP raster drawing and image editor; the X Window System graphical-display system; the LibreOffice office suite; and the TeX and LaTeX typesetting systems.

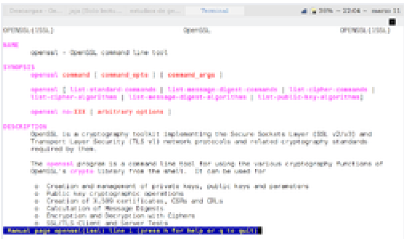
Free Software



Blender, a 3D computer graphics software.

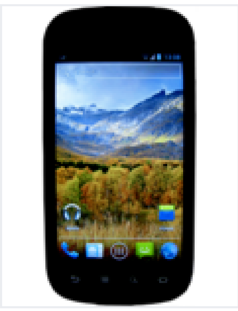


KDE Plasma desktop on Debian.

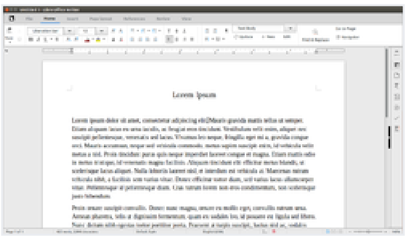


OpenSSL's manual page.

Creating a 3D car racing game using the Blender Game Engine.



Replicant smartphone OS, an Android-based system that is 100% free software.



LibreOffice is a free multi-platform office suite.

History

From the 1950s up until the early 1970s, it was normal for computer users to have the *software freedoms* associated with free software, which was typically public-domain software.[10] Software was commonly shared by individuals who used computers and by hardware manufacturers who welcomed the fact that people were making software that made their hardware useful. Organizations of users and suppliers, for example, SHARE, were formed to facilitate exchange of software. As software was often written in an interpreted language such as BASIC, the source code was distributed to use these programs. Software was also shared and distributed as printed source code (Type-in program) in computer magazines (like *Creative Computing*, *SoftSide*, *Compute!*, *Byte*, etc.) and books, like the bestseller *BASIC Computer Games*.[25] By the early 1970s, the picture changed: software costs were dramatically increasing, a growing software industry was competing with the hardware manufacturer's bundled software products (free in that the cost was included in the hardware cost), leased machines required software support while providing no revenue for software, and some customers able to better meet their own needs did not want the costs of "free" software bundled with hardware product costs. In *United States vs. IBM*, filed January 17, 1969, the government charged that bundled software was anti-competitive.[26] While some software might always be free, there would henceforth be a growing amount of software produced primarily for sale. In the 1970s and early 1980s, the software industry began using technical measures (such as only distributing binary copies of computer programs) to prevent computer users from being able to study or adapt the software applications as they saw fit. In 1980, copyright law was extended to computer programs.

In 1983, Richard Stallman, one of the original authors of the popular Emacs program and a longtime member of the hacker community at the MIT Artificial Intelligence Laboratory, announced the GNU Project, the purpose of which was to produce a completely non-proprietary Unix-compatible operating system, saying that he had become frustrated with the shift in climate surrounding the computer world and its users. In his initial declaration of the project and its purpose, he specifically cited as a motivation his opposition to being asked to agree to non-disclosure agreements and restrictive licenses which prohibited the free sharing of potentially profitable in-development software, a prohibition directly contrary to the traditional hacker ethic. Software development for the GNU operating system began in January 1984, and the Free Software Foundation (FSF) was founded in October 1985. He developed a free software definition and the concept of "copyleft", designed to ensure *software freedom* for all. Some non-software industries are beginning to use techniques similar to those used in free software development for their research and development process; scientists, for example, are looking towards more open development processes, and hardware such as microchips are beginning to be developed with specifications released under copyleft licenses (*see the OpenCores project, for instance*). Creative Commons and the free-culture movement have also been largely influenced by the free software movement.

1980s: Foundation of the GNU Project

In 1983, Richard Stallman, longtime member of the hacker community at the MIT Artificial Intelligence Laboratory, announced the GNU Project, saying that he had become frustrated with the effects of the change in culture of the computer industry and its users.[27] Software development for the GNU operating system began in January 1984, and the Free Software Foundation (FSF) was founded in October 1985.

An article outlining the project and its goals was published in March 1985 titled the *GNU Manifesto*. The manifesto included significant explanation of the GNU philosophy, *Free Software Definition* and "copyleft" ideas.

1990s: Release of the Linux kernel

The Linux kernel, started by Linus Torvalds, was released as freely modifiable source code in 1991. The first licence was a proprietary software licence. However, with version 0.12 in February 1992, he relicensed the project under the GNU General Public License.[28] Much like Unix, Torvalds' kernel attracted the attention of volunteer programmers. FreeBSD and NetBSD (both derived from 386BSD) were released as free software when the *USL v. BSDi* lawsuit was settled out of court in 1993. OpenBSD forked from NetBSD in 1995. Also in 1995, The Apache HTTP Server, commonly referred to as Apache, was released under the Apache License 1.0.

Licensing

All free-software licenses must grant users all the freedoms discussed above. However, unless the applications' licenses are compatible, combining programs by mixing source code or directly linking binaries is problematic, because of license technicalities. Programs indirectly connected together may avoid this problem.

Copyleft, a novel use of copyright law to ensure that works remain unrestricted, originates in the world of free software.[29]

The majority of free software falls under a small set of licenses. The most popular of these licenses are:[30][31]

- The MIT License
- The GNU General Public License v2 (GPLv2)
- The Apache License
- The GNU General Public License v3 (GPLv3)
- The BSD License
- The GNU Lesser General Public License (LGPL)
- The Mozilla Public License (MPL)
- The Eclipse Public License

The Free Software Foundation and the Open Source Initiative both publish lists of licenses that they find to comply with their own definitions of free software and open-source software respectively:

- List of FSF approved software licenses
- List of OSI approved software licenses

The FSF list is not prescriptive: free-software licenses can exist that the FSF has not heard about, or considered important enough to write about. So it is possible for a license to be free and not in the FSF list. The OSI list only lists licenses that have been submitted, considered and approved. All open-source licenses must meet the Open Source Definition in order to be officially recognized as open source software. Free software, on the other hand, is a more informal classification that does not rely on official recognition. Nevertheless, software licensed under licenses that do not meet the Free Software Definition cannot rightly be considered free software.

Apart from these two organizations, the Debian project is seen by some to provide useful advice on whether particular licenses comply with their Debian Free Software Guidelines. Debian does not publish a list of *approved* licenses, so its judgments have to be tracked by checking what software they have allowed into their software archives. That is summarized at the Debian web site.[32]

It is rare that a license announced as being in-compliance with the FSF guidelines does not also meet the Open Source Definition, although the reverse is not necessarily true (for example, the NASA Open Source Agreement is an OSI-approved license, but non-free according to FSF).

There are different categories of free software.

- Public-domain software: the copyright has expired, the work was not copyrighted (released without copyright notice before 1988), or the author has released the software onto the public domain with a waiver statement (in countries where this is possible). Since public-domain software lacks copyright protection, it may be freely incorporated into any work, whether proprietary or free. The FSF recommends the CC0 public domain dedication for this purpose.[33]
- Permissive licenses, also called BSD-style because they are applied to much of the software distributed with the BSD operating systems. The author retains copyright solely to disclaim warranty and require proper attribution of modified works, and permits redistribution and *any* modification, even closed-source ones.
- Copyleft licenses, with the GNU General Public License being the most prominent: the author retains copyright and permits redistribution under the restriction that all such redistribution is licensed under the same license. Additions and modifications by others must also be licensed under the same "copyleft" license whenever they are distributed with part of the original licensed product. This is also known as a *viral*, *protective*, or *reciprocal* license.

Proponents of permissive and copyleft licenses disagree on whether software freedom should be viewed as a negative or positive liberty. Due to their restrictions on distribution, not everyone considers copyleft licenses to be free.[34] Conversely, a permissive license may provide an incentive to create non-free software by reducing the cost of developing restricted software. Since this is incompatible with the spirit of software freedom, many people consider permissive licenses to be less free than copyleft licenses.[35]

Security and reliability

There is debate over the security of free software in comparison to proprietary software, with a major issue being security through obscurity. A popular quantitative test in computer security is to use relative counting of known unpatched security flaws. Generally, users of this method advise avoiding products that lack fixes for known security flaws, at least until a fix is available.

Free software advocates strongly believe that this methodology is biased by counting more vulnerabilities for the free software systems, since their source code is accessible and their community is more forthcoming about what problems exist as a part of full disclosure,[39][40] and proprietary software systems can have undisclosed societal drawbacks, such as disenfranchising less fortunate would-be users of free programs. As users can analyse and trace the source code, many more people with no commercial constraints can inspect the code and find bugs and loopholes than a corporation would find practicable. According to Richard Stallman, user access to the source code makes deploying free software with undesirable hidden spyware functionality far more difficult than for proprietary software.[41]

Some quantitative studies have been done on the subject.[42][43][44][45]

Although nearly all computer viruses only affect Microsoft Windows,[36][37][38] antivirus software such as ClamTk (shown here) is still provided for Linux and other Unix-based systems, so that users can detect malware that might infect Windows hosts.

Binary blobs and other proprietary software

In 2006, OpenBSD started the first campaign against the use of binary blobs in kernels. Blobs are usually freely distributable device drivers for hardware from vendors that do not reveal driver source code to users or developers. This restricts the users' freedom effectively to modify the software and distribute modified versions. Also, since the blobs are undocumented and may have bugs, they pose a security risk to any operating system whose kernel includes them. The proclaimed aim of the campaign against blobs is to collect hardware documentation that allows developers to write free software drivers for that hardware, ultimately enabling all free operating systems to become or remain blob-free.

The issue of binary blobs in the Linux kernel and other device drivers motivated some developers in Ireland to launch gNewSense, a Linux-based distribution with all the binary blobs removed. The project received support from the Free Software Foundation and stimulated the creation, headed by the Free Software Foundation Latin America, of the Linux-libre kernel.[46] As of October 2012, Trisquel is the most popular FSF endorsed Linux distribution ranked by Distrowatch (over 12 months).[47] While Debian is not endorsed by the FSF and does not use Linux-libre, it is also a popular distribution available without kernel blobs by default since 2011.[46]

The Linux community uses the term "blob" to refer to all nonfree firmware in a kernel whereas OpenBSD uses the term to refer to device drivers. The FSF does not consider OpenBSD to be blob free under the Linux community's definition of blob.[48]

Business model

Selling software under any free-software licence is permissible, as is commercial use. This is true for licenses with or without copyleft.[18][49][50]

Since free software may be freely redistributed, it is generally available at little or no fee. Free software business models are usually based on adding value such as customization, accompanying hardware, support, training, integration, or certification.[18] Exceptions exist however, where the user is charged to obtain a copy of the free application itself.[51]

Fees are usually charged for distribution on compact discs and bootable USB drives, or for services of installing or maintaining the operation of free software. Development of large, commercially used free software is often funded by a combination of user donations, crowdfunding, corporate contributions, and tax money. The SELinux project at the United States National Security Agency is an example of a federally funded free-software project.

Proprietary software, on the other hand, tends to use a different business model, where a customer of the proprietary application pays a fee for a license to legally access and use it. This license may grant the customer the ability to configure some or no parts of the software themselves. Often some level of support is included in the purchase of proprietary software, but additional support services (especially for enterprise applications) are usually available for an additional fee. Some proprietary software vendors will also customize software for a fee.[52]

The Free Software Foundation encourages selling free software. As the Foundation has written, "distributing free software is an opportunity to raise funds for development. Don't waste it!".[7] For example, the FSF's own recommended license (the GNU GPL) states that "[you] may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee."[53]

Microsoft CEO Steve Ballmer stated in 2001 that "open source is not available to commercial companies. The way the license is written, if you use any open-source software, you have to make the rest of your software open source."[54] This misunderstanding is based on a requirement of copyleft licenses (like the GPL) that if one distributes modified versions of software, they must release the source and use the same license. This requirement does not extend to other software from the same developer.[55] The claim of incompatibility between commercial companies and free software is also a misunderstanding. There are several large companies, e.g. Red Hat and IBM (IBM acquired RedHat in 2019),[56] which do substantial commercial business in the development of free software.

Economic aspects and adoption

Free software played a significant part in the development of the Internet, the World Wide Web and the infrastructure of dot-com companies.[57][58] Free software allows users to cooperate in enhancing and refining the programs they use; free software is a pure public good rather than a private good. Companies that contribute to free software increase commercial innovation.[59]

"We migrated key functions from Windows to Linux because we needed an operating system that was stable and reliable – one that would give us in-house control. So if we needed to patch, adjust, or adapt, we could."

Official statement of the United Space Alliance, which manages the computer systems for the International Space Station (ISS), regarding

The economic viability of free software has been recognized by large corporations such as IBM, Red Hat, and Sun Microsystems.[62][63][64][65][66] Many companies whose core business is not in the IT sector choose free software for their Internet information and sales sites, due to the lower initial capital investment and ability to freely customize the application packages. Most companies in the software business include free software in their commercial products if the licenses allow that.[18]

Free software is generally available at no cost and can result in permanently lower TCO (total cost of ownership) compared to proprietary software.[67] With free software, businesses can fit software to their specific needs by changing the software themselves or by hiring programmers to modify it for them. Free software often has no warranty, and more importantly, generally does not assign legal liability to anyone. However, warranties are permitted between any two parties upon the condition of the software and its usage. Such an agreement is made separately from the free software license.

their May 2013 decision to migrate ISS computer systems from Windows to Linux[60][61]

A report by Standish Group estimates that adoption of free software has caused a drop in revenue to the proprietary software industry by about $60 billion per year.[68] Eric S. Raymond argued that the term *free software* is too ambiguous and intimidating for the business community. Raymond promoted the term *open-source software* as a friendlier alternative for the business and corporate world.[69]

See also

Free and open-source software portal

- Definition of Free Cultural Works
- Digital rights
- Free content
- List of formerly proprietary software
- List of free software project directories
- List of free software for Web 2.0 Services
- Open format
- Open standard
- Open-source hardware
- Outline of free software
- Category:Free software lists and comparisons
- Appropriate Technology
- Sustainable Development
- Gratis versus libre

Notes

a. Access to source code is a necessary but insufficient condition, according to both the Free Software and Open Source definitions.

References

1. GNU Project. "What is free software?" (https://www.gnu.org/philosophy/free-sw.en.html). Free Software Foundation. Archived (https://web.archive.org/web/20231115065058/https://www.gnu.org/philosophy/free-sw.en.html) from the original on Nov 15, 2023.
2. "Richard Stallman" (http://www.internethalloffame.org/inductees/richard-stallman). *Internet Hall of Fame*. Retrieved 26 March 2017.
3. "Free Software Movement" (https://www.gnu.org/philosophy/free-software-intro.en.html). *GNU*. Retrieved 2021-01-11.
4. "Philosophy of the GNU Project" (https://www.gnu.org/philosophy/philosophy.en.html). *GNU*. Retrieved 2021-01-11.
5. "What is free software and why is it so important for society?" (https://www.fsf.org/about/what-is-free-software). *Free Software Foundation*. Retrieved 2021-01-11.
6. Stallman, Richard M. (2015). *Free Software Free Society: Selected Essays of Richard M. Stallman, 3rd Edition* (https://www.gnu.org/doc/fsfs3-hardcover.pdf) (PDF).
7. Selling Free Software (https://www.gnu.org/philosophy/selling.html) (GNU)
8. Stallman, Richard (27 September 1983). "Initial Announcement" (https://www.gnu.org/gnu/initial-announcement.html). *GNU Project*. Free Software Foundation.

9. Stallman, Richard. "Words to Avoid (or Use with Care) Because They Are Loaded or Confusing: Access" (https://www.gnu.org/philosophy/words-to-avoid.en.html#Access). *www.gnu.org*.
10. Shea, Tom (1983-06-23). "Free software - Free software is a junkyard of software spare parts" (https://books.google.com/books?id=yy8EAAAAMBAJ&q=us+government+public+domain+software&pg=PA31). *InfoWorld*. Retrieved 2016-02-10. "*"In contrast to commercial software is a large and growing body of free software that exists in the public domain. Public-domain software is written by microcomputer hobbyists (also known as "hackers") many of whom are professional programmers in their work life. [...] Since everybody has access to source code, many routines have not only been used but dramatically improved by other programmers."*"
11. Levi, Ran. "Richard Stallman and The History of Free Software and Open Source". *Curious Minds Podcast*.
12. "GNU" (https://cs.stanford.edu/people/eroberts/cs181/projects/open-source/gnu.htm). *cs.stanford.edu*. Retrieved 2017-10-17.
13. Rosen, David (May 16, 2010). "Open-source software is not always freeware" (http://blog.wolfire.com/2010/05/Open-source-software-is-not-always-freeware). wolfire.com. Retrieved 2016-01-18.
14. "Definition of GRATIS" (https://www.merriam-webster.com/dictionary/gratis). *www.merriam-webster.com*. Retrieved 2023-05-08.
15. Dixon, Rod (2004). *Open Source Software Law* (https://books.google.com/books?id=9b_vVPf53xcC&q=%22free+software%22+freeware&pg=PA4). Artech House. p. 4. ISBN 978-1-58053-719-3. Retrieved 2009-03-16.
16. Graham, Lawrence D. (1999). *Legal battles that shaped the computer industry* (https://books.google.com/books?id=c6IS3RnN6qAC&q=%22Legal+battles+that+shaped+the+computer+industry%22+%22from+the+beginning+of+the+computer+age%22&pg=PA175). Greenwood Publishing Group. p. 175. ISBN 978-1-56720-178-9. Retrieved 2009-03-16.
17. Sullivan, John (17 July 2008). "The Last Mile is Always the Hardest" (https://web.archive.org/web/20141028230334/http://www.fsf.org/bulletin/2008/spring/the-last-mile-is-always-the-hardest/). *fsf.org*. Archived from the original (https://www.fsf.org/bulletin/2008/spring/the-last-mile-is-always-the-hardest/) on 28 October 2014. Retrieved 29 December 2014.
18. Popp, Dr. Karl Michael (2015). *Best Practices for commercial use of open source software*. Norderstedt, Germany: Books on Demand. ISBN 978-3738619096.
19. Stallman, Richard. "Why "Open Source" misses the point of Free Software" (https://www.gnu.org/philosophy/open-source-misses-the-point.html). *GNU Project*. Free Software Foundation.
20. Stallman, Richard (2013-05-14). "The advantages of free software" (https://www.gnu.org/philosophy/practical.html). Free Software Foundation. Retrieved 2013-08-12.
21. Stallman, Richard. "What is the Free Software Foundation?" (https://www.gnu.org/bulletins/bull1.txt). *GNU's Bulletin*. Vol. 1, no. 1. p. 8.
22. Free Software Foundation. "What is free software?" (https://www.gnu.org/philosophy/free-sw.html). Retrieved 14 December 2011.
23. "Four Freedoms" (https://fsfe.org/freesoftware/#freedoms). *fsfe.org*. Retrieved March 22, 2022.
24. Perens, Bruce. "Debian's "Social Contract" with the Free Software Community" (http://fsfe.org/freesoftware/transcripts/rms-fs-2006-03-09.en.html). *debian-announce mailing list*.
25. Ahl, David. "David H. Ahl biography from Who's Who in America" (http://www.swapmeetdave.com/Ahl/DHAbio.htm). Retrieved 2009-11-23.
26. Fisher, Franklin M.; McKie, James W.; Mancke, Richard B. (1983). *IBM and the U.S. Data Processing Industry: An Economic History*. Praeger. ISBN 0-03-063059-2.

27. Williams, Sam (2002). *Free as in Freedom: Richard Stallman's Crusade for Free Software* (https://archive.org/details/freeasinfreedomr00will). O'Reilly Media. ISBN 0-596-00287-4.
28. "Release notes for Linux kernel 0.12" (https://www.kernel.org/pub/linux/kernel/Historic/old-versions/RELNOTES-0.12). Kernel.org.
29. Carver, Brian W. (2005-04-05). "Share and Share Alike: Understanding and Enforcing Open Source and Free Software Licenses". *Berkeley Technology Law Journal*. **20**: 39. SSRN 1586574 (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1586574).
30. "Top 20 licenses" (https://web.archive.org/web/20160719043600/http://www.blackducksoftware.com/resources/data/top-20-licenses). Black Duck Software. 19 November 2015. Archived from the original (http://www.blackducksoftware.com/resources/data/top-20-licenses) on 19 July 2016. Retrieved 19 November 2015. "1. MIT license 24%, 2. GNU General Public License (GPL) 2.0 23%, 3. Apache License 16%, 4. GNU General Public License (GPL) 3.0 9%, 5. BSD License 2.0 (3-clause, New or Revised) License 6%, 6. GNU Lesser General Public License (LGPL) 2.1 5%, 7. Artistic License (Perl) 4%, 8. GNU Lesser General Public License (LGPL) 3.0 2%, 9. Microsoft Public License 2%, 10. Eclipse Public License (EPL) 2%"
31. Balter, Ben (2015-03-09). "Open source license usage on GitHub.com" (https://github.com/blog/1964-license-usage-on-github-com). github.com. Retrieved 2015-11-21. " "1 MIT 44.69%, 2 Other 15.68%, 3 GPLv2 12.96%, 4 Apache 11.19%, 5 GPLv3 8.88%, 6 BSD 3-clause 4.53%, 7 Unlicense 1.87%, 8 BSD 2-clause 1.70%, 9 LGPLv3 1.30%, 10 AGPLv3 1.05%"
32. "License information" (http://www.debian.org/legal/licenses/). *Debian*. 2020-09-03.
33. "Various Licenses and Comments about Them" (https://www.gnu.org/licenses/license-list.html#CC0). *GNU Operating System*. 12 January 2022.
34. Palmer, Doug (2003-02-15). "Why Not Use the GPL? Thoughts on Free and Open-Source Software" (https://www.charvolant.org/doug/gpl/html/gpl.html). *www.charvolant.org*. Archived (https://web.archive.org/web/20200124140248/https://www.charvolant.org/doug/gpl/html/gpl.html) from the original on 2020-01-24. Retrieved 2020-01-24.
35. Stallman, Richard (2021-12-25). "The BSD License Problem" (https://www.gnu.org/licenses/bsd.en.html). Free Software Foundation. Retrieved 2024-03-29.
36. Mookhey, K.K.; Burghate, Nilesh (2005). *Linux: Security, Audit and Control Features* (https://books.google.com/books?id=-kD0sxQ0EkIC&pg=PA128). ISACA. p. 128. ISBN 9781893209787.
37. Toxen, Bob (2003). *Real World Linux Security: Intrusion Prevention, Detection, and Recovery* (https://books.google.com/books?id=_-1jwRwNaEoC&pg=PA365). Prentice Hall Professional. p. 365. ISBN 9780130464569.
38. Noyes, Katherine (Aug 3, 2010). "Why Linux Is More Secure Than Windows" (https://web.archive.org/web/20130901151841/http://www.pcworld.com/article/202452/why_linux_is_more_secure_than_windows.html). *PCWorld*. Archived from the original (https://www.pcworld.com/article/202452/why_linux_is_more_secure_than_windows.html) on 2013-09-01.
39. "Firefox more secure than MSIE after all" (http://news.cnet.com/8301-10784_3-6047727-7.html). *CNET*. News.com.
40. "The Benefits of Open Source" (http://www.albion.com/security/intro-7.html). Retrieved 19 March 2015.
41. "Transcript where Stallman explains about spyware" (http://fsfe.org/freesoftware/transcripts/rms-fs-2006-03-09.en.html). *Fsfe - Free Software Foundation Europe*.
42. David A. Wheeler: Why Open Source Software / Free Software (OSS/FS, FLOSS, or FOSS)? Look at the Numbers! (http://www.dwheeler.com/oss_fs_why.html#security) 2007
43. Michelle Delio: Linux: Fewer Bugs Than Rivals (https://www.wired.com/software/coolapps/news/2004/12/66022) Wired 2004

44. Barton P. Miller; David Koski; Cjin Pheow Lee; Vivekananda Maganty; Ravi Murthy; Ajitkumar Natarajan; Jeff Steidl (11 April 1995). Fuzz Revisited: A Re-examination of the Reliability of UNIX Utilities and Services (https://www.researchgate.net/publication/239668581) (Report). Madison, WI: University of Wisconsin: Computer Sciences Department. Archived (https://web.archive.org/web/20100621162832/http://ftp.cs.wisc.edu/pub/paradyn/technical_papers/fuzz-revisited.pdf) (PDF) from the original on 21 June 2010. "*...The reliability of the basic utilities from GNU and Linux were noticeably better than those of the commercial systems*"
45. Miller, Barton P.; Cooksey, Gregory; Moore, Fredrick (2006). "An empirical study of the robustness of MacOS applications using random testing" (https://web.archive.org/web/20100621163055/http://ftp.cs.wisc.edu/pub/paradyn/technical_papers/Fuzz-MacOS.pdf) (PDF). *Proceedings of the 1st international workshop on Random testing - RT '06*. New York, New York, USA: ACM Press. pp. 1, 2. doi:10.1145/1145735.1145743 (https://doi.org/10.1145%2F1145735.1145743). ISBN 159593457X. Archived from the original (http://ftp.cs.wisc.edu/pub/paradyn/technical_papers/Fuzz-MacOS.pdf) (PDF) on 21 June 2010. "We are back again, this time testing... Apple's Mac OS X. [...] While the results were reasonable, we were disappointed to find that the reliability was no better than that of the Linux/GNU tools tested in 1995. We were less sure what to expect when testing the GUI- based applications; the results turned out worse than we expected."
46. "Links to Other Free Software Sites - GNU Project - Free Software Foundation" (https://www.gnu.org/links/links.html#FreeGNULinuxDistributions). Retrieved 19 March 2015.
47. "DistroWatch Page Hit Ranking" (https://web.archive.org/web/20111007074633/http://distrowatch.com/stats.php?section=popularity). DistroWatch. 30 October 2012. Archived from the original (http://distrowatch.com/stats.php?section=popularity) on 7 October 2011. Retrieved 30 October 2012.
48. "Explaining Why We Don't Endorse Other Systems" (https://www.gnu.org/distros/common-distros.html).
49. "BSD license definition" (http://www.linfo.org/bsdlicense.html). Retrieved 19 March 2015.
50. "Why you should use a BSD style license for your Open Source Project" (http://www.freebsd.org/doc/en/articles/bsdl-gpl/article.html). Retrieved 19 March 2015.
51. "[libreplanet-discuss] Is there any software that is libre but not gratis" (https://lists.gnu.org/archive/html/libreplanet-discuss/2016-02/msg00227.html). *lists.gnu.org*.
52. Andy Dornan. "The Five Open Source Business Models" (https://web.archive.org/web/20091010195844/http://www.informationweek.com/blog/main/archives/2008/01/the_five_open_s.html). Archived from the original (http://www.informationweek.com/blog/main/archives/2008/01/the_five_open_s.html) on October 10, 2009.
53. GNU General Public License, section 4. (https://www.gnu.org/licenses/gpl.html#section4) gnu.org
54. "Ballmer calling open source a 'cancer', saying it's 'not available to commercial companies' " (https://web.archive.org/web/20010615205548/http://suntimes.com/output/tech/cst-fin-micro01.html). *Chicago Sun-Times*. 1 June 2001. Archived from the original (http://suntimes.com/output/tech/cst-fin-micro01.html) on 2001-06-15.
55. "Licenses" (https://choosealicense.com/licenses/). *Choose a License*. Retrieved 2022-10-19.
56. "IBM Closes Landmark Acquisition of Red Hat for $34 Billion; Defines Open, Hybrid Cloud Future" (https://newsroom.ibm.com/2019-07-09-IBM-Closes-Landmark-Acquisition-of-Red-Hat-for-34-Billion-Defines-Open-Hybrid-Cloud-Future). *IBM Newsroom*. Retrieved 2022-10-19.
57. Netcraft (14 March 2023). "Web Server Usage Survey" (http://news.netcraft.com/archives/category/web-server-survey/).

58. The Apache Software Foundation. "Apache Strategy in the New Economy" (https://web.archive.org/web/20080216004731/http://www.unc.edu/~mohrmana/apache.pdf) (PDF). Archived from the original (http://www.unc.edu/~mohrmana/apache.pdf) (PDF) on 2008-02-16.
59. Waring, Teresa; Maddocks, Philip (1 October 2005). "Open Source Software implementation in the UK public sector: Evidence from the field and implications for the future" (http://doai.io/10.1016/j.ijinfomgt.2005.06.00). *International Journal of Information Management*. **25** (5): 411–428. doi:10.1016/j.ijinfomgt.2005.06.002 (https://doi.org/10.1016%2Fj.ijinfomgt.2005.06.002). "In addition OSS's development process is creating innovative products that are reliable, secure, practical and have high usability and performance ratings. Users are now not only benefiting from the OSS revolution but also from the improved proprietary software development that is being forced upon suppliers in order to maintain competitive advantage."
60. Gunter, Joel (May 10, 2013). "International Space Station to boldly go with Linux over Windows" (https://www.telegraph.co.uk/technology/news/10049444/International-Space-Station-to-boldly-go-with-Linux-over-Windows.html). *The Telegraph*. Archived (https://ghostarchive.org/archive/20220111/https://www.telegraph.co.uk/technology/news/10049444/International-Space-Station-to-boldly-go-with-Linux-over-Windows.html) from the original on 2022-01-11.
61. Bridgewater, Adrian (May 13, 2013). "International Space Station adopts Debian Linux, drops Windows & Red Hat into airlock" (http://www.computerweekly.com/blogs/open-source-insider/2013/05/international-space-station-adopts-debian-linux-drop-windows-red-hat-into-airlock.html). *Computer Weekly*.
62. "IBM launches biggest Linux lineup ever" (https://web.archive.org/web/19991110114228/http://www.ibm.com/news/1999/03/02.phtml). IBM. 1999-03-02. Archived from the original (http://www.ibm.com/news/1999/03/02.phtml) on 1999-11-10.
63. Hamid, Farrah (2006-05-24). "IBM invests in Brazil Linux Tech Center" (https://lwn.net/Articles/185602/). LWN.net.
64. "Interview: The Eclipse code donation" (https://web.archive.org/web/20091218093727/http://www.ibm.com/developerworks/linux/library/l-erick.html). IBM. 2001-11-01. Archived from the original (http://www.ibm.com/developerworks/linux/library/l-erick.html) on 2009-12-18.
65. "Sun begins releasing Java under the GPL" (http://www.fsf.org/news/fsf-welcomes-gpl-java.html). Free Software Foundation. November 15, 2006. Retrieved 2007-09-23.
66. Rishab Aiyer Ghosh (November 20, 2006). "Study on the: Economic impact of open source software on innovation and the competitiveness of the Information and Communication Technologies (ICT) sector in the EU" (http://ec.europa.eu/enterprise/sectors/ict/files/2006-11-20-flossimpact_en.pdf) (PDF). European Union. p. 51. Retrieved 2007-01-25.
67. "Total cost of ownership of open source software: a report for the UK Cabinet Office supported by OpenForum Europe" (http://eprints.lse.ac.uk/39826/). Retrieved 19 March 2015.
68. "Open Source" (https://web.archive.org/web/20120118001419/http://standishgroup.com/newsroom/open_source.php). *Standish Newsroom*. Standishgroup.com. 2008-04-16. Archived from the original (http://standishgroup.com/newsroom/open_source.php) on 2012-01-18. Retrieved 2010-08-22.
69. Eric S. Raymond. "Eric S. Raymond's initial call to start using the term open source software, instead of free software" (http://www.catb.org/~esr/open-source.html).

Further reading

- Puckette, Miller. "Who Owns our Software?: A first-person case study." eContact (September 2009). Montréal: CEC (http://cec.sonus.ca/econtact/11_3/puckette_ownership.html)

- Hancock, Terry. "The Jargon of Freedom: 60 Words and Phrases with Context". Free Software Magazine. 2010-20-24 (http://www.freesoftwaremagazine.com/articles/jargon_freedom_60_words_and_phrases_context) Archived (https://web.archive.org/web/20120606070818/http://www.freesoftwaremagazine.com/articles/jargon_freedom_60_words_and_phrases_context) 2012-06-06 at the Wayback Machine
- Stallman, Richard M. (2010) [2002]. *Free Software Free Society: Selected Essays of Richard M. Stallman, 2nd Edition* (http://shop.fsf.org/product/free-software-free-society-2/). GNU Press. ISBN 978-0-9831592-0-9.

External links

WIKIPEDIA
The Free Encyclopedia

Gratis versus *libre*

Free Beer being sold for 500 yen at Isummit 2008. This contradicts the usual definition and instead illustrates *"Free as in freedom":* recipe and label shared openly under CC BY-SA.

The adjective ***free*** in English is commonly used in one of two meanings: "at no monetary cost" (***gratis***) or "with little or no restriction" (***libre***). This ambiguity can cause issues where the distinction is important, as it often is in dealing with laws concerning the use of information, such as copyright and patents.

The terms *gratis* and *libre* may be used to categorise intellectual property like computer programs, according to the licenses and legal restrictions that cover them, especially in the free software and open source communities, as well as the broader free culture movement. For example, they are used to distinguish "freeware" (software *gratis*) from free software (software *libre*).

Free software advocate and GNU founder Richard Stallman advocates usage of the slogan: "Think free as in free speech, not free beer."[1] This basically means: "Think free as in *libre*, not *gratis*."

Gratis

Gratis (/ˈɡrɑːtɪs/) in English is adopted from the various Romance and Germanic languages, ultimately descending from the plural ablative and dative form of the first-declension noun *grātia* in Latin. It means "free" in the sense that some goods or service is supplied without need for payment, even though it may have value.

Libre

Libre (/ˈliːbrə/) in English is adopted from the various Romance languages, ultimately descending from the Latin word *līber*; its origin is closely related to *liberty*. It denotes "the state of being free", as in "liberty" or "having freedom". The *Oxford English Dictionary* (OED) considers *libre* to be obsolete,[2] but the word has come back into limited[a] use. Unlike *gratis*, *libre* appears in few English dictionaries,[a] although there is no other English single-word adjective signifying "liberty" exclusively, without also meaning "at no monetary cost".

"Free beer" and "freedom of speech" distinction

In software development, where the marginal cost of an additional unit is zero, it is common for developers to make software available at no cost. One of the early and basic forms of this model is called freeware. With freeware, software is licensed only for personal use and the developer does not gain any monetary payment.

With the advent of the free software movement, license schemes were created to give developers more freedom in terms of code sharing, commonly called open source or free and open-source software (called FLOSS, FOSS, or F/OSS). As the English adjective *free* does not distinguish between "for free" and "liberty", the phrases "free as in freedom of speech" (*libre*, free software) and "free as in free beer" (*gratis*, freeware) were adopted. Many in the free software movement feel strongly about the *freedom* to use the software, make modifications, etc., whether or not this freely usable software is to be exchanged for money. Therefore, this distinction became important.

> "Free software" means software that respects users' freedom and community. Roughly, it means that the users have the freedom to run, copy, distribute, study, change and improve the software. Thus, "free software" is a matter of liberty, not price. To understand the concept, you should think of "free" as in "free speech," not as in "free beer". We sometimes call it "libre software," borrowing the French or Spanish word for "free" as in freedom, to show we do not mean the software is gratis.
>
> —The Free Software Foundation[3]

These phrases have become common, along with *gratis* and *libre*, in the software development and computer law fields for encapsulating this distinction.[b] The distinction is similar to the distinction made in political science between positive liberty and negative liberty. Like "free beer", positive liberty promises equal access by all without cost or regard to income, of a given good (assuming the good exists). Like "free speech", negative liberty safeguards the right to use of something (in this case, speech) without regard to whether in a given case there is a cost involved for this use.[c]

Uses in open-access academic publishing

In order to reflect real-world differences in the degree of open access, the distinction between gratis open access and libre open access was added in 2006 by Peter Suber and Stevan Harnad, two of the co-drafters of the original Budapest Open Access Initiative definition of open access publishing.[4] Gratis open access refers to online access free of charge (which Wikipedia indicates with the icon), and libre open access refers to online access free of charge plus some additional re-use rights (Wikipedia icon).[4] Libre open access is equivalent to the definition of open access in the Budapest Open Access Initiative, the Bethesda Statement on Open Access Publishing and the Berlin Declaration on Open Access to Knowledge in the Sciences and Humanities. The re-use rights of libre OA are often specified by various specific Creative Commons licenses;[5] these almost all require attribution of authorship to the original authors.[4][6]

Comparison with use in software

The original gratis/libre distinction concerns software (i.e., code), with which users can potentially do two kinds of things: 1. access and use it; and 2. modify and re-use it. "Gratis" pertains to being *able* to access and use the code, without a price-barrier, while "libre" pertains to being *allowed* to modify and re-use the code, without a permission barrier. The target content of the open access movement, however, is not software but published, peer-reviewed research journal article texts.[7]

1. **Source code accessibility and use.** For published research articles, the case for making their text accessible free for all online (Gratis) is even stronger than it is for software code, because in the case of software, some developers may wish to give their code away for free, while others may wish to sell it, whereas in the case of published research article texts, *all* their authors, without exception, give them away for free: None seek or get royalties or fees from their sale.[8] On the contrary, any access-denial to potential users means loss of potential research impact (downloads, citations) for the author's research—and researcher-authors' employment, salary, promotion and funding depends in part on the uptake and impact of their research.
2. **Source code modifiability and re-use.** For published research articles, the case for allowing text modification and re-use is much weaker than for software code, because, unlike software, the *text* of a research article is not intended for modification and re-use. (In contrast, the *content* of research articles is and always was intended for modification and re-use: that is how research progresses.) There are no copyright barriers to modifying, developing, building upon and re-using an author's ideas and findings, once they have been published, as long as the author and published source are credited—but modifications to the published text are another matter. Apart from verbatim quotation, scholarly/scientific authors are not in general interested in allowing other authors to create "mashups" of their texts. Researcher-authors are all happy to make their texts available for harvesting and indexing for search as well as data-mining, but not for re-use in altered form (without the permission of the author).

The formal analogy between open software and open access has been made,[9] along with the generalization of the gratis/libre distinction from the one field to the other.

See also

Free and open-source software portal

- Alternative terms for free software
- Comparison of free and open-source software licenses
- Free Beer (free as freedom, not gratis)
- Free software movement
- Freedom isn't free
- Gift economy
- Information wants to be free
- No such thing as a free lunch
- Open content
- Open-source license

Footnotes

a. The Onelook dictionary website (http://www.onelook.com/) finds about 5 monoglot English dictionaries including "libre"; about 30 include "gratis"
b. For example, the free software definition (https://www.gnu.org/philosophy/free-sw.html) clarifies the distinction in this way.
c. A quote from the GNU free software definition was used in a section on positive and negative liberty by Guinevere Nell in *Rediscovering Fire: Basic Economic Lessons From the Soviet Experiment*, Algora, 2010.

References

1. Lessig, Lawrence (September 2006). "Free, as in beer" (https://www.wired.com/2006/09/free-as-in-beer/). *Wired*. Archived (https://web.archive.org/web/20240725183246/https://www.wired.com/2006/09/free-as-in-beer/) from the original on 25 July 2024. Retrieved 18 March 2009.
2. OED.com (http://www.oed.com/viewdictionaryentry/Entry/107929) Archived (https://web.archive.org/web/20141112091240/http://www.oed.com/viewdictionaryentry/Entry/107929) 12 November 2014 at the Wayback Machine, OED definition of *libre*: "Obs. Of the will: Free". Access to the OED is libre, but not gratis.
3. "What is free software?" (https://www.gnu.org/philosophy/free-sw.html). *GNU Operating System*. Free Software Foundation, Inc. Archived (https://web.archive.org/web/20131014132149/http://www.gnu.org/philosophy/free-sw.html) from the original on 14 October 2013. Retrieved 11 May 2018.
4. Suber, Peter. 2008."Gratis and Libre Open Access" (http://nrs.harvard.edu/urn-3:HUL.InstRepos:4322580) Archived (https://wayback.archive-it.org/all/20170310160505/https://dash.harvard.edu/handle/1/4322580) 10 March 2017 at Archive-It. Retrieved on 2011-12-03.
5. Suber 2012, pp. 68–69
6. Suber, Peter (2012). *Open access* (https://archive.org/details/9780262517638OpenAccess). MIT Press. pp. 7 (https://archive.org/details/9780262517638OpenAccess/page/n29)-8. ISBN 9780262517638.
7. Swan, Alma (2012). "Policy guidelines for the development and promotion of open access" (https://unesdoc.unesco.org/ark:/48223/pf0000215863). *UNESCO*. Archived (https://web.archive.org/web/20190414001646/https://unesdoc.unesco.org/ark:/48223/pf0000215863) from the original on 14 April 2019. Retrieved 14 April 2019.
8. Harnad, Stevan (2003) For Whom the Gate Tolls (http://dlc.dlib.indiana.edu/dlc/bitstream/handle/10535/4406/For_Whom_the_Gate_Tolls.pdf?sequence=1) *Journal of Postgraduate Medicine* 49: 337-342
9. Suber, Peter (2008) Gratis and libre open access (http://legacy.earlham.edu/~peters/fos/newsletter/08-02-08.htm#gratis-libre) Archived (https://web.archive.org/web/20130115144052/http://www.earlham.edu/~peters/fos/newsletter/08-02-08.htm#gratis-libre) 15 January 2013 at the Wayback Machine *SPARC Open Access Newsletter*, August 2, 2008

Sources

- *Free as in Speech and Beer*, book by Darren Wershler-Henry
- Stallman's discussion of FreeAsInBeer (http://wiki.c2.com/?FreeAsInBeer)

External links

- The dictionary definition of *libre* at Wiktionary
- The dictionary definition of *gratis* at Wiktionary
- The dictionary definition of *free of charge* at Wiktionary
- The dictionary definition of *free* at Wiktionary

Wikipedia
The Free Encyclopedia

Long-term support

Long-term support (**LTS**) is a product lifecycle management policy in which a stable release of computer software is maintained for a longer period of time than the standard edition. The term is typically reserved for open-source software, where it describes a software edition that is supported for months or years longer than the software's standard edition. This is often called an **extended-support release**.

Short-term support (STS) is a term that distinguishes the support policy for the software's standard edition. STS software has a comparatively short life cycle, and may be afforded new features that are omitted from the LTS edition to avoid potentially compromising the stability or compatibility of the LTS release.[1]

Characteristics

LTS applies the tenets of reliability engineering to the software development process and software release life cycle. Long-term support extends the period of software maintenance; it also alters the type and frequency of software updates (patches) to reduce the risk, expense, and disruption of software deployment, while promoting the dependability of the software. It does not necessarily imply technical support.

At the beginning of a long-term support period, the software developers impose a feature freeze: They make patches to correct software bugs and vulnerabilities, but do not introduce new features that may cause regression. The software maintainer either distributes patches individually, or packages them in maintenance releases, point releases, or service packs. At the conclusion of the support period, the product either reaches end-of-life, or receives a reduced level of support for a period of time (e.g., high-priority security patches only).[2]

Rationale

Before upgrading software, a decision-maker might consider the risk and cost of the upgrade.[3]

As software developers add new features and fix software bugs, they may introduce new bugs or break old functionality.[4] When such a flaw occurs in software, it is called a *regression*.[4] Two ways that a software publisher or maintainer can reduce the risk of regression are to release major updates less frequently, and to allow users to test an alternate, updated version of the software.[3][5] LTS software applies these two risk-reduction strategies. The LTS edition of the software is published in parallel with the STS (short-term support) edition. Since major updates to the STS edition are published more frequently, it offers LTS users a preview of changes that might be incorporated into the LTS edition when those changes are judged to be of sufficient quality.

While using older versions of software may avoid the risks associated with upgrading, it may introduce the risk of losing support for the old software.[6] Long-term support addresses this by assuring users and administrators that the software will be maintained for a specific period of time, and that updates selected for publication will carry a significantly reduced risk of regression.[2] The maintainers of LTS software only publish updates that either have low IT risk or that reduce IT risk (such as security patches). Patches for LTS software are published with the understanding that installing them is less risky than not installing them.

Software with separate LTS versions

This table only lists software that have a specific LTS version in addition to a normal release cycle. Many projects, such as CentOS, provide a long period of support for every release.

Software	Software type	Date of first LTS release	LTS period	STS period	Notes
Blender	Computer graphics	3 June 2020 (v2.83)	2 years[7]		
ChromeOS	Operating system	March 2022	6 months	4 weeks	Chrome Enterprise and Education Help Center on Long-term Support (LTS) on ChromeOS (https://support.google.com/chrome/a/answer/11333726)
Collabora Online	Office Suite	2 June 2016 [8]	1 year	Varies	Web-based, enterprise-ready edition of LibreOffice, its STS is typically a month.[9]
Collabora Online for Desktop	Office Suite	11 May 2013 [10]	3 years		For Windows, macOS and Linux, enterprise-ready edition of LibreOffice. "LTS support for 3 years as standard, with up to 5 years if required."[11] Collabora Online for Mobile (Android, iOS and ChromeOS) have no LTS they receive rolling updates, their STS is a bit longer than Collabora Online.
Debian GNU/Linux	Linux distribution	1 June 2014[12]	5 years	3 years	LTS (no cost) is provided by "a separate group of volunteers and companies interested in making it a success."[13] Partial paid (for some versions) Extended long-term support (ELTS), for 2 extra years over the 5 of LTS, provided by Freexian.
Deno	Runtime system	November 2024 (v2.1)	6 months	4 weeks	
Django	Application framework	23 March 2012 (v1.4)	3 years[14]	16 months	
Firefox	Web browser	31 January 2012 (v10.0)	1 year	4 weeks	Mozilla's LTS term is "Extended

					Support Release" (ESR) (see Firefox#Extended Support Release).
Joomla	CMS	January 2008 (v1.5)	2 years, 3 months[15]	7 months	Since Joomla! is a web application, long-term support also implies support for legacy web browsers.
Laravel	Application framework	9 June 2015 (v5.1)[16]	3 years[17]	1 year	For LTS releases, bug fixes are provided for 2 years and security fixes are provided for 3 years. For general releases, bug fixes are provided for 6 months and security fixes are provided for 1 year.[18]
Linux kernel	Kernel	11 October 2008 (v2.6.27)	Varies, 6, 10+ years[19][20][21]	Varies	Linux kernel v2.6.16 and v2.6.27, were unofficially supported in LTS fashion[22] before a 2011 working group in the Linux Foundation started a formal Long Term Support Initiative.[23][24] The LTS support period was increased to 6 years; Linux kernel 4.4 will have 6 years of support before being taken over by the "Civil Infrastructure Platform" (CIP) project that plans to maintain it for a minimum of 10 years under "SLTS (Super Long Term Support)" (the CIP has only, for now, decided to maintain for 64-bit x86-64 and 32-bit ARM; while 64-bit ARM hardware support is also planned).[25] "The use cases CIP project is targeting have a life cycle of between 25 and 50 years." and the CIP envisions 15+

					years of support.[26][27][28]
Linux Mint	Linux distribution	8 June 2008	5 years[29]	6 months	As of version 13 the LTS period increased from three years to five, since Linux Mint derives from Ubuntu. Version 16 was the last non-LTS version.
Java	Virtual machine and runtime environment	25 September 2018 (v11)	6 years (more for older versions or depending on vendor)	6 months	Java 17 is supported for 6 years, e.g. by Microsoft. Java 8 is supported for up to 16 years by Oracle. All versions prior to Java 9 were supported for long periods of time (4 years or more).[30]
Moodle	LMS	12 May 2014 (v2.7)[31]	3 years[32]	18 months[32]	
Matomo	Web analytics	3 February 2016 (v2.16)[33][34]	≥12 months[33]	~4 weeks[35]	
Node.js	Runtime system	12 October 2015 (v4.2.0)[36]	18 months	12 months	
Symfony	Application framework	June 2013	3 years	8 months	
Tiki-wiki	Wiki/CMS	May 2009 (Tiki3)	5 years	6 months	Every third version is a Long Term Support (LTS) version.
Trisquel 7.0[37]	Linux distribution	2014-11-04	5 years	1 year	Linux-libre (kernel) 3.13, GNOME fallback 3.12 and Abrowser or GNU IceCat
TYPO3	CMS	January 2011 (v4.5 LTS)[38]	3 years (min.)	Varies	TYPO3 is a web application stewarded by the TYPO3 Association.
Ubuntu	Linux distribution	1 June 2006 (Ubuntu 6.06 LTS)[39]	5 years,[40] 10 years with ESM,[41] 12 years with Legacy Support[42]	9 months[1]	A new LTS version is released every two years. From 2006 through 2011, LTS support for the desktop was for approximately two years, and for servers five, but LTS versions are now supported for five years for both.[39][40] Extended Security

					Maintenance (ESM) is available for an additional 5 years on Ubuntu 14.04 and subsequent LTS releases[43] and Legacy Support for a further 2 years beyond ESM.[44]
Windows 10	Operating system	29 July 2015 (v10.0.10240)[45]	10 years[46]	18 months (previously 8–12 months)[46]	The Long-Term Servicing Channel (LTSC) (previously Long-Term Servicing Branch) releases of Windows 10 are supported for 10 years for mission critical machines. The LTSC release gets monthly security updates; the updates to the LTSC release bring little to no feature changes. Every 2–3 years, a new major LTSC release is published, but businesses may opt to stay on their current LTSC version until its end-of-life. The LTSC release is available only for businesses running the Windows 10 Enterprise edition. Regular consumers on the Semi-Annual Channel (SAC) get new versions of the operating system approximately every six months (previously every four months) while business customers get upgraded to new versions of SAC approximately four months after Microsoft released the SAC release for regular consumers (previously a separate release is done approximately every eight months).[46]
Windows 11	Operating system	5 October 2021 (v10.0.22000.258)	3 years (Enterprise and	2 years	"Windows 11 feature updates will

			Education editions)		release in the second half of the calendar year and will come with 24 months of support for Home, Pro, Pro for Workstations, and Pro Education editions. Windows 11 will come with 36 months of support for Enterprise and Education editions."[47]
Zabbix	Network monitoring software	21 May 2012 (2.0)[48]	5 years[49]	6 months	Dot-zero versions (3.0, 4.0, 5.0, etc) are LTS releases that have "Full support" for three years, and "Limited support" (e.g., security update) for an addition two, for a total of five years. Standard releases (5.2, 5.4, etc) are released every six months and are only supported until the next software release (plus an extra month for security fixes).[49]

1.^ The support period for Ubuntu's parent distribution, Debian, is one year after the release of the next stable version.[50][51] Since Debian 6.0 "Squeeze", LTS support (bug fixes and security patches) was added to all version releases.[52] The total LTS support time is generally around 5 years for every version.[53][54] Due to the irregular release cycle of Debian, support times might vary from that average[54] and the LTS support is done not by the Debian team but by a separate group of volunteers.[55]

See also

Free and open-source software portal

Linux portal

- Backporting
- Branching (version control)
- Computer security policy
- DevOps
- Disaster recovery plan
- Enterprise risk management
- Enterprise software
- ISO/IEC 12207, an international standard for software life-cycle processes
- Semantic Versioning
- Software quality
- Software reliability testing
- Total cost of ownership

- Vulnerability management

References

1. "Release and support cycle" (https://docs.joomla.org/release_and_support_cycle). *Joomla! Documentation*. Retrieved 2017-06-20.
2. von Hagen, William (2009). "The Ubuntu Linux Project § Ubuntu Update and Maintenance Commitments" (https://books.google.com/books?id=sl6GkVpZ5bEC&pg=PA9). *Ubuntu 8.10 Linux Bible*. Wiley Publishing. pp. 9–10. ISBN 978-0-470-29420-8.
3. Rozanski, Nick; Woods, Eóin (2012) [2011]. "The Operational Viewpoint § Functional Migration" (https://books.google.com/books?id=nXRF77-gxRkC&pg=PA395). *Software Systems Architecture: Working with Stakeholders Using Viewpoints and Perspectives* (2nd ed.). Addison-Wesley. p. 395. ISBN 978-0-321-71833-4 – via Google Books.
4. Desikan, Srinivasan; Gopalaswamy, Ramesh (2008) [2006]. "What Is Regression Testing?" (https://books.google.com/books?id=Yt2yRW6du9wC&pg=PA194). *Software Testing: Principles and Practice*. Pearson Education. p. 194. ISBN 978-81-7758-121-8 – via Google Books.
5. Black, Rex (2007). "Three Other Regression Strategies" (https://books.google.com/books?id=N6EKCAAAQBAJ&pg=PA43). *Pragmatic Software Testing: Becoming an Effective and Efficient Test Professional*. Wiley Publishing. pp. 43–44. ISBN 978-0-470-12790-2 – via Google Books.
6. May, Rod (1998). "COTS in Safety-Related Systems: Issues and an Illustration" (https://books.google.com/books?id=sIzhBwAAQBAJ&pg=PA220). In Redmill, Felix; Anderson, Tom (eds.). *Industrial Perspectives of Safety-critical Systems: Proceedings of the Sixth Safety-Critical Systems Symposium, Birmingham 1998*. Springer-Verlag. p. 220. doi:10.1007/978-1-4471-1534-2 (https://doi.org/10.1007%2F978-1-4471-1534-2). ISBN 978-3-540-76189-1. S2CID 31255136 (https://api.semanticscholar.org/CorpusID:31255136) – via Google Books.
7. Roosendaal, Ton (20 May 2020). "Long-term Support Pilot" (https://code.blender.org/2020/05/long-term-support-pilot/). Blender Developers Blog.
8. "Collabora Productivity releases Collabora Online 1.0 "Engine" for Hosters and Clouds" (https://www.collaboraoffice.com/press-releases/collabora-productivity-releases-collabora-online-cool-1-0-engine-for-hosters-and-clouds/) (Press release). Collabora Productivity. 2 June 2016. Retrieved 15 June 2016.
9. Bärwaldt, Eric (2020-09-01). "Working Together Tools for collaborative office work" (https://www.linux-magazine.com/Issues/2020/238/Collabora-and-OnlyOffice/(language)/eng-US). *Linux Magazine* (238/2020).
10. "Collabora Office repository" (https://cgit.freedesktop.org/libreoffice/help/log/?h=distro/collabora/cp-4.0). Collabora Productivity. 11 May 2013. Retrieved 18 May 2021.
11. "Collabora Office The enterprise-ready edition of LibreOffice" (https://www.collaboraoffice.com/solutions/collabora-office/). *Collabora Office*. "We offer LTS support for 3 years as standard, with up to 5 years if required. Incremental updates via MSP installers and software repositories. No installation or redeployment required."
12. "Debian -- News -- Long term support for Debian 6.0 Announced" (https://www.debian.org/News/2014/20140424.html). *www.debian.org*. Retrieved 2020-05-25.
13. "LTS - Debian Wiki" (https://wiki.debian.org/LTS). *wiki.debian.org*. Retrieved 2021-10-18.
14. Graham, Tim (25 June 2015). "Django's Roadmap" (https://www.djangoproject.com/weblog/2015/jun/25/roadmap/). *djangoproject.com*. Django Software Foundation.
15. van Geest, M.; et al. (22 August 2011). "Release and support cycle" (http://docs.joomla.org/Release_and_support_cycle). *Joomla! Documentation*. Joomla! Project Team. Retrieved 23 June 2013.

16. Barnes, Eric L. (1 May 2015). "Everything we know about Laravel 5.1 – Updated" (https://laravel-news.com/2015/04/laravel-5-1/). *laravel-news.com*. Laravel-news.
17. Barnes, Eric L. (30 April 2015). "Laravel announces v5.1 will be LTS" (https://laravel-news.com/2015/05/laravel-announces-v5-1-will-be-lts/). *laravel-news.com*. Laravel-news.
18. "Support Policy" (https://laravel.com/docs/5.2/releases#support-policy). *laravel.com*.
19. CIP (2016-10-13). "Civil Infrastructure Platform Announces First Super Long Term Support Kernel at Embedded Linux Conference Europe" (https://www.cip-project.org/blog/2016/10/13/civil-infrastructure-platform-announces-first-super-long-term-support-kernel-at-embedded-linux-conference-europe). *Civil Infrastructure Platform*. Retrieved 2019-01-16.
20. "Renesas taps new 10-year SLTS kernel from the Civil Infrastructure Platform" (https://linuxgizmos.com/renesas-taps-new-10-year-slts-kernel-from-the-civil-infrastructure-platform/). *LinuxGizmos.com*. 2017-10-16. Retrieved 2019-01-16.
21. "Super long-term kernel support [LWN.net]" (https://lwn.net/Articles/749530/). *lwn.net*. Retrieved 2019-01-16.
22. Bunk, Adrian (11 October 2008). "Linux 2.6.27 will be a longtime supported kernel" (https://marc.info/?l=linux-kernel&m=122375909403298). *Linux kernel* (Mailing list). Retrieved 23 June 2013.
23. Larabel, Michael (26 October 2011). "Linux Foundation Backs Long-Term Support Kernels" (https://www.phoronix.com/scan.php?page=news_item&px=MTAwNjA). *Phoronix*. Phoronix Media. Retrieved 12 August 2013.
24. "What is LTSI?" (https://web.archive.org/web/20141005231742/https://ltsi.linuxfoundation.org/what-is-ltsi). *linuxfoundation.org*. The Linux Foundation. Archived from the original (https://ltsi.linuxfoundation.org/what-is-ltsi) on 5 October 2014. Retrieved 12 August 2013.
25. CIP (2018-10-23). "Renesas RZ/G2M-96CE board adopted as Arm64 reference board for the next CIP SLTS Kernel" (https://www.cip-project.org/blog/2018/10/23/renesas-rz-g2m-96ce-board-adopted-as-arm64-reference-board-for-the-next-cip-slts-kernel). *Civil Infrastructure Platform*. Retrieved 2019-01-16.
26. "The Civil Infrastructure Platform" (https://wiki.linuxfoundation.org/_media/civilinfrastructureplatform/whitepaper_short.pdf) (PDF). *Linux Foundation*.
27. "Renesas Electronics Enables Long-Term Support for Embedded Industrial Linux Developers with New RZ/G Linux Platform" (https://www.renesas.com/us/en/about/press-room/renesas-electronics-enables-long-term-support-embedded-industrial-linux-developers-new-rzg-linux). *www.renesas.com*. 2017-10-16. Retrieved 2023-11-28.
28. "Civil Infrastructure Platform Announces First Super Long Term Support Kernel at Embedded Linux Conference Europe" (https://www.cip-project.org/blog/2016/10/13/civil-infrastructure-platform-announces-first-super-long-term-support-kernel-at-embedded-linux-conference-europe). *Civil Infrastructure Platform*. 2016-10-13. Retrieved 2023-11-28.
29. "Linux Mint Releases" (https://web.archive.org/web/20131217042310/http://www.linuxmint.com/oldreleases.php). *linuxmint.com*. Archived from the original (http://www.linuxmint.com/oldreleases.php) on 17 December 2013. Retrieved 14 December 2013.
30. "Oracle Java SE Support Roadmap" (https://www.oracle.com/technetwork/java/javase/eol-135779.html). Oracle Corporation. 2018-09-25. Retrieved 2018-09-27.
31. Connolly, Dec (2014-05-13). "Moodle 2.7 - the version you've been waiting for?" (https://www.tituslearning.com/moodle-2-7-the-version-youve-been-waiting-for/). *Titus Learning*. Retrieved 2024-02-25.
32. "Moodle 4.1 (LTS)" (https://moodledev.io/general/releases/4.1). *moodledev.io*. March 2023. Retrieved 2023-07-03.
33. Aubry, Matthieu (January 11, 2016). "Announcing Long Term Support in Matomo 2 – The analytics platform for your mission critical projects" (https://matomo.org/blog/2016/01/announcing-long-term-support-in-piwik-2-the-analytics-platform-for-your-mission-critical-projects/). *Official Matomo Blog*. Retrieved 2018-08-28.

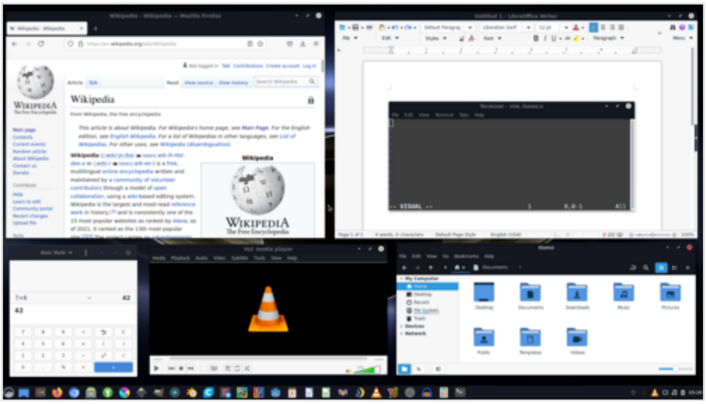
34. "Milestones - matomo-org/matomo" (https://github.com/matomo-org/matomo/milestones?state=closed). *GitHub*. Retrieved 2018-08-28.
35. "When is the next release of Matomo? What is the release schedule?" (https://matomo.org/faq/new-to-piwik/faq_18926/). *Matomo.org*. Retrieved 2018-08-28.
36. "Node v4.2.0 (LTS)" (https://nodejs.org/en/blog/release/v4.2.0/). 2015-10-12. Retrieved 2023-11-28.
37. "Trisquel 7.0 LTS Belenos" (https://trisquel.info/es/trisquel-70-lts-belenos). *Trisquel.info* (in Spanish). 2014-04-11.
38. Baschny, Ernesto; et al. (26 January 2011). "TYPO3 4.5" (http://wiki.typo3.org/TYPO3_4.5). *TYPO3Wiki*. TYPO3 Association. Retrieved 23 June 2013.
39. Brockmeier, Joe (1 June 2006). "Mark Shuttleworth on Ubuntu Long Term Support" (https://www.linux.com/news/mark-shuttleworth-ubuntu-long-term-support). *Linux.com*. Linux Foundation. Retrieved 20 June 2013.
40. Darra, Clive; et al. (23 May 2006). "LTS" (https://wiki.ubuntu.com/LTS). *Ubuntu Wiki*. Canonical Ltd. Retrieved 20 June 2013.
41. "Ubuntu Server 20.04 LTS: Stability, security and more" (https://ubuntu.com/blog/ubuntu-server-20-04).
42. "Canonical Extends Ubuntu LTS Support To 12 Years For Ubuntu Pro Customers" (https://www.phoronix.com/news/Ubuntu-LTS-Pro-12-Years). *Phoronix*. Retrieved 2024-07-27.
43. "Ubuntu 14.04 and 16.04 lifecycle extended to ten years" (https://ubuntu.com/blog/ubuntu-14-04-and-16-04-lifecycle-extended-to-ten-years). *Ubuntu*. Retrieved 2021-10-09.
44. "Canonical expands Long Term Support to 12 years starting with Ubuntu 14.04 LTS" (https://ubuntu.com/blog/canonical-expands-long-term-support-to-12-years-starting-with-ubuntu-14-04-lts). *Ubuntu*. Retrieved 2024-07-27.
45. "Windows 10 release information" (https://technet.microsoft.com/windows/release-info.aspx). *technet.microsoft.com*. Retrieved 2018-03-10.
46. "Overview of Windows as a service" (https://docs.microsoft.com/windows/deployment/update/waas-overview). *technet.microsoft.com*. Retrieved 2016-03-16.
47. "Windows lifecycle and servicing update" (https://techcommunity.microsoft.com/t5/windows-it-pro-blog/windows-lifecycle-and-servicing-update/ba-p/2493043). *TECHCOMMUNITY.MICROSOFT.COM*. 2021-06-28. Retrieved 2021-10-18.
48. "Zabbix release list" (http://zabbix.org/wiki/Zabbix_releases).
49. "Zabbix Life Cycle & Release Policy" (https://www.zabbix.com/life_cycle_and_release_policy).
50. Kern, Philipp; Piat, Franklin; Simmons, Geoff; et al. (19 April 2006). "Point Releases" (https://wiki.debian.org/DebianReleases/PointReleases). *Debian Wiki*. Debian Project. Retrieved 27 June 2013.
51. "Debian security FAQ" (https://www.debian.org/security/faq#lifespan). *Debian.org*. Debian Project. Retrieved 27 June 2013.
52. "[SECURITY] [DSA 2907-1] Announcement of long term support for Debian oldstable" (https://lists.debian.org/debian-security-announce/2014/msg00082.html). *lists.debian.org*. Retrieved 2020-05-22.
53. "LTS - Debian Wiki" (https://wiki.debian.org/LTS/). Debian Project. Retrieved 21 May 2020.
54. "Debian Version History" (https://web.archive.org/web/20200117212830/https://www.debian.org/doc/manuals/project-history/ch-releases.en.html). *Debian.org*. Archived from the original (https://www.debian.org/doc/manuals/project-history/ch-releases.en.html) on 17 January 2020.
55. "LTS/FAQ - Debian Wiki" (https://wiki.debian.org/LTS/FAQ). *wiki.debian.org*. Retrieved 2020-05-22.

Further reading

- Arnuphaptrairong, Tharwon (2011). "Top Ten Lists of Software Project Risks: Evidence from the Literature Survey" (http://www.iaeng.org/publication/IMECS2011/IMECS2011_pp732-737.pdf) (PDF). *Proceedings of the International MultiConference of Engineers and Computer Scientists 2011*. **1**. ISBN 978-988-18210-3-4. ISSN 2078-0966 (https://search.worldcat.org/issn/2078-0966). Retrieved 7 August 2013.
- "What makes software dependable?" (https://www.bcs.org/content/conWebDoc/9933). *bcs.org*. BCS. 5 February 2007. Retrieved 12 August 2013.
- Brenner, Bill. "Outdated software is risky business" (https://searchsecurity.techtarget.com/news/1034019/Outdated-software-is-risky-business). *Search Security*. TechTarget. Retrieved 12 August 2013.
- "Deploying software: Principles to ensure frequent, low-risk deployments" (https://www.gov.uk/service-manual/making-software/deployment.html). *Government Service Design Manual*. Government Digital Service. Retrieved 12 August 2013.
- Graydon, Patrick J.; Knight, John C.; Xiang Yin (2010). "Practical Limits on Software Dependability: A Case Study" (http://www.cs.virginia.edu/~pjg2e/documents/2010.Ada-Europe.pdf) (PDF). Retrieved 12 August 2013.
- Humble, Jez (16 February 2012). "Four Principles of Low-Risk Software Releases" (https://www.informit.com/articles/article.aspx?p=1833567). *InformIT*. Pearson Education. Retrieved 12 August 2013.
- Littlewood, Bev; Strigini, Lorenzo. "Software Reliability and Dependability: A Roadmap" (http://laser.cs.umass.edu/courses/cs521-621/papers/LittlewoodStrigini.pdf) (PDF). Retrieved 12 August 2013.
- Weiss, Aaron (8 June 2012). "The Hidden Security Risks of Legacy Software" (https://www.esecurityplanet.com/patches/the-hidden-security-risks-of-legacy-software.html). *eSecurity Planet*. Retrieved 12 August 2013.
- Ghosh, Shona (19 December 2013). "Windows XP: Microsoft's ticking time bomb" (https://www.pcpro.co.uk/features/386077/windows-xp-microsoft-s-ticking-time-bomb). *PC Pro*. Dennis Publishing. Retrieved 2 January 2014.

Open-source software

Open-source software (**OSS**) is computer software that is released under a license in which the copyright holder grants users the rights to use, study, change, and distribute the software and its source code to anyone and for any purpose.[1][2] Open-source software may be developed in a collaborative, public manner. Open-source software is a prominent example of open collaboration, meaning any capable user is able to participate online in development, making the number of possible contributors indefinite. The ability to examine the code facilitates public trust in the software.[3]



A screenshot of Manjaro Linux running the Cinnamon desktop environment, Firefox accessing Wikipedia which uses MediaWiki, LibreOffice Writer, Vim, GNOME Calculator, VLC and Nemo file manager, all of which are open-source software

Open-source software development can bring in diverse perspectives beyond those of a single company. A 2024 estimate of the value of open-source software to firms is $8.8 trillion, as firms would need to spend 3.5 times the amount they currently do without the use of open source software.[4]

Open-source code can be used for studying and allows capable end users to adapt software to their personal needs in a similar way user scripts and custom style sheets allow for web sites, and eventually publish the modification as a fork for users with similar preferences, and directly submit possible improvements as pull requests.

Definitions

The Open Source Initiative's (OSI) definition is recognized by several governments internationally[5] as the standard or *de facto* definition. OSI uses *The Open Source Definition* to determine whether it considers a software license open source. The definition was based on the Debian Free Software Guidelines, written and adapted primarily by Bruce Perens.[6][7][8] Perens did not base his writing on the "four freedoms" from the Free Software Foundation (FSF), which were only widely available later.[9]

Under Perens' definition, *open source* is a broad software license that makes source code available to the general public with relaxed or non-existent restrictions on the use and modification of the code. It is an explicit "feature" of open source that it puts very few restrictions on the use or distribution by any organization or user, in order to enable the rapid evolution of the software.[10]

According to Feller et al. (2005), the terms "free software" and "open-source software" should be applied to any "software products distributed under terms that allow users" to use, modify, and redistribute the software "in any manner they see fit, without requiring that they pay the author(s) of the software a royalty or fee for engaging in the listed activities."[11]

Despite initially accepting it,[12] Richard Stallman of the FSF now flatly opposes the term "Open Source" being applied to what they refer to as "free software". Although he agrees that the two terms describe "almost the same category of software", Stallman considers equating the terms incorrect and misleading.[13] Stallman also opposes the professed pragmatism of the Open Source Initiative, as he fears that the free software ideals of freedom and community are threatened by compromising on the FSF's idealistic standards for software freedom.[14] The FSF considers free software to be a subset of open-source software, and Richard Stallman explained that DRM software, for example, can be developed as open source, despite that it does not give its users freedom (it restricts them), and thus does not qualify as free software.[13]

The logo of the Open Source Initiative

Open-source software development

Development model

In his 1997 essay *The Cathedral and the Bazaar*, open-source influential contributor Eric S. Raymond suggests a model for developing OSS known as the *bazaar* model.[15] Raymond likens the development of software by traditional methodologies to building a cathedral, with careful isolated work by individuals or small groups.[15] He suggests that all software should be developed using the bazaar style, with differing agendas and approaches.[15]

In the traditional model of development, which he called the *cathedral* model, development takes place in a centralized way.[15] Roles are clearly defined.[15] Roles include people dedicated to designing (the architects), people responsible for managing the project, and people responsible for implementation.[15] Traditional software engineering follows the cathedral model.[15]

The bazaar model, however, is different.[15] In this model, roles are not clearly defined.[15] Some proposed characteristics of software developed using the bazaar model should exhibit the following patterns:[16]

Users should be treated as co-developers: The users are treated like co-developers and so they should have access to the source code of the software.[16] Furthermore, users are encouraged to submit additions to the software, code fixes for the software, bug reports, documentation, etc. Having more co-developers increases the rate at which the software evolves.[16] Linus's law states that given enough eyeballs all bugs are shallow.[16] This means that if many users view the source code, they will eventually find all bugs and suggest how to fix them.[16] Some users have advanced programming skills, and furthermore, each user's machine provides an additional testing environment.[16] This new testing environment offers the ability to find and fix a new bug.[16]

Early releases: The first version of the software should be released as early as possible so as to increase one's chances of finding co-developers early.[16]

Frequent integration: Code changes should be integrated (merged into a shared code base) as often as possible so as to avoid the overhead of fixing a large number of bugs at the end of the project life cycle.[16][17] Some open-source projects have nightly builds where integration is done automatically.[16]

Several versions: There should be at least two versions of the software.[16] There should be a buggier version with more features and a more stable version with fewer features.[16] The buggy version (also called the development version) is for users who want the immediate use of the latest features and are willing to accept the risk of using code that is not yet thoroughly tested.[16] The users can then act as co-developers, reporting bugs and providing bug fixes.[16][18]

High modularization: The general structure of the software should be modular allowing for parallel development on independent components.[16]

Dynamic decision-making structure: There is a need for a decision-making structure, whether formal or informal, that makes strategic decisions depending on changing user requirements and other factors.[16] Compare with extreme programming.[16]

The process of Open source development begins with a requirements elicitation where developers consider if they should add new features or if a bug needs to be fixed in their project.[18] This is established by communicating with the OSS community through avenues such as bug reporting and tracking or mailing lists and project pages.[18] Next, OSS developers select or are assigned to a task and identify a solution. Because there are often many different possible routes for solutions in OSS, the best solution must be chosen with careful consideration and sometimes even peer feedback.[18] The developer then begins to develop and commit the code.[18] The code is then tested and reviewed by peers.[18] Developers can edit and evolve their code through feedback from continuous integration.[18] Once the leadership and community are satisfied with the whole project, it can be partially released and user instruction can be documented.[18] If the project is ready to be released, it is frozen, with only serious bug fixes or security repairs occurring.[18] Finally, the project is fully released and only changed through minor bug fixes.[18]

Advantages

Open source implementation of a standard can increase adoption of that standard.[19] This creates developer loyalty as developers feel empowered and have a sense of ownership of the end product.[20]

Moreover, lower costs of marketing and logistical services are needed for OSS.[21] OSS can be a tool to promote a company's image, including its commercial products.[22] The OSS development approach has helped produce reliable, high quality software quickly and inexpensively.[21]

Open source development offers the potential to quicken innovation and create of social value.[23] In France for instance, a policy that incentivized government to favor free open-source software increased to nearly 600,000 OSS contributions per year, generating social value by increasing the quantity and quality of open-source software.[23] This policy also led to an estimated increase of up to 18% of tech startups and a 14% increase in the number of people employed in the IT sector.[23]

OSS can be highly reliable when it has thousands of independent programmers testing and fixing bugs of the software.[16] Open source is not dependent on the company or author that originally created it.[24] Even if the company fails, the code continues to exist and be developed by its users.[24]

OSS is flexible because modular systems allow programmers to build custom interfaces, or add new abilities to it and it is innovative since open-source programs are the product of collaboration among a large number of different programmers.[16] The mix of divergent perspectives, corporate objectives, and personal goals speeds up innovation.[25]

Moreover, free software can be developed in accordance with purely technical requirements.[26] It does not require thinking about commercial pressure that often degrades the quality of the software.[26] Commercial pressures make traditional software developers pay more attention to customers' requirements than to security requirements, since such features are somewhat invisible to the customer.[26]

Development tools

In open-source software development, tools are used to support the development of the product and the development process itself.[18]

Version control systems such as Centralized Version control system (CVCS) and the distributed version control system (DVCS) are examples of tools, often open source, that help manage the source code files and the changes to those files for a software project in order to foster collaboration.[27] CVCS are centralized with a central repository while DVCS are decentralized and have a local repository for every user.[27] Concurrent Versions System (CVS) and later Subversion (SVN) and Git are examples of CVCS.[27] The repositories are hosted and published on source-code-hosting facilities such as GitHub.[27]

Open-source projects use utilities such as issue trackers to organize open-source software development. Commonly used bug trackers include Bugzilla and Redmine.[18]

Tools such as mailing lists and IRC provide means of coordination and discussion of bugs among developers.[18] Project web pages, wiki pages, roadmap lists and newsgroups allow for the distribution of project information that focuses on end users.[18]

Opportunities for participation

Contributing

The basic roles OSS participants can fall into multiple categories, beginning with leadership at the center of the project who have control over its execution.[28] Next are the core contributors with a great deal of experience and authority in the project who may guide the other contributors.[28] Non-core contributors have less experience and authority, but regularly contribute and are vital to the project's development.[28] New contributors are the least experienced but with mentorship and guidance can become regular contributors.[28]

Some possible ways of contributing to open-source software include such roles as programming, user interface design and testing, web design, bug triage, accessibility design and testing, UX design, code testing, and security review and testing.[28] However, there are several ways of contributing to OSS projects even without coding skills.[28] For example, some less technical ways of participating are documentation writing and editing, translation, project management, event organization and coordination, marketing, release management, community management, and public relations and outreach.[28]

Funding is another way that individuals and organizations choose to contribute to open source projects. Groups like Open Collective provide a means for individuals to contribute monthly to supporting their favorite projects.[29] Organizations like the Sovereign Tech Fund is able to contribute to millions to supporting the tools the German Government uses.[30] The National Science Foundation established a Pathways to Enable Open-Source Ecosystems (POSE) program to support open source innovation.[31]

Industry participation

The adoption of open-source software by industry is increasing over time.[32] OSS is popular in several industries such as telecommunications, aerospace, healthcare, and media & entertainment due to the benefits it provides.[33] Adoption of OSS is more likely in larger organizations and is dependent on the company's IT usage, operating efficiencies, and the productivity of employees.[32]

Industries are likely to use OSS due to back-office functionality, sales support, research and development, software features, quick deployment, portability across platforms and avoidance of commercial license management.[32] Additionally, lower cost for hardware and ownership are also important benefits.[32]

Prominent organizations

Organizations that contribute to the development and expansions of free and open-source software movements exist all over the world.[28] These organizations are dedicated to goals such as teaching and spreading technology.[28] As listed by a former vice president of the Open Source Initiative, some American organizations include the Free Software Foundation, Software Freedom Conservancy, the Open Source Initiative and Software in the Public Interest.[28] Within Europe some notable organizations are Free Software Foundation Europe, open-source projects EU (OSP) and OpenForum Europe (OFE).[28] One Australian organization is Linux Australia while Asia has Open source Asia and FOSSAsia.[28] Free and open source software for Africa (FOSSFA) and OpenAfrica are African organizations and Central and South Asia has such organizations as FLISOL and GRUP de usuarios de software libre Peru.[28] Outside of these, many more organizations dedicated to the advancement of open-source software exist.[28]

Legal and economic issues

Licensing

FOSS products are generally licensed under two types of licenses: permissive licensing and copyleft licensing.[34] Both of these types of licenses are different than proprietary licensing in that they can allow more users access to the software and allow for the creation of derivative works as specified by the terms of the specific license, as each license has its own rules.[34] Permissive licenses allow recipients of the software to implement the author's copyright rights without having to use the same license for distribution.[34] Examples of this type of license include the BSD, MIT, and Apache licenses.[34] Copyleft licenses are different in that they require recipients to use the same license for at least some parts of the distribution of their works.[34] Strong copyleft licenses require all derivative works to use the same license while weak copyleft licenses require the use of the same license only under certain conditions.[34] Examples of this type of license include the GNU family of licenses, and the MPL and EPL licenses.[34] The similarities between these two categories of licensing include that they provide a broad grant of copyright rights, require that recipients preserve copyright notices, and that a copy of the license is provided to recipients with the code.[34]

One important legal precedent for open-source software was created in 2008, when the Jacobson v Katzer case enforced terms of the Artistic license, including attribution and identification of modifications.[34] The ruling of this case cemented enforcement under copyright law when the conditions of the license were not followed.[34] Because of the similarity of the Artistic license to other open-source software licenses, the ruling created a precedent that applied widely.[34]

Examples of free-software license / open-source licenses include Apache licenses, BSD licenses, GNU General Public Licenses, GNU Lesser General Public License, MIT License, Eclipse Public License and Mozilla Public License.[34]

Legal issues

Several gray areas exist within software regulation that have great impact on open-source software, such as if software is a good or service, what can be considered a modification, governance through contract vs license, ownership and right of use.[34] While there have been developments on these issues, they often lead to even more questions.[34] The existence of these uncertainties in regulation has a negative impact on industries involved in technologies as a whole.[34]

Within the legal history of software as a whole, there was much debate on whether to protect it as intellectual property under patent law, copyright law or establishing a unique regulation.[34] Ultimately, copyright law became the standard with computer programs being considered a form of literary work, with some tweaks of unique regulation.[34]

Software is generally considered source code and object code, with both being protectable, though there is legal variety in this definition.[34] Some jurisdictions attempt to expand or reduce this conceptualization for their own purposes.[34] For example, The European Court of Justice defines a computer program as not including the functionality of a program, the programing language, or the format of data files.[34] By limiting protections of the different aspects of software, the law favors an open-source approach to software use.[34] The US especially has an open approach to software, with most open-source licenses originating there.[34] However, this has increased the focus on patent rights within these licenses, which has seen backlash from the OSS community, who prefer other forms of IP protection.[34]

Another issue includes technological protection measures (TPM) and digital rights management (DRM) techniques which were internationally legally recognized and protected in the 1996 World Intellectual Property Organization (WIPO) Treaty.[34] Open source software proponents disliked these technologies as they constrained end-users potentially beyond copyright law.[34] Europe responded to such complaints by putting TPM under legal controls, representing a victory for OSS supporters.[34]

Economic/business implications

In open-source communities, instead of owning the software produced, the producer owns the development of the evolving software.[35] In this way, the future of the software is open, making ownership or intellectual property difficult within OSS.[35] Licensing and branding can prevent others from stealing it, preserving its status as a public good.[35] Open source software can be considered a public good as it is available to everyone and does not decrease in value for others when downloaded by one person.[35] Open source software is unique in that it becomes more valuable as it is used and contributed to, instead of diminishing the resource. This is explained by concepts such as investment in reputation and network effects.[35]

The economic model of open-source software can be explained as developers contribute work to projects, creating public benefits.[35] Developers choose projects based on the perceived benefits or costs, such as improved reputation or value of the project.[35] The motivations of developers can come from many different places and reasons, but the important takeaway is that money is not the only or even most important incentivization.[35]

Because economic theory mainly focuses on the consumption of scarce resources, the OSS dynamic can be hard to understand. In OSS, producers become consumers by reaping the rewards of contributing to a project.[35] For example, a developer becomes well regarded by their peers for a successful contribution to an OSS project.[35] The social benefits and interactions of OSS are difficult to account for in economic models as well.[35] Furthermore, the innovation of technology creates constantly changing value discussions and outlooks, making economic model unable to predict social behavior.[35]

Although OSS is theoretically challenging in economic models, it is explainable as a sustainable social activity that requires resources.[35] These resources include time, money, technology and contributions.[35] Many developers have used technology funded by organizations such as universities and governments, though these same organizations benefit from the work done by OSS.[35] As OSS grows, hybrid systems containing OSS and proprietary systems are becoming more common.[35]

Throughout the mid 2000s, more and more tech companies have begun to use OSS.[24] For example, Dell's move of selling computers with Linux already installed.[24] Microsoft itself has launched a Linux-based operating system despite previous animosity with the OSS movement.[24] Despite these developments, these companies tend to only use OSS for certain purposes, leading to worries that OSS is being taken advantage of by corporations and not given anything in return.[24]

Government uses

While many governments are interested in implementing and promoting open-source software due to the many benefits provided, a huge issue to be considered is cybersecurity.[36] While accidental vulnerabilities are possible, so are attacks by outside agents.[36] Because of these fears, governmental interest in contributing to the governance of software has become more prominent.[36] However, these are the broad strokes of the issue, with each country having their own specific politicized interactions with open-source software and their goals for its implementation.[36] For example, the United States has focused on national security in regard to open-source software implementation due to the perceived threat of the increase of open-source software activity in countries like China and Russia, with the Department of Defense considering multiple criteria for using OSS.[36] These criteria include: if it comes from and is maintained by trusted sources, whether it will continue to be maintained, if there are dependencies on sub-components in the software, component security and integrity, and foreign governmental influence.[36]

Another issue for governments in regard to open source is their investments in technologies such as operating systems, semiconductors, cloud, and artificial intelligence.[36] These technologies all have implications for global cooperation, again opening up security issues and political consequences.[36] Many countries have to balance technological innovation with technological dependence in these partnerships.[36] For example, after China's open-source dependent company Huawei was prevented from using Google's Android system in 2019, they began to create their own alternative operating system: Harmony OS.[36]

Germany recently established a Sovereign Tech Fund, to help support the governance and maintenance of the software that they use.

Open software movement

History

In the early days of computing, such as the 1950s and into the 1960s, programmers and developers shared software to learn from each other and evolve the field of computing.[37] For example, Unix included the operating system source code for users.[37] Eventually, the commercialization of software in the years 1970–1980 began to prevent this practice.[37] However, academics still often developed software collaboratively.[37]

In response, the open-source movement was born out of the work of skilled programmer enthusiasts, widely referred to as hackers or hacker culture.[38] One of these enthusiasts, Richard Stallman, was a driving force behind the free software movement, which would later allow for the open-source movement.[17] In 1984, he resigned from MIT to create a free operating system, GNU, after the programmer culture in his lab was stifled by proprietary software preventing source code from being shared and improved upon.[17] GNU was UNIX compatible, meaning that the programmer enthusiasts would still be familiar with how it worked.[17] However, it quickly became apparent that there was some confusion with the label Stallman had chosen of free software, which he described as free as in free speech, not free beer, referring to the meaning of free as freedom rather than price.[17] He later expanded this concept of freedom to the four essential freedoms.[17] Through GNU, open-source norms of incorporating others' source code, community bug fixes and suggestions of code for new features appeared.[17] In 1985, Stallman founded the Free Software Foundation (FSF) to promote changes in software and to help write GNU.[17] In order to prevent his work from being used in proprietary software, Stallman created the concept of copyleft, which allowed the use of his work by anyone, but under specific terms.[17] To do this, he created the GNU General Public License (GNU GPL) in 1989, which was updated in 1991.[17] In 1991, GNU was combined with the Linux kernel written by Linus Torvalds, as a kernel was missing in GNU.[39] The operating system is now usually referred to as Linux.[17] Throughout this whole period, there were many other free software projects and licenses around at the time, all with different ideas of what the concept of free software was and should be, as well as the morality of proprietary software, such as Berkeley Software Distribution, TeX, and the X Window System.[40]

As free software developed, the Free Software Foundation began to look how to bring free software ideas and perceived benefits to the commercial software industry.[40] It was concluded that FSF's social activism was not appealing to companies and they needed a way to rebrand the free software movement to emphasize the business potential of sharing and collaborating on software source code.[40] The term open source was suggested by Christine Peterson in 1998 at a meeting of supporters of free software.[17] Many in the group felt the name free software was confusing to newcomers and holding back industry interest and they readily accepted the new designation of open source, creating the Open Source Initiative (OSI) and the OSI definition of what open source software is.[17] The Open Source Initiative's (OSI) definition is now recognized by several governments internationally as the standard or *de facto* definition.[39] The definition was based on the Debian Free Software Guidelines, written and adapted primarily by Bruce Perens.[41] The OSI definition differed from the free software definition in that it allows the inclusion of proprietary software and allows more liberties in its licensing.[17] Some, such as Stallman, agree more with the original concept of free software as a result because it takes a strong moral stance against proprietary software, though there is much overlap between the two movements in terms of the operation of the software.[17]

While the Open Source Initiative sought to encourage the use of the new term and evangelize the principles it adhered to, commercial software vendors found themselves increasingly threatened by the concept of freely distributed software and universal access to an application's source code, with an executive of Microsoft calling open source an intellectual property destroyer in 2001.[42] However, while free and open-source software (FOSS) has historically played a role outside of mainstream private software development, companies as large as Microsoft have begun to develop official open source presences on the Internet.[42] IBM, Oracle, and State Farm are just a few of the companies with a serious public stake in today's competitive open source market, marking a significant shift in the corporate philosophy concerning the development of FOSS.[42]

Future

The future of the open source software community, and the free software community by extension, has become successful if not confused about what it stands for.[24] For example, Android and Ubuntu are examples milestones of success in the open source software rise to prominence from the sidelines of technological innovation as it existed in the early 2000s.[24] However, some in the community consider them failures in their representation of OSS due to issues such as the downplaying of the OSS center of Android by Google and its partners, the use of an Apache license that allowed forking and resulted in a loss of opportunities for collaboration within Android, the prioritization of convenience over freedom in Ubuntu, and features within Ubuntu that track users for marketing purposes.[24]

The use of OSS has become more common in business with 78% of companies reporting that they run all or part of their operations on FOSS.[24] The popularity of OSS has risen to the point that Microsoft, a once detractor of OSS, has included its use in their systems.[24] However, this success has raised concerns that will determine the future of OSS as the community must answer questions such as what OSS is, what should it be, and what should be done to protect it, if it even needs protecting.[24] All in all, while the free and open source revolution has slowed to a perceived equilibrium in the market place, that does not mean it is over as many theoretical discussions must take place to determine its future.[24]

Comparisons with other software licensing/development models

Closed source / proprietary software

Open source software differs from proprietary software in that it is publicly available, the license requires no fees, modifications and distributions are allowed under license specifications.[43] All of this works to prevent a monopoly on any OSS product, which is a goal of proprietary software.[43] Proprietary software limits their customers' choices to either committing to using that software, upgrading it or switching to other software, forcing customers to have their software preferences impacted by their monetary cost.[43] The ideal case scenario for the proprietary software vendor would be a lock-in, where the customer does not or cannot switch software due to these costs and continues to buy products from that vendor.[43]

Within proprietary software, bug fixes can only be provided by the vendor, moving platforms requires another purchase and the existence of the product relies on the vendor, who can discontinue it at any point.[38] Additionally, proprietary software does not provide its source code and cannot be altered by users.[17] For businesses, this can pose a security risk and source of frustration, as they cannot specialize the product to their needs, and there may be hidden threats or information leaks within the software that they cannot access or change.[17]

Free software

Under OSI's definition, open source is a broad software license that makes source code available to the general public with relaxed or non-existent restrictions on the use and modification of the code.[44] It is an explicit feature of open source that it puts very few restrictions on the use or distribution by any organization or user, in order to enable the rapid evolution of the software.[44]

Richard Stallman, leader of the Free software movement and member of the free software foundation opposes the term open source being applied to what they refer to as free software.[13] Although he agrees that the two terms describe almost the same category of software, Stallman considers equating the terms incorrect and misleading.[13] He believes that the main difference is that by choosing one term over the other lets others know about what one's goals are: development (open source) or a social stance (free software).[45] Nevertheless, there is significant overlap between open source software and free software.[13] Stallman also opposes the professed pragmatism of the Open Source Initiative, as he fears that the free software ideals of freedom and community are threatened by compromising on the FSF's idealistic standards for software freedom.[45] The FSF considers free software to be a subset of open-source software, and Richard Stallman explained that DRM software, for example, can be developed as open source, despite how it restricts its users, and thus does not qualify as free software.[13]

The FSF said that the term open source fosters an ambiguity of a different kind such that it confuses the mere availability of the source with the freedom to use, modify, and redistribute it.[13] On the other hand, the term free software was criticized for the ambiguity of the word free, which was seen as discouraging for business adoption, and for the historical ambiguous usage of the term.[45]

Developers have used the alternative terms *Free and Open Source Software* (FOSS), or *Free/Libre and Open Source Software* (FLOSS), consequently, to describe open-source software that is also free software.[28]

Source-available software

Software can be distributed with source code, which is a code that is readable.[46] Software is source available when this source code is available to be seen.[46] However to be source available or FOSS, the source code does not need to be accessible to all, just the users of that software.[46] While all FOSS software is source available because this is a requirement made by the Open Source Definition, not all source available software is FOSS.[46] For example, if the software does not meet other aspects of the Open Source Definition such as permitted modification or redistribution, even if the source code is available, the software is not FOSS.[46]

Open-sourcing

A recent trend within software companies is open sourcing, or transitioning their previous proprietary software into open source software through releasing it under an open-source license.[47][48] Examples of companies who have done this are Google, Microsoft and Apple.[47] Additionally, open sourcing can refer to programming open source software or installing open source software.[48] Open sourcing can be beneficial in multiple ways, such as attracting more external contributors who bring new perspectives and problem solving capabilities.[47] The downsides of open sourcing include the work that has to be done to maintaining the new community, such as making the base code easily understandable, setting up communication channels for new developers and creating documentation to allow new developers to easily join.[47] However, a review of several open sourced projects found that although a newly open sourced project attracts many newcomers, a great amount are likely to soon leave the project and their forks are also likely to not be impactful.[47]

Other

Other concepts that may share some similarities to open source are shareware, public domain software, freeware, and software viewers/readers that are freely available but do not provide source code.[17] However, these differ from open source software in access to source code, licensing, copyright and fees.[17]

Society and culture

Demographics

Despite being able to collaborate internationally, open source software contributors were found to mostly be located in large clusters such as Silicon Valley that largely collaborate within themselves.[49] Possible reasons for this phenomenon may be that the OSS contributor demographic largely works in software, meaning that the OSS geographic location is closely related to that dispersion and collaborations could be encouraged through work and social networks.[49] Code acceptance can be impacted by status within these social network clusters, creating unfair predispositions in code acceptance based on location.[50]

Barriers to international collaboration also include linguistic or cultural differences.[51] Furthermore, each country has been shown to have a higher acceptance rate for code from contributors within their country except India, indicating a bias for culturally similar collaborators.[51]

In 2021, the countries with the highest open source software contributions included the United States, China, Germany, India, and the UK, in that order.[49] The counties with the highest OSS developers per capita from a study in 2021 include, in order, Iceland, Switzerland, Norway, Sweden, and Finland, while in 2008 the countries with top amount of estimated contributors in SourceForge were the United States, Germany, United Kingdom, Canada and France.[49][51] Though there have been several studies done on the distribution and contributions of OSS developers, this is still an open field that can be measured in several different ways.[51] For instance, Information and communication technology participation, population, wealth and proportion of access to the internet have been shown to be correlated with OSS contributions.[51]

Although gender diversity has been found to enhance team productivity, women still face biases while contributing to open source software projects when their gender is identifiable.[52] In 2002, only 1.5% of international open-source software developers were women, while women made up 28% of tech industry roles, demonstrating their low representation in the software field.[53] Despite OSS contributions having no prerequisites, this gender bias may continue to exist due to the common belief of contributors that gender should not matter, and the quality of code should be the only consideration for code acceptance, preventing the community from addressing the systemic disparities in female representation.[38] However, a more recent figure of female OSS participation internationally calculated across 2005 to 2021 is 9.8%, with most being recent contributors, indicating that female participation may be growing.[54]

Motivations

There are many motivations for contributing to the OSS community.[28] For one, it is an opportunity to learn and practice multiple skills such as coding and other technology related abilities, but also fundamental skills such as communication and collaboration and practical skills needed to excel in technology related fields such as issue tracking or version control.[28] Instead of learning through a classroom or a job, learning through contributing to OSS allows participants to learn at their own pace and follow what interests them.[28] When contributing to OSS, the contributor can learn the current industry best practices, technology and trends and even have the opportunity to contribute to the next big innovation as OSS grows increasingly popular within the tech field.[28] Contributing to OSS without payment means there is no threat of being fired, though reputations can take a hit.[28] On the other hand, a huge motivation to contribute to OSS is the reputation gained as one grows one's public portfolio.[28]

Disparities

Even though programming was originally seen as a female profession, there remains a large gap in computing.[55] Social identity tends to be a large concern as women in the tech industry face insecurity about attracting unwanted male attention and harassment or being unfeminine in their technology knowledge, having a large impact on confidence.[38] Some male tech participants make clear that they believe women fitting in within the culture is impossible, furthering the insecurity for women and their place in the tech industry.[52] Additionally, even in a voluntary contribution environment like open source software, women tend to end up doing the less technical aspects of projects, such as manual testing or documentation despite women and men showing the same productivity in OSS contributions.[52] Explicit biases include longer feedback time, more scrutinization of code and lower acceptance rate of code.[52] Specifically in the open-source software community, women report that sexually offensive language is common and the women's identity as female is given more attention that as an OSS contributor[38] Bias is hard to address due to the belief that gender should not matter, with most contributors feeling that women getting special treatment is unfair and success should be dependent on skill, preventing any changes to be more inclusive.[38]

Adoption and application

Key projects

Open source software projects are built and maintained by a network of programmers, who may often be volunteers, and are widely used in free as well as commercial products.[56]

Unix: Unix is an operating system created by AT&T that began as a precursor to open source software in that the free and open-source software revolution began when developers began trying to create operating systems without Unix code.[24] Unix was created in the 1960s, before the commercialization of software and before the concept of open source software was necessary, therefore it was not considered a true open source software project.[24] It started as a research project before being commercialized in the mid 1980s.[24] Before its commercialization, it represented many of the ideals held by the Free and Open source software revolution, including the decentralized collaboration of global users, rolling releases and a community culture of distaste towards proprietary software.[24]

BSD: Berkeley Software Distribution (BSD) is an operating system that began as a variant of Unix in 1978 that mixed Unix code with code from Berkeley labs to increase functionality.[24] As BSD was focused on increasing functionality, it would publicly share its greatest innovations with the main Unix operating system.[24] This is an example of the free public code sharing that is a central characteristic of FOSS today.[24] As Unix became commercialized in the 1980s, developers or members of the community who did not support proprietary software began to focus on BSD and turning it into an operating system that did not include any of Unix's code.[24] The final version of BSD was released in 1995.[24]

GNU: GNU is a free operating system created by Richard Stallman in 1984 with its name meaning Gnu's Not Unix.[24] The idea was to create a Unix alternative operating system that would be available for anyone to use and allow programmers to share code freely between them.[24] However, the goal of GNU was not to only replace Unix, but to make a superior version that had more technological capabilities.[24] It was released before the philosophical beliefs of the Free and Open source software revolution were truly defined.[24] Because of its creation by prominent FOSS programmer Richard Stallman, GNU was heavily involved in FOSS activism, with one of the greatest achievements of GNU being the creation of the GNU General Public License or GPL, which allowed developers to release software that could be legally shared and modified.[24]

Linux: Linux is an operating system kernel that was introduced in 1991 by Linus Torvalds.[24] Linux was inspired by making a better version of the for profit operating service Minux.[24] It was radically different than what other hackers were producing at the time due to it being totally free of cost and being decentralized.[24] Later, Linux was put under the GPL license, allowing people to make money with Linux and bringing Linux into the FOSS community.[24]

Apache: Apache began in 1995 as a collaboration between a group of developers releasing their own web server due to their frustration with NCSA HTTPd code base.[24] The name Apache was used because of the several patches they applied to this code base.[24] Within a year of its release, it became the worldwide leading web server.[24] Soon, Apache came out with its own license, creating discord in the greater FOSS community, though ultimately proving successful.[24] The Apache license allowed permitted members to directly access source code, a marked difference from GNU and Linux's approaches.[24]

Extensions for non-software use

While the term open source applied originally only to the source code of software, it is now being applied to many other areas such as open-source ecology, a movement to decentralize technologies so that any human can use them.[13][57] However, it is often misapplied to other areas that have different and competing principles, which overlap only partially.[38]

The same principles that underlie open-source software can be found in many other ventures, such as open source, open content, and open collaboration.[58][3]

This "culture" or ideology takes the view that the principles apply more generally to facilitate concurrent input of different agendas, approaches, and priorities, in contrast with more centralized models of development such as those typically used in commercial companies.[15]

Value

More than 90 percent of companies use open-source software as a component of their proprietary software.[59] The decision to use open-source software, or even engage with open-source projects to improve existing open-source software, is typically a pragmatic business decision.[60][61] When proprietary software is in direct competition with an open-source alternative, research has found conflicting results on the effect of the competition on the proprietary product's price and quality.[62]

For decades, some companies have made servicing of an open-source software product for enterprise users their business model. These companies control an open-source software product, and instead of charging for licensing or use, charge for improvements, integration, and other servicing.[63] Software as a service (SaaS) products based on open-source components are increasingly common.[64]

Open-source software is preferred for scientific applications, because it increases transparency and aids in the validation and acceptance of scientific results.[65]

See also

foss ***Free and open-source software portal***

- Comparison of free and open-source software licenses
- Free software

- Free-software license
- Free software movement
- List of free and open-source software packages
- Free content
- Open-source hardware
- Open Source Initiative
- Open-source license
- Open-source software advocacy
- Open Source Software Institute
- Open-source software security
- Open-source video game
- All articles with titles containing "Open source"
- Proprietary software
- Shared Source Initiative
- Timeline of free and open-source software
- Software composition analysis
- Digital public goods

References

1. St. Laurent, Andrew M. (2008). *Understanding Open Source and Free Software Licensing* (https://books.google.com/books?id=04jG7TTLujoC&pg=PT18). O'Reilly Media. p. 4. ISBN 978-0-596-55395-1. Archived (https://web.archive.org/web/20230422145617/https://books.google.com/books?id=04jG7TTLujoC&pg=PT18) from the original on 22 April 2023. Retrieved 21 March 2023.
2. Corbly, James Edward (25 September 2014). "The Free Software Alternative: Freeware, Open Source Software, and Libraries" (http://ejournals.bc.edu/ojs/index.php/ital/article/view/5105). *Information Technology and Libraries*. **33** (3): 65. doi:10.6017/ital.v33i3.5105 (https://doi.org/10.6017%2Fital.v33i3.5105). ISSN 2163-5226 (https://search.worldcat.org/issn/2163-5226). Archived (https://web.archive.org/web/20210501023728/https://ejournals.bc.edu/index.php/ital/article/view/5105) from the original on 1 May 2021. Retrieved 28 April 2021.
3. Levine, Sheen S.; Prietula, Michael J. (30 December 2013). "Open Collaboration for Innovation: Principles and Performance". *Organization Science*. **25** (5): 1414–1433. arXiv:1406.7541 (https://arxiv.org/abs/1406.7541). doi:10.1287/orsc.2013.0872 (https://doi.org/10.1287%2Forsc.2013.0872). ISSN 1047-7039 (https://search.worldcat.org/issn/1047-7039). S2CID 6583883 (https://api.semanticscholar.org/CorpusID:6583883).
4. Hoffmann, Manuel; Nagle, Frank; Zhou, Yanuo (2024). "The Value of Open Source Software" (https://dx.doi.org/10.2139/ssrn.4693148). *SSRN Electronic Journal*. doi:10.2139/ssrn.4693148 (https://doi.org/10.2139%2Fssrn.4693148). ISSN 1556-5068 (https://search.worldcat.org/issn/1556-5068).
5. "International Authority & Recognition" (https://opensource.org/authority). Opensource.org. 21 April 2015. Archived (https://web.archive.org/web/20190723150523/https://opensource.org/authority) from the original on 23 July 2019. Retrieved 7 December 2017.
6. Perens, Bruce. Open Sources: Voices from the Open Source Revolution (http://www.oreilly.com/catalog/opensources/book/perens.html) Archived (https://web.archive.org/web/20140915025222/http://oreilly.com/catalog/opensources/book/perens.html) 15 September 2014 at the Wayback Machine. O'Reilly Media. 1999.
7. Dibona, Chris; Ockman, Sam (January 1999). The Open Source Definition by Bruce Perens (https://archive.org/details/isbn_9781565925823). O'Reilly. ISBN 978-1-56592-582-3.

8. "The Open Source Definition" (http://opensource.org/docs/osd). 7 July 2006. Archived (https://web.archive.org/web/20131015144021/http://opensource.org/docs/osd) from the original on 15 October 2013. Retrieved 24 August 2008., The Open Source Definition according to the Open Source Initiative
9. "How Many Open Source Licenses Do You Need? – Slashdot" (http://news.slashdot.org/comments.pl?sid=1129863&cid=26875815). *News.slashdot.org*. 16 February 2009. Archived (https://web.archive.org/web/20130717074714/http://news.slashdot.org/comments.pl?sid=1129863&cid=26875815) from the original on 17 July 2013. Retrieved 25 March 2012.
10. Open Source Initiative (24 July 2006). "The Open Source Definition (Annotated)" (https://opensource.org/osd-annotated). *opensource.org*. Archived (https://web.archive.org/web/20210505064040/https://opensource.org/osd-annotated) from the original on 5 May 2021. Retrieved 22 July 2016.
11. Feller, Joseph; Fitzgerald, Brian; Hissam, Scott; Lakhani, Karim R. (2005). "Introduction". *Perspectives on Free and Open Source Software*. Cambridge, MA: The MIT Press. pp. xvii. ISBN 0-262-06246-1.
12. Tiemann, Michael. "History of the OSI" (https://web.archive.org/web/20060924132022/http://www.opensource.org/docs/history.php). Open Source Initiative. Archived from the original (http://www.opensource.org/docs/history.php) on 24 September 2006. Retrieved 13 May 2014.
13. Stallman, Richard (2007). "Why Open Source Misses the Point of Free Software" (https://www.gnu.org/philosophy/open-source-misses-the-point.html).
14. Stallman, Richard (19 June 2007). "Why "Free Software" is better than "Open Source" " (https://www.gnu.org/philosophy/free-software-for-freedom.html). *Philosophy of the GNU Project*. Free Software Foundation. Archived (https://web.archive.org/web/20210327080246/https://www.gnu.org/philosophy/free-software-for-freedom.html) from the original on 27 March 2021. Retrieved 23 July 2007.
15. Raymond, Eric (2005). "The Cathedral and the Bazaar (originally published in Volume 3, Number 3, March 1998)" (https://doi.org/10.5210%2Ffm.v0i0.1472). *First Monday*. doi:10.5210/fm.v0i0.1472 (https://doi.org/10.5210%2Ffm.v0i0.1472). ISSN 1396-0466 (https://search.worldcat.org/issn/1396-0466).
16. Robles, Gregorio (2006). "Empirical Software Engineering Research on Free/Libre/Open Source Software". *2006 22nd IEEE International Conference on Software Maintenance* (https://ieeexplore.ieee.org/document/4021360). pp. 347–350. doi:10.1109/icsm.2006.25 (https://doi.org/10.1109%2Ficsm.2006.25). ISBN 0-7695-2354-4. S2CID 6589566 (https://api.semanticscholar.org/CorpusID:6589566). Retrieved 21 November 2023.
17. Napoleao, Bianca M.; Petrillo, Fabio; Halle, Sylvain (2020). "Open Source Software Development Process: A Systematic Review" (https://ieeexplore.ieee.org/document/9233046). *2020 IEEE 24th International Enterprise Distributed Object Computing Conference (EDOC)*. IEEE. pp. 135–144. arXiv:2008.05015 (https://arxiv.org/abs/2008.05015). doi:10.1109/EDOC49727.2020.00025 (https://doi.org/10.1109%2FEDOC49727.2020.00025). ISBN 978-1-7281-6473-1.
18. Napoleao, Bianca M.; Petrillo, Fabio; Halle, Sylvain (2020). "Open Source Software Development Process: A Systematic Review" (https://ieeexplore.ieee.org/document/9233046). *2020 IEEE 24th International Enterprise Distributed Object Computing Conference (EDOC)*. IEEE. pp. 135–144. arXiv:2008.05015 (https://arxiv.org/abs/2008.05015). doi:10.1109/EDOC49727.2020.00025 (https://doi.org/10.1109%2FEDOC49727.2020.00025). ISBN 978-1-7281-6473-1.
19. US Department of Defense. "Open Source Software FAQ" (https://dodcio.defense.gov/Open-Source-Software-FAQ/). *Chief Information Officer*. Archived (https://web.archive.org/web/20160828150638/http://dodcio.defense.gov/Open-Source-Software-FAQ/) from the original on 28 August 2016. Retrieved 22 July 2016.

20. Sharma, Srinarayan; Vijayan Sugumaran; Balaji Rajagopalan (2002). "A framework for creating hybrid-open source software communities" (http://www.cin.ufpe.br/~in953/lectures/papers/ISJAFrameworkForCreatingHybrid-OpenSourceSoftwareCommunities.pdf) (PDF). *Information Systems Journal*. **12**: 7–25. doi:10.1046/j.1365-2575.2002.00116.x (https://doi.org/10.1046%2Fj.1365-2575.2002.00116.x). S2CID 5815589 (https://api.semanticscholar.org/CorpusID:5815589). Archived (https://web.archive.org/web/20081030014215/http://www.cin.ufpe.br/~in953/lectures/papers/ISJAFrameworkForCreatingHybrid-OpenSourceSoftwareCommunities.pdf) (PDF) from the original on 30 October 2008. Retrieved 8 September 2008.
21. Reynolds, Carl; Jeremy Wyatt (February 2011). "Open Source, Open Standards, and Health Care Information Systems" (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3221346). *Journal of Medical Internet Research*. **13** (1): e24. doi:10.2196/jmir.1521 (https://doi.org/10.2196%2Fjmir.1521). PMC 3221346 (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3221346). PMID 21447469 (https://pubmed.ncbi.nlm.nih.gov/21447469).
22. Landry, John; Rajiv Gupta (September 2000). "Profiting from Open Source". *Harvard Business Review*. doi:10.1225/F00503 (https://doi.org/10.1225%2FF00503) (inactive 1 November 2024).
23. Nagle, Frank (3 March 2019). "Government Technology Policy, Social Value, and National Competitiveness" (https://www.cin.ufpe.br/~in953/lectures/papers/ISJAFrameworkForCreatingHybrid-OpenSourceSoftwareCommunities.pdf) (PDF). *Information Systems Journal*. **12**. doi:10.2139/ssrn.3355486 (https://doi.org/10.2139%2Fssrn.3355486). S2CID 85509685 (https://api.semanticscholar.org/CorpusID:85509685). SSRN 3355486 (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3355486).
24. Tozzi, Christopher (2017). *For Fun and Profit: A History of the Free and Open Source Software Revolution*. United States: MIT Press. ISBN 978-0-262-34118-9.
25. Plotkin, Hal (December 1998). "What (and Why) you should know about open-source software". *Harvard Management Update*: 8–9.
26. Payne, Christian (February 2002). "On the Security of Open Source Software". *Information Systems Journal*. **12** (1): 61–78. doi:10.1046/j.1365-2575.2002.00118.x (https://doi.org/10.1046%2Fj.1365-2575.2002.00118.x). S2CID 8123076 (https://api.semanticscholar.org/CorpusID:8123076).
27. Zolkifli, Nazatul Nurlisa; Ngah, Amir; Deraman, Aziz (2018). "Version Control System: A Review" (https://doi.org/10.1016%2Fj.procs.2018.08.191). *Procedia Computer Science*. **135**: 408–415. doi:10.1016/j.procs.2018.08.191 (https://doi.org/10.1016%2Fj.procs.2018.08.191).
28. Brasseur, V. M. (2018). *Forge your future with open source: build your skills, build your network, build the future of technology*. The pragmatic programmers. Raleigh, North Carolina: The Pragmatic Bookshelf. ISBN 978-1-68050-301-2.
29. "Open Source" (https://blog.opencollective.com/tag/open-source/). *Open Collective*. 20 October 2022. Retrieved 28 May 2024.
30. "Technologies" (https://www.sovereigntechfund.de/tech). *Sovereign Tech Fund*. Retrieved 28 May 2024.
31. "NSF invests over $26 million in open-source projects | NSF - National Science Foundation" (https://new.nsf.gov/tip/updates/nsf-invests-over-26m-open-source-projects). *new.nsf.gov*. 25 October 2023. Retrieved 28 May 2024.
32. Spinellis, Diomidis; Giannikas, Vaggelis (2012). "Organizational adoption of open source software" (https://linkinghub.elsevier.com/retrieve/pii/S0164121211002512). *Journal of Systems and Software*. **85** (3): 666–682. doi:10.1016/j.jss.2011.09.037 (https://doi.org/10.1016%2Fj.jss.2011.09.037).

33. Zhang, Yiming; Malhotra, Baljeet; Chen, Cheng (2018). "Industry-Wide Analysis of Open Source Security" (https://ieeexplore.ieee.org/document/8514185). *2018 16th Annual Conference on Privacy, Security and Trust (PST)*. IEEE. pp. 1–10. doi:10.1109/PST.2018.8514185 (https://doi.org/10.1109%2FPST.2018.8514185). ISBN 978-1-5386-7493-2. S2CID 53234981 (https://api.semanticscholar.org/CorpusID:53234981).
34. Brock, Amanda (2023). *Open Source Law, Policy and Practice* (2nd ed.). UK: Oxford University Press. ISBN 978-0-19-886234-5.
35. Wynants, M., & Cornelis, J. (Eds.). (2005). *How open is the future? : Economic, social and cultural scenarios inspired by free and open-source software*. ASP.
36. Pannier, Alice (2022). *Software Power: The Economic and Geopolitical Implications of Open Source Software*. Études de l'Ifri. ISBN 979-10-373-0641-8.
37. Maracke, Catharina (2019). "Free and Open Source Software and FRAND-based patent licenses: How to mediate between Standard Essential Patent and Free and Open Source Software" (https://doi.org/10.1111%2Fjwip.12114). *The Journal of World Intellectual Property*. **22** (3–4): 78–102. doi:10.1111/jwip.12114 (https://doi.org/10.1111%2Fjwip.12114). ISSN 1422-2213 (https://search.worldcat.org/issn/1422-2213).
38. Bretthauer, David (2001). "Open Source Software: A History" (https://link.gale.com/apps/doc/A84594448/AONE?u=anon~ec73e203&sid=googleScholar&xid=c0f93b7e). *Information Technology and Libraries*. **21** (1).
39. "International Authority & Recognition" (https://opensource.org/authority/). *Open Source Initiative*. 21 April 2015. Retrieved 18 December 2023.
40. Fogel, Karl (2006). *Producing open source software: how to run a successful free software project* (1. Aufl., [Nachdr.] ed.). Beijing Köln: O'Reilly. ISBN 978-0-596-00759-1.
41. Kelty, Christopher (2008). *Two Bits: The Cultural Significance of Free Software*. Duke University Press. ISBN 978-0-8223-8900-2.
42. Miller, Keith W.; Voas, Jeffrey; Costello, Tom (2010). "Free and Open Source Software" (https://dx.doi.org/10.1109/mitp.2010.147). *IT Professional*. **12** (6): 14–16. doi:10.1109/mitp.2010.147 (https://doi.org/10.1109%2Fmitp.2010.147). ISSN 1520-9202 (https://search.worldcat.org/issn/1520-9202). S2CID 265508713 (https://api.semanticscholar.org/CorpusID:265508713).
43. Zhu, Kevin Xiaoguo; Zhou, Zach Zhizhong (2012). "Research Note —Lock-In Strategy in Software Competition: Open-Source Software vs. Proprietary Software" (https://pubsonline.informs.org/doi/10.1287/isre.1110.0358). *Information Systems Research*. **23** (2): 536–545. doi:10.1287/isre.1110.0358 (https://doi.org/10.1287%2Fisre.1110.0358). ISSN 1047-7047 (https://search.worldcat.org/issn/1047-7047).
44. "The Open Source Definition (Annotated)" (https://opensource.org/definition-annotated/). *Open Source Initiative*. 24 July 2006. Retrieved 18 December 2023.
45. Stallman, Richard M.; Gay, Joshua (2002). *Free software, free society*. Boston (Mass.): Free software foundation. ISBN 978-1-882114-98-6.
46. Fortunato, Laura; Galassi, Mark (17 May 2021). "The case for free and open source software in research and scholarship" (https://royalsocietypublishing.org/doi/10.1098/rsta.2020.0079). *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*. **379** (2197). Bibcode:2021RSPTA.37900079F (https://ui.adsabs.harvard.edu/abs/2021RSPTA.37900079F). doi:10.1098/rsta.2020.0079 (https://doi.org/10.1098%2Frsta.2020.0079). ISSN 1364-503X (https://search.worldcat.org/issn/1364-503X). PMID 33775148 (https://pubmed.ncbi.nlm.nih.gov/33775148). S2CID 232387092 (https://api.semanticscholar.org/CorpusID:232387092).

47. Pinto, Gustavo; Steinmacher, Igor; Dias, Luiz Felipe; Gerosa, Marco (2018). "On the challenges of open-sourcing proprietary software projects" (http://link.springer.com/10.1007/s10664-018-9609-6). *Empirical Software Engineering*. **23** (6): 3221–3247. doi:10.1007/s10664-018-9609-6 (https://doi.org/10.1007%2Fs10664-018-9609-6). ISSN 1382-3256 (https://search.worldcat.org/issn/1382-3256). S2CID 254467440 (https://api.semanticscholar.org/CorpusID:254467440).
48. Ågerfalk; Fitzgerald (2008). "Outsourcing to an Unknown Workforce: Exploring Opensurcing as a Global Sourcing Strategy" (https://dx.doi.org/10.2307/25148845). *MIS Quarterly*. **32** (2): 385. doi:10.2307/25148845 (https://doi.org/10.2307%2F25148845). ISSN 0276-7783 (https://search.worldcat.org/issn/0276-7783). JSTOR 25148845 (https://www.jstor.org/stable/25148845).
49. Wachs, Johannes; Nitecki, Mariusz; Schueller, William; Polleres, Axel (March 2002). "The Geography of Open Source Software: Evidence from GitHub" (https://linkinghub.elsevier.com/retrieve/pii/S0040162522000105). *Technological Forecasting and Social Change*. **176**: 121478. arXiv:2107.03200 (https://arxiv.org/abs/2107.03200). doi:10.1016/j.techfore.2022.121478 (https://doi.org/10.1016%2Fj.techfore.2022.121478).
50. Rastogi, Ayushi; Nagappan, Nachiappan; Gousios, Georgios; van der Hoek, André (11 October 2018). "Relationship between geographical location and evaluation of developer contributions in github" (https://dl.acm.org/doi/10.1145/3239235.3240504). *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement* (http://resolver.tudelft.nl/uuid:3b3ad2bd-d865-4a78-b5f0-cdb849d2ee50). ACM. pp. 1–8. doi:10.1145/3239235.3240504 (https://doi.org/10.1145%2F3239235.3240504). ISBN 978-1-4503-5823-1. S2CID 215822439 (https://api.semanticscholar.org/CorpusID:215822439).
51. Gonzalez-Barahona, Jesus M.; Robles, Gregorio; Andradas-Izquierdo, Roberto; Ghosh, Rishab Aiyer (August 2008). "Geographic origin of libre software developers" (https://linkinghub.elsevier.com/retrieve/pii/S0167624508000395). *Information Economics and Policy*. **20** (4): 356–363. doi:10.1016/j.infoecopol.2008.07.001 (https://doi.org/10.1016%2Fj.infoecopol.2008.07.001).
52. Bosu, Amiangshu; Sultana, Kazi Zakia (2019). "Diversity and Inclusion in Open Source Software (OSS) Projects: Where do We Stand?" (https://ieeexplore.ieee.org/document/8870179). *2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. IEEE. pp. 1–11. doi:10.1109/ESEM.2019.8870179 (https://doi.org/10.1109%2FESEM.2019.8870179). ISBN 978-1-7281-2968-6. S2CID 197640269 (https://api.semanticscholar.org/CorpusID:197640269).
53. Nafus, Dawn (June 2012). " 'Patches don't have gender': What is not open in open source software" (http://journals.sagepub.com/doi/10.1177/1461444811422887). *New Media & Society*. **14** (4): 669–683. doi:10.1177/1461444811422887 (https://doi.org/10.1177%2F1461444811422887). ISSN 1461-4448 (https://search.worldcat.org/issn/1461-4448). S2CID 206727320 (https://api.semanticscholar.org/CorpusID:206727320).
54. Trinkenreich, Bianca; Wiese, Igor; Sarma, Anita; Gerosa, Marco; Steinmacher, Igor (31 October 2022). "Women's Participation in Open Source Software: A Survey of the Literature" (https://dl.acm.org/doi/10.1145/3510460). *ACM Transactions on Software Engineering and Methodology*. **31** (4): 1–37. arXiv:2105.08777 (https://arxiv.org/abs/2105.08777). doi:10.1145/3510460 (https://doi.org/10.1145%2F3510460). ISSN 1049-331X (https://search.worldcat.org/issn/1049-331X). S2CID 234778104 (https://api.semanticscholar.org/CorpusID:234778104).
55. Albusays, Khaled; Bjorn, Pernille; Dabbish, Laura; Ford, Denae; Murphy-Hill, Emerson; Serebrenik, Alexander; Storey, Margaret-Anne (April 2021). "The Diversity Crisis in Software Development" (https://ieeexplore.ieee.org/document/9354402). *IEEE Software*. **38** (2): 19–25. doi:10.1109/MS.2020.3045817 (https://doi.org/10.1109%2FMS.2020.3045817). ISSN 0740-7459 (https://search.worldcat.org/issn/0740-7459).

56. Popp, Karl Michael, ed. (2020). *Best practices for commercial use of open source software: business models, processes and tools for managing open source software*. Synomic Academy. Norderstedt: BoD – Books on Demand. ISBN 978-3-7386-1909-6.
57. Powers, Stephen M.; Hampton, Stephanie E. (2019). "Open science, reproducibility, and transparency in ecology" (https://doi.org/10.1002%2Feap.1822). *Ecological Applications*. **29** (1): e01822. Bibcode:2019EcoAp..29E1822P (https://ui.adsabs.harvard.edu/abs/2019EcoAp..29E1822P). doi:10.1002/eap.1822 (https://doi.org/10.1002%2Feap.1822). ISSN 1051-0761 (https://search.worldcat.org/issn/1051-0761). PMID 30362295 (https://pubmed.ncbi.nlm.nih.gov/30362295).
58. Cheliotis, Giorgos (2009). "From open source to open content: Organization, licensing and decision processes in open cultural production" (https://dx.doi.org/10.1016/j.dss.2009.02.006). *Decision Support Systems*. **47** (3): 229–244. doi:10.1016/j.dss.2009.02.006 (https://doi.org/10.1016%2Fj.dss.2009.02.006). ISSN 0167-9236 (https://search.worldcat.org/issn/0167-9236).
59. Butler *et al.* 2022, p. 1.
60. Butler *et al.* 2022, p. 11152.
61. Davila 2015, p. 7.
62. Zhou & Choudhary 2022, p. 731.
63. August *et al.* 2021, pp. 1–2.
64. August *et al.* 2021, p. 1.
65. Morin *et al.* 2012, Compatibility, Proliferation, Fragmentation, and Directionality.

Further reading

- Androutsellis-Theotokis, Stephanos; Spinellis, Diomidis; Kechagia, Maria; Gousios, Georgios (2010). "Open source software: A survey from 10,000 feet" (http://www.dmst.aueb.gr/dds/pubs/jrnl/2010-TOMS-OSS-Survey/html/ASKG10.pdf) (PDF). *Foundations and Trends in Technology, Information and Operations Management*. **4** (3–4): 187–347. doi:10.1561/0200000026 (https://doi.org/10.1561%2F0200000026). ISBN 978-1-60198-484-5.
- August, Terrence; Chen, Wei; Zhu, Kevin (2021). "Competition Among Proprietary and Open-Source Software Firms: The Role of Licensing in Strategic Contribution". *Management Science*. **67** (5): 3041–3066. doi:10.1287/mnsc.2020.3674 (https://doi.org/10.1287%2Fmnsc.2020.3674).
- Birkinbine, Benjamin J. (2020). *Incorporating the Digital Commons: Corporate Involvement in Free and Open Source Software* (https://library.oapen.org/handle/20.500.12657/37226). University of Westminster Press. hdl:20.500.12657/37226 (https://hdl.handle.net/20.500.12657%2F37226). ISBN 978-1-912656-43-1.
- Butler, Simon; Gamalielsson, Jonas; Lundell, Björn; Brax, Christoffer; Mattsson, Anders; Gustavsson, Tomas; Feist, Jonas; Kvarnström, Bengt; Lönroth, Erik (2022). "Considerations and challenges for the adoption of open source components in software-intensive businesses". *Journal of Systems and Software*. **186**. Elsevier BV: 111152. doi:10.1016/j.jss.2021.111152 (https://doi.org/10.1016%2Fj.jss.2021.111152). ISSN 0164-1212 (https://search.worldcat.org/issn/0164-1212).
- Coleman, E. Gabriella. *Coding Freedom: The Ethics and Aesthetics of Hacking* (Princeton UP, 2012)
- Davila, Jacinto (2015). "The Political Logic of Free, Libre, Open Source Software". *Societal Benefits of Freely Accessible Technologies and Knowledge Resources*. IGI Global. pp. 1–24. ISBN 978-1-4666-8337-2.

- Fadi P. Deek; James A. M. McHugh (2008). *Open Source: Technology and Policy* (https://archive.org/details/opensourcetechno0000deek). Cambridge: Cambridge University Press. ISBN 978-0-511-36775-5.
- Chris DiBona and Sam Ockman and Mark Stone, ed. (1999). *Open Sources: Voices from the Open Source Revolution* (https://archive.org/details/isbn_9781565925823). O'Reilly. ISBN 978-1-56592-582-3.
- Joshua Gay, ed. (2002). *Free Software, Free Society: Selected Essays of Richard M. Stallman* (https://archive.org/details/freesoftwarefree00rich). Boston: GNU Press, Free Software Foundation. ISBN 978-1-882114-98-6.
- *Understanding FOSS | editor = Sampathkumar Coimbatore India (https://books.google.com/books?id=2JwU71HSkikC&q=k.s.sampathkumar)*
- Benkler, Yochai (2002), "Coase's Penguin, or, Linux and The Nature of the Firm." Yale Law Journal 112.3 (Dec 2002): p367(78) (http://www.benkler.org/CoasesPenguin.PDF) (in Adobe pdf format)
- v. Engelhardt, Sebastian (2008). " "The Economic Properties of Software", Jena Economic Research Papers, Volume 2 (2008), Number 2008-045" (https://ideas.repec.org/p/jrp/jrpwrp/2008-045.html) (PDF). *Jena Economics Research Papers*.
- Lerner, J. & Tirole, J. (2002): 'Some simple economics on open source', Journal of Industrial Economics 50(2), p 197–234
- Välimäki, Mikko (2005). *The Rise of Open Source Licensing: A Challenge to the Use of Intellectual Property in the Software Industry* (https://web.archive.org/web/20090304002920/http://pub.turre.com/openbook_valimaki.pdf) (PDF). Turre Publishing. Archived from the original (http://pub.turre.com/openbook_valimaki.pdf) (PDF) on 4 March 2009.
- Morin, Andrew; Urban, Jennifer; Sliz, Piotr (2012). "A Quick Guide to Software Licensing for the Scientist-Programmer" (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3406002). *PLOS Computational Biology*. **8** (7): e1002598. Bibcode:2012PLSCB...8E2598M (https://ui.adsabs.harvard.edu/abs/2012PLSCB...8E2598M). doi:10.1371/journal.pcbi.1002598 (https://doi.org/10.1371%2Fjournal.pcbi.1002598). ISSN 1553-7358 (https://search.worldcat.org/issn/1553-7358). PMC 3406002 (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3406002). PMID 22844236 (https://pubmed.ncbi.nlm.nih.gov/22844236).
- Polley, Barry (11 December 2007). "Open Source Discussion Paper – version 1.0" (https://web.archive.org/web/20180223161836/https://nzoss.org.nz/system/files/moj_oss_strategy_1.0.pdf) (PDF). *New Zealand Open Source Society*. New Zealand Ministry of Justice. Archived from the original (http://nzoss.org.nz/system/files/moj_oss_strategy_1.0.pdf) (PDF) on 23 February 2018. Retrieved 12 December 2007.
- Rossi, M. A. (2006): Decoding the free/open-source software puzzle: A survey of theoretical and empirical contributions, in J. Bitzer P. Schröder, eds, 'The Economics of Open Source Software Development', p 15–55.
- Open Sources: Voices from the Open Source Revolution (http://www.oreilly.com/catalog/opensources/book/toc.html) — an online book containing essays from prominent members of the open-source community
- Berry, D M (2004). The Contestation of Code: A Preliminary Investigation into the Discourse of the Free Software and Open Software Movement, Critical Discourse Studies, Volume 1(1). (https://web.archive.org/web/20060912222653/http://opensource.mit.edu/papers/berry1.pdf)
- Schrape, Jan-Felix (2017). "Open Source Projects as Incubators of Innovation. From Niche Phenomenon to Integral Part of the Software Industry" (http://www.uni-stuttgart.de/soz/oi/publikationen/soi_2017_3_Schrape.Open.Source.Projects.Incubators.Innovation.pdf) (PDF). Stuttgart: Research Contributions to Organizational Sociology and Innovation Studies 2017-03.
- *Sustainable Open Source (https://wiki.evolveum.com/display/midPoint/Sustainable+Open+Source)*, a Confluence article providing guidelines for fair participation in the open source

ecosystem, by Radovan Semancik

- Zhou, Zach Zhizhong; Choudhary, Vidyanand (2022). "Impact of Competition from Open Source Software on Proprietary Software". *Production and Operations Management*. **31** (2): 731–742. doi:10.1111/poms.13575 (https://doi.org/10.1111%2Fpoms.13575).

External links

- The Open Source Initiative's definition of open source (https://opensource.org/docs/definition.php)
- Free / Open Source Research Community (https://web.archive.org/web/20050828004359/http://opensource.mit.edu/online_papers.php) — Many online research papers about Open Source

WIKIPEDIA
The Free Encyclopedia

Open-source software development

Open-source software development (OSSD) is the process by which open-source software, or similar software whose source code is publicly available, is developed by an **open-source software project**. These are software products available with its source code under an open-source license to study, change, and improve its design. Examples of some popular open-source software products are Mozilla Firefox, Google Chromium, Android, LibreOffice and the VLC media player.

History

In 1997, Eric S. Raymond wrote *The Cathedral and the Bazaar*.[1] In this book, Raymond makes the distinction between two kinds of software development. The first is the conventional closed-source development. This kind of development method is, according to Raymond, like the building of a cathedral; central planning, tight organization and one process from start to finish. The second is the progressive open-source development, which is more like "a great babbling bazaar of differing agendas and approaches out of which a coherent and stable system could seemingly emerge only by a succession of miracles." The latter analogy points to the discussion involved in an open-source development process.

Differences between the two styles of development, according to Bar and Fogel, are in general the handling (and creation) of bug reports and feature requests, and the constraints under which the programmers are working.[2] In closed-source software development, the programmers are often spending a lot of time dealing with and creating bug reports, as well as handling feature requests. This time is spent on creating and prioritizing further development plans. This leads to part of the development team spending a lot of time on these issues, and not on the actual development. Also, in closed-source projects, the development teams must often work under management-related constraints (such as deadlines, budgets, etc.) that interfere with technical issues of the software. In open-source software development, these issues are solved by integrating the users of the software in the development process, or even letting these users build the system themselves.

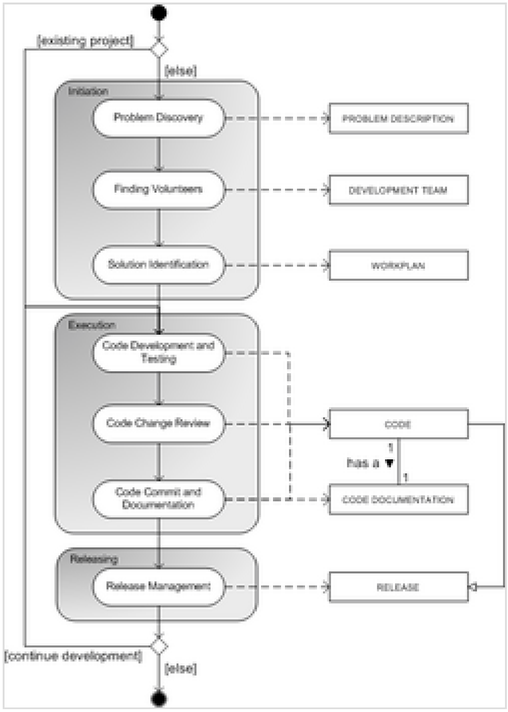
Model

Open-source software development can be divided into several phases. The phases specified here are derived from *Sharma et al*.[3] A diagram displaying the process-data structure of open-source software development is shown on the right. In this picture, the phases of open-source software development are displayed, along with the corresponding data elements. This diagram is made using the meta-modeling and meta-process modeling techniques.

Starting an open-source project

There are several ways in which work on an open-source project can start:

1. An individual who senses the need for a project announces the intent to develop a project in public.
2. A developer working on a limited but working codebase, releases it to the public as the first version of an open-source program.
3. The source code of a mature project is released to the public.
4. A well-established open-source project can be forked by an interested outside party.



Process-Data Model for open-source software development

Eric Raymond observed in his essay *The Cathedral and the Bazaar* that announcing the intent for a project is usually inferior to releasing a working project to the public.

It's a common mistake to start a project when contributing to an existing similar project would be more effective (NIH syndrome). To start a successful project it is very important to investigate what's already there. The process starts with a choice between the adopting of an existing project, or the starting of a new project. If a new project is started, the process goes to the Initiation phase. If an existing project is adopted, the process goes directly to the Execution phase.

Types of open-source projects

Several types of open-source projects exist. First, there is the garden variety of software programs and libraries, which consist of standalone pieces of code. Some might even be dependent on other open-source projects. These projects serve a specified purpose and fill a definite need. Examples of this type of project include the Linux kernel, the Firefox web browser and the LibreOffice office suite of tools.

Distributions are another type of open-source project. Distributions are collections of software that are published from the same source with a common purpose. The most prominent example of a "distribution" is an operating system. There are many Linux distributions (such as Debian, Fedora Core, Mandriva, Slackware, Ubuntu etc.) which ship the Linux kernel along with many user-land components. There are other distributions, like ActivePerl (https://web.archive.org/web/20160331201814/http://www.activestate.com/activeperl), the Perl programming language for various operating systems, and Cygwin distributions of open-source programs for Microsoft Windows.

Other open-source projects, like the BSD derivatives, maintain the source code of an entire operating system, the kernel and all of its core components, in one revision control system; developing the entire system together as a single team. These operating system development projects closely integrate their tools, more so than in the other distribution-based systems.

Finally, there is the book or standalone document project. These items usually do not ship as part of an open-source software package. Linux Documentation Project hosts many such projects that document various aspects of the Linux operating system. There are many other examples of this type of open-source project.

Methods

It is hard to run an open-source project following a more traditional software development method like the waterfall model, because in these traditional methods it is not allowed to go back to a previous phase. In open-source software development, requirements are rarely gathered before the start of the project; instead they are based on early releases of the software product, as Robbins describes.[4] Besides requirements, often volunteer staff is attracted to help develop the software product based on the early releases of the software. This networking effect is essential according to Abrahamsson et al.: “if the introduced prototype gathers enough attention, it will gradually start to attract more and more developers”. However, Abrahamsson et al. also point out that the community is very harsh, much like the business world of closed-source software: “if you find the customers you survive, but without customers you die”.[5]

Fuggetta[6] argues that “rapid prototyping, incremental and evolutionary development, spiral lifecycle, rapid application development, and, recently, extreme programming and the agile software process can be equally applied to proprietary and open source software”. He also pinpoints Extreme Programming as an extremely useful method for open source software development. More generally, all Agile programming methods are applicable to open-source software development, because of their iterative and incremental character. Other Agile methods are equally useful for both open and closed source software development: Internet-Speed Development, for example is suitable for open-source software development because of the distributed development principle it adopts. Internet-Speed Development uses geographically distributed teams to ‘work around the clock’. This method, mostly adopted by large closed-source firms, (because they're the only ones which afford development centers in different time zones), works equally well in open source projects because a software developed by a large group of volunteers shall naturally tend to have developers spread across all time zones.

Tools

Communication channels

Developers and users of an open-source project are not all necessarily working on the project in proximity. They require some electronic means of communications. Email is one of the most common forms of communication among open-source developers and users. Often, electronic mailing lists are used to make sure e-mail messages are delivered to all interested parties at once. This ensures that at least one of the members can reply to it. In order to communicate in real time, many projects use an instant messaging method such as IRC. Web forums have recently become a common way for users to get help with problems they encounter when using an open-source product. Wikis have become common as a communication medium for developers and users.[7]

Version control systems

In OSS development the participants, who are mostly volunteers, are distributed amongst different geographic regions so there is need for tools to aid participants to collaborate in the development of source code.

During early 2000s, Concurrent Versions System (CVS) was a prominent example of a source code collaboration tool being used in OSS projects. CVS helps manage the files and codes of a project when several people are working on the project at the same time. CVS allows several people to work on the same file at the same time. This is done by moving the file into the users' directories and then merging the files when the users are done. CVS also enables one to easily retrieve a previous version of a file. During mid 2000s, The Subversion revision control system (SVN) was created to replace CVS. It is quickly gaining ground as an OSS project version control system.[7]

Many open-source projects are now using distributed revision control systems, which scale better than centralized repositories such as SVN and CVS. Popular examples are git, used by the Linux kernel,[8] and Mercurial, used by the Python programming language.

Bug trackers and task lists

Most large-scale projects require a bug tracking system to keep track of the status of various issues in the development of the project.

Testing and debugging tools

Since OSS projects undergo frequent integration, tools that help automate testing during system integration are used. An example of such tool is Tinderbox. Tinderbox enables participants in an OSS project to detect errors during system integration. Tinderbox runs a continuous build process and informs users about the parts of source code that have issues and on which platform(s) these issues arise.[7]

A debugger is a computer program that is used to debug (and sometimes test or optimize) other programs. GNU Debugger (GDB) is an example of a debugger used in open-source software development. This debugger offers remote debugging, what makes it especially applicable to open-source software development.

A memory leak tool or memory debugger is a programming tool for finding memory leaks and buffer overflows. A memory leak is a particular kind of unnecessary memory consumption by a computer program, where the program fails to release memory that is no longer needed. Examples of memory leak detection tools used by Mozilla are the XPCOM Memory Leak tools. Validation tools are used to check if pieces of code conform to the specified syntax. An example of a validation tool is Splint.

Package management

A package management system is a collection of tools to automate the process of installing, upgrading, configuring, and removing software packages from a computer. The Red Hat Package Manager (RPM) for .rpm and Advanced Packaging Tool (APT) for .deb file format, are package management systems used by a number of Linux distributions.

Publicizing a project

Software directories and release logs:

1. The Free Software Directory

Articles:

1. Linux Weekly News
2. IBM developerWorks

See also

Free and open-source software portal

- Business models for open-source software
- Government Open Code Collaborative
- Open-source software security
- Software composition analysis
- Software development process
- Release management
- Software engineering
- Metamodeling

References

1. Raymond, E.S. (1999). *The Cathedral & the Bazaar*. O'Reilly Retrieved from http://www.catb.org/~esr/writings/cathedral-bazaar/.
2. Bar, M. & Fogel, K. (2003). *Open Source Development with CVS*, 3rd Edition. Paraglyph Press. (ISBN 1-932111-81-6)
3. Sharma, S., Sugumaran, V. & Rajagopalan, B. (2002). *A framework for creating hybrid-open source software communities*. Information Systems Journal 12 (1), 7 – 25.
4. Robbins, J. E. (2003). *Adopting Open Source Software Engineering (OSSE) Practices by Adopting OSSE Tools*. Making Sense of the Bazaar: Perspectives on Open Source and Free Software, Fall 2003.
5. Abrahamsson, P, Salo, O. & Warsta, J. (2002). *Agile software development methods: Review and Analysis*. VTT Publications.
6. Fuggetta, Alfonso (2003). "Open source software––an evaluation". *Journal of Systems and Software*. **66** (1): 77–90. doi:10.1016/S0164-1212(02)00065-1 (https://doi.org/10.1016%2FS0164-1212%2802%2900065-1).
7. "Tim Berners-Lee on the Web at 25: the past, present and future" (https://www.wired.co.uk/magazine/archive/2014/03/web-at-25/tim-berners-lee). *Wired UK*.
8. "The Greatness of Git - Linux Foundation" (https://www.linuxfoundation.org/blog/blog/the-greatness-of-git). *www.linuxfoundation.org*. Retrieved 2023-08-25.

Further reading

- Kavanagh, Paul (2004). *Open source software: implementation and management*. Software development. Amsterdam Boston: Elsevier Digital Press. ISBN 978-1-55558-320-0.
- Feller, Joseph, ed. (2005). *Perspectives on free and open source software*. Cambridge, Mass: MIT Press. ISBN 978-0-262-06246-6.

- Koch, Stefan, ed. (2005). *Free, open source software development*. Hershey, Pa.: Idea Group Publ. ISBN 978-1-59140-370-8.
- Fogel, Karl (2005). *Producing open source software: how to run a successful free software project* (https://www.worldcat.org/title/ocm62322583) (1st ed.). Beijing ; Sebastopol, CA: O'Reilly. ISBN 978-0-596-00759-1. OCLC 62322583 (https://search.worldcat.org/oclc/62322583).
- Muir, Scott P. (2005). *Open Source Software*. Library Hi Tech. Mark Leggott. Bradford: Emerald Publishing Limited. ISBN 978-1-84544-877-6.
- Feller, Joseph (2007). *Open Source Development, Adoption and Innovation: IFIP Working Group 2. 13 on Open Source Software, June 11-14, 2007, Limerick, Ireland*. IFIP Advances in Information and Communication Technology Ser. Brian Fitzgerald, Walt Scacchi, Alberto Sillitti. New York, NY: Springer. ISBN 978-0-387-72485-0.
- Sowe, Sulayman K.; Stamelos, Ioannis G.; Samoladas, Ioannis M., eds. (2008). *Emerging free and open source software practices* (https://www.worldcat.org/title/ocm84838909). Hershey: IGI Pub. ISBN 978-1-59904-210-7. OCLC 84838909 (https://search.worldcat.org/oclc/84838909).
- Fogel, Karl (2009). *Producing Open Source Software: How to Run a Successful Free Software Project*. Sebastopol: O'Reilly Media, Inc. ISBN 978-0-596-00759-1.
- Engard, Nicole C. (2010). *Practical open source software for libraries*. Chandos information professional series. Oxford: Chandos publishing. ISBN 978-1-84334-585-5.
- Tucker, Allen B.; Morelli, Ralph; de Silva, Chamindra (2012). *Software Development: An Open Source Approach*. Chapman and Hall/CRC Innovations in Software Engineering and Software Development Ser. Boca Raton: Chapman and Hall/CRC. ISBN 978-1-4398-1290-7.
- Haff, Gordon (2021). *How Open Source Ate Software: Understand the Open Source Movement and So Much More* (2nd ed.). Berkeley, CA: Apress L. P. ISBN 978-1-4842-6799-8.

External links

Timeline of free and open-source software

This article presents a timeline of events related to popular free/open-source software. For a narrative explaining the overall development, see the related history of free and open-source software.

The Achievements column documents achievements a project attained at some point in time (not necessarily when it was first released).

1970s

Date	Project	Event	Achievements
1976	Emacs	The original EMACS was a set of Editor MACroS for the TECO editor written in 1976 by Richard Stallman, initially together with Guy L. Steele Jr. Later in 1984 the GNU Emacs was released under a GNU General Public License.[1]	Longest continuously-developed GNU project
1978	BSD	Bill Joy started compiling the first Berkeley Software Distribution (1BSD), which was released on March 9, 1978.	

1980s

Date	Project	Event	Achievements
1982	TeX	Originally written by Donald Knuth in 1978, the new version of TeX was rewritten from scratch and was published in 1982.[2]	One of the longest continuously-developed open source projects
1983, September	GNU Project	Announced by Richard Stallman on Usenet as a project to create a "Free Unix"[3]	Became the standard userland for Linux (c. 1991); USENIX Lifetime Achievement Award (2001)
1984	X Window System	X originated at MIT in 1984. The current protocol version, X11, appeared in September 1987. The X.Org Foundation now leads the X project, with the current reference implementation, X.org Server, available as free software under the MIT License and similar permissive licenses.	Most popular windowing system implementation for desktop Linux and all Unix operating systems, excluding Mac OS X
1985	POSTGRES	Michael Stonebraker returned to Berkeley in 1985, and began a post-Ingres project to address the problems with contemporary database systems that had become increasingly clear during the early 1980s.	
1987	GCC	Written by Richard Stallman with contributions from others as the C compiler for the GNU Project. Later the project would be known as the GNU Compiler Collection.	
1987	Perl	Perl, the dynamic programming language was created by Larry Wall and first released in 1987.	
1988	Bash	Brian Fox began coding Bash on January 10, 1988, after Richard Stallman became dissatisfied with the lack of progress being made by a prior developer.	
1989	BSD	Networking Release 1 (Net/1) was made available to non-licensees of AT&T code and was freely redistributable under the terms of the BSD license.	

1990s

Date	Project	Event	Achievements
1990	Zsh	Paul Falstad wrote the first version of Zsh in 1990.	
1991	Linux kernel	Started by Linus Torvalds, Since the initial release of its source code in 1991, it would grow from a small number of C files under a license prohibiting commercial distribution to its state in 2007 of about 290 megabytes of source under the GNU General Public License.	Many, including: Most popular kernel used by top 500 supercomputers. Most popular kernel in mobile devices sold in 2013.
1991	Python	First released by Guido van Rossum in 1991.	
1992	386BSD	386BSD was written mainly by Berkeley alumni Lynne Jolitz and William Jolitz. The 386BSD releases made to the public beginning in 1992.	
1992	Samba	Andrew Tridgell developed the first version of Samba in 1992, at the Australian National University.	
1993, March	NetBSD	The project began as a result of frustration within the 386BSD developer community with the pace and direction of the operating system's development. The four founders of the NetBSD project were Chris Demetriou, Theo de Raadt, Adam Glass and Charles Hannum.	
1993	Lua	Lua was created in 1993 by Roberto Ierusalimschy, Luiz Henrique de Figueiredo, and Waldemar Celes, members of the Computer Graphics Technology Group (Tecgraf) at the Pontifical Catholic University of Rio de Janeiro, in Brazil.	
1993, August	R	First released by Ross Ihaka and Robert Gentleman at the University of Auckland, NZ	
1993, Dec	FreeBSD	FreeBSD's development began in 1993 with a quickly growing, unofficial patchkit maintained by users of the 386BSD operating system. The first official release was FreeBSD 1.0 in December 1993.	
1993	Wine	Bob Amstadt (the initial project leader) and Eric Youngdale started the project in 1993 as a way to run Windows applications on Linux.	Now able to run vast numbers of Windows applications and video games
1994, March	*Linux Journal*	First issue of the first computer magazine dedicated to Linux.	
1994, March	BSD	4.4BSD-Lite was released that no longer require a USL source license.	
1995, June	PHP	Originally created by Rasmus Lerdorf in 1994, it was released publicly in June 1995.	Formed part of the most popular web development stack (LAMP) in the 1990s and 2000s
1995	GIMP	Created by Spencer Kimball and Peter Mattis, the project originally stood for General Image Manipulation Program.	Used by Hollywood, in the forked form of CinePaint (formerly known as Film Gimp)
1995	Ruby	Created by Yukihiro Matsumoto, the programming language drew greater attention in the 2000s due to	Became extremely popular with internet

		the Ruby on Rails web development framework	startups
1996	Apache	The first version of the Apache web server was created by Robert McCool, who was heavily involved with the NCSA web server, known simply as NCSA HTTPd.	Most popular web server
1996	KDE	KDE was founded in 1996 by Matthias Ettrich, who was then a student at the Eberhard Karls University of Tübingen.	
1997, August	GNOME	The initial project leaders for GNOME were Miguel de Icaza and Federico Mena.	
1999, August	OpenOffice.org	Originally developed as the proprietary software application suite StarOffice by the German company StarDivision, the code was purchased in 1999 by Sun Microsystems. The code was made available free of charge in August 1999. On July 19, 2000, Sun Microsystems announced that it was making the source code of StarOffice available for download under both the LGPL and the Sun Industry Standards Source License (SISSL)	

2000s

Date	Project	Event	Achievements
2000	LLVM	Compiler toolkit, started at the University of Illinois at Urbana–Champaign. Initially a research project and known as "Low-Level Virtual Machine".	Adopted by Apple as their primary compilation platform for Mac OS X
2001	Free Software Foundation Europe	Founded to support free software and oppose software patents in Europe	Theodor Heuss Medal (2010)
2002	Blender	Formerly proprietary software, released as open source in 2002 after a crowdfunding campaign	
2002	MediaWiki	There was no name for the project, until the Wikimedia Foundation was announced in June 2003, when name MediaWiki was coined by a Wikipedia contributor.	Integral to the development of Wikipedia
2003, February	New Zealand Open Source Society	New Zealand Open Source Society (NZOSS), a non-profit organization and incorporated society began with a suggestive letter by David Lane to the government, along with 400 supporters signatures to begin the advancement of open software in New Zealand.	
2003, April	Firefox	Descended from the Mozilla Application Suite, the project started as an experimental branch of the Mozilla Project. Originally titled Phoenix, then renamed as Firebird, the project was finally named Mozilla Firefox. The version 1.0 was released on November 9, 2004.	The second most popular web browser in the world until 2012.[4]
2003, May	WordPress	a free and open-source content management system (CMS) written in PHP and paired with a MySQL or MariaDB database.	Most popular content management system in the world
2004	Ubuntu	a user friendly linux distro	
2005	Git	Created by Linux founder Linus Torvalds	World's most popular distributed revision control system
2008, September	Chromium	Released by Google	Forms the majority of the code in Google Chrome, the most popular web browser in the world
2008	Android	Released by Google	Most popular mobile platform in the world
2009	ChromiumOS	Released by Google	Has since enjoyed popular use in types of devices known as Chromebooks and Chromeboxes

By the 2000s the number of open source software packages in wide use was so large that it would be infeasible to make a definitive list.

2010s

Date	Project	Event	Achievements
2010, March	Linaro	Founded	
2010	LibreOffice	LibreOffice is released; a free open office suite including applications such as word processing, spreadsheets, drawing and database.	Available in over 100 languages.
2010	Android	Becomes most popular smartphone operating system (OS),[5] later became most popular general purpose OS overall.	
2011	Git	Microsoft survey of 1,000 software developers reveals that Git is the most popular version control system among developers[6]	
2011	Bootstrap	Free CSS and JavaScript development starting kit, released by Twitter	Becomes most popular repository on GitHub (2012)
2012	Google Chrome, based on Chromium	Overtakes Internet Explorer to become most widely used web browser, according to StatCounter	
2013	Firefox OS	Mobile phone operating system, released by Mozilla Foundation	
2013 (Q2)	Android	Overtakes iOS to become most popular tablet operating system[7]	
2013, September	SteamOS	Valve's new Linux-based operating system for its Steambox consoles, intended to promote Linux gaming and spread Linux adoption in the high-end video game sector	
2014, February	Vue.js	The JavaScript framework Vue.js is released	Becomes most popular JavaScript framework on GitHub
2019, July	Debian	Debian 10 "Buster" is released[8]	

2020s

Date	Project	Event	Achievements
2020, June	Linux	The Linux operating systems market share breaks the 3% marker for the first time in June 2020, reaching 3.57% in July 2020.[9][10]	

See also

- History of free and open-source software
- List of free and open-source software packages
- Timeline of programming languages – many programming languages are open source

- Timeline of operating systems – quite a few operating systems are open source
- Timeline of Linux adoption
- Timeline of the open-access movement
- History of Linux
  - Linux kernel § History

References

1. Yu-wei, Lin (June 2005), "Epistemologically Multiple Actor-Centred System: or EMACS at Work!" (http://www.hswworld.com/pdfs/frontier16.pdf#page=25) (PDF), *System Design Frontier*, vol. 2, no. 6, pp. 25–35, retrieved April 21, 2013.
2. Beebe, Nelson H. F. (2003), "25 Years of TEX and METAFONT: Looking Back and Looking Forward" (https://web.archive.org/web/20160304002219/http://cta.tug.org/TUGboat/Articles/tb25-1/beebe-2003keynote.pdf) (PDF), *TUGboat*, vol. 25, no. 1, pp. 7–30, archived from the original (http://cta.tug.org/TUGboat/Articles/tb25-1/beebe-2003keynote.pdf) (PDF) on March 4, 2016, retrieved April 21, 2013.
3. Richard Stallman. "new Unix implementation" (https://www.gnu.org/gnu/initial-announcement.en.html).
4. "Desktop Browser Market Share Worldwide" (https://gs.statcounter.com/browser-market-share/desktop/worldwide/#monthly-200901-201807). *statcounter.com*.
5. Hachman, Mark (October 5, 2010). "Nielsen: Android Is Most Popular Smartphone OS | News & Opinion" (https://www.pcmag.com/article2/0,2817,2370259,00.asp). PCMag.com. Retrieved August 12, 2013.
6. Ravi Mandalia (July 20, 2011). "Microsoft Survey Reveals GitHub, Git Most Popular among Developers" (http://www.itproportal.com/2011/07/20/microsoft-survey-reveals-github-git-most-popular-among-developers/). *IT Pro Portal*. Retrieved September 28, 2013.
7. Devin Coldewey (September 27, 2013). "Android overtakes iPad in tablet race" (http://www.nbcnews.com/technology/android-overtakes-ipad-tablet-race-8C11278308). NBC News. Retrieved September 28, 2013.
8. "Debian 10: Playing catch-up with the rest of the Linux world (that's a good thing)" (https://arstechnica.com/information-technology/2019/09/debian-10-playing-catch-up-with-the-rest-of-the-linux-world-thats-a-good-thing/). *Ars Technica*. September 11, 2019. Retrieved September 1, 2020.
9. Moore, Mike. "Microsoft may finally have some encouraging news for Windows 10 users" (https://www.techradar.com/news/microsoft-may-finally-have-some-encouraging-news-for-windows-10-users). *TechRadar*. Retrieved September 1, 2020.
10. Popa, Bogdan. "Windows Grows as Linux Declines a Little in July 2020" (https://news.softpedia.com/news/windows-grows-as-linux-declines-a-little-in-july-2020-530703.shtml). *softpedia*. Retrieved September 1, 2020.

External links

- A brief history of open-source software (https://web.archive.org/web/20080407115446/http://eu.conecta.it/paper/brief_history_open_source.html)

List of bioinformatics software

(Redirected from Bioinformatics software)

The **list of bioinformatics software tools** can be split up according to the license used:

- List of proprietary bioinformatics software
- List of open-source bioinformatics software

Alternatively, here is a categorization according to the respective bioinformatics subfield specialized on:

- **Sequence analysis software**
  - List of sequence alignment software
  - List of alignment visualization software
  - Alignment-free sequence analysis
  - De novo sequence assemblers
  - List of gene prediction software
  - List of disorder prediction software
  - List of Protein subcellular localization prediction tools
  - List of phylogenetics software
  - List of phylogenetic tree visualization software
  - Category:Metagenomics_software
- **Structural biology software**
  - List of molecular graphics systems
  - List of protein-ligand docking software
  - List of RNA structure prediction software
  - List of software for protein model error verification
  - List of protein secondary structure prediction programs
  - List of protein structure prediction software
  - Category:Molecular dynamics software
  - Structural alignment software
- **Other**
  - Compression of genomic sequencing data
  - Bioinformatics workflow management system
  - List of genetic engineering software
  - List of systems biology visualization software
  - List of systems biology modelling software
  - 2D gel analysis software
  - List of mass spectrometry software

Codec

A **codec** is a computer hardware or software component that encodes or decodes a data stream or signal.[1][2][3] *Codec* is a portmanteau of **coder/decoder**.[4]

In electronic communications, an **endec** is a device that acts as both an encoder and a decoder on a signal or data stream,[5] and hence is a type of codec. *Endec* is a portmanteau of **encoder/decoder**.

A coder or encoder encodes a data stream or a signal for transmission or storage, possibly in encrypted form, and the decoder function reverses the encoding for playback or editing. Codecs are used in videoconferencing, streaming media, and video editing applications.

History

Originally, in the mid-20th century, a codec was a hardware device that coded analog signals into digital form using pulse-code modulation (PCM). Later, the term was also applied to software for converting between digital signal formats, including companding functions.

Examples

An audio codec converts analog audio signals into digital signals for transmission or encodes them for storage. A receiving device converts the digital signals back to analog form using an audio decoder for playback. An example of this is the codecs used in the sound cards of personal computers. A video codec accomplishes the same task for video signals.

When implementing the Infrared Data Association (IrDA) protocol, an endec may be used between the UART and the optoelectronic systems.[6]

Compression

In addition to encoding a signal, a codec may also compress the data to reduce transmission bandwidth or storage space. Compression codecs are classified primarily into lossy codecs and lossless codecs.

Lossless codecs are often used for archiving data in compressed form while retaining all information present in the original stream. If preserving the original quality of the stream is more important than eliminating the correspondingly larger data sizes, lossless codecs are preferred. This is especially true if the data is to undergo further processing (for example, editing) in which case the repeated application of processing (encoding and decoding) on lossy codecs will degrade the quality of the resulting data such that it is no longer identifiable (visually, audibly, or both). Using more than one codec or encoding scheme successively can also degrade quality significantly. The decreasing cost of storage capacity and network bandwidth has a tendency to reduce the need for lossy codecs for some media.

Many popular codecs are lossy. They reduce quality in order to maximize compression. Often, this type of compression is virtually indistinguishable from the original uncompressed sound or images, depending on the codec and the settings used.[7] The most widely used lossy data compression technique in digital media is based on the discrete cosine transform (DCT), used in compression standards such as JPEG images, H.26x and MPEG video, and MP3 and AAC audio. Smaller data sets ease the strain on relatively expensive storage sub-systems such as non-volatile memory and hard disk, as well as write-once-read-many formats such as CD-ROM, DVD, and Blu-ray Disc. Lower data rates also reduce cost and improve performance when the data is transmitted, e.g., over the internet.

Media codecs

Two principal techniques are used in codecs, pulse-code modulation and delta modulation. Codecs are often designed to emphasize certain aspects of the media to be encoded. For example, a digital video (using a DV codec) of a sports event needs to encode motion well but not necessarily exact colors, while a video of an art exhibit needs to encode color and surface texture well.

Audio codecs for cell phones need to have very low latency between source encoding and playback. In contrast, audio codecs for recording or broadcasting can use high-latency audio compression techniques to achieve higher fidelity at a lower bit rate.

There are thousands of audio and video codecs, ranging in cost from free to hundreds of dollars or more. This variety of codecs can create compatibility and obsolescence issues. The impact is lessened for older formats, for which free or nearly-free codecs have existed for a long time. The older formats are often ill-suited to modern applications, however, such as playback on small portable devices. For example, raw uncompressed PCM audio (44.1 kHz, 16-bit stereo, as represented on an audio CD or in a .wav or .aiff file) has long been a standard across multiple platforms, but its transmission over networks is slow and expensive compared with more modern compressed formats, such as Opus and MP3.

Many multimedia data streams contain both audio and video, and often some metadata that permits synchronization of audio and video. Each of these three streams may be handled by different programs, processes, or hardware; but for the multimedia data streams to be useful in stored or transmitted form, they must be encapsulated together in a container format.

Lower bitrate codecs allow more users, but they also have more distortion. Beyond the initial increase in distortion, lower bit rate codecs also achieve their lower bit rates by using more complex algorithms that make certain assumptions, such as those about the media and the packet loss rate. Other codecs may not make those same assumptions. When a user with a low bitrate codec talks to a user with another codec, additional distortion is introduced by each transcoding.

Audio Video Interleave (AVI) is sometimes erroneously described as a codec, but AVI is actually a container format, while a codec is a software or hardware tool that encodes or decodes audio or video into or from some audio or video format. Audio and video encoded with many codecs might be put into an AVI container, although AVI is not an ISO standard. There are also other well-known container formats, such as Ogg, ASF, QuickTime, RealMedia, Matroska, and DivX Media Format. MPEG transport stream, MPEG program stream, MP4, and ISO base media file format are examples of container formats that are ISO standardized.

Malware

Fake codecs are used when an online user takes a type of codec and installs viruses and other malware into whatever data is being compressed and uses it as a disguise. This disguise appears as a codec download through a pop-up alert or ad. When a user goes to click or download that codec, the malware is then installed on the computer. Once a fake codec is installed it is often used to access private data, corrupt an entire computer system or to keep spreading the malware. One of the previous most used ways to spread malware was fake AV pages and with the rise of codec technology, both have been used in combination to take advantage of online users.[8] This combination allows fake codecs to be automatically downloaded to a device through a website linked in a pop-up ad, virus/codec alerts or articles as well.

See also

- Audio signal processing
- Comparison of audio coding formats
- Comparison of video codecs
- Comparison of video container formats
- Digital signal processing
- List of codecs
- List of open-source codecs

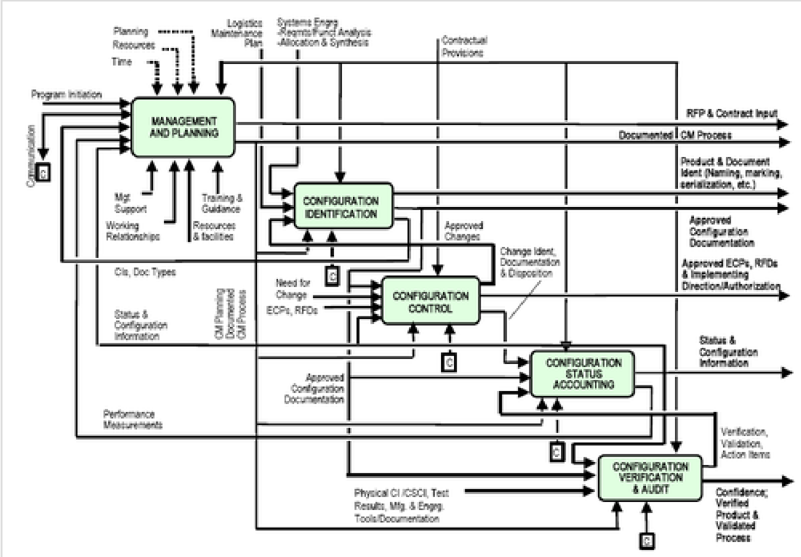
References

1. "Using codecs" (https://support.microsoft.com/en-us/help/15070/windows-media-player-codecs-frequently-asked-questions). Microsoft. Archived (https://web.archive.org/web/20100906114538/https://www.microsoft.com/windows/windowsmedia/player/faq/codec.mspx) from the original on September 6, 2010. Retrieved 2009-12-21.
2. Siegchrist, Gretchen. "About.com - Codec" (https://web.archive.org/web/20150405161202/http://desktopvideo.about.com/od/glossary/g/codec.htm). About.com. Archived from the original (http://desktopvideo.about.com/od/glossary/g/codec.htm) on April 5, 2015. Retrieved 2009-12-21.
3. "Ubuntu Documentation - What is a codec?" (https://web.archive.org/web/20120219101738/https://help.ubuntu.com/8.04/musicvideophotos/C/codecs.html). Ubuntu Documentation Team. Archived from the original (https://help.ubuntu.com/8.04/musicvideophotos/C/codecs.html) on February 19, 2012. Retrieved 2009-12-21.
4. "Codec – Definition of Codec by Merriam-Webster" (https://www.merriam-webster.com/dictionary/codec). *Merriam-Webster*. Retrieved 2019-01-15.
5. Rouse, Margaret (24 October 2016). "Encoder/Decoder" (https://www.techopedia.com/definition/11463/encoderdecoder-endec). *Techopedia*. Retrieved 27 December 2023.
6. "IrDA SIR ENDEC functional description" (http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ddi0183g/I31065.html).
7. "Audio quality of aac vs. mp3 vs. wma vs. ogg encoders" (http://soundexpert.org/). SoundExpert. Retrieved 2010-07-25. "above 5.0 – all sound artifacts will be beyond threshold of human perception with corresponding perception margin"
8. "Fake Video Codecs Still Going Strong" (https://www.zscaler.com/blogs/security-research/fake-video-codecs-still-going-strong). Retrieved 2022-01-09.

Wikipedia
The Free Encyclopedia

Configuration management

Configuration management (**CM**) is a management process for establishing and maintaining consistency of a product's performance, functional, and physical attributes with its requirements, design, and operational information throughout its life.[1][2] The CM process is widely used by military engineering organizations to manage changes throughout the system lifecycle of complex systems, such as weapon systems, military vehicles, and information systems. Outside the military, the CM process is also used with IT service management as defined by ITIL, and with other domain models in the civil engineering and other industrial engineering segments such as roads, bridges, canals, dams, and buildings.[3][4][5]



Top level Configuration Management Activity model

Introduction

CM applied over the life cycle of a system provides visibility and control of its performance, functional, and physical attributes. CM verifies that a system performs as intended, and is identified and documented in sufficient detail to support its projected life cycle. The CM process facilitates orderly management of system information and system changes for such beneficial purposes as to revise capability; improve performance, reliability, or maintainability; extend life; reduce cost; reduce risk and liability; or correct defects. The relatively minimal cost of implementing CM is returned manyfold in cost avoidance. The lack of CM, or its ineffectual implementation, can be very expensive and sometimes can have such catastrophic consequences such as failure of equipment or loss of life.

CM emphasizes the functional relation between parts, subsystems, and systems for effectively controlling system change. It helps to verify that proposed changes are systematically considered to minimize adverse effects. Changes to the system are proposed, evaluated, and implemented using a standardized, systematic approach that ensures consistency, and proposed changes are evaluated in terms of their anticipated impact on the entire system. CM verifies that changes are carried out as prescribed and that documentation of items and systems reflects their true configuration. A complete CM program includes provisions for the storing, tracking, and updating of all system information on a component, subsystem, and system basis.[6]

A structured CM program ensures that documentation (e.g., requirements, design, test, and acceptance documentation) for items is accurate and consistent with the actual physical design of the item. In many cases, without CM, the documentation exists but is not consistent with the item itself. For this reason, engineers, contractors, and management are frequently forced to develop documentation reflecting the actual status of the item before they can proceed with a change. This reverse engineering process is wasteful in terms of human and other resources and can be minimized or eliminated using CM.

History

Configuration Management originated in the United States Department of Defense in the 1950s as a technical management discipline for hardware material items—and it is now a standard practice in virtually every industry. The CM process became its own technical discipline sometime in the late 1960s when the DoD developed a series of military standards called the "480 series" (i.e., MIL-STD-480, MIL-STD-481 and MIL-STD-483) that were subsequently issued in the 1970s. In 1991, the "480 series" was consolidated into a single standard known as the MIL–STD–973 that was then replaced by MIL–HDBK–61 pursuant to a general DoD goal that reduced the number of military standards in favor of industry technical standards supported by standards developing organizations (SDO).[7] This marked the beginning of what has now evolved into the most widely distributed and accepted standard on CM, ANSI–EIA–649–1998.[8] Now widely adopted by numerous organizations and agencies, the CM discipline's concepts include systems engineering (SE), Integrated Logistics Support (ILS), Capability Maturity Model Integration (CMMI), ISO 9000, Prince2 project management method, COBIT, ITIL, product lifecycle management, and Application Lifecycle Management. Many of these functions and models have redefined CM from its traditional holistic approach to technical management. Some treat CM as being similar to a librarian activity, and break out change control or change management as a separate or stand alone discipline.

Overview

CM is the practice of handling changes systematically so that a system maintains its integrity over time. CM implements the policies, procedures, techniques, and tools that manage, evaluate proposed changes, track the status of changes, and maintain an inventory of system and support documents as the system changes. CM programs and plans provide technical and administrative direction to the development and implementation of the procedures, functions, services, tools, processes, and resources required to successfully develop and support a complex system. During system development, CM allows program management to track requirements throughout the life-cycle through acceptance and operations and maintenance. As changes inevitably occur in the requirements and design, they must be approved and documented, creating an accurate record of the system status. Ideally the CM process is applied throughout the system lifecycle. Most professionals mix up or get confused with Asset management (AM, see also ISO/IEC 19770), where it inventories the assets on hand. The key difference between CM and AM is that the former does not manage the financial accounting aspect but on service that the system supports or in other words, that the later (AM) is trying to realize value from an IT asset.[9][10][11]

The CM process for both hardware- and software-configuration items comprises five distinct disciplines as established in the MIL–HDBK–61A[12] and in ANSI/EIA-649. Members of an organization interested in applying a standard change-management process will employ these disciplines as policies and procedures for establishing baselines, manage and control change, and monitor and assess the effectiveness and correctness of progress. The IEEE 12207 process IEEE 12207.2 also has these activities and adds "Release management and delivery". The five disciplines are:

1. CM Planning and Management: a formal document and plan to guide the CM program that includes items such as:
   - Personnel
   - Responsibilities and resources
   - Training requirements
   - Administrative meeting guidelines, including a definition of procedures and tools
   - Baselining processes
   - Configuration control and configuration-status accounting
   - Naming conventions
   - Audits and reviews
   - Subcontractor/vendor CM requirements
2. Configuration Identification (CI): consists of setting and maintaining baselines, which define the system or subsystem architecture, components, and any developments at any point in time. It is the basis by which changes to any part of a system are identified, documented, and later tracked through design, development, testing, and final delivery. CI incrementally establishes and maintains the definitive current basis for Configuration Status Accounting (CSA) of a system and its configuration items (CIs) throughout their lifecycle (development, production, deployment, and operational support) until disposal.
3. Configuration Control: includes the evaluation of all change-requests and change-proposals, and their subsequent approval or disapproval. It covers the process of controlling modifications to the system's design, hardware, firmware, software, and documentation.
4. Configuration Status Accounting: includes the process of recording and reporting configuration item descriptions (e.g., hardware, software, firmware, etc.) and all departures from the baseline during design and production. In the event of suspected problems, the verification of baseline configuration and approved modifications can be quickly determined.
5. Configuration Verification and Audit: an independent review of hardware and software for the purpose of assessing compliance with established performance requirements, commercial and appropriate military standards, and functional, allocated, and product baselines. Configuration audits verify that the system and subsystem configuration documentation complies with the functional and physical performance characteristics before acceptance into an architectural baseline.

Software

The software configuration management (SCM) process is looked upon by practitioners as the best solution to handling changes in software projects. It identifies the functional and physical attributes of software at various points in time, and performs systematic control of changes to the identified attributes for the purpose of maintaining software integrity and traceability throughout the software development life cycle.

The SCM process further defines the need to trace changes, and the ability to verify that the final delivered software has all of the planned enhancements that are supposed to be included in the release. It identifies four procedures that must be defined for each software project to ensure that a sound SCM process is implemented. They are:

1. Configuration identification
2. Configuration control
3. Configuration status accounting
4. Configuration audits

These terms and definitions change from standard to standard, but are essentially the same.

- Configuration identification is the process of identifying the attributes that define every aspect of a configuration item. A configuration item is a product (hardware and/or software) that has an end-user purpose. These attributes are recorded in configuration documentation and baselined. Baselining an attribute forces formal configuration change control processes to be effected in the event that these attributes are changed.
- Configuration change control is a set of processes and approval stages required to change a configuration item's attributes and to re-baseline them.
- Configuration status accounting is the ability to record and report on the configuration baselines associated with each configuration item at any moment of time.
- Configuration audits are broken into functional and physical configuration audits. They occur either at delivery or at the moment of effecting the change. A functional configuration audit ensures that functional and performance attributes of a configuration item are achieved, while a physical configuration audit ensures that a configuration item is installed in accordance with the requirements of its detailed design documentation.

Configuration management database

ITIL specifies the use of a configuration management system (CMS) or configuration management database (CMDB) as a means of achieving industry best practices for Configuration Management. CMDBs are used to track Configuration Items (CIs) and the dependencies between them, where CIs represent the things in an enterprise that are worth tracking and managing, such as but not limited to computers, software, software licenses, racks, network devices, storage, and even the components within such items. CMS helps manage a federated collection of CMDBs.

The benefits of a CMS/CMDB includes being able to perform functions like root cause analysis, impact analysis, change management, and current state assessment for future state strategy development.

Configuration Management (CM) is an ITIL-specific ITSM process that tracks all of the individual CIs in an IT system which may be as simple as a single server, or as complex as the entire IT department. In large organizations a configuration manager may be appointed to oversee and manage the CM process. In ITIL version 3, this process has been renamed as *Service Asset and Configuration Management*.

Information assurance

For information assurance, CM can be defined as the management of security features and assurances through control of changes made to hardware, software, firmware, documentation, test, test fixtures, and test documentation throughout the life cycle of an information system.[13] CM for information assurance, sometimes referred to as *secure configuration management* (SCM), relies upon performance, functional, and physical attributes of IT platforms and products and their environments to determine the appropriate security features and assurances that are used to measure a system configuration state. For example, configuration requirements may be different for a network firewall that functions as part of an organization's Internet boundary versus one that functions as an internal local network firewall.

Maintenance systems

Configuration management is used to maintain an understanding of the status of complex assets with a view to maintaining the highest level of serviceability for the lowest cost. Specifically, it aims to ensure that operations are not disrupted due to the asset (or parts of the asset) overrunning limits of planned lifespan or below quality levels.

In the military, this type of activity is often classed as "mission readiness", and seeks to define which assets are available and for which type of mission; a classic example is whether aircraft on board an aircraft carrier are equipped with bombs for ground support or missiles for defense.

Operating system configuration management

Configuration management can be used to maintain OS configuration files.[14] Many of these systems utilize Infrastructure as Code to define and maintain configuration.[15]

The Promise theory of configuration maintenance was developed by Mark Burgess,[16][17][18] with a practical implementation on present day computer systems in the software CFEngine able to perform real time repair as well as preventive maintenance.

Preventive maintenance

Understanding the "as is" state of an asset and its major components is an essential element in preventive maintenance as used in maintenance, repair, and overhaul and enterprise asset management systems.

Complex assets such as aircraft, ships, industrial machinery etc. depend on many different components being serviceable. This serviceability is often defined in terms of the amount of usage the component has had since it was new, since fitted, since repaired, the amount of use it has had over its life and several other limiting factors. Understanding how near the end of their life each of these components is has been a major undertaking involving labor-intensive record keeping until recent developments in software.

Predictive maintenance

Many types of component use electronic sensors to capture data which provides live condition monitoring. This data is analyzed on board or at a remote location by computer to evaluate its current serviceability and increasingly its likely future state using algorithms which predict potential future failures based on previous examples of failure through field experience and modeling. This is the basis for "predictive maintenance".

Availability of accurate and timely data is essential in order for CM to provide operational value and a lack of this can often be a limiting factor. Capturing and disseminating the operating data to the various support organizations is becoming an industry in itself.

The consumers of this data have grown more numerous and complex with the growth of programs offered by original equipment manufacturers (OEMs). These are designed to offer operators guaranteed availability and make the picture more complex with the operator managing the asset but the OEM taking on the liability to ensure its serviceability.

Standards

A number of standards support or include configuration management,[19] including:

- ANSI/EIA-649-1998 National Consensus Standard for Configuration Management
- EIA-649-A 2004 National Consensus Standard for Configuration Management
- SAE EIA-649-C 2019 Global Consensus Configuration Management Standard
- ISO 10007 Quality management systems – Guidelines for configuration management
- Federal Standard 1037C
- GEIA Standard 836–2002 Configuration Management Data Exchange and Interoperability
- IEEE 829 Standard for Software Test Documentation
- *IEEE Standard for Configuration Management in Systems and Software Engineering*. 2012. doi:10.1109/IEEESTD.2012.6170935 (https://doi.org/10.1109%2FIEEESTD.2012.6170935). ISBN 978-0-7381-7232-3.
- MIL-STD-973 Configuration Management (cancelled on 20 September 2000)[20]
- NATO STANAG 4427 Configuration Management in Systems Life Cycle Management including
- NATO ACMP 2000 Policy on Configuration Management
- NATO ACMP 2009 Guidance on Configuration Management[21]
- NATO ACMP 2100 Configuration Management Contractual Requirements
- CMMI CMMI for Development, Version 1.2 Configuration Management
- CMII-100E CMII Standard for Enterprise Configuration Management[22]
- Extended List of Configuration Management & Related Standards[23]
- ITIL Service Asset and Configuration Management
- ISO 20000:1 2011& 2018 Service Management System.
- ECSS-M-ST-40C Rev.1 Configuration and information management[24]

Guidelines

- IEEE 828-2012 Standard for Configuration Management in Systems and Software Engineering,[25] published date:2012-03-16
- ISO 10007:2017 Quality management – Guidelines for configuration management[26]
- NATO ACMP-2009 – Guidance on configuration management[21]
- ANSI/EIA-632-1998 Processes for Engineering a System
- ANSI/EIA-649-1998 National Consensus Standard for Configuration Management
- GEIA-HB-649 – Implementation Guide for Configuration Management
- EIA-836 Consensus Standard for Configuration Management Data Exchange and Interoperability
- MIL-HDBK-61B Configuration Management Guidance,[27] 7 April 2020
- MIL-STD-3046 Configuration Management,[28] 6 March 2013 and canceled on June 1, 2015
- Defense Acquisition Guidebook,[29] elements of CM at 4.3.7 SE Processes, attributes of CM at 5.1.7 Lifecycle support
- Systems Engineering Fundamentals, Chapter 10 Configuration Management[30]
- Configuration Management Plan United States Dept. of Defense Acquisition document[31]

Construction

More recently configuration management has been applied to large construction projects which can often be very complex and have a huge number of details and changes that need to be documented. Construction agencies such as the Federal Highway Administration have used configuration management for their infrastructure projects.[32] There are construction-based configuration management tools that aim to document change orders and RFIs in order to ensure a project stays on schedule and on budget. These programs can also store information to aid in the maintenance and modification of the infrastructure when it is completed. One such application, CCSNet, was tested in a case study funded by the Federal Transportation Administration (FTA) in which the efficacy of configuration management was measured through comparing the approximately 80% complete construction of the Los Angeles County Metropolitan Transit Agency (LACMTA) first and second segments of the Red Line, a $5.3 billion rail construction project. This study yielded results indicating a benefit to using configuration management on projects of this nature.[33]

See also

- Change detection
- Configuration lifecycle management
- Granular Configuration Automation
- Comparison of open source configuration management software
- Dependency
- List of software engineering topics
- Interchangeable parts
- Continuous configuration automation
- System configuration
- Systems management

References

1. "*MIL-HDBK-61A, ""Military Handbook: Configuration Management Guidance*" (https://web.archive.org/web/20120320093634/https://assist.daps.dla.mil/quicksearch/basic_profile.cfm?ident_number=202239). Department of Defense. 7 February 2001. Archived from the original (https://assist.daps.dla.mil/quicksearch/basic_profile.cfm?ident_number=202239) on 20 March 2012. Retrieved 24 March 2012.
2. "*ANSI/EIA-649B, ""National Consensus Standard for Configuration Management*" (https://archive.today/20120801125835/http://www.geia.org/ANSI-EIA-649-A-Standard----NATIONAL-CONSENSUS-STANDARD-FOR-CONFIGURATION-MANAGEMENT-). TechAmerica. 1 April 2011. Archived from the original (http://www.geia.org/ANSI-EIA-649-A-Standard----NATIONAL-CONSENSUS-STANDARD-FOR-CONFIGURATION-MANAGEMENT-) on 1 August 2012. Retrieved 24 March 2012.
3. "History and Heritage of Civil Engineering" (https://web.archive.org/web/20070216235716/http://live.asce.org/hh/index.mxml?versionChecked=true). *ASCE*. Archived from the original (http://live.asce.org/hh/index.mxml?versionChecked=true) on 16 February 2007. Retrieved 8 August 2007.

4. "Institution of Civil Engineers What is Civil Engineering" (https://web.archive.org/web/20060923023137/http://www.ice.org.uk/downloads//little_book_full_colour.pdf) (PDF). *ICE*. Archived from the original (http://www.ice.org.uk/downloads//little_book_full_colour.pdf) (PDF) on 23 September 2006. Retrieved 22 September 2007.
5. "*Configuration Management and the Federal Transportation Administration (FTA) National Lessons Learned Program*" (https://archive.today/20120907165342/http://www.pacotechnologies.com/Pages/FTA-National-Lessons-Learned-Program.aspx). *Federal Transportation Administration*. Archived from the original (http://www.pacotechnologies.com/Pages/FTA-National-Lessons-Learned-Program.aspx) on 7 September 2012. Retrieved 22 September 2007.
6. "*Systems Engineering Fundamentals*" (https://web.archive.org/web/20060211165311/http://www.dau.mil/pubs/pdf/SEFGuide%2001-01.pdf) (PDF). Defense Acquisition University Press. January 2001. Archived from the original (http://www.dau.mil/pubs/pdf/SEFGuide%2001-01.pdf) (PDF) on 11 February 2006. Retrieved 25 March 2012.
7. "*Memorandum, Specifications and Standards – A New Way of Doing Business*" (https://web.archive.org/web/20131021123641/https://acc.dau.mil/CommunityBrowser.aspx?id=32397). Secretary of Defense. 29 June 1994. Archived from the original (https://acc.dau.mil/CommunityBrowser.aspx?id=32397) on 21 October 2013. Retrieved 23 March 2012.
8. "*Configuration Management Compliance Validation: Critical Review and Technology Assessment(CR/TA)Report*" (http://www.cs.wcupa.edu/~yang/495ppt/config_management.pdf) (PDF). Defense Technical Information Center. Archived (https://ghostarchive.org/archive/20221009/http://www.cs.wcupa.edu/~yang/495ppt/config_management.pdf) (PDF) from the original on 9 October 2022. Retrieved 14 May 2001.
9. Atlassian. "Guide to configuration management databases (CMDBs)" (https://www.atlassian.com/itsm/it-asset-management/cmdb). *Atlassian*. Retrieved 20 July 2021.
10. Galusha, C. (June 2001). "Getting started with IT asset management" (https://ieeexplore.ieee.org/document/939973). *IT Professional*. **3** (3): 37–40. doi:10.1109/6294.939973 (https://doi.org/10.1109%2F6294.939973).
11. "The ISO 19770-1 standard: A guide to implementing IT asset management" (https://blog.shi.com/business-of-it/iso-19770-1-standard-guide-implementing-asset-management/). *The SHI Hub*. 30 January 2018. Retrieved 20 July 2021.
12. "Military Handbook: Configuration Management Guidance" (http://www.product-lifecycle-management.com/download/MIL-HDBK-61B%20%28Draft%29.pdf) (PDF). Department of Defense: United States of America. p. iii–iv. Retrieved 21 July 2016. "4. CM LIFE CYCLE MANAGEMENT AND PLANNING [...] 5. CONFIGURATION IDENTIFICATION [...] 6. CONFIGURATION CONTROL [...] 7. CONFIGURATION STATUS ACCOUNTING [...] 8. CONFIGURATION VERIFICATION AND AUDIT [...] 9. DATA MANAGEMENT [...]"
13. National Information Systems Security Glossary
14. C. Lueninghoener. "Getting Started with Configuration Management. ;login: issue: April 2011, Volume 36, Number 2" (https://www.usenix.org/system/files/login/articles/105457-Lueninghoener.pdf) (PDF). Archived (https://ghostarchive.org/archive/20221009/https://www.usenix.org/system/files/login/articles/105457-Lueninghoener.pdf) (PDF) from the original on 9 October 2022. Retrieved 23 November 2012.
15. Loschwitz, Martin (14 November 2014). "Choosing between the leading open source configuration managers" (http://www.admin-magazine.com/Archive/2014/23/Choosing-between-the-leading-open-source-configuration-managers). *Admin Network & Security*. Lawrence, Kansas: Linux New Media USA LLC.
16. M. Burgess, Cfengine: a site configuration engine, USENIX Computing systems, Vol8, No. 3 1995 [1] (https://www.usenix.org/legacy/publications/compsystems/1995/sum_burgess.pdf)

17. M. Burgess, On the theory of system administration, Science of Computer Programming 49, 2003. p1-46 pdf (http://research.iu.hio.no/papers/sysadmtheory3.pdf) Archived (https://web.archive.org/web/20110724180113/http://research.iu.hio.no/papers/sysadmtheory3.pdf) 24 July 2011 at the Wayback Machine
18. M. Burgess, Configurable immunity for evolving human-computer systems, Science of Computer Programming 51 2004, p197-213 pdf (http://research.iu.hio.no/papers/immunity.pdf) Archived (https://web.archive.org/web/20120303062114/http://research.iu.hio.no/papers/immunity.pdf) 3 March 2012 at the Wayback Machine
19. "NISTIR 7339 Analysis of Standards for Lifecycle Management of Systems for US Army" (https://web.archive.org/web/20161221103821/http://www.mel.nist.gov/msidlibrary/doc/NISTIR_7339.pdf) (PDF). National Institute of Standards and Technology. August 2006. Archived from the original (http://www.mel.nist.gov/msidlibrary/doc/NISTIR_7339.pdf) (PDF) on 21 December 2016. Retrieved 25 November 2015.
20. "ASSIST-QuickSearch - Basic Profile" (https://web.archive.org/web/20110927135629/https://assist.daps.dla.mil/quicksearch/basic_profile.cfm?ident_number=106772). 27 September 2011. Archived from the original (https://assist.daps.dla.mil/quicksearch/basic_profile.cfm?ident_number=106772) on 27 September 2011.
21. [2] (http://nso.nato.int/nso/nsdd/APdetails.html?APNo=2422&LA=EN)
22. "Standards for CM | Institute of Configuration Management" (https://web.archive.org/web/20120502113244/http://www.icmhq.com/cmii-standardsforcm.html). 2 May 2012. Archived from the original (http://www.icmhq.com/cmii-standardsforcm.html) on 2 May 2012.
23. "Configuration Management Standards: an extensive list of CM and related industry standards" (http://cmpic.com/configuration-management-standards.htm). *CMPIC - The Configuration Management Process Improvement Center*.
24. "ECSS-M-ST-40C Rev.1 – Configuration and information management (6 March 2009) | European Cooperation for Space Standardization" (https://ecss.nl/standard/ecss-m-st-40c-rev-1-configuration-and-information-management/). *ecss.nl*.
25. "IEEE 828-2012 - IEEE Standard for Configuration Management in Systems and Software Engineering" (https://standards.ieee.org/ieee/828/5367/). *IEEE*.
26. "ISO 10007:2017(en) Quality management — Guidelines for configuration management" (https://www.iso.org/obp/ui/#iso:std:iso:10007:ed-3:v1:en). *iso.org*. Retrieved 29 November 2023.
27. "ASSIST-QuickSearch Document Details" (https://quicksearch.dla.mil/qsDocDetails.aspx?ident_number=202239). Quicksearch.dla.mil. Retrieved 28 August 2022.
28. "ASSIST-QuickSearch Document Details" (https://quicksearch.dla.mil/qsDocDetails.aspx?ident_number=279266). Quicksearch.dla.mil. Retrieved 28 August 2022.
29. "Defense Acquisition Guidebook [DAG]" (https://web.archive.org/web/20130213203151/https://dag.dau.mil/Pages/Default.aspx). 13 February 2013. Archived from the original (https://dag.dau.mil/Pages/Default.aspx) on 13 February 2013.
30. "Archived copy" (https://web.archive.org/web/20170131231503/http://www.dau.mil/publications/publicationsdocs/sefguide%2001-01.pdf) (PDF). *www.dau.mil*. Archived from the original (http://www.dau.mil/publications/publicationsdocs/sefguide%2001-01.pdf) (PDF) on 31 January 2017. Retrieved 11 January 2022.
31. "Configuration Management Plan" (https://acqnotes.com/acqnote/careerfields/configuration-management-plan). *AcqNotes*.
32. "*Configuration Management for Transportation Management Systems Handbook*" (http://www.everyspec.com/DOT/FHWA-OP-04-013_SEP2003_41209/). *Federal Highway Administration*. Retrieved 28 March 2012.
33. "*Configuration Management Case Study*" (https://web.archive.org/web/20160826174139/http://www.pacotechnologies.com/Pages/Case-Study.aspx). *PACO Technologies, Inc*. Archived from the original (http://www.pacotechnologies.com/Pages/Case-Study.aspx) on 26 August 2016. Retrieved 28 March 2012.

WIKIPEDIA
The Free Encyclopedia

Device driver

In the context of an operating system, a **device driver** is a computer program that operates or controls a particular type of device that is attached to a computer or automaton.[1] A driver provides a software interface to hardware devices, enabling operating systems and other computer programs to access hardware functions without needing to know precise details about the hardware being used.

A driver communicates with the device through the computer bus or communications subsystem to which the hardware connects. When a calling program invokes a routine in the driver, the driver issues commands to the device (drives it). Once the device sends data back to the driver, the driver may invoke routines in the original calling program.

Drivers are hardware dependent and operating-system-specific. They usually provide the interrupt handling required for any necessary asynchronous time-dependent hardware interface.[2]

Purpose

The main purpose of device drivers is to provide abstraction by acting as a translator between a hardware device and the applications or operating systems that use it.[1] Programmers can write higher-level application code independently of whatever specific hardware the end-user is using. For example, a high-level application for interacting with a serial port may simply have two functions for "send data" and "receive data". At a lower level, a device driver implementing these functions would communicate to the particular serial port controller installed on a user's computer. The commands needed to control a 16550 UART are much different from the commands needed to control an FTDI serial port converter, but each hardware-specific device driver abstracts these details into the same (or similar) software interface.

Development

Writing a device driver requires an in-depth understanding of how the hardware and the software works for a given platform function. Because drivers require low-level access to hardware functions in order to operate, drivers typically operate in a highly privileged environment and can cause system operational issues if something goes wrong. In contrast, most user-level software on modern operating systems can be stopped without greatly affecting the rest of the system. Even drivers executing in user mode can crash a system if the device is erroneously programmed. These factors make it more difficult and dangerous to diagnose problems.[3]

The task of writing drivers thus usually falls to software engineers or computer engineers who work for hardware-development companies. This is because they have better information than most outsiders about the design of their hardware. Moreover, it was traditionally considered in the hardware manufacturer's interest to guarantee that their clients can use their hardware in an optimal way. Typically, the Logical Device Driver (LDD) is written by the operating system vendor, while the Physical Device Driver (PDD) is implemented by the device vendor. However, in recent years, non-vendors have written numerous device drivers for proprietary devices, mainly for use with free and open source operating systems. In such cases, it is important that the hardware manufacturer provide information on how the device communicates. Although this information can instead be learned by reverse engineering, this is much more difficult with hardware than it is with software.

Windows uses a combination of driver and minidriver, where the full class/port driver is provided with the operating system, and miniclass/miniport drivers are developed by vendors and implement hardware- or function-specific subset of the full driver stack.[4] Miniport model is used by NDIS, WDM, WDDM, WaveRT, StorPort, WIA, and HID drivers; each of them uses device-specific APIs and still requires the developer to handle tedious device management tasks.

Microsoft has attempted to reduce system instability due to poorly written device drivers by creating a new framework for driver development, called Windows Driver Frameworks (WDF). This includes User-Mode Driver Framework (UMDF) that encourages development of certain types of drivers—primarily those that implement a message-based protocol for communicating with their devices—as user-mode drivers. If such drivers malfunction, they do not cause system instability. The Kernel-Mode Driver Framework (KMDF) model continues to allow development of kernel-mode device drivers but attempts to provide standard implementations of functions that are known to cause problems, including cancellation of I/O operations, power management, and plug-and-play device support.

Apple has an open-source framework for developing drivers on macOS, called I/O Kit.

In Linux environments, programmers can build device drivers as parts of the kernel, separately as loadable modules, or as user-mode drivers (for certain types of devices where kernel interfaces exist, such as for USB devices). Makedev includes a list of the devices in Linux, including ttyS (terminal), lp (parallel port), hd (disk), loop, and sound (these include mixer, sequencer, dsp, and audio).[5]

Microsoft Windows .sys files and Linux .ko files can contain loadable device drivers. The advantage of loadable device drivers is that they can be loaded only when necessary and then unloaded, thus saving kernel memory.

Privilege levels

Depending on the operating system, device drivers may be permitted to run at various different privilege levels. The choice of which level of privilege the drivers are in is largely decided by the type of kernel an operating system uses. An operating system that uses a monolithic kernel, such as the Linux kernel, will typically run device drivers with the same privilege as all other kernel objects. By contrast, a system designed around microkernel, such as Minix, will place drivers as processes independent from the kernel but that use it for essential input-output functionalities and to pass messages between user programs and each other.[6] On Windows NT, a system with a hybrid kernel, it is common for device drivers to run in either kernel-mode or user-mode.[7]

The most common mechanism for segregating memory into various privilege levels is via protection rings. On many systems, such as those with x86 and ARM processors, switching between rings imposes a performance penalty, a factor that operating system developers and embedded software engineers consider when creating drivers for devices which are preferred to be run with low latency, such as network interface cards. The primary benefit of running a driver in user mode is improved stability since a poorly written user-mode device driver cannot crash the system by overwriting kernel memory.[8]

Applications

Because of the diversity of modern hardware and operating systems, drivers operate in many different environments.[9] Drivers may interface with:

- Printers
- Video adapters
- Network cards
- Sound cards
- PC chipsets
- Power and battery management
- Local buses of various sorts—in particular, for bus mastering on modern systems
- Low-bandwidth I/O buses of various sorts (for pointing devices such as mice, keyboards, etc.)
- Computer storage devices such as hard disk, CD-ROM, and floppy disk buses (ATA, SATA, SCSI, SAS)
- Implementing support for different file systems
- Image scanners
- Digital cameras
- Digital terrestrial television tuners
- Radio frequency communication transceiver adapters for wireless personal area networks as used for short-distance and low-rate wireless communication in home automation, (such as example Bluetooth Low Energy (BLE), Thread, Zigbee, and Z-Wave).
- IrDA adapters

Common levels of abstraction for device drivers include:

- For hardware:
  - Interfacing directly
  - Writing to or reading from a device control register
  - Using some higher-level interface (e.g. Video BIOS)
  - Using another lower-level device driver (e.g. file system drivers using disk drivers)
  - Simulating work with hardware, while doing something entirely different[10]
- For software:
  - Allowing the operating system direct access to hardware resources
  - Implementing only primitives
  - Implementing an interface for non-driver software (e.g. TWAIN)
  - Implementing a language, sometimes quite high-level (e.g. PostScript)

So choosing and installing the correct device drivers for given hardware is often a key component of computer system configuration.[11]

Virtual device drivers

Virtual device drivers represent a particular variant of device drivers. They are used to emulate a hardware device, particularly in virtualization environments, for example when a DOS program is run on a Microsoft Windows computer or when a guest operating system is run on, for example, a Xen host. Instead of enabling the guest operating system to dialog with hardware, virtual device drivers take the opposite role and emulates a piece of hardware, so that the guest operating system and its drivers running inside a virtual machine can have the illusion of accessing real hardware. Attempts by the guest operating system to access the hardware are routed to the virtual device driver in the host operating system as e.g., function calls. The virtual device driver can also send simulated processor-level events like interrupts into the virtual machine.

Virtual devices may also operate in a non-virtualized environment. For example, a virtual network adapter is used with a virtual private network, while a virtual disk device is used with iSCSI. A good example for virtual device drivers can be Daemon Tools.

There are several variants of virtual device drivers, such as VxDs, VLMs, and VDDs.

Open source drivers

- Graphics device driver
- Printers: CUPS
- RAIDs: CCISS[12] (Compaq Command Interface for SCSI-3 Support[13])
- Scanners: SANE
- Video: Vidix, Direct Rendering Infrastructure

Solaris descriptions of commonly used device drivers:

- fas: Fast/wide SCSI controller
- hme: Fast (10/100 Mbit/s) Ethernet
- isp: Differential SCSI controllers and the SunSwift card
- glm: (Gigabaud Link Module[14]) UltraSCSI controllers
- scsi: Small Computer Serial Interface (SCSI) devices
- sf: soc+ or social Fiber Channel Arbitrated Loop (FCAL)
- soc: SPARC Storage Array (SSA) controllers and the control device
- social: Serial optical controllers for FCAL (soc+)

APIs

- Windows Display Driver Model (WDDM) – the graphic display driver architecture for Windows Vista and later.
- Unified Audio Model (UAM)[15]
- Windows Driver Foundation (WDF)
- Declarative Componentized Hardware (DCH) - Universal Windows Platform driver[16]
- Windows Driver Model (WDM)

- Network Driver Interface Specification (NDIS) – a standard network card driver API
- Advanced Linux Sound Architecture (ALSA) – the standard Linux sound-driver interface
- Scanner Access Now Easy (SANE) – a public-domain interface to raster-image scanner-hardware
- Installable File System (IFS) – a filesystem API for IBM OS/2 and Microsoft Windows NT
- Open Data-Link Interface (ODI) – network card API similar to NDIS
- Uniform Driver Interface (UDI) – a cross-platform driver interface project
- Dynax Driver Framework (dxd) – C++ open source cross-platform driver framework for KMDF and IOKit[17]

Identifiers

A device on the PCI bus or USB is identified by two IDs which consist of two bytes each. The vendor ID identifies the vendor of the device. The device ID identifies a specific device from that manufacturer/vendor.

A PCI device has often an ID pair for the main chip of the device, and also a subsystem ID pair that identifies the vendor, which may be different from the chip manufacturer.

Security

Computers often have many diverse and customized device drivers running in their operating system (OS) kernel which often contain various bugs and vulnerabilities, making them a target for exploits.[18] A *Bring Your Own Vulnerable Driver* (BYOVD) attacker installs any signed, old third-party driver with known vulnerabilities that allow malicious code to be inserted into the kernel.[19] Drivers that may be vulnerable include those for WiFi and Bluetooth,[20][21] gaming/graphics drivers,[22] and drivers for printers.[23]

There is a lack of effective kernel vulnerability detection tools, especially for closed-source OSes such as Microsoft Windows[24] where the source code of the device drivers is mostly proprietary and not available to examine,[25] and drivers often have many privileges.[26][27][28][29]

A group of security researchers considers the lack of isolation as one of the main factors undermining kernel security,[30] and published an isolation framework to protect operating system kernels, primarily the monolithic Linux kernel whose drivers they say get ~80,000 commits per year.[31][32]

An important consideration in the design of a kernel is the support it provides for protection from faults (fault tolerance) and from malicious behaviours (security). These two aspects are usually not clearly distinguished, and the adoption of this distinction in the kernel design leads to the rejection of a hierarchical structure for protection.[33]

The mechanisms or policies provided by the kernel can be classified according to several criteria, including: static (enforced at compile time) or dynamic (enforced at run time); pre-emptive or post-detection; according to the protection principles they satisfy (e.g., Denning[34][35]); whether they are hardware supported or language based; whether they are more an open mechanism or a binding policy; and many more.

See also

- Driver (software)
- Class driver
- Device driver synthesis and verification
- Driver wrapper
- Free software
- Firmware
- Loadable kernel module
- Makedev
- Microcontroller
- Open-source hardware
- Printer driver
- Replicant (operating system)
- udev (userspace /dev)

References

1. "What is all device driver?" (http://searchenterprisedesktop.techtarget.com/definition/device-driver). *WhatIs.com*. TechTarget. Archived (https://web.archive.org/web/20210213094525/https://searchenterprisedesktop.techtarget.com/definition/device-driver) from the original on 13 February 2021. Retrieved 19 March 2018.
2. EMC Education Services (2010). *Information Storage and Management: Storing, Managing, and Protecting Digital Information* (https://books.google.com/books?id=sCCfRAj3aCgC&q=device+driver+hardware+dependent). John Wiley & Sons. ISBN 9780470618332. Archived (https://web.archive.org/web/20210213094529/https://books.google.com/books?id=sCCfRAj3aCgC&q=device+driver+hardware+dependent) from the original on 2021-02-13. Retrieved 2020-11-10.
3. Burke, Timothy (1995). *Writing device drivers: tutorial and reference* (https://books.google.com/books?id=9aFQAAAAMAAJ&q=writing+a+device+driver). Digital Press. ISBN 9781555581411. Archived (https://web.archive.org/web/20210126050637/https://books.google.com/books?id=9aFQAAAAMAAJ&q=writing+a+device+driver) from the original on 2021-01-26. Retrieved 2016-08-05.
4. "Choosing a driver model" (https://docs.microsoft.com/en-us/windows-hardware/drivers/gettingstarted/choosing-a-driver-model). Microsoft. Retrieved 2021-03-30.
5. "MAKEDEV — Linux Command — Unix Command" (http://linux.about.com/od/commands/l/blcmdl8_MAKEDEV.htm). Linux.about.com. 2009-09-11. Archived (https://web.archive.org/web/20090430115615/http://linux.about.com/od/commands/l/blcmdl8_MAKEDEV.htm) from the original on 2009-04-30. Retrieved 2009-09-17.
6. Tanenbaum, Andrew; Woodhull, Albert (2006). *Operating Systems, Design and Implementation* (3rd. ed.). Upper Saddle River, NJ: Pearson Pretence Hall. p. 256. ISBN 0-13-142938-8.
7. Yosifovich, Pavel; Ionescu, Alex; Russinovich, Mark; Solomon, David (2017). *Windows Internals, Part 1* (Seventh ed.). Redmond, Washington: Microsoft Press. ISBN 978-0-7356-8418-8.

8. "Introduction to the User-Mode Driver Framework (UMDF)" (http://blogs.msdn.com/iliast/archive/2006/10/10/Introduction-to-the-User_2D00_Mode-Driver-Framework.aspx). Microsoft. 2006-10-10. Archived (https://web.archive.org/web/20100107015247/http://blogs.msdn.com/iliast/archive/2006/10/10/Introduction-to-the-User_2D00_Mode-Driver-Framework.aspx) from the original on 2010-01-07. Retrieved 2008-03-04.
9. Deborah Morley (2009). *Understanding Computers 2009: Today and Tomorrow* (https://books.google.com/books?id=tfU7RtI7OX8C&q=applications+device+driver). Cengage Learning. ISBN 9780324830132. Archived (https://web.archive.org/web/20210609160421/https://books.google.com/books?id=tfU7RtI7OX8C&q=applications+device+driver) from the original on 2021-06-09. Retrieved 2020-11-10.
10. *Computer Peripherals and Interfaces* (https://books.google.com/books?id=uK01xKT_mAMC&q=Simulating+work+with+hardware%2C+while+doing+something+entirely+different&pg=SA5-PA8). Technical Publications Pune. January 2008. pp. 5–8. ISBN 978-8184314748. Retrieved 2016-05-03.
11. "What are Device Drivers and why do we need them?" (http://www.drivers.com/update/drivers-news/what-are-device-drivers-and-why-do-we-need-them/). *drivers.com*. April 17, 2015. Archived (https://web.archive.org/web/20161120050252/http://www.drivers.com/update/drivers-news/what-are-device-drivers-and-why-do-we-need-them/) from the original on November 20, 2016. Retrieved March 19, 2018.
12. "CCISS" (https://web.archive.org/web/20100821112730/http://sourceforge.net/projects/cciss/). *SourceForge*. 2010. Archived from the original (http://sourceforge.net/projects/cciss/) on 2010-08-21. Retrieved 2010-08-11. "Drivers for the HP (previously Compaq) Smart Array controllers which provide hardware RAID capability."
13. Russell, Steve; et al. (2003-10-21). *Abbreviations and acronyms* (http://www.redbooks.ibm.com/redbooks/pdfs/sg246852.pd). IBM International Technical Support Organization. p. 207. ISBN 0-7384-2684-9. Retrieved 2011-08-14. {{cite book}}: |work= ignored (help)
14. "US Patent 5969841 - Gigabaud link module with received power detect signal" (https://web.archive.org/web/20110612215844/http://www.patentstorm.us/patents/5969841.html). PatentStorm LLC. Archived from the original (http://www.patentstorm.us/patents/5969841.html) on 2011-06-12. Retrieved 2009-09-08. "An improved Gigabaud Link Module (GLM) is provided for performing bi-directional data transfers between a host device and a serial transfer medium."
15. "Unified Audio Model (Windows CE 5.0)" (https://msdn.microsoft.com/en-us/library/ms894071.aspx). *Microsoft Developer Network*. 14 September 2012. Archived (https://web.archive.org/web/20170622222123/https://msdn.microsoft.com/en-us/library/ms894071.aspx) from the original on 2017-06-22. Retrieved 2016-09-19.
16. Dell US. "What are DCH drivers and why do you need to know about them? | Dell US" (https://www.dell.com/support/article/en-us/sln318535/what-are-dch-drivers-and-why-do-you-need-to-know-about-them?lang=en). *www.dell.com*. Retrieved 2020-10-29.
17. "dxd - dynax driver framework: Main Page" (https://dxd.dynax.at/). *dxd.dynax.at*. Archived (https://web.archive.org/web/20160529042017/https://dxd.dynax.at/) from the original on 2016-05-29. Retrieved 2016-09-19.
18. Talebi, Seyed Mohammadjavad Seyed; Tavakoli, Hamid; Zhang, Hang; Zhang, Zheng; Sani, Ardalan Amiri; Qian, Zhiyun (2018). *Charm: Facilitating Dynamic Analysis of Device Drivers of Mobile Systems* (https://www.usenix.org/conference/usenixsecurity18/presentation/talebi). pp. 291–307. ISBN 9781939133045. Archived (https://web.archive.org/web/20221105175041/https://www.usenix.org/conference/usenixsecurity18/presentation/talebi) from the original on 5 November 2022. Retrieved 5 November 2022.

19. Goodin, Dan (14 October 2022). "How a Microsoft blunder opened millions of PCs to potent malware attacks" (https://arstechnica.com/information-technology/2022/10/how-a-microsoft-blunder-opened-millions-of-pcs-to-potent-malware-attacks/). *Ars Technica*. Archived (https://web.archive.org/web/20221108164935/https://arstechnica.com/information-technology/2022/10/how-a-microsoft-blunder-opened-millions-of-pcs-to-potent-malware-attacks/) from the original on 8 November 2022. Retrieved 8 November 2022.
20. Ridley, Jacob (9 February 2022). "You're going to want to update your Wi-Fi and Bluetooth drivers today" (https://www.pcgamer.com/its-a-really-good-idea-to-update-your-wi-fi-and-bluetooth-drivers-today/). *PC Gamer*. Archived (https://web.archive.org/web/20221105175040/https://www.pcgamer.com/its-a-really-good-idea-to-update-your-wi-fi-and-bluetooth-drivers-today/) from the original on 5 November 2022. Retrieved 5 November 2022.
21. "Wireless 'BlueBorne' Attacks Target Billions of Bluetooth Devices" (https://threatpost.com/wireless-blueborne-attacks-target-billions-of-bluetooth-devices/127921/). *threatpost.com*. Archived (https://web.archive.org/web/20221105175043/https://threatpost.com/wireless-blueborne-attacks-target-billions-of-bluetooth-devices/127921/) from the original on 5 November 2022. Retrieved 5 November 2022.
22. Spadafora, Anthony (12 January 2022). "Installing gaming drivers might leave your PC vulnerable to cyberattacks" (https://www.techradar.com/news/installing-gaming-drivers-might-leave-your-pc-vulnerable-to-cyberattacks). *TechRadar*. Archived (https://web.archive.org/web/20221105175046/https://www.techradar.com/news/installing-gaming-drivers-might-leave-your-pc-vulnerable-to-cyberattacks) from the original on 5 November 2022. Retrieved 5 November 2022.
23. "HP patches vulnerable driver lurking in printers for 16 years" (https://www.zdnet.com/article/hp-patches-vulnerable-printer-driver-impacting-millions-of-devices/). *ZDNET*. Archived (https://web.archive.org/web/20221105175042/https://www.zdnet.com/article/hp-patches-vulnerable-printer-driver-impacting-millions-of-devices/) from the original on 5 November 2022. Retrieved 5 November 2022.
24. Pan, Jianfeng; Yan, Guanglu; Fan, Xiaocao (2017). *Digtool: A {Virtualization-Based} Framework for Detecting Kernel Vulnerabilities* (https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/pan). USENIX Association. pp. 149–165. ISBN 9781931971409. Retrieved 5 November 2022.
25. King, Bertel (18 June 2022). "Closed Source vs. Open Source Hardware Drivers: Why It Matters" (https://www.makeuseof.com/closed-source-vs-open-source-hardware-drivers/). *MUO*. Archived (https://web.archive.org/web/20221105175040/https://www.makeuseof.com/closed-source-vs-open-source-hardware-drivers/) from the original on 5 November 2022. Retrieved 5 November 2022.
26. Branscombe, Mary (7 April 2022). "How Microsoft blocks vulnerable and malicious drivers in Defender, third-party security tools and in Windows 11" (https://www.techrepublic.com/article/how-microsoft-blocks-vulnerable-malicious-drivers-defender-third-party-security-tools-windows-11/). *TechRepublic*. Archived (https://web.archive.org/web/20221105175043/https://www.techrepublic.com/article/how-microsoft-blocks-vulnerable-malicious-drivers-defender-third-party-security-tools-windows-11/) from the original on 5 November 2022. Retrieved 5 November 2022.
27. Goodin, Dan (5 October 2022). "No fix in sight for mile-wide loophole plaguing a key Windows defense for years" (https://arstechnica.com/information-technology/2022/10/no-fix-in-sight-for-mile-wide-loophole-plaguing-a-key-windows-defense-for-years/). *Ars Technica*. Archived (https://web.archive.org/web/20221105175043/https://arstechnica.com/information-technology/2022/10/no-fix-in-sight-for-mile-wide-loophole-plaguing-a-key-windows-defense-for-years/) from the original on 5 November 2022. Retrieved 5 November 2022.

28. Davenport, Corbin. " "Bring Your Own Vulnerable Driver" Attacks Are Breaking Windows" (https://www.howtogeek.com/820374/bring-your-own-vulnerable-driver-attacks-are-breaking-windows/). *How-To Geek*. Archived (https://web.archive.org/web/20221105175043/https://www.howtogeek.com/820374/bring-your-own-vulnerable-driver-attacks-are-breaking-windows/) from the original on 5 November 2022. Retrieved 5 November 2022.
29. "Windows 10 Security Alert: Vulnerabilities Found in Over 40 Drivers" (https://www.bleepingcomputer.com/news/security/windows-10-security-alert-vulnerabilities-found-in-over-40-drivers/). *BleepingComputer*. Archived (https://web.archive.org/web/20221105175043/https://www.bleepingcomputer.com/news/security/windows-10-security-alert-vulnerabilities-found-in-over-40-drivers/) from the original on 5 November 2022. Retrieved 5 November 2022.
30. "Fine-grained kernel isolation" (https://mars-research.github.io/projects/kernel-isolation/). *mars-research.github.io*. Archived (https://web.archive.org/web/20220915120154/https://mars-research.github.io/projects/kernel-isolation/) from the original on 15 September 2022. Retrieved 15 September 2022.
31. Fetzer, Mary. "Automatic device driver isolation protects against bugs in operating systems" (https://techxplore.com/news/2022-08-automatic-device-driver-isolation-bugs.html). *Pennsylvania State University via techxplore.com*. Archived (https://web.archive.org/web/20220915120154/https://techxplore.com/news/2022-08-automatic-device-driver-isolation-bugs.html) from the original on 15 September 2022. Retrieved 15 September 2022.
32. Huang, Yongzhe; Narayanan, Vikram; Detweiler, David; Huang, Kaiming; Tan, Gang; Jaeger, Trent; Burtsev, Anton (2022). "KSplit: Automating Device Driver Isolation" (https://mars-research.github.io/doc/ksplit-osdi22.pdf) (PDF). Archived (https://web.archive.org/web/20220915120155/https://mars-research.github.io/doc/ksplit-osdi22.pdf) (PDF) from the original on 15 September 2022. Retrieved 15 September 2022.
33. Wulf 1974 pp.337–345
34. Denning 1976
35. Swift 2005, p.29 quote: "isolation, resource control, decision verification (checking), and error recovery."

External links

- Windows Hardware Dev Center (https://developer.microsoft.com/en-us/windows/hardware)
- Linux Hardware Compatibility Lists and Linux Drivers (https://web.archive.org/web/20190819193521/https://www.linux-drivers.org/)
- Understanding Modern Device Drivers(Linux) (http://pages.cs.wisc.edu/~kadav/study/study.pdf)
- BinaryDriverHowto, Ubuntu. (https://help.ubuntu.com/community/BinaryDriverHowto/)
- Linux Drivers Source (https://github.com/torvalds/linux/tree/master/drivers)

WIKIPEDIA
The Free Encyclopedia

Graphics software

In computer graphics, **graphics software** refers to a program or collection of programs that enable a person to manipulate images or models visually on a computer.[1]

Computer graphics can be classified into two distinct categories: raster graphics and vector graphics, with further 2D and 3D variants. Many graphics programs focus exclusively on either vector or raster graphics, but there are a few that operate on both. It is simple to convert from vector graphics to raster graphics, but going the other way is harder. Some software attempts to do this.

In addition to static graphics, there are animation and video editing software. Different types of software are often designed to edit different types of graphics such as video, photos, and vector-based drawings. The exact sources of graphics may vary for different tasks, but most can read and write files.[2]

Most graphics programs have the ability to import and export one or more graphics file formats, including those formats written for a particular computer graphics program. Such programs include, but are not limited to: GIMP, Adobe Photoshop, CorelDRAW, Microsoft Publisher, Picasa, etc.[3]

The use of a swatch is a palette of active colours that are selected and rearranged by the preference of the user. A swatch may be used in a program or be part of the universal palette on an operating system. It is used to change the colour of a text or image and in video editing. Vector graphics animation can be described as a series of mathematical transformations that are applied in sequence to one or more shapes in a scene. Raster graphics animation works in a similar fashion to film-based animation, where a series of still images produces the illusion of continuous movement.

History

SuperPaint was one of the earliest graphics software applications, first conceptualized in 1972 and achieving its first stable image in 1973[4]

Fauve Matisse (later Macromedia xRes) was a pioneering program of the early 1990s, notably introducing layers in customer software.[5]

Currently Adobe Photoshop is one of the most used and best-known graphics programs in the Americas, having created more custom hardware solutions in the early 1990s, but was initially subject to various litigation. GIMP is a popular open-source alternative to Adobe Photoshop.

See also

- Comparison of raster graphics editors
- Comparison of vector graphics editors
- List of 2D graphics software
- List of 2D animation software

- List of raster graphics editors
- Graphic art software
- Image morphing software
- Image conversion
- imc FAMOS (1987), graphical data analysis
- Raster graphics editor
- Vector graphics editor

References

1. "What is Graphics Software?" (https://www.geeksforgeeks.org/what-is-graphics-software/). *GeeksforGeeks*. 2022-08-09. Retrieved 2023-10-18.
2. "Software Graphic: graphics software" (https://elearning.reb.rw/mod/forum/discuss.php?d=3834). *elearning.reb.rw*. Retrieved 2023-10-09.
3. "Software Graphic: Graphics Software" (https://elearning.reb.rw/mod/forum/discuss.php?d=3837). *elearning.reb.rw*. Retrieved 2023-10-09.
4. Alonso, Bogar (2013-05-28). "The 1970s Graphics Program That Spurred Space Exploration, Computer Picassos and Pixar" (https://www.vice.com/en/article/the-1970s-graphics-program-that-spurred-space-exploration-computer-picassos-and-pixar/). *Vice*. Retrieved 2023-10-09.
5. Macromedia Matisse (http://pages.swcp.com/rtoads/vivisect/software/matisse.html) Archived (https://web.archive.org/web/20170201140928/http://pages.swcp.com/rtoads/vivisect/software/matisse.html) 2017-02-01 at the Wayback Machine, Reign of Toads (http://pages.swcp.com/rtoads/) Archived (https://web.archive.org/web/20170211053135/http://pages.swcp.com/rtoads/) 2017-02-11 at the Wayback Machine

Medical software

(Redirected from Health software)

Medical software is any software item or system used within a medical context.[1][2][3] This can include:

- Standalone software used for diagnostic or therapeutic purposes.
- Software used by health care providers to reduce paperwork and offer digital services to patients, e.g., a patient portal.
- Software embedded in a medical device (often referred to as "medical device software").
- Software that drives a medical device or determines how it is used.
- Software that acts as an accessory to a medical device.
- Software used in the design, production, and testing of a medical device (or)
- Software that provides quality control management of a medical device.

History

Medical software has been in use since at least since the 1960s,[4] a time when the first computerized information-handling system in the hospital sphere was being considered by Lockheed.[5][6] As computing became more widespread and useful in the late 1970s and 1980s, the concept of "medical software" as a data and operations management tool in the medical industry—including in the physician's office—became more prevalent.[7][8] Medical software became more prominent in medical devices in fields such as nuclear medicine, cardiology, and medical robotics by the early 1990s, prompting additional scrutiny of the "safety-critical" nature of medical software in the research and legislative communities, in part fueled by the Therac-25 radiation therapy device scandal.[9][10]

The development of the ISO 9000-3 standard[9] as well as the European Medical Devices Directive in 1993[1] helped bring some harmonization of existing laws with medical devices and their associated software, and the addition of IEC 62304 in 2006 further cemented how medical device software should be developed and tested.[11] The U.S. Food and Drug Administration (FDA) has also offered guidance and driven regulation on medical software, particularly embedded in and used as medical devices.[2][12][13][14] There was an expansion of medical software innovation with the adoption of electronic health records (EHR) and availability of electronic clinical data. In the United States, substantial resources were allocated, starting with the HITECH Act of 2009. [15]

Medical device software

The global IEC 62304 standard on the software life cycle processes of medical device software states it is a "software system that has been developed for the purpose of being incorporated into the medical device being developed or that is intended for use as a medical device in its own right."[11] In the U.S., the FDA states that "any software that meets the legal definition of a [medical] device" is considered medical device software.[16] A similar "software can be a medical device" interpretation was also made by the European Union in 2007 with an update to its European Medical Devices Directive, when "used specifically for diagnostic and/or therapeutic purposes."[17]

Due to the broad scope covered by these terms, manifold classifications can be proposed for various medical software, based for instance on their technical nature (embedded in a device or standalone), on their level of safety (from the most trivial to the most safety-critical ones), or on their primary function (treatment, education, diagnostics, and/or data management).

A key distinction in medical software classification is between Software in a Medical Device (SiMD)and Software as a Medical Device (SaMD). SiMD refers to software that is essential for a medical device to function, such as control software for robotic surgical systems or firmware in diagnostic instruments. SaMD, on the other hand, operates independently of a hardware device and is designed to fulfill a medical purpose on its own.

A portable heart rate variability device is an example of a medical device that contains medical device software.

Software as a medical device

The dramatic increase in smartphone usage in the twenty-first century triggered the emergence of thousands of standalone health- and medical-related software apps, many falling into a gray or borderline area in terms of regulation.[18] While software embedded into a medical device was being addressed, medical software separate from medical hardware—referred to by the International Medical Device Regulators Forum (IMDRF) as "software as a medical device" or "SaMD"[19]—was falling through existing regulatory cracks.

In the U.S., the FDA eventually released new draft guidance in July 2011 on "mobile medical applications," with members of the legal community, such as Keith Barritt, speculating it should be read to imply "as applicable to all software, since the test for determining whether a mobile application is a regulated mobile 'medical' application is the same test one would use to determine if any software is regulated."[20] Examples of mobile apps potentially covered by the guidance included those that regulate an installed pacemaker or those that analyze images for cancerous lesions, X-rays and MRI, graphic data such as EEG waveforms as well as bedside monitors, urine analyzers, glucometer, stethoscopes, spirometers, BMI calculators, heart rate monitors, and body fat calculators.[21]

By the time its final guidance was released in late 2013, however, members of Congress began to be concerned about how the guidance would be used in the future, in particular with what it would mean to the SOFTWARE Act legislation that had recently been introduced.[22] Around the same time, the IMDRF was working on a more global perspective of SaMD with the release of its Key Definitions in December 2013, focused on "[establishing] a common framework for regulators to incorporate converged controls into their regulatory approaches for SaMD."[19] Aside from "not [being] necessary for a hardware medical device to achieve its intended medical purpose," the IMDRF also found that SaMD also could not drive a medical device, though it could be used as a module of or interfaced with one.[19] The group further developed quality management system principles for SaMD in 2015.[23]

Software in a medical device

Software in a medical device (SiMD) refers to software that is integral to the operation of a physical medical device. Unlike SaMD, which functions independently, SiMD is embedded within or necessary for the device's intended medical purpose. Examples include software that controls an artificial cardiac pacemaker, manages infusion pumps, or operates imaging systems like MRI machines.

The development and maintenance of SiMD are governed by international standards to ensure safety and efficacy. IEC 62304:2006 outlines the life cycle requirements for medical device software, establishing a framework for processes, activities, and tasks throughout the software's life cycle. Additionally, ISO 13485:2016 specifies requirements for a quality management system in the design and manufacture of medical devices, including software components.

International standards

IEC 62304 has become the benchmark standard for the development of medical device software, standalone or otherwise, in both the E.U. and the U.S.[3][24] Innovation in software technologies has led key industry leaders and government regulators to recognize the emergence of numerous standalone medical software products that operate as medical devices. This has been reflected in regulatory changes in the E.U. (European Medical Devices Directive[1]) and the U.S. (various FDA guidance documents[2][12][13][22]). Additionally, quality management system requirements for manufacturing a software medical device, as is the case with any medical device, are described in the U.S. Quality Systems Regulation[25] of the FDA and also in ISO 13485:2016.

Software technology manufacturers that operate within the software medical device space conduct mandatory development of their products in accordance with those requirements. Furthermore, though not mandatory, they may elect to obtain certification from a notified body, having implemented such quality system requirements as described within international standards such as ISO 13485:2016.

Further reading

- Babelotzky, W; Bohrt, C.; Choudhuri, J.; Handorn, B.; Heidenreich, G.; Neuder, K.; Neumann, G.; Prinz, T.; Rösch, A.; Spyra, G.; Stephan, S.; Wenner, H.; Wufka, M. (2018) *Development and Production of Medical Software : Standards in Medical Engineering* (https://books.google.com/books?id=CKjPvQEACAAJ&q=Development+and+Production+of+Medical+Software+:+Standards+in+Medical+Engineering). VDE VERLAG GMBH. pp. 1-207. ISBN 9783800745388.
- Becchetti, C.; Neri, A. (2013). "Chapter 6: Medical Software" (https://books.google.com/books?id=P3VeLRVMdQ8C&pg=PT275). *Medical Instrument Design and Development: From Requirements to Market Placements*. Chichester, U.K.: John Wiley & Sons Ltd. pp. 359–418. ISBN 9781119952404.
- Degoulet, P.; Fieschi, M. (2012). "Chapter 2: Medical Software Development" (https://books.google.com/books?id=obfDBAAAQBAJ&pg=PA26). *Introduction to Clinical Informatics*. New York: Springer Science & Business Media. pp. 19–34. ISBN 9781461268659.

See also

- Health informatics
- Health information technology
- Category: Medical software

External links

Media related to Medical software at Wikimedia Commons

References

1. Becchetti, C.; Neri, A. (2013). "Chapter 6: Medical Software" (https://books.google.com/books?id=P3VeLRVMdQ8C&pg=PT275). *Medical Instrument Design and Development: From Requirements to Market Placements*. Chichester, U.K.: John Wiley & Sons Ltd. pp. 359–418. ISBN 9781119952404.
2. Vogel, D.A. (2011). "Chapter 3: The FDA Software Validation Regulations and Why You Should Validate Software Anyway" (https://books.google.com/books?id=LYxH-zUSOTgC&pg=PA27). *Medical Device Software Verification, Validation, and Compliance*. Boston, MA: Artech House. pp. 27–36. ISBN 9781596934238.
3. Jetley, R.; Sudarsan, S.; R., Sampath; Ramaswamy, S. (2013). "Medical Software - Issues and Best Practices" (https://books.google.com/books?id=5I25BQAAQBAJ&pg=PA69). *Distributed Computing and Internet Technology: 9th International Conference, ICDCIT 2013, Bhubaneswar, India, February 5-8, 2013, Proceedings*. Hyderabad, India: Springer. pp. 69–91. ISBN 9783642360718.
4. "Radar and Electronics" (https://books.google.com/books?id=3WpVAAAAYAAJ&q=%22medical+software%22). Radar and Electronics Association. March 1963. Retrieved 26 April 2016.
5. *Lockheed Hospital Information System* (https://books.google.com/books?id=8vsFGwAACAAJ). Lockheed Aircraft Corporation. 1965. p. 82.
6. Gall, John E.; Norwood, Donald D.; El Camino Hospital (1977). *Demonstration and evaluation of a total hospital information system* (https://books.google.com/books?id=6RA7YzUXYg8C). NCHSR research summary series. U.S. Dept. of Health, Education, and Welfare, Public Health Service, Health Resources Administration, National Center for Health Services Research. p. 38.
7. Zimmerman, J.; Rector, A. (1978). *Computers for the Physician's Office* (https://books.google.com/books?id=8FprAAAAMAAJ). Forest Grove, OR: Research Studies Press. p. 305. ISBN 0893550078.
8. Freedman, E.; Hecht, E.; Whiteside, D. (1985). "Consultants Perspective on Medical Office Computerization". *Computers in Healthcare, Volume 6* (https://books.google.com/books?id=xW1LAAAAYAAJ). Englewood: Cardiff Publishing Company.
9. Cosgriff, P.S. (1994). "Quality assurance of medical software". *Journal of Medical Engineering & Technology*. **18** (1): 1–10. doi:10.3109/03091909409030782 (https://doi.org/10.3109%2F03091909409030782). PMID 8006924 (https://pubmed.ncbi.nlm.nih.gov/8006924).
10. Jones, P.; Jetley, R.; Abraham, J. (9 February 2010). "A Formal Methods-based verification approach to medical device software analysis" (https://www.embedded.com/design/prototyping-and-development/4008888/A-Formal-Methods-based-verification-approach-to-medical-device-software-analysis). *Embedded*. UBM. Retrieved 26 April 2016.

11. International Electrotechnical Commission (2006). "Medical device software – Software life cycle processes" (https://webstore.iec.ch/preview/info_iec62304%7Bed1.0%7Den_d.pdf) (PDF). *International Standard IEC 62304, First Edition 2006-05*. International Electrotechnical Commission. Retrieved 26 April 2016.
12. Office of Device Evaluation, Center for Devices and Radiological Health (9 September 1999). "Guidance for Industry, FDA Reviewers and Compliance on Off-The-Shelf Software Use in Medical Devices" (https://www.fda.gov/downloads/MedicalDevices/.../ucm073779.pdf) (PDF). U.S. Food and Drug Administration. Retrieved 26 April 2016.
13. Center for Devices; Radiological Health (11 May 2005). "Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices" (https://www.fda.gov/RegulatoryInformation/Guidances/ucm089543.htm). U.S. Food and Drug Administration. Retrieved 26 April 2016.
14. "Doctors near me" (https://www.smartclinics.com.au/doctors-near-me/).
15. "HITECH Act of 2009" (https://www.healthit.gov/topic/laws-regulation-and-policy/health-it-legislation).
16. Murray Jr., J.F. (March 2010). "CDRH Regulated Software: An Introduction" (https://www.fda.gov/downloads/Training/CDRHLearn/UCM209129.pdf) (PDF). U.S. Food and Drug Administration. Retrieved 26 April 2016.
17. "Directive 2007/47/ED of the European Parliament and of the Council" (http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2007:247:0021:0055:en:PDF) (PDF). *Official Journal of the European Union*. European Union. 5 September 2007. Retrieved 26 April 2016.
18. "Technology in healthcare: how software improves medical practice?" (https://www.labsexplorer.com/c/technology-in-healthcare-how-software-improves-medical-practice_216). *Labs Explorer*. Retrieved 5 May 2020.
19. Spanou, D. (9 December 2013). "Software as a Medical Device (SaMD): Key Definitions" (http://www.imdrf.org/docs/imdrf/final/technical/imdrf-tech-131209-samd-key-definitions-140901.pdf) (PDF). International Medical Device Regulators Forum. p. 9. Retrieved 26 April 2016.
20. "New FDA Draft Guidance Sheds Light On Regulation of 'Mobile Medical Apps' and Other Software" (https://web.archive.org/web/20131127030705/http://www.fr.com/files/Uploads/Documents/Barritt%2C%20Keith.%20BNA%20Medical%20Devices%20Law%20%26%20Industry%20Report.%20New%20FDA%20Draft%20Guidance.%2008-10-11.pdf) (PDF). *Medical Devices Law & Industry Report*. **5** (16): 1–3. August 2011. Archived from the original (http://www.fr.com/files/Uploads/Documents/Barritt,%20Keith.%20BNA%20Medical%20Devices%20Law%20&%20Industry%20Report.%20New%20FDA%20Draft%20Guidance.%2008-10-11.pdf) (PDF) on 27 November 2013. Retrieved 26 April 2016.
21. Yetisen, A.K.; Martinez-Hurtado, J.L.; Vasconcellos, F.C.; Simsekler, M.C.E.; Akram, M.S.; Lowe, C.R. (2014). "The regulation of mobile medical applications". *Lab on a Chip*. **14** (5): 833–840. doi:10.1039/C3LC51235E (https://doi.org/10.1039%2FC3LC51235E). PMID 24425070 (https://pubmed.ncbi.nlm.nih.gov/24425070).
22. Slabodkin, G. (20 November 2013). "Congress, FDA at odds over software as a medical device" (http://www.fiercemobilehealthcare.com/story/congress-fda-odds-over-software-medical-device/2013-11-20). *Fierce Mobile Healthcare*. Questex, LLC.
23. Mezher, M. (8 April 2015). "IMDRF Proposes QMS Principles for Software as a Medical Device" (https://www.raps.org/Regulatory-Focus/News/2015/04/08/21936/IMDRF-Proposes-QMS-Principles-for-Software-as-a-Medical-Device/). *Regulatory Focus*. Regulatory Affairs Professionals Society. Retrieved 26 April 2016.
24. Rust, P.; Flood, D.; McCaffery, F. (2015). "Software Process Improvement and Roadmapping – A Roadmap for Implementing IEC 62304 in Organizations Developing and Maintaining Medical Device Software" (https://books.google.com/books?id=u3zMCQAAQBAJ&pg=PA19). In Rout, T.; C'Connor, R.V.; Dorling, A. (eds.). *Software Process Improvement and Capability Determination*. Cham, Switzerland: Springer. pp. 19–32. doi:10.1007/978-3-319-19860-6_3 (https://doi.org/10.1007%2F978-3-319-19860-6_3). ISBN 9783319198606.

25. "Quality System (QS) Regulation/Medical Device Good Manufacturing Practices" (https://www.fda.gov/medicaldevices/deviceregulationandguidance/postmarketrequirements/qualitysystemsregulations/default.htm). U.S. Food and Drug Administration. 30 June 2014. Retrieved 26 April 2016.

Mathematical software

Mathematical software is software used to model, analyze or calculate numeric, symbolic or geometric data.[1]

Evolution of mathematical software

Numerical analysis and symbolic computation had been in most important place of the subject, but other kind of them is also growing now. A useful mathematical knowledge of such as algorism which exist before the invention of electronic computer, helped to mathematical software developing. On the other hand, by the growth of computing power (such as seeing on Moore's law), the new treatment (for example, a new kind of technique such as data assimilation which combined numerical analysis and statistics) needing conversely the progress of the mathematical science or applied mathematics.
The progress of mathematical information presentation such as TeX or MathML[2] will demand to evolution form *formula manipulation language* to true *mathematics manipulation language* (notwithstanding the problem that whether mathematical theory is inconsistent or not). And popularization of general purpose mathematical software, special purpose mathematical software[3] so called *one purpose software* which used special subject will alive with adapting for environment progress at normalization of platform. So the diversity of mathematical software will be kept.

Solvers

A solver is a piece of mathematical software, possibly in the form of a stand-alone computer program or as a software library, that 'solves' a mathematical problem. A solver takes problem descriptions in some sort of generic form and calculates their solution. In a solver, the emphasis is on creating a program or library that can easily be applied to other problems of similar type.

Software calculator

A software calculator allows the user to perform simple mathematical operations, like addition, multiplication, exponentiation and trigonometry. Data input is typically manual, and the output is a text label.

Computer algebra systems

Many mathematical suites are computer algebra systems that use symbolic mathematics. They are designed to solve classical algebra equations and problems in human readable notation.

Statistics

Many tools are available for statistical analysis of data. See also Comparison of statistical packages.

Theorem provers and proof assistants

Optimization software

Geometry

Numerical analysis

The Netlib repository contains various collections of software routines for numerical problems, mostly in Fortran and C. Commercial products implementing many different numerical algorithms include the IMSL, NMath and NAG libraries; a free alternative is the GNU Scientific Library. A different approach is taken by the Numerical Recipes library, where emphasis is placed on clear understanding of algorithms.

Many computer algebra systems (listed above) can also be used for numerical computations.

Music mathematics software

Music mathematics software utilizes mathematics to analyze or synthesize musical symbols and patterns.

- Musimat (by Gareth Loy)[4]

Websites

A growing number of mathematical software is available in web browsers, without the need to download or install any code.[5]

Programming libraries

Low-level mathematical libraries intended for use within other programming languages:

- GNU Multiple Precision Arithmetic Library (GMP), for high-performance arbitrary-precision arithmetic.
- Class Library for Numbers, a high-level C++ library for arbitrary-precision arithmetic.
- AMD Core Math Library, a software development library released by AMD
- Boost, for C++

List of mathematical software

- Desmos
- GeoGebra
- GNU Octave
- gnuplot
- KCalc
- Maple (software)
- Mathematica
- Mathcad
- Maxima (software)
- Microsoft Mathematics

- MuPAD
- MATLAB
- R (programming language)
- SageMath
- SciPy
- Matlab Simulink
- SymPy
- TeX
- TI-Nspire
- Wolfram Alpha

Mathematical notation software

- AUCTeX
- Authorea
- Apache OpenOffice Math
- AsciiMath
- Calligra Words - Formula editor
- CoCalc
- GeoGebra
- GNOME LaTeX
- GNU TeXmacs
- Gummi
- KaTeX
- Kile
- LaTeX
- LibreOffice Math
- LyX
- MathJax

- MathML

- MathType
- Notepad++
- Overleaf
- Scientific WorkPlace
- TeX
- Texmaker
- TeXnicCenter
- TeXShop
- TeXstudio
- TeXworks
- Verbosus
- Vim
- Visual Studio Code - LaTeX Workshop (https://github.com/James-Yu/latex-workshop/wiki)
- WinEdt
- WinFIG
- WinShell

Mathematical art software

- Apophysis
- Electric Sheep
- Fractint
- Fyre
- Mandelbulb
- MilkDrop
- openPlaG
- Ultra Fractal
- XaoS
- R / R Mandelbrot sets
- Sterling
- Bryce
- Picogen
- Terragen
- GeoGebra
- Desmos

- Grapher
- Winplot
- Processing
- Grasshopper 3D
- P5.js
- matplotlib

- gnuplot
- Inkscape spirograph
- Wolfram Mathematica
- Houdini
- MATLAB
- TouchDesigner
- Unity (with math plugins)[6][7][8][9][10]
- LaTeX (PGF/TikZ, PGFPlots)
- Manim
- D3.js

See also

- Computational mathematics
- Computer-Based Math
- Comparison of formula editors
- Gödel's incompleteness theorems
- List of information graphics software
- Manim - open-source Python mathematical animation and visualisation software
- Plot (graphics)
- Time complexity

References

1. See, e.g., the editorial charter (http://toms.acm.org/Charter.html) of the *ACM Transactions on Mathematical Software* or the problem taxonomy (http://gams.nist.gov/cgi-bin/serve.cgi) of the National Institute of Standards and Technology Guide to Available Mathematical Software (both retrieved 2015-02-15).
2. Both MathML and TeX may be only simple a kind of computer language which enable also to present the mathematical formula. However they also may be the mathematical software if the term of *software* interpreted as whole technology on how to use computer, at most wide sense.
3. Included your written script code on the general purpose mathematical software.
4. Musimathics website, freeware download (http://www.musimat.com/)
5. Internet Accessible Mathematical Computation (http://icm.mcs.kent.edu/research/iamc.html), Institute for Computational Mathematics, Kent State University, retrieved 2015-02-15.
6. https://assetstore.unity.com/packages/tools/utilities/ultimate-math-library-228481
7. https://assetstore.unity.com/packages/tools/gui/math-equation-writer-199520
8. https://assetstore.unity.com/packages/tools/game-toolkits/mathplus-library-72197
9. https://www.evilwizardstudios.com/smithy/
10. https://assetstore.unity.com/packages/3d/environments/neurofractals-pack-124436

External links

- Calcular (https://calcularboletadehonorarios.cl/) Database on mathematical software

WIKIPEDIA
The Free Encyclopedia

Productivity software

(Redirected from Office suite)

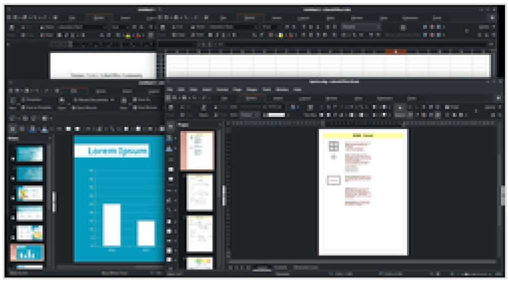
Productivity software (also called **personal productivity software** or **office productivity software**[1]) is application software used for producing information (such as documents, presentations, worksheets, databases, charts, graphs, digital paintings, electronic music and digital video).[2] Its names arose from it increasing productivity, especially of individual office workers, from typists to knowledge workers, although its scope is now wider than that. Office suites, which brought word processing, spreadsheet, and relational database programs to the desktop in the 1980s, are the core example of productivity software. They revolutionized the office with the magnitude of the productivity increase they brought as compared with the pre-1980s office environments of typewriters, paper filing, and handwritten lists and ledgers. In the United States, some 78% of "middle-skill" occupations (those that call for more than a high school diploma but less than a bachelor's degree) now require the use of productivity software.[3] In the 2010s, productivity software had become even more consumerized than it already was, as computing became ever more integrated into daily personal life.

Details

Productivity software traditionally runs directly on a computer. For example, Commodore Plus/4 model of computer contained in ROM for applications of productivity software. Productivity software is one of the reasons people use personal computers.

Office suite

An **office suite** is a bundle of productivity software (a software suite) intended to be used by office workers. The components are generally distributed together, have a consistent user interface and usually can interact with each other, sometimes in ways that the operating system would not normally allow.[4]



LibreOffice, an example of an office suite, showing Writer, Calc, Impress and Draw

The earliest office suite for personal computers was MicroPro International's StarBurst in the early 1980s, comprising the WordStar word processor, the CalcStar spreadsheet and the DataStar database software.[5] Other suites arose in the 1980s, and Microsoft Office came to dominate the market in the 1990s,[6] a position it retains as of 2024.

During the 1990s, office suite products gained popularity by offering bundles of applications that, when bought as part of a suite, effectively discounted the individual applications, with four or five applications being bundled for the price of two applications bought separately. When faced with such potential savings, customers could be "tempted by the suite, rather than the value of a particular product", and by 1994 more than 60 percent of the sales of Microsoft Word and around 70 percent of the sales of Microsoft

Excel were as part of sales of Microsoft Office. Such considerations had an impact on vendors of individual applications, often smaller companies, raising concerns that office suites were "stifling innovation", and even established vendors such as Borland and WordPerfect were having to adapt to the suite phenomenon, Borland ultimately deciding to sell its Quattro Pro spreadsheet to WordPerfect as the latter sought to assemble its own suite product. The dominant suite vendors, Microsoft and Lotus, downplayed competition and innovation concerns, claiming that users were still able to exercise choice and that "user-driven development" was guiding the evolution of office suites. Another view was that component-based software would eventually emerge, focusing development on more specialised components used by productivity software, empowering "a plethora of third-party developers", and that a "mix and match" approach of such components would adapt to the user's way of working.[7]

Office suite components

The base components of office suites are:

- Word processor
- Spreadsheet
- Presentation program

Other components include:

- Database software
- Graphics suite (raster graphics editor, vector graphics editor, image viewer)
- Desktop publishing software
- Formula editor
- Diagramming software
- Email client
- Communication software
- Personal information manager
- Notetaking
- Groupware
- Project management software
- Table (information)
- Web log analysis software

See also

- Integrated software
- List of office suites
- List of collaborative software
- List of personal information managers
- List of PDF software
- List of software that supports Office Open XML
- List of software that supports OpenDocument
- Comparison of office suites
- Comparison of word processors
- Comparison of spreadsheet software

- Comparison of note-taking software
- Online office suite
- Online spreadsheet
- Online word processor
- Wireless clicker

References

1. "Office Productivity Software" (https://www.pcmag.com/encyclopedia/term/66246/office-productivity-software). *PC Magazine Encyclopedia*. Ziff Davis. Retrieved 30 November 2014.
2. "Productivity Software" (https://www.pcmag.com/encyclopedia/term/49780/productivity-software). *PC Magazine Encyclopedia*. Ziff Davis. Retrieved 30 November 2014.
3. Crunched by the Numbers: The Digital Skills Gap in the Workforce (http://burning-glass.com/research/digital-skills-gap/), Burning Glass Technologies, March 2015
4. "Office Suite" (https://www.pcmag.com/encyclopedia/term/48311/office-suite). *PC Magazine Encyclopedia*. Ziff Davis. Retrieved 30 November 2014.
5. John C. Dvorak. "Whatever Happened to Wordstar?" (http://www.dvorak.org/blog/whatever-happened-to-wordstar-2/). Retrieved 2015-08-22.
6. *A Brief History of Computing,* by Gerard O'Regan, p. 87 (https://books.google.com/books?id=X7kWyJXB7z0C&pg=PA87)
7. Johnstone, Helen (July 1994). "A Bitter-Suite Situation". *Personal Computer World*. p. 236.

External links

WIKIPEDIA
The Free Encyclopedia

Operating system

An **operating system** (**OS**) is system software that manages computer hardware and software resources, and provides common services for computer programs.

Time-sharing operating systems schedule tasks for efficient use of the system and may also include accounting software for cost allocation of processor time, mass storage, peripherals, and other resources.

For hardware functions such as input and output and memory allocation, the operating system acts as an intermediary between programs and the computer hardware,[1][2] although the application code is usually executed directly by the hardware and frequently makes system calls to an OS function or is interrupted by it. Operating systems are found on many devices that contain a computer – from cellular phones and video game consoles to web servers and supercomputers.

As of September 2024, Android is the most popular operating system with a 46% market share, followed by Microsoft Windows at 26%, iOS and iPadOS at 18%, macOS at 5%, and Linux at 1%. Android, iOS, and iPadOS are mobile operating systems, while Windows, macOS, and Linux are desktop operating systems.[3] Linux distributions are dominant in the server and supercomputing sectors. Other specialized classes of operating systems (special-purpose operating systems),[4][5] such as embedded and real-time systems, exist for many applications. Security-focused operating systems also exist. Some operating systems have low system requirements (e.g. light-weight Linux distribution). Others may have higher system requirements.

Some operating systems require installation or may come pre-installed with purchased computers (OEM-installation), whereas others may run directly from media (i.e. live CD) or flash memory (i.e. a LiveUSB from a USB stick).

Definition and purpose

An operating system is difficult to define,[6] but has been called "the layer of software that manages a computer's resources for its users and their applications".[7] Operating systems include the software that is always running, called a kernel—but can include other software as well.[6][8] The two other types of programs that can run on a computer are system programs—which are associated with the operating system, but may not be part of the kernel—and applications—all other software.[8]

There are three main purposes that an operating system fulfills:[9]

- Operating systems allocate resources between different applications, deciding when they will receive central processing unit (CPU) time or space in memory.[9] On modern personal computers, users often want to run several applications at once. In order to ensure that one program cannot monopolize the computer's limited hardware resources, the operating system gives each application a share of the resource, either in time (CPU) or space (memory).[10][11] The operating system also must isolate applications from each other to protect them from errors and security vulnerabilities in another application's code, but enable communications between different applications.[12]

- Operating systems provide an interface that abstracts the details of accessing hardware details (such as physical memory) to make things easier for programmers.[9][13] Virtualization also enables the operating system to mask limited hardware resources; for example, virtual memory can provide a program with the illusion of nearly unlimited memory that exceeds the computer's actual memory.[14]
- Operating systems provide common services, such as an interface for accessing network and disk devices. This enables an application to be run on different hardware without needing to be rewritten.[15] Which services to include in an operating system varies greatly, and this functionality makes up the great majority of code for most operating systems.[16]

Types of operating systems

Multicomputer operating systems

With multiprocessors multiple CPUs share memory. A multicomputer or cluster computer has multiple CPUs, each of which has its own memory. Multicomputers were developed because large multiprocessors are difficult to engineer and prohibitively expensive;[17] they are universal in cloud computing because of the size of the machine needed.[18] The different CPUs often need to send and receive messages to each other;[19] to ensure good performance, the operating systems for these machines need to minimize this copying of packets.[20] Newer systems are often multiqueue—separating groups of users into separate queues—to reduce the need for packet copying and support more concurrent users.[21] Another technique is remote direct memory access, which enables each CPU to access memory belonging to other CPUs.[19] Multicomputer operating systems often support remote procedure calls where a CPU can call a procedure on another CPU,[22] or distributed shared memory, in which the operating system uses virtualization to generate shared memory that does not physically exist.[23]

Distributed systems

A distributed system is a group of distinct, networked computers—each of which might have their own operating system and file system. Unlike multicomputers, they may be dispersed anywhere in the world.[24] Middleware, an additional software layer between the operating system and applications, is often used to improve consistency. Although it functions similarly to an operating system, it is not a true operating system.[25]

Embedded

Embedded operating systems are designed to be used in embedded computer systems, whether they are internet of things objects or not connected to a network. Embedded systems include many household appliances. The distinguishing factor is that they do not load user-installed software. Consequently, they do not need protection between different applications, enabling simpler designs. Very small operating systems might run in less than 10 kilobytes,[26] and the smallest are for smart cards.[27] Examples include Embedded Linux, QNX, VxWorks, and the extra-small systems RIOT and TinyOS.[28]

Real-time

A real-time operating system is an operating system that guarantees to process events or data by or at a specific moment in time. Hard real-time systems require exact timing and are common in manufacturing, avionics, military, and other similar uses.[28] With soft real-time systems, the occasional missed event is acceptable; this category often includes audio or multimedia systems, as well as smartphones.[28] In order for hard real-time systems be sufficiently exact in their timing, often they are just a library with no protection between applications, such as eCos.[28]

Hypervisor

A hypervisor is an operating system that runs a virtual machine. The virtual machine is unaware that it is an application and operates as if it had its own hardware.[14][29] Virtual machines can be paused, saved, and resumed, making them useful for operating systems research, development,[30] and debugging.[31] They also enhance portability by enabling applications to be run on a computer even if they are not compatible with the base operating system.[14]

Library

A *library operating system* (libOS) is one in which the services that a typical operating system provides, such as networking, are provided in the form of libraries and composed with a single application and configuration code to construct a unikernel: [32] a specialized (only the absolute necessary pieces of code are extracted from libraries and bound together [33]), single address space, machine image that can be deployed to cloud or embedded environments.

The operating system code and application code are not executed in separated protection domains (there is only a single application running, at least conceptually, so there is no need to prevent interference between applications) and OS services are accessed via simple library calls (potentially inlining them based on compiler thresholds), without the usual overhead of context switches, [34] in a way similarly to embedded and real-time OSes. Note that this overhead is not negligible: to the direct cost of mode switching it's necessary to add the indirect pollution of important processor structures (like CPU caches, the instruction pipeline, and so on) which affects both user-mode and kernel-mode performance. [35]

History

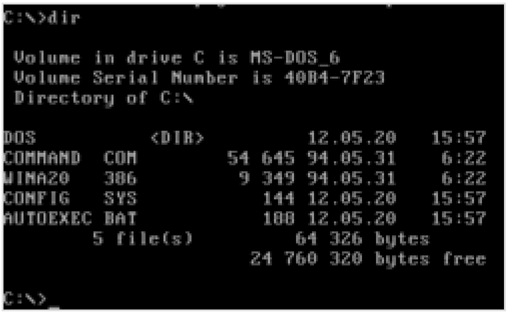
The first computers in the late 1940s and 1950s were directly programmed either with plugboards or with machine code inputted on media such as punch cards, without programming languages or operating systems.[36] After the introduction of the transistor in the mid-1950s, mainframes began to be built. These still needed professional operators[36] who manually do what a modern operating system would do, such as scheduling programs to run,[37] but mainframes still had rudimentary operating systems such as Fortran Monitor System (FMS) and IBSYS.[38] In the 1960s, IBM introduced the first series of intercompatible computers (System/360). All of them ran the same operating system—OS/360—which consisted of millions of lines of assembly language that had thousands of bugs. The OS/360 also was the first popular operating system to support multiprogramming, such that the CPU could be put to use on one job while another was waiting on input/output (I/O). Holding multiple jobs in memory necessitated memory partitioning and safeguards against one job accessing the memory allocated to a different one.[39]

IBM System/360 Model 50 operator's console and CPU; the operator's console is a terminal used by the operating system to communicate with the operator.

Around the same time, teleprinters began to be used as terminals so multiple users could access the computer simultaneously. The operating system MULTICS was intended to allow hundreds of users to access a large computer. Despite its limited adoption, it can be considered the precursor to cloud computing. The UNIX operating system originated as a development of MULTICS for a single user.[40] Because UNIX's source code was available, it became the basis of other, incompatible operating systems, of which the most successful were AT&T's System V and the University of California's Berkeley Software Distribution (BSD).[41] To increase compatibility, the IEEE released the POSIX standard for operating system application programming interfaces (APIs), which is supported by most UNIX systems. MINIX was a stripped-down version of UNIX, developed in 1987 for educational uses, that inspired the commercially available, free software Linux. Since 2008, MINIX is used in controllers of most Intel microchips, while Linux is widespread in data centers and Android smartphones.[42]

Microcomputers

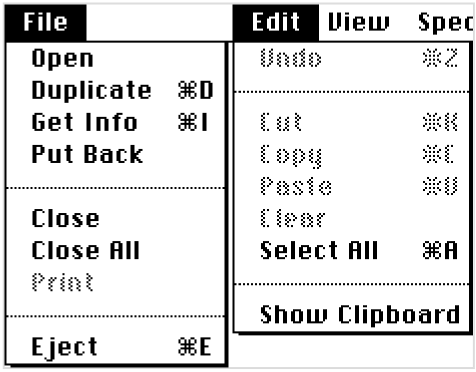
The invention of large scale integration enabled the production of personal computers (initially called microcomputers) from around 1980.[43] For around five years, the CP/M (Control Program for Microcomputers) was the most popular operating system for microcomputers.[44] Later, IBM bought the DOS (Disk Operating System) from Microsoft. After modifications requested by IBM, the resulting system was called MS-DOS (MicroSoft Disk Operating System) and was widely used on IBM microcomputers. Later versions increased their sophistication, in part by borrowing features from UNIX.[44]



Command-line interface of the MS-DOS operating system

Apple's Macintosh was the first popular computer to use a graphical user interface (GUI). The GUI proved much more user friendly than the text-only command-line interface earlier operating systems had used. Following the success of Macintosh, MS-DOS was updated with a GUI overlay called Windows. Windows later was rewritten as a stand-alone operating system, borrowing so many features from another (VAX VMS) that a large legal settlement was paid.[45] In the twenty-first century, Windows continues to be popular on personal computers but has less market share of servers. UNIX operating systems, especially Linux, are the most popular on enterprise systems and servers but are also used on mobile devices and many other computer systems.[46]

On mobile devices, Symbian OS was dominant at first, being usurped by BlackBerry OS (introduced 2002) and iOS for iPhones (from 2007). Later on, the open-source Android operating system (introduced 2008), with a Linux kernel and a C library (Bionic) partially based on BSD code, became most popular.[47]

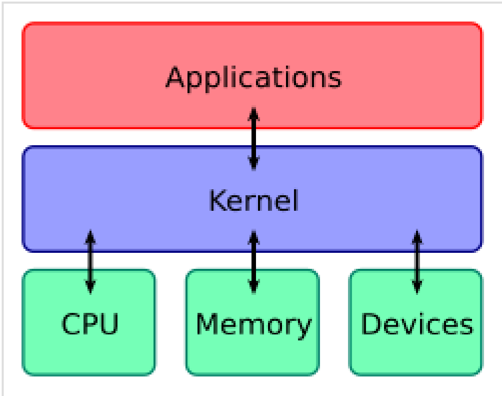


Graphical user interface of a Macintosh

Components

The components of an operating system are designed to ensure that various parts of a computer function cohesively. With the de facto obsoletion of DOS, all user software must interact with the operating system to access hardware.

Kernel



A kernel connects the application software to the hardware of a computer.

The kernel is the part of the operating system that provides protection between different applications and users. This protection is key to improving reliability by keeping errors isolated to one program, as well as security by limiting the power of malicious software and protecting private data, and ensuring that one program cannot monopolize the computer's resources.[48] Most operating systems have two modes of operation:[49] in user mode, the hardware checks that the software is only executing legal instructions, whereas the kernel has unrestricted powers and is not subject to these checks.[50] The kernel also manages memory for other processes and controls access to input/output devices.[51]

Program execution

The operating system provides an interface between an application program and the computer hardware, so that an application program can interact with the hardware only by obeying rules and procedures programmed into the operating system. The operating system is also a set of services which simplify development and execution of application programs. Executing an application program typically involves the creation of a process by the operating system kernel, which assigns memory space and other resources, establishes a priority for the process in multi-tasking systems, loads program binary code into memory, and initiates execution of the application program, which then interacts with the user and with hardware devices. However, in some systems an application can request that the operating system execute another application within the same process, either as a subroutine or in a separate thread, e.g., the **LINK** and **ATTACH** facilities of OS/360 and successors.

Interrupts

An interrupt (also known as an abort, exception, *fault*, signal,[52] or *trap*)[53] provides an efficient way for most operating systems to react to the environment. Interrupts cause the central processing unit (CPU) to have a control flow change away from the currently running program to an interrupt handler, also known as an interrupt service routine (ISR).[54][55] An interrupt service routine may cause the central processing unit (CPU) to have a context switch.[56][a] The details of how a computer processes an interrupt vary from architecture to architecture, and the details of how interrupt service routines behave vary from operating system to operating system.[57] However, several interrupt functions are common.[57] The architecture and operating system must:[57]

1. transfer control to an interrupt service routine.
2. save the state of the currently running process.
3. restore the state after the interrupt is serviced.

Software interrupt

A software interrupt is a message to a process that an event has occurred.[52] This contrasts with a *hardware interrupt* — which is a message to the central processing unit (CPU) that an event has occurred.[58] Software interrupts are similar to hardware interrupts — there is a change away from the currently running process.[59] Similarly, both hardware and software interrupts execute an interrupt service routine.

Software interrupts may be normally occurring events. It is expected that a time slice will occur, so the kernel will have to perform a context switch.[60] A computer program may set a timer to go off after a few seconds in case too much data causes an algorithm to take too long.[61]

Software interrupts may be error conditions, such as a malformed machine instruction.[61] However, the most common error conditions are division by zero and accessing an invalid memory address.[61]

Users can send messages to the kernel to modify the behavior of a currently running process.[61] For example, in the command-line environment, pressing the *interrupt character* (usually Control-C) might terminate the currently running process.[61]

To generate *software interrupts* for x86 CPUs, the INT assembly language instruction is available.[62] The syntax is `INT X`, where `X` is the offset number (in hexadecimal format) to the interrupt vector table.

Signal

To generate *software interrupts* in Unix-like operating systems, the `kill(pid,signum)` system call will send a signal to another process.[63] `pid` is the process identifier of the receiving process. `signum` is the signal number (in mnemonic format)[b] to be sent. (The abrasive name of `kill` was chosen because early implementations only terminated the process.)[64]

In Unix-like operating systems, *signals* inform processes of the occurrence of asynchronous events.[63] To communicate asynchronously, interrupts are required.[65] One reason a process needs to asynchronously communicate to another process solves a variation of the classic reader/writer problem.[66] The writer receives a pipe from the shell for its output to be sent to the reader's input stream.[67] The command-line syntax is `alpha | bravo`. `alpha` will write to the pipe when its computation is ready and then sleep in the wait queue.[68] `bravo` will then be moved to the ready queue and soon will read from its input stream.[69] The kernel will generate *software interrupts* to coordinate the piping.[69]

Signals may be classified into 7 categories.[63] The categories are:

1. when a process finishes normally.
2. when a process has an error exception.
3. when a process runs out of a system resource.
4. when a process executes an illegal instruction.
5. when a process sets an alarm event.
6. when a process is aborted from the keyboard.
7. when a process has a tracing alert for debugging.

Hardware interrupt

Input/output (I/O) devices are slower than the CPU. Therefore, it would slow down the computer if the CPU had to wait for each I/O to finish. Instead, a computer may implement interrupts for I/O completion, avoiding the need for polling or busy waiting.[70]

Some computers require an interrupt for each character or word, costing a significant amount of CPU time. Direct memory access (DMA) is an architecture feature to allow devices to bypass the CPU and access main memory directly.[71] (Separate from the architecture, a device may perform direct memory access[c] to and from main memory either directly or via a bus.)[72][d]

Input/output

Interrupt-driven I/O

When a computer user types a key on the keyboard, typically the character appears immediately on the screen. Likewise, when a user moves a mouse, the cursor immediately moves across the screen. Each keystroke and mouse movement generates an *interrupt* called *Interrupt-driven I/O*. An interrupt-driven I/O occurs when a process causes an interrupt for every character[72] or word[73] transmitted.

Direct memory access

Devices such as hard disk drives, solid-state drives, and magnetic tape drives can transfer data at a rate high enough that interrupting the CPU for every byte or word transferred, and having the CPU transfer the byte or word between the device and memory, would require too much CPU time. Data is, instead, transferred between the device and memory independently of the CPU by hardware such as a channel or a direct memory access controller; an interrupt is delivered only when all the data is transferred.[74]

If a computer program executes a system call to perform a block I/O *write* operation, then the system call might execute the following instructions:

- Set the contents of the CPU's registers (including the program counter) into the process control block.[75]

- Create an entry in the device-status table.[76] The operating system maintains this table to keep track of which processes are waiting for which devices. One field in the table is the memory address of the process control block.
- Place all the characters to be sent to the device into a memory buffer.[65]
- Set the memory address of the memory buffer to a predetermined device register.[77]
- Set the buffer size (an integer) to another predetermined register.[77]
- Execute the machine instruction to begin the writing.
- Perform a context switch to the next process in the ready queue.

While the writing takes place, the operating system will context switch to other processes as normal. When the device finishes writing, the device will *interrupt* the currently running process by *asserting* an interrupt request. The device will also place an integer onto the data bus.[78] Upon accepting the interrupt request, the operating system will:

- Push the contents of the program counter (a register) followed by the status register onto the call stack.[57]
- Push the contents of the other registers onto the call stack. (Alternatively, the contents of the registers may be placed in a system table.)[78]
- Read the integer from the data bus. The integer is an offset to the interrupt vector table. The vector table's instructions will then:
    - Access the device-status table.
    - Extract the process control block.
    - Perform a context switch back to the writing process.

When the writing process has its time slice expired, the operating system will:[79]

- Pop from the call stack the registers other than the status register and program counter.
- Pop from the call stack the status register.
- Pop from the call stack the address of the next instruction, and set it back into the program counter.

With the program counter now reset, the interrupted process will resume its time slice.[57]

Memory management

Among other things, a multiprogramming operating system kernel must be responsible for managing all system memory which is currently in use by the programs. This ensures that a program does not interfere with memory already in use by another program. Since programs time share, each program must have independent access to memory.

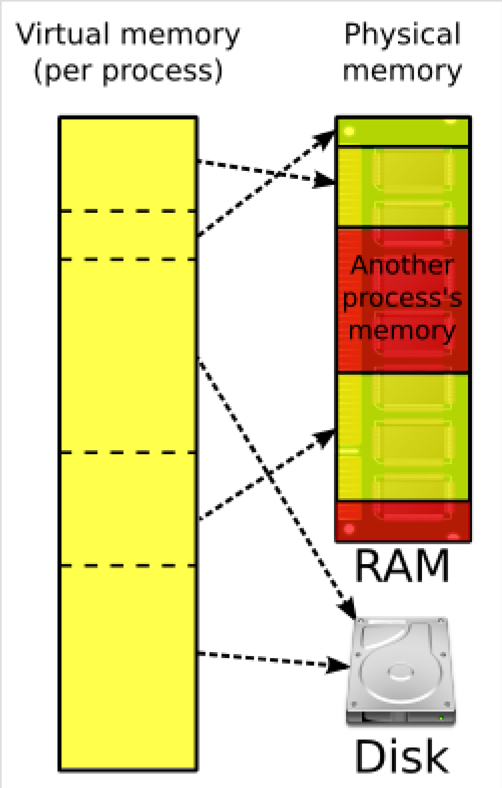
Cooperative memory management, used by many early operating systems, assumes that all programs make voluntary use of the kernel's memory manager, and do not exceed their allocated memory. This system of memory management is almost never seen anymore, since programs often contain bugs which can cause them to exceed their allocated memory. If a program fails, it may cause memory used by one or more other programs to be affected or overwritten. Malicious programs or viruses may purposefully alter another program's memory, or may affect the operation of the operating system itself. With cooperative memory management, it takes only one misbehaved program to crash the system.

Memory protection enables the kernel to limit a process' access to the computer's memory. Various methods of memory protection exist, including memory segmentation and paging. All methods require some level of hardware support (such as the 80286 MMU), which does not exist in all computers.

In both segmentation and paging, certain protected mode registers specify to the CPU what memory address it should allow a running program to access. Attempts to access other addresses trigger an interrupt, which causes the CPU to re-enter supervisor mode, placing the kernel in charge. This is called a segmentation violation or Seg-V for short, and since it is both difficult to assign a meaningful result to such an operation, and because it is usually a sign of a misbehaving program, the kernel generally resorts to terminating the offending program, and reports the error.

Windows versions 3.1 through ME had some level of memory protection, but programs could easily circumvent the need to use it. A general protection fault would be produced, indicating a segmentation violation had occurred; however, the system would often crash anyway.

Virtual memory



Many operating systems can "trick" programs into using memory scattered around the hard disk and RAM as if it is one continuous chunk of memory, called virtual memory.

The use of virtual memory addressing (such as paging or segmentation) means that the kernel can choose what memory each program may use at any given time, allowing the operating system to use the same memory locations for multiple tasks.

If a program tries to access memory that is not accessible[e] memory, but nonetheless has been allocated to it, the kernel is interrupted . This kind of interrupt is typically a page fault.

When the kernel detects a page fault it generally adjusts the virtual memory range of the program which triggered it, granting it access to the memory requested. This gives the kernel discretionary power over where a particular application's memory is stored, or even whether or not it has been allocated yet.

In modern operating systems, memory which is accessed less frequently can be temporarily stored on a disk or other media to make that space available for use by other programs. This is called swapping, as an area of memory can be used by multiple programs, and what that memory area contains can be swapped or exchanged on demand.

Virtual memory provides the programmer or the user with the perception that there is a much larger amount of RAM in the computer than is really there.[80]

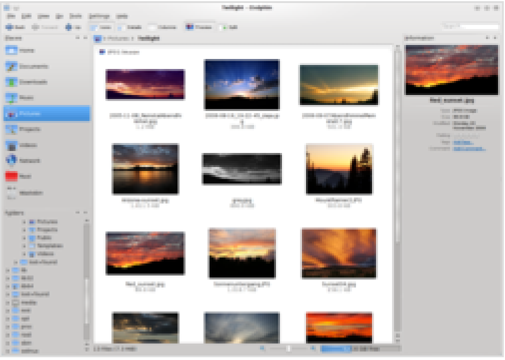
Concurrency

Concurrency refers to the operating system's ability to carry out multiple tasks simultaneously.[81] Virtually all modern operating systems support concurrency.[82]

Threads enable splitting a process' work into multiple parts that can run simultaneously.[83] The number of threads is not limited by the number of processors available. If there are more threads than processors, the operating system kernel schedules, suspends, and resumes threads, controlling when each thread runs and how much CPU time it receives.[84] During a context switch a running thread is suspended, its state is saved into the thread control block and stack, and the state of the new thread is loaded in.[85] Historically, on many systems a thread could run until it relinquished control (cooperative multitasking). Because this model can allow a single thread to monopolize the processor, most operating systems now can interrupt a thread (preemptive multitasking).[86]

Threads have their own thread ID, program counter (PC), a register set, and a stack, but share code, heap data, and other resources with other threads of the same process.[87][88] Thus, there is less overhead to create a thread than a new process.[89] On single-CPU systems, concurrency is switching between processes. Many computers have multiple CPUs.[90] Parallelism with multiple threads running on different CPUs can speed up a program, depending on how much of it can be executed concurrently.[91]

File system



File systems allow users and programs to organize and sort files on a computer, often through the use of directories (or folders).

Permanent storage devices used in twenty-first century computers, unlike volatile dynamic random-access memory (DRAM), are still accessible after a crash or power failure. Permanent (non-volatile) storage is much cheaper per byte, but takes several orders of magnitude longer to access, read, and write.[92][93] The two main technologies are a hard drive consisting of magnetic disks, and flash memory (a solid-state drive that stores data in electrical circuits). The latter is more expensive but faster and more durable.[94][95]

File systems are an abstraction used by the operating system to simplify access to permanent storage. They provide human-readable filenames and other metadata, increase performance via amortization of accesses, prevent multiple threads from accessing the same section of memory, and include checksums to identify corruption.[96] File systems are composed of files (named collections of data, of an arbitrary size) and directories (also called folders) that list human-readable filenames and other directories.[97] An absolute file path begins at the root directory and lists subdirectories divided by punctuation, while a relative path defines the location of a file from a directory.[98][99]

System calls (which are sometimes wrapped by libraries) enable applications to create, delete, open, and close files, as well as link, read, and write to them. All these operations are carried out by the operating system on behalf of the application.[100] The operating system's efforts to reduce latency include storing recently requested blocks of memory in a cache and prefetching data that the application has not asked for, but might need next.[101] Device drivers are software specific to each input/output (I/O) device that enables the operating system to work without modification over different hardware.[102][103]

Another component of file systems is a dictionary that maps a file's name and metadata to the data block where its contents are stored.[104] Most file systems use directories to convert file names to file numbers. To find the block number, the operating system uses an index (often implemented as a tree).[105] Separately, there is a free space map to track free blocks, commonly implemented as a bitmap.[105] Although any free block can be used to store a new file, many operating systems try to group together files in the same directory to maximize performance, or periodically reorganize files to reduce fragmentation.[106]

Maintaining data reliability in the face of a computer crash or hardware failure is another concern.[107] File writing protocols are designed with atomic operations so as not to leave permanent storage in a partially written, inconsistent state in the event of a crash at any point during writing.[108] Data corruption is addressed by redundant storage (for example, RAID—redundant array of inexpensive disks)[109][110] and checksums to detect when data has been corrupted. With multiple layers of checksums and backups of a file, a system can recover from multiple hardware failures. Background processes are often used to detect and recover from data corruption.[110]

Security

Security means protecting users from other users of the same computer, as well as from those who seeking remote access to it over a network.[111] Operating systems security rests on achieving the CIA triad: confidentiality (unauthorized users cannot access data), integrity (unauthorized users cannot modify data), and availability (ensuring that the system remains available to authorized users, even in the event of a denial of service attack).[112] As with other computer systems, isolating security domains—in the case of operating systems, the kernel, processes, and virtual machines—is key to achieving security.[113] Other ways to increase security include simplicity to minimize the attack surface, locking access to resources by default, checking all requests for authorization, principle of least authority (granting the minimum privilege essential for performing a task), privilege separation, and reducing shared data.[114]

Some operating system designs are more secure than others. Those with no isolation between the kernel and applications are least secure, while those with a monolithic kernel like most general-purpose operating systems are still vulnerable if any part of the kernel is compromised. A more secure design features microkernels that separate the kernel's privileges into many separate security domains and reduce the consequences of a single kernel breach.[115] Unikernels are another approach that improves security by minimizing the kernel and separating out other operating systems functionality by application.[115]

Most operating systems are written in C or C++, which create potential vulnerabilities for exploitation. Despite attempts to protect against them, vulnerabilities are caused by buffer overflow attacks, which are enabled by the lack of bounds checking.[116] Hardware vulnerabilities, some of them caused by CPU optimizations, can also be used to compromise the operating system.[117] There are known instances of operating system programmers deliberately implanting vulnerabilities, such as back doors.[118]

Operating systems security is hampered by their increasing complexity and the resulting inevitability of bugs.[119] Because formal verification of operating systems may not be feasible, developers use operating system hardening to reduce vulnerabilities,[120] e.g. address space layout randomization, control-flow integrity,[121] access restrictions,[122] and other techniques.[123] There are no restrictions on who can contribute code to open source operating systems; such operating systems have transparent change histories and distributed governance structures.[124] Open source developers strive to work collaboratively to find and eliminate security vulnerabilities, using code review and type checking to expunge malicious code.[125][126] Andrew S. Tanenbaum advises releasing the source code of all operating systems, arguing that it prevents developers from placing trust in secrecy and thus relying on the unreliable practice of security by obscurity.[127]

User interface

A user interface (UI) is essential to support human interaction with a computer. The two most common user interface types for any computer are

- command-line interface, where computer commands are typed, line-by-line,
- graphical user interface (GUI) using a visual environment, most commonly a combination of the window, icon, menu, and pointer elements, also known as WIMP.

For personal computers, including smartphones and tablet computers, and for workstations, user input is typically from a combination of keyboard, mouse, and trackpad or touchscreen, all of which are connected to the operating system with specialized software.[128] Personal computer users who are not software developers or coders often prefer GUIs for both input and output; GUIs are supported by most personal computers.[129] The software to support GUIs is more complex than a command line for input and plain text output. Plain text output is often preferred by programmers, and is easy to support.[130]

Operating system development as a hobby

A hobby operating system may be classified as one whose code has not been directly derived from an existing operating system, and has few users and active developers.[131]

In some cases, hobby development is in support of a "homebrew" computing device, for example, a simple single-board computer powered by a 6502 microprocessor. Or, development may be for an architecture already in widespread use. Operating system development may come from entirely new concepts, or may commence by modeling an existing operating system. In either case, the hobbyist is her/his own developer, or may interact with a small and sometimes unstructured group of individuals who have like interests.

Examples of hobby operating systems include Syllable and TempleOS.

Diversity of operating systems and portability

If an application is written for use on a specific operating system, and is ported to another OS, the functionality required by that application may be implemented differently by that OS (the names of functions, meaning of arguments, etc.) requiring the application to be adapted, changed, or otherwise maintained.

This cost in supporting operating systems diversity can be avoided by instead writing applications against software platforms such as Java or Qt. These abstractions have already borne the cost of adaptation to specific operating systems and their system libraries.

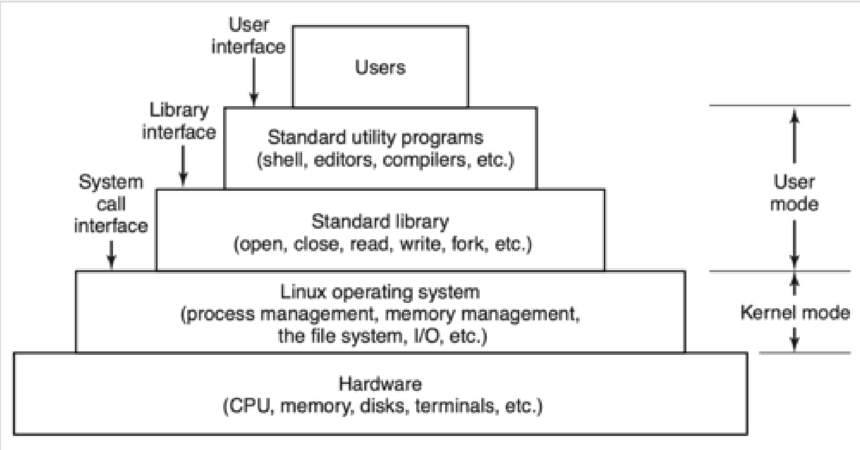
Another approach is for operating system vendors to adopt standards. For example, POSIX and OS abstraction layers provide commonalities that reduce porting costs.

Popular operating systems

As of September 2024, Android (based on the Linux kernel) is the most popular operating system with a 46% market share, followed by Microsoft Windows at 26%, iOS and iPadOS at 18%, macOS at 5%, and Linux at 1%. Android, iOS, and iPadOS are mobile operating systems, while Windows, macOS, and Linux are desktop operating systems.[3]

Linux

Linux is a free software distributed under the GNU General Public License (GPL), which means that all of its derivatives are legally required to release their source code.[132] Linux was designed by programmers for their own use, thus emphasizing simplicity and consistency, with a small number of basic elements that can be combined in nearly unlimited ways, and avoiding redundancy.[133]

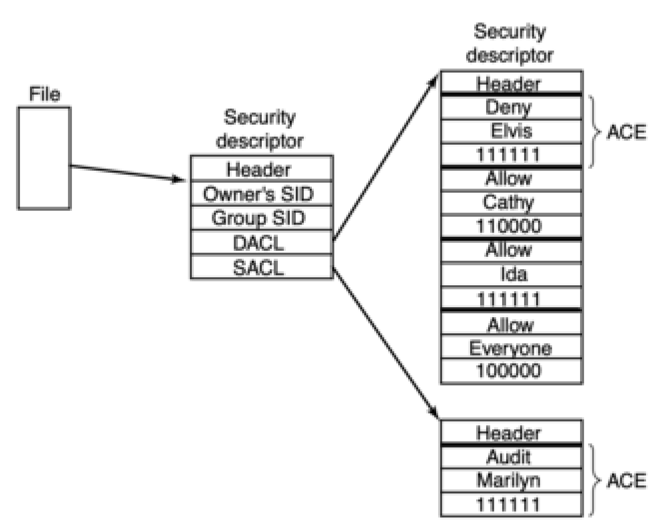


Layers of a Linux system

Its design is similar to other UNIX systems not using a microkernel.[134] It is written in C[135] and uses UNIX System V syntax, but also supports BSD syntax. Linux supports standard UNIX networking features, as well as the full suite of UNIX tools, while supporting multiple users and employing preemptive multitasking. Initially of a minimalist design, Linux is a flexible system that can work in under 16 MB of RAM, but still is used on large multiprocessor systems.[134] Similar to other UNIX systems, Linux distributions are composed of a kernel, system libraries, and system utilities.[136] Linux has a graphical user interface (GUI) with a desktop, folder and file icons, as well as the option to access the operating system via a command line.[137]

Android is a partially open-source operating system closely based on Linux and has become the most widely used operating system by users, due to its popularity on smartphones and, to a lesser extent, embedded systems needing a GUI, such as "smart watches, automotive dashboards, airplane seatbacks, medical devices, and home appliances".[138] Unlike Linux, much of Android is written in Java and uses object-oriented design.[139]

Microsoft Windows



Security descriptor for a file that is read-only by default, specified no access for Elvis, read/write access for Cathy, and full access for Ida, the owner of the file[140]

Windows is a proprietary operating system that is widely used on desktop computers, laptops, tablets, phones, workstations, enterprise servers, and Xbox consoles.[141] The operating system was designed for "security, reliability, compatibility, high performance, extensibility, portability, and international support"—later on, energy efficiency and support for dynamic devices also became priorities.[142]

Windows Executive works via kernel-mode objects for important data structures like processes, threads, and sections (memory objects, for example files).[143] The operating system supports demand paging of virtual memory, which speeds up I/O for many applications. I/O device drivers use the Windows Driver Model.[143] The NTFS file system has a master table and each file is represented as a record with metadata.[144] The scheduling includes preemptive multitasking.[145] Windows has many security features;[146] especially important are the use of access-control lists and integrity levels. Every process has an authentication token and each object is given a security descriptor. Later releases have added even more security features.[144]

See also

- Comparison of operating systems
- DBOS
- Interruptible operating system
- List of operating systems
- List of pioneers in computer science
- Glossary of operating systems terms
- Microcontroller
- Network operating system
- Object-oriented operating system
- Lisp machine
- Operating System Projects
- System Commander
- System image
- Timeline of operating systems

Notes

a. Modern CPUs provide instructions (e.g. SYSENTER) to invoke selected kernel services without an interrupts. Visit https://wiki.osdev.org/SYSENTER for more information.

b. Examples include SIGINT, SIGSEGV, and SIGBUS.
c. often in the form of a DMA chip for smaller systems and I/O channels for larger systems
d. Modern motherboards have a DMA controller. Additionally, a device may also have one. Visit SCSI RDMA Protocol.
e. There are several reasons that the memory might be inaccessible
   - The address might be out of range
   - The address might refer to a page or segment that has been moved to a backing store
   - The address might refer to memory that has restricted access due to, e.g., key, ring.

References

1. Stallings (2005). *Operating Systems, Internals and Design Principles*. Pearson: Prentice Hall. p. 6.
2. Dhotre, I.A. (2009). *Operating Systems*. Technical Publications. p. 1.
3. "Operating System Market Share Worldwide" (https://gs.statcounter.com/os-market-share). *StatCounter Global Stats*. Retrieved 20 December 2024.
4. "VII. Special-Purpose Systems - Operating System Concepts, Seventh Edition [Book]" (https://www.oreilly.com/library/view/operating-system-concepts/9780471694663/pt07.html). *www.oreilly.com*. Archived (https://web.archive.org/web/20210613190049/https://www.oreilly.com/library/view/operating-system-concepts/9780471694663/pt07.html) from the original on 13 June 2021. Retrieved 8 February 2021.
5. "Special-Purpose Operating Systems - RWTH AACHEN UNIVERSITY Institute for Automation of Complex Power Systems - English" (https://www.acs.eonerc.rwth-aachen.de/cms/E-ON-ERC-ACS/Studium/Lehrveranstaltungen/~lrhs/Spezial-Betriebssysteme/?lidx=1). *www.acs.eonerc.rwth-aachen.de*. Archived (https://web.archive.org/web/20210614034001/https://www.acs.eonerc.rwth-aachen.de/cms/E-ON-ERC-ACS/Studium/Lehrveranstaltungen/~lrhs/Spezial-Betriebssysteme/?lidx=1) from the original on 14 June 2021. Retrieved 8 February 2021.
6. Tanenbaum & Bos 2023, p. 4.
7. Anderson & Dahlin 2014, p. 6.
8. Silberschatz et al. 2018, p. 6.
9. Anderson & Dahlin 2014, p. 7.
10. Anderson & Dahlin 2014, pp. 9–10.
11. Tanenbaum & Bos 2023, pp. 6–7.
12. Anderson & Dahlin 2014, p. 10.
13. Tanenbaum & Bos 2023, p. 5.
14. Anderson & Dahlin 2014, p. 11.
15. Anderson & Dahlin 2014, pp. 7, 9, 13.
16. Anderson & Dahlin 2014, pp. 12–13.
17. Tanenbaum & Bos 2023, p. 557.
18. Tanenbaum & Bos 2023, p. 558.
19. Tanenbaum & Bos 2023, p. 565.
20. Tanenbaum & Bos 2023, p. 562.
21. Tanenbaum & Bos 2023, p. 563.
22. Tanenbaum & Bos 2023, p. 569.
23. Tanenbaum & Bos 2023, p. 571.
24. Tanenbaum & Bos 2023, p. 579.

25. Tanenbaum & Bos 2023, p. 581.
26. Tanenbaum & Bos 2023, pp. 37–38.
27. Tanenbaum & Bos 2023, p. 39.
28. Tanenbaum & Bos 2023, p. 38.
29. Silberschatz et al. 2018, pp. 701.
30. Silberschatz et al. 2018, pp. 705.
31. Anderson & Dahlin 2014, p. 12.
32. Madhavapeddy, Anil; Scott, David J (November 2013). "Unikernels: Rise of the Virtual Library Operating System: What if all the software layers in a virtual appliance were compiled within the same safe, high-level language framework?" (https://doi.org/10.1145/2557963.2566628). *Queue*. Vol. 11, no. 11. New York, NY, USA: ACM. pp. 30–44. doi:10.1145/2557963.2566628 (https://doi.org/10.1145%2F2557963.2566628). ISSN 1542-7730 (https://search.worldcat.org/issn/1542-7730). Retrieved 7 August 2024.
33. "Build Process - Unikraft" (https://unikraft.org/docs/concepts/build-process). Archived (https://web.archive.org/web/20240422183734/https://unikraft.org/docs/concepts/build-process) from the original on 22 April 2024. Retrieved 8 August 2024.
34. "Leave your OS at home: the rise of library operating systems" (https://www.sigarch.org/leave-your-os-at-home-the-rise-of-library-operating-systems/). ACM SIGARCH. 14 September 2017. Archived (https://web.archive.org/web/20240301072916/https://www.sigarch.org/leave-your-os-at-home-the-rise-of-library-operating-systems/) from the original on 1 March 2024. Retrieved 7 August 2024.
35. Soares, Livio Baldini; Stumm, Michael (4 October 2010). *FlexSC: Flexible System Call Scheduling with Exception-Less System Calls* (https://www.usenix.org/conference/osdi10/flexsc-flexible-system-call-scheduling-exception-less-system-calls). OSDI '10, 9th USENIX Symposium on Operating System Design and Implementation (https://www.usenix.org/legacy/events/osdi10/). USENIX. Retrieved 9 August 2024. p. 2: "Synchronous implementation of system calls negatively impacts the performance of system intensive workloads, both in terms of the *direct* costs of mode switching and, more interestingly, in terms of the *indirect* pollution of important processor structures which affects both user-mode and kernel-mode performance. A motivating example that quantifies the impact of system call pollution on application performance can be seen in Figure 1. It depicts the user-mode instructions per cycles (kernel cycles and instructions are ignored) of one of the SPEC CPU 2006 benchmarks (Xalan) immediately before and after a `pwrite` system call. There is a significant drop in instructions per cycle (IPC) due to the system call, and it takes up to 14,000 cycles of execution before the IPC of this application returns to its previous level. As we will show, this performance degradation is mainly due to interference caused by the kernel on key processor structures."
36. Tanenbaum & Bos 2023, p. 8.
37. Arpaci-Dusseau, Remzi; Arpaci-Dusseau, Andrea (2015). *Operating Systems: Three Easy Pieces* (http://pages.cs.wisc.edu/~remzi/OSTEP/). Archived (https://web.archive.org/web/20160725012948/http://pages.cs.wisc.edu/~remzi/OSTEP/) from the original on 25 July 2016. Retrieved 25 July 2016.
38. Tanenbaum & Bos 2023, p. 10.
39. Tanenbaum & Bos 2023, pp. 11–12.
40. Tanenbaum & Bos 2023, pp. 13–14.
41. Tanenbaum & Bos 2023, pp. 14–15.
42. Tanenbaum & Bos 2023, p. 15.
43. Tanenbaum & Bos 2023, pp. 15–16.
44. Tanenbaum & Bos 2023, p. 16.
45. Tanenbaum & Bos 2023, p. 17.

46. Tanenbaum & Bos 2023, p. 18.
47. Tanenbaum & Bos 2023, pp. 19–20.
48. Anderson & Dahlin 2014, pp. 39–40.
49. Tanenbaum & Bos 2023, p. 2.
50. Anderson & Dahlin 2014, pp. 41, 45.
51. Anderson & Dahlin 2014, pp. 52–53.
52. Kerrisk, Michael (2010). *The Linux Programming Interface*. No Starch Press. p. 388. ISBN 978-1-59327-220-3. "A signal is a notification to a process that an event has occurred. Signals are sometimes described as software interrupts."
53. Hyde, Randall (1996). "Chapter Seventeen: Interrupts, Traps and Exceptions (Part 1)" (https://www.plantation-productions.com/Webster/www.artofasm.com/DOS/ch17/CH17-1.html#HEADING1-0). *The Art Of Assembly Language Programming*. No Starch Press. Archived (https://web.archive.org/web/20211222205623/https://www.plantation-productions.com/Webster/www.artofasm.com/DOS/ch17/CH17-1.html#HEADING1-0) from the original on 22 December 2021. Retrieved 22 December 2021. "The concept of an interrupt is something that has expanded in scope over the years. The 80x86 family has only added to the confusion surrounding interrupts by introducing the int (software interrupt) instruction. Indeed, different manufacturers have used terms like exceptions, faults, aborts, traps and interrupts to describe the phenomena this chapter discusses. Unfortunately there is no clear consensus as to the exact meaning of these terms. Different authors adopt different terms to their own use."
54. Tanenbaum, Andrew S. (1990). *Structured Computer Organization, Third Edition* (https://archive.org/details/structuredcomput00tane/page/308). Prentice Hall. p. 308 (https://archive.org/details/structuredcomput00tane/page/308). ISBN 978-0-13-854662-5. "Like the trap, the interrupt stops the running program and transfers control to an interrupt handler, which performs some appropriate action. When finished, the interrupt handler returns control to the interrupted program."
55. Silberschatz, Abraham (1994). *Operating System Concepts, Fourth Edition*. Addison-Wesley. p. 32. ISBN 978-0-201-50480-4. "When an interrupt (or trap) occurs, the hardware transfers control to the operating system. First, the operating system preserves the state of the CPU by storing registers and the program counter. Then, it determines which type of interrupt has occurred. For each type of interrupt, separate segments of code in the operating system determine what action should be taken."
56. Silberschatz, Abraham (1994). *Operating System Concepts, Fourth Edition*. Addison-Wesley. p. 105. ISBN 978-0-201-50480-4. "Switching the CPU to another process requires saving the state of the old process and loading the saved state for the new process. This task is known as a context switch."
57. Silberschatz, Abraham (1994). *Operating System Concepts, Fourth Edition*. Addison-Wesley. p. 31. ISBN 978-0-201-50480-4.
58. Silberschatz, Abraham (1994). *Operating System Concepts, Fourth Edition*. Addison-Wesley. p. 30. ISBN 978-0-201-50480-4. "Hardware may trigger an interrupt at any time by sending a signal to the CPU, usually by way of the system bus."
59. Kerrisk, Michael (2010). *The Linux Programming Interface*. No Starch Press. p. 388. ISBN 978-1-59327-220-3. "Signals are analogous to hardware interrupts in that they interrupt the normal flow of execution of a program; in most cases, it is not possible to predict exactly when a signal will arrive."
60. Kerrisk, Michael (2010). *The Linux Programming Interface*. No Starch Press. p. 388. ISBN 978-1-59327-220-3. "Among the types of events that cause the kernel to generate a signal for a process are the following: A software event occurred. For example, ... the process's CPU time limit was exceeded[.]"
61. Kerrisk, Michael (2010). *The Linux Programming Interface*. No Starch Press. p. 388. ISBN 978-1-59327-220-3.

62. "Intel® 64 and IA-32 Architectures Software Developer's Manual" (https://www.intel.com/content/dam/www/public/us/en/documents/manuals/64-ia-32-architectures-software-developer-instruction-set-reference-manual-325383.pdf) (PDF). Intel Corporation. September 2016. p. 610. Archived (https://web.archive.org/web/20220323231921/https://www.intel.com/content/dam/www/public/us/en/documents/manuals/64-ia-32-architectures-software-developer-instruction-set-reference-manual-325383.pdf) (PDF) from the original on 23 March 2022. Retrieved 5 May 2022.
63. Bach, Maurice J. (1986). *The Design of the UNIX Operating System*. Prentice-Hall. p. 200. ISBN 0-13-201799-7.
64. Kerrisk, Michael (2010). *The Linux Programming Interface*. No Starch Press. p. 400. ISBN 978-1-59327-220-3.
65. Tanenbaum, Andrew S. (1990). *Structured Computer Organization, Third Edition* (https://archive.org/details/structuredcomput00tane/page/308). Prentice Hall. p. 308 (https://archive.org/details/structuredcomput00tane/page/308). ISBN 978-0-13-854662-5.
66. Silberschatz, Abraham (1994). *Operating System Concepts, Fourth Edition*. Addison-Wesley. p. 182. ISBN 978-0-201-50480-4.
67. Haviland, Keith; Salama, Ben (1987). *UNIX System Programming*. Addison-Wesley Publishing Company. p. 153. ISBN 0-201-12919-1.
68. Haviland, Keith; Salama, Ben (1987). *UNIX System Programming*. Addison-Wesley Publishing Company. p. 148. ISBN 0-201-12919-1.
69. Haviland, Keith; Salama, Ben (1987). *UNIX System Programming*. Addison-Wesley Publishing Company. p. 149. ISBN 0-201-12919-1.
70. Tanenbaum, Andrew S. (1990). *Structured Computer Organization, Third Edition* (https://archive.org/details/structuredcomput00tane/page/292). Prentice Hall. p. 292 (https://archive.org/details/structuredcomput00tane/page/292). ISBN 978-0-13-854662-5.
71. IBM (September 1968), "Main Storage" (http://bitsavers.org/pdf/ibm/360/princOps/A22-6821-7_360PrincOpsDec67.pdf#page=8) (PDF), *IBM System/360 Principles of Operation* (http://bitsavers.org/pdf/ibm/360/princOps/A22-6821-7_360PrincOpsDec67.pdf) (PDF), Eighth Edition, p. 7, archived (https://web.archive.org/web/20220319083255/http://bitsavers.org/pdf/ibm/360/princOps/A22-6821-7_360PrincOpsDec67.pdf) (PDF) from the original on 19 March 2022, retrieved 13 April 2022
72. Tanenbaum, Andrew S. (1990). *Structured Computer Organization, Third Edition* (https://archive.org/details/structuredcomput00tane/page/294). Prentice Hall. p. 294 (https://archive.org/details/structuredcomput00tane/page/294). ISBN 978-0-13-854662-5.
73. "Program Interrupt Controller (PIC)" (http://bitsavers.org/pdf/dec/pdp7/F-75_PDP-7userHbk_Jun65.pdf#page=62) (PDF). *Users Handbook - PDP-7* (http://bitsavers.org/pdf/dec/pdp7/F-75_PDP-7userHbk_Jun65.pdf) (PDF). Digital Equipment Corporation. 1965. pp. 48 (http://bitsavers.org/pdf/dec/pdp7/F-75_PDP-7userHbk_Jun65.pdf#page=63). F-75. Archived (https://web.archive.org/web/20220510164742/http://bitsavers.org/pdf/dec/pdp7/F-75_PDP-7userHbk_Jun65.pdf) (PDF) from the original on 10 May 2022. Retrieved 20 April 2022.
74. *PDP-1 Input-Output Systems Manual* (http://bitsavers.org/pdf/dec/pdp1/F25_PDP1_IO.pdf) (PDF). Digital Equipment Corporation. pp. 19–20. Archived (https://web.archive.org/web/20190125050839/http://bitsavers.org/pdf/dec/pdp1/F25_PDP1_IO.pdf) (PDF) from the original on 25 January 2019. Retrieved 16 August 2022.
75. Silberschatz, Abraham (1994). *Operating System Concepts, Fourth Edition*. Addison-Wesley. p. 32. ISBN 978-0-201-50480-4.
76. Silberschatz, Abraham (1994). *Operating System Concepts, Fourth Edition*. Addison-Wesley. p. 34. ISBN 978-0-201-50480-4.
77. Tanenbaum, Andrew S. (1990). *Structured Computer Organization, Third Edition* (https://archive.org/details/structuredcomput00tane/page/295). Prentice Hall. p. 295 (https://archive.org/details/structuredcomput00tane/page/295). ISBN 978-0-13-854662-5.

78. Tanenbaum, Andrew S. (1990). *Structured Computer Organization, Third Edition* (https://archive.org/details/structuredcomput00tane/page/309). Prentice Hall. p. 309 (https://archive.org/details/structuredcomput00tane/page/309). ISBN 978-0-13-854662-5.
79. Tanenbaum, Andrew S. (1990). *Structured Computer Organization, Third Edition* (https://archive.org/details/structuredcomput00tane/page/310). Prentice Hall. p. 310 (https://archive.org/details/structuredcomput00tane/page/310). ISBN 978-0-13-854662-5.
80. Stallings, William (2008). *Computer Organization & Architecture*. New Delhi: Prentice-Hall of India Private Limited. p. 267. ISBN 978-81-203-2962-1.
81. Anderson & Dahlin 2014, p. 129.
82. Silberschatz et al. 2018, p. 159.
83. Anderson & Dahlin 2014, p. 130.
84. Anderson & Dahlin 2014, p. 131.
85. Anderson & Dahlin 2014, pp. 157, 159.
86. Anderson & Dahlin 2014, p. 139.
87. Silberschatz et al. 2018, p. 160.
88. Anderson & Dahlin 2014, p. 183.
89. Silberschatz et al. 2018, p. 162.
90. Silberschatz et al. 2018, pp. 162–163.
91. Silberschatz et al. 2018, p. 164.
92. Anderson & Dahlin 2014, pp. 492, 517.
93. Tanenbaum & Bos 2023, pp. 259–260.
94. Anderson & Dahlin 2014, pp. 517, 530.
95. Tanenbaum & Bos 2023, p. 260.
96. Anderson & Dahlin 2014, pp. 492–493.
97. Anderson & Dahlin 2014, p. 496.
98. Anderson & Dahlin 2014, pp. 496–497.
99. Tanenbaum & Bos 2023, pp. 274–275.
100. Anderson & Dahlin 2014, pp. 502–504.
101. Anderson & Dahlin 2014, p. 507.
102. Anderson & Dahlin 2014, p. 508.
103. Tanenbaum & Bos 2023, p. 359.
104. Anderson & Dahlin 2014, p. 545.
105. Anderson & Dahlin 2014, p. 546.
106. Anderson & Dahlin 2014, p. 547.
107. Anderson & Dahlin 2014, pp. 589, 591.
108. Anderson & Dahlin 2014, pp. 591–592.
109. Tanenbaum & Bos 2023, pp. 385–386.
110. Anderson & Dahlin 2014, p. 592.
111. Tanenbaum & Bos 2023, pp. 605–606.
112. Tanenbaum & Bos 2023, p. 608.
113. Tanenbaum & Bos 2023, p. 609.
114. Tanenbaum & Bos 2023, pp. 609–610.
115. Tanenbaum & Bos 2023, p. 612.
116. Tanenbaum & Bos 2023, pp. 648, 657.
117. Tanenbaum & Bos 2023, pp. 668–669, 674.
118. Tanenbaum & Bos 2023, pp. 679–680.

119. Tanenbaum & Bos 2023, pp. 605, 617–618.
120. Tanenbaum & Bos 2023, pp. 681–682.
121. Tanenbaum & Bos 2023, p. 683.
122. Tanenbaum & Bos 2023, p. 685.
123. Tanenbaum & Bos 2023, p. 689.
124. Richet & Bouaynaya 2023, p. 92.
125. Richet & Bouaynaya 2023, pp. 92–93.
126. Berntsso, Strandén & Warg 2017, pp. 130–131.
127. Tanenbaum & Bos 2023, p. 611.
128. Tanenbaum & Bos 2023, pp. 396, 402.
129. Tanenbaum & Bos 2023, pp. 395, 408.
130. Tanenbaum & Bos 2023, p. 402.
131. Holwerda, Thom (20 December 2009). "My OS Is Less Hobby than Yours" (https://www.osnews.com/story/22638/my-os-is-less-hobby-than-yours/). *OS News*. Retrieved 4 June 2024.
132. Silberschatz et al. 2018, pp. 779–780.
133. Tanenbaum & Bos 2023, pp. 713–714.
134. Silberschatz et al. 2018, p. 780.
135. Vaughan-Nichols, Steven (2022). "Linus Torvalds prepares to move the Linux kernel to modern C" (https://www.zdnet.com/article/linus-torvalds-prepares-to-move-the-linux-kernel-to-modern-c/). *ZDNET*. Retrieved 7 February 2024.
136. Silberschatz et al. 2018, p. 781.
137. Tanenbaum & Bos 2023, pp. 715–716.
138. Tanenbaum & Bos 2023, pp. 793–794.
139. Tanenbaum & Bos 2023, p. 793.
140. Tanenbaum & Bos 2023, pp. 1021–1022.
141. Tanenbaum & Bos 2023, p. 871.
142. Silberschatz et al. 2018, p. 826.
143. Tanenbaum & Bos 2023, p. 1035.
144. Tanenbaum & Bos 2023, p. 1036.
145. Silberschatz et al. 2018, p. 821.
146. Silberschatz et al. 2018, p. 827.

Further reading

- Anderson, Thomas; Dahlin, Michael (2014). *Operating Systems: Principles and Practice*. Recursive Books. ISBN 978-0-9856735-2-9.
- Auslander, M. A.; Larkin, D. C.; Scherr, A. L. (September 1981). "The Evolution of the MVS Operating System". *IBM Journal of Research and Development*. **25** (5): 471–482. doi:10.1147/rd.255.0471 (https://doi.org/10.1147%2Frd.255.0471). ISSN 0018-8646 (https://search.worldcat.org/issn/0018-8646).
- Berntsson, Petter Sainio; Strandén, Lars; Warg, Fredrik (2017). *Evaluation of Open Source Operating Systems for Safety-Critical Applications*. Springer International Publishing. pp. 117–132. ISBN 978-3-319-65948-0.
- Deitel, Harvey M.; Deitel, Paul; Choffnes, David (25 December 2015). *Operating Systems* (https://archive.org/details/modernoperatings00tane). Pearson/Prentice Hall. ISBN 978-0-13-092641-8.
- Bic, Lubomur F.; Shaw, Alan C. (2003). *Operating Systems*. Pearson: Prentice Hall.

- Silberschatz, Avi; Galvin, Peter; Gagne, Greg (2008). *Operating Systems Concepts*. John Wiley & Sons. ISBN 978-0-470-12872-5.
- O'Brien, J. A., & Marakas, G. M.(2011). *Management Information Systems*. 10e. McGraw-Hill Irwin.
- Leva, Alberto; Maggio, Martina; Papadopoulos, Alessandro Vittorio; Terraneo, Federico (2013). *Control-based Operating System Design*. IET. ISBN 978-1-84919-609-3.
- Richet, Jean-Loup; Bouaynaya, Wafa (2023). "Understanding and Managing Complex Software Vulnerabilities: An Empirical Analysis of Open-Source Operating Systems" (https://www.cairn.info/revue-systemes-d-information-et-management-2023-1-page-87.htm.). *Systèmes d'information & management*. **28** (1): 87–114. doi:10.54695/sim.28.1.0087 (https://doi.org/10.54695%2Fsim.28.1.0087) (inactive 1 November 2024).
- Silberschatz, Abraham; Galvin, Peter B.; Gagne, Greg (2018). *Operating System Concepts* (https://archive.org/details/operating-system-concepts-10th) (10 ed.). Wiley. ISBN 978-1-119-32091-3.
- Tanenbaum, Andrew S.; Bos, Herbert (2023). *Modern Operating Systems, Global Edition*. Pearson Higher Ed. ISBN 978-1-292-72789-9.

External links

- Multics History (http://www.cbi.umn.edu/iterations/haigh.html) and the history of operating systems

Video game

A **video game**[a] or **computer game** is an electronic game that involves interaction with a user interface or input device (such as a joystick, controller, keyboard, or motion sensing device) to generate visual feedback from a display device, most commonly shown in a video format on a television set, computer monitor, flat-panel display or touchscreen on handheld devices, or a virtual reality headset. Most modern video games are audiovisual, with audio complement delivered through speakers or headphones, and sometimes also with other types of sensory feedback (e.g., haptic technology that provides tactile sensations). Some video games also allow microphone and webcam inputs for in-game chatting and livestreaming.

First-generation *Pong* console at the Computerspielemuseum Berlin

Video games are typically categorized according to their hardware platform, which traditionally includes arcade video games, console games, and computer games (which includes LAN games, online games, and browser games). More recently, the video game industry has expanded onto mobile gaming through mobile devices (such as smartphones and tablet computers), virtual and augmented reality systems, and remote cloud gaming. Video games are also classified into a wide range of genres based on their style of gameplay and target audience.

The first video game prototypes in the 1950s and 1960s were simple extensions of electronic games using video-like output from large, room-sized mainframe computers. The first consumer video game was the arcade video game *Computer Space* in 1971, which took inspiration from the earlier 1962 computer game *Spacewar!*. In 1972 came the now-iconic video game *Pong* and the first home console, the Magnavox Odyssey. The industry grew quickly during the "golden age" of arcade video games from the late 1970s to early 1980s but suffered from the crash of the North American video game market in 1983 due to loss of publishing control and saturation of the market. Following the crash, the industry matured, was dominated by Japanese companies such as Nintendo, Sega, and Sony, and established practices and methods around the development and distribution of video games to prevent a similar crash in the future, many of which continue to be followed. In the 2000s, the core industry centered on "AAA" games, leaving little room for riskier experimental games. Coupled with the availability of the Internet and digital distribution, this gave room for independent video game development (or "indie games") to gain prominence into the 2010s. Since then, the commercial importance of the video game industry has been increasing. The emerging Asian markets and proliferation of smartphone games in particular are altering player demographics towards casual gaming and increasing monetization by incorporating games as a service.

Today, video game development requires numerous skills, vision, teamwork, and liaisons between different parties, including developers, publishers, distributors, retailers, hardware manufacturers, and other marketers, to successfully bring a game to its consumers. As of 2020, the global video game market had estimated annual revenues of US$159 billion across hardware, software, and services, which is three times the size of the global music industry and four times that of the film industry in 2019,[1] making it a formidable heavyweight across the modern entertainment industry. The video game market is also a major influence behind the electronics industry, where personal computer component, console, and peripheral sales, as well as consumer demands for better game performance, have been powerful driving factors for hardware design and innovation.

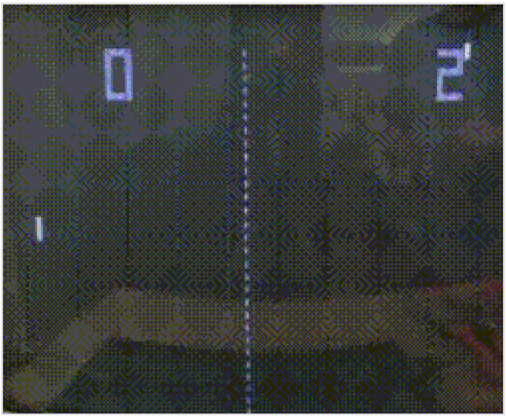
Origins

Early video games use interactive electronic devices with various display formats. The earliest example is from 1947—a "cathode-ray tube amusement device" was filed for a patent on 25 January 1947, by Thomas T. Goldsmith Jr. and Estle Ray Mann, and issued on 14 December 1948, as U.S. Patent 2455992.[2] Inspired by radar display technology, it consists of an analog device allowing a user to control the parabolic arc of a dot on the screen to simulate a missile being fired at targets, which are paper drawings fixed to the screen.[3] Other early examples include Christopher Strachey's draughts game, the Nimrod computer at the 1951 Festival of Britain; *OXO*, a tic-tac-toe computer game by Alexander S. Douglas for the EDSAC in 1952; *Tennis for Two*, an electronic interactive game engineered by William Higinbotham in 1958; and *Spacewar!*, written by Massachusetts Institute of Technology students Martin Graetz, Steve Russell, and Wayne Wiitanen's on a DEC PDP-1 computer in 1962. Each game has different means of display: NIMROD has a panel of lights to play the game of Nim,[4] *OXO* has a graphical display to play tic-tac-toe,[5] *Tennis for Two* has an oscilloscope to display a side view of a tennis court,[3] and *Spacewar!* has the DEC PDP-1's vector display to have two spaceships battle each other.[6]

Tennis for Two (1958), an early analog computer game that used an oscilloscope for a display

Spacewar! (1962), an early mainframe computer game, pictured running on a PDP-1 computer

These inventions laid the foundation for modern video games. In 1966, while working at Sanders Associates, Ralph H. Baer devised a system to play a basic table tennis game on a television screen. With the company's approval, Baer created the prototype known as the "Brown Box". Sanders patented Baer's innovations and licensed them to Magnavox, which commercialized the technology as the first home video game console, the Magnavox Odyssey, released in 1972.[7][8] Separately, Nolan Bushnell and Ted Dabney, inspired by seeing *Spacewar!* running at Stanford University, devised a similar version running in a smaller coin-operated arcade cabinet using a less expensive computer. This was released as *Computer Space*, the first arcade video game, in 1971.[9] Bushnell and Dabney went on to form Atari, Inc., and with Allan Alcorn, created their second arcade game in 1972, the hit ping pong-style *Pong*, which was directly inspired by the table tennis game on the Odyssey. Atari made a home version of *Pong*, which was released by Christmas 1975.[3] The success of the Odyssey and *Pong*, both as an arcade game and home machine, launched the video game industry.[10][11] Both Baer and Bushnell have been titled "Father of Video Games" for their contributions.[12][13]



Pong (1972), one of the earliest arcade video games

Terminology

The term "video game" was developed to describe electronic games played on a video display rather than on a teletype printer, audio speaker, or similar device.[14] This also distinguished from handheld electronic games such as *Merlin*, which commonly used LED lights for indicators not in combination for imaging purposes.[15]

Ralph H. Baer in 2009

Nolan Bushnell in 2013

"Computer game" may also be used as a descriptor, as all these types of games essentially require the use of a computer processor; in some cases, it is used interchangeably with "video game".[16] Particularly in the United Kingdom and Western Europe, this is common due to the historic relevance of domestically produced microcomputers. Other terms used include digital game, for example, by the Australian Bureau of Statistics.[17] The term "computer game" can also refer to PC games, which are played primarily on personal computers or other flexible hardware systems, to distinguish them from console games, arcade games, or mobile games.[15][14]

Other terms, such as "television game", "telegame", or "TV game", had been used in the 1970s and early 1980s, particularly for home gaming consoles that rely on connection to a television set.[18] However, these terms were also used interchangeably with "video game" in the 1970s, primarily due to "video" and "television" being synonymous.[19] In Japan, where consoles like the Odyssey were first imported and then made within the country by the large television manufacturers such as Toshiba and Sharp Corporation, such games are known as "TV games", "TV geemu", or "terebi geemu".[20] The term "TV game" is still commonly used into the 21st century.[20][21] "Electronic game" may also be used to refer to video games, but this also incorporates devices like early handheld electronic games that lack any video output.[16]

The first appearance of the term "video game" emerged around 1973. The Oxford English Dictionary cited a 10 November 1973 *BusinessWeek* article as the first printed use of the term.[22] Though Bushnell believed the term came from a vending magazine review of *Computer Space* in 1971,[23] a review of the major vending magazines *Vending Times* and *Cashbox* showed that the term may have come even earlier, appearing first in a letter dated July 10, 1972. In the letter, Bushnell uses the term "video game" twice.[24] Per video game historian Keith Smith, the sudden appearance suggested that the term had been proposed and readily adopted by those in the field. Around March 1973, Ed Adlum, who ran *Cashbox*'s coin-operated section until 1972 and then later founded *RePlay Magazine*, covering the coin-op amusement field, in 1975, used the term in an article in March 1973. In a September 1982 issue of *RePlay*, Adlum is credited with first naming these games as "video games": "RePlay's Eddie Adlum worked at 'Cash Box' when 'TV games' first came out. The personalities in those days were Bushnell, his sales manager Pat Karns, and a handful of other 'TV game' manufacturers like Henry Leyser and the McEwan brothers. It seemed awkward to call their products 'TV games', so borrowing a word from *Billboard*'s description of movie jukeboxes, Adlum started to refer to this new breed of amusement machine as 'video games.' The phrase stuck." Adlum explained in 1985 that up until the early 1970s, amusement arcades typically had non-video arcade games such as pinball machines and electro-mechanical games. With the arrival of video games in arcades during the early 1970s, there was initially some confusion in the arcade industry over what term should be used to describe the new games. He "wrestled with descriptions of this type of game," alternating between "TV game" and "television game" but "finally woke up one day" and said, "What the hell... video game!"[25]

Definition

While many games readily fall into a clear, well-understood definition of video games, new genres and innovations in game development have raised the question of what are the essential factors of a video game that separate the medium from other forms of entertainment.

The introduction of interactive films in the 1980s with games like *Dragon's Lair*, featured games with full motion video played off a form of media but only limited user interaction.[26] This had required a means to distinguish these games from more traditional board games that happen to also use external media, such as the *Clue VCR Mystery Game* which required players to watch VCR clips between turns. To distinguish between these two, video games are considered to require some interactivity that affects the visual display.[15]

Most video games tend to feature some type of victory or winning conditions, such as a scoring mechanism or a final boss fight. The introduction of walking simulators (adventure games that allow for exploration but lack any objectives) like *Gone Home*, and empathy games (video games that tend to focus on emotion) like *That Dragon, Cancer* brought the idea of games that did not have any such type of winning condition and raising the question of whether these were actually games.[27] These are still commonly justified as video games as they provide a game world that the player can interact with by some means.[28]

The lack of any industry definition for a video game by 2021 was an issue during the case *Epic Games v. Apple* which dealt with video games offered on Apple's iOS App Store. Among concerns raised were games like *Fortnite Creative* and *Roblox* which created metaverses of interactive experiences, and whether the larger game and the individual experiences themselves were games or not in relation to fees that Apple charged for the App Store. Judge Yvonne Gonzalez Rogers, recognizing that there was yet an industry standard definition for a video game, established for her ruling that "At a bare minimum, video games appear to require some level of interactivity or involvement between the player and the medium" compared to passive entertainment like film, music, and television, and "videogames are also generally graphically rendered or animated, as opposed to being recorded live or via motion capture as in films or television".[29] Rogers still concluded that what is a video game "appears highly eclectic and diverse".[29]

Video game terminology

Freedoom, a clone of the first-person shooter *Doom*. Common elements include a heads-up display along the bottom that includes the player's remaining health and ammunition.

The gameplay experience varies radically between video games, but many common elements exist. Most games will launch into a title screen and give the player a chance to review options such as the number of players before starting a game. Most games are divided into levels which the player must work the avatar through, scoring points, collecting power-ups to boost the avatar's innate attributes, all while either using special attacks to defeat enemies or moves to avoid them. This information is relayed to the player through a type of on-screen user interface such as a heads-up display atop the rendering of the game itself. Taking damage will deplete their avatar's health, and if that falls to zero or if the avatar otherwise falls into an impossible-to-escape location, the player will lose one of their lives. Should they lose all their lives without gaining an extra life or "1-UP", then the player will reach the "game over" screen. Many levels as well as the game's finale end with a type of boss character the player must defeat to continue on. In some games, intermediate points between levels will offer save points where the player can create a saved game on storage media to restart the game should they lose all their lives or need to stop the game and restart at a later time. These also may be in the form of a passage that can be written down and reentered at the title screen.

Product flaws include software bugs which can manifest as glitches which may be exploited by the player; this is often the foundation of speedrunning a video game. These bugs, along with cheat codes, Easter eggs, and other hidden secrets that were intentionally added to the game can also be exploited.[30][31][32][33] On some consoles, cheat cartridges allow players to execute these cheat codes, and user-developed trainers allow similar bypassing for computer software games. Both of which might make the game easier, give the player additional power-ups, or change the appearance of the game.[31]

Components

To distinguish from electronic games, a video game is generally considered to require a platform, the hardware which contains computing elements, to process player interaction from some type of input device and displays the results to a video output display.[34]

Platform

Video games require a platform, a specific combination of electronic components or computer hardware and associated software, to operate.[35] The term system is also commonly used. These platforms may include multiple brandsheld by platform holders, such as Nintendo or Sony, seeking to gain larger market shares.[36][37][38][39][40] Games are typically designed to be played on one or a limited number of platforms, and exclusivity to a platform or brand is used by platform holders as a competitive edge in the video game market.[41] However, games may be developed for alternative platforms than intended, which are described as ports or conversions. These also may be remasters - where most of the original game's source code is reused and art assets, models, and game levels are updated for modern systems – and remakes, where in addition to asset improvements, significant reworking of the original game and possibly from scratch is performed.[42]

Arcade video game machines at the Sugoi arcade game hall in Malmi, Helsinki, Finland

The list below is not exhaustive and excludes other electronic devices capable of playing video games such as PDAs and graphing calculators.

Various gaming consoles at the Computer Games Museum in Berlin

PC games

PC games involve a player interacting with a personal computer (PC) connected to a video monitor.[43] Personal computers are not dedicated game platforms, so there may be differences running the same game on different hardware. Also, the openness allows some features to developers like reduced software cost,[44] increased flexibility, increased innovation, emulation, creation of modifications or mods, open hosting for online gaming (in which a person plays a video game with people who are in a different household) and others. A gaming computer is a PC or laptop intended specifically for gaming, typically using high-performance, high-cost components. In addition to personal computer gaming, there also exist games that work on mainframe computers and other similarly shared systems, with users logging in remotely to use the computer.

Home console

A console game is played on a home console, a specialized electronic device that connects to a common television set or composite video monitor. Home consoles are specifically designed to play games using a dedicated hardware environment, giving developers a concrete hardware target for development and assurances of what features will be available, simplifying development compared to PC game development. Usually consoles only run games developed for it, or games from other platform made by the same company, but never games developed by its direct competitor, even if the same game is available on different platforms. It often comes with a specific game controller. Major console platforms include Xbox, PlayStation and Nintendo.

Handheld console

A handheld game console is a small, self-contained electronic device that is portable and can be held in a user's hands. It features the console, a small screen, speakers and buttons, joystick or other game controllers in a single unit. Like consoles, handhelds are dedicated platforms, and share almost the same characteristics. Handheld hardware usually is less powerful than PC or console hardware. Some handheld games from the late 1970s and early 1980s could only play one game. In the 1990s and 2000s, a number of handheld games used cartridges, which enabled them to be used to play many different games. The handheld console has waned in the 2010s as mobile device gaming has become a more dominant factor.

Arcade video game

An arcade video game generally refers to a game played on an even more specialized type of electronic device that is typically designed to play only one game and is encased in a special, large coin-operated cabinet which has one built-in console, controllers (joystick, buttons, etc.), a CRT screen, and audio amplifier and speakers. Arcade games often have brightly painted logos and images relating to the theme of the game. While most arcade games are housed in a vertical cabinet, which the user typically stands in front of to play, some arcade games use a tabletop approach, in which the display screen is housed in a table-style cabinet with a see-through table top. With table-top games, the users typically sit to play. In the 1990s and 2000s, some arcade games offered players a choice of multiple games. In the 1980s, video arcades were businesses in which game players could use a number of arcade video games. In the 2010s, there are far fewer video arcades, but some movie theaters and family entertainment centers still have them.

The PlayStation 2 is the best-selling video game console, with over 155 million units sold.[45]

Browser game

A browser game takes advantages of standardizations of technologies for the functionality of web browsers across multiple devices providing a cross-platform environment. These games may be identified based on the website that they appear, such as with Miniclip games. Others are named based on the programming platform used to develop them, such as Java and Flash games.

A police-themed arcade game in which players use a light gun

Mobile game

With the introduction of smartphones and tablet computers standardized on the iOS and Android operating systems, mobile gaming has become a significant platform. These games may use unique features of mobile devices that are not necessary present on other platforms, such as accelerometers, global positioning information and camera devices to support augmented reality gameplay.

Cloud gaming

Cloud gaming requires a minimal hardware device, such as a basic computer, console, laptop, mobile phone or even a dedicated hardware device connected to a display with good Internet connectivity that connects to hardware systems by the cloud gaming provider. The game is computed and rendered on the remote hardware, using a number of predictive methods to reduce the network latency between player input and output on their display device. For example, the Xbox Cloud Gaming and PlayStation Now platforms use dedicated custom server blade hardware in cloud computing centers.

Virtual reality

Virtual reality (VR) games generally require players to use a special head-mounted unit that provides stereoscopic screens and motion tracking to immerse a player within virtual environment that responds to their head movements. Some VR systems include control units for the player's hands as to provide a direct way to interact with the virtual world. VR systems generally require a separate computer, console, or other processing device that couples with the head-mounted unit.

Emulation

An emulator enables games from a console or otherwise different system to be run in a type of virtual machine on a modern system, simulating the hardware of the original and allows old games to be played. While emulators themselves have been found to be legal in United States case law, the act of obtaining the game software that one does not already own may violate copyrights. However, there are some official releases of emulated software from game manufacturers, such as Nintendo with its Virtual Console or Nintendo Switch Online offerings.

Players using the PlayStation VR headsets in 2017

Backward compatibility

Backward compatibility is similar in nature to emulation in that older games can be played on newer platforms, but typically directly though hardware and built-in software within the platform. The PlayStation 2 popularized the trend by having the capability of playing past generation games from the PlayStation via inserting the original game media into the newer console, while Nintendo's Wii could play GameCube titles as well in the same manner.[46][47][48]

Game media

Early arcade games, home consoles, and handheld games were dedicated hardware units with the game's logic built into the electronic componentry of the hardware. Since then, most video game platforms are considered programmable, having means to read and play multiple games distributed on different types of media or formats. Physical formats include ROM cartridges, magnetic storage including magnetic-tape data storage and floppy discs, optical media formats including CD-ROM and DVDs, and flash memory cards. Furthermore digital distribution over the Internet or other communication methods as well as cloud gaming alleviate the need for any physical media. In some cases, the media serves as the direct read-only memory for the game, or it may be the form of installation media that is used to write the main assets to the player's platform's local storage for faster loading periods and later updates.

An unlabeled game cartridge for the Nintendo Entertainment System

Games can be extended with new content and software patches through either expansion packs which are typically available as physical media, or as downloadable content nominally available via digital distribution. These can be offered freely or can be used to monetize a game following its initial release. Several games offer players the ability to create user-generated content to share with others to play. Other games, mostly those on personal computers, can be extended with user-created modifications or mods that alter or add onto the game; these often are unofficial and were developed by players from reverse engineering of the game, but other games provide official support for modding the game.[49]

Input device

A North American Super NES game controller from the early 1990s

Video game can use several types of input devices to translate human actions to a game. Most common are the use of game controllers like gamepads and joysticks for most consoles, and as accessories for personal computer systems along keyboard and mouse controls. Common controls on the most recent controllers include face buttons, shoulder triggers, analog sticks, and directional pads ("d-pads"). Consoles typically include standard controllers which are shipped or bundled with the console itself, while peripheral controllers are available as a separate purchase from the console manufacturer or third-party vendors.[50] Similar control sets are built into handheld consoles and onto arcade cabinets. Newer technology improvements have incorporated additional technology into the controller or the game platform, such as touchscreens and motion detection sensors that give more options for how the player interacts with the game. Specialized controllers may be used for certain genres of games, including racing wheels, light guns and dance pads. Digital cameras and motion detection can capture movements of the player as input into the game, which can, in some cases, effectively eliminate the control, and on other systems such as virtual reality, are used to enhance immersion into the game.

Display and output

Handheld units, like the Game Boy, include built-in output screens and sound speakers.

By definition, all video games are intended to output graphics to an external video display, such as cathode-ray tube televisions, newer liquid-crystal display (LCD) televisions and built-in screens, projectors or computer monitors, depending on the type of platform the game is played on. Features such as color depth, refresh rate, frame rate, and screen resolution are a combination of the limitations of the game platform and display device and the program efficiency of the game itself. The game's output can range from fixed displays using LED or LCD elements, text-based games, two-dimensional and three-dimensional graphics, and augmented reality displays.

The game's graphics are often accompanied by sound produced by internal speakers on the game platform or external speakers attached to the platform, as directed by the game's programming. This often will include sound effects tied to the player's actions to provide audio feedback, as well as background music for the game.

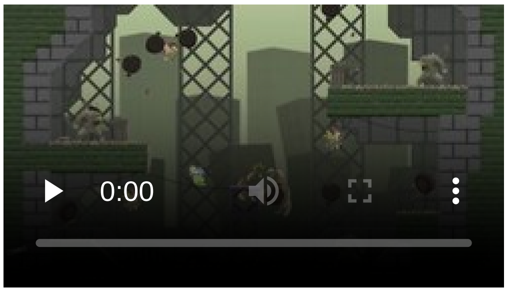
Some platforms support additional feedback mechanics to the player that a game can take advantage of. This is most commonly haptic technology built into the game controller, such as causing the controller to shake in the player's hands to simulate a shaking earthquake occurring in game.

Classifications

Video games are frequently classified by a number of factors related to how one plays them.

Genre

A video game, like most other forms of media, may be categorized into genres. However, unlike film or television which use visual or narrative elements, video games are generally categorized into genres based on their gameplay interaction, since this is the primary means which one interacts with a video game.[51][52][53] The narrative setting does not impact gameplay; a shooter game is still a shooter game, regardless of whether it takes place in a fantasy world or in outer space.[54][55] An exception is the horror game genre, used for games that are based on narrative elements of horror fiction, the supernatural, and psychological horror.[56]



Dustforce is representative of the platform game genre as its gameplay involves jumping between platforms.

Genre names are normally self-describing in terms of the type of gameplay, such as action game, role playing game, or shoot 'em up, though some genres have derivations from influential works that have defined that genre, such as roguelikes from *Rogue*,[57] Grand Theft Auto clones from *Grand Theft Auto III*,[58] and battle royale games from the film *Battle Royale*.[59] The names may shift over time as players, developers and the media come up with new terms; for example, first-person shooters were originally called "Doom clones" based on the 1993 game.[60] A hierarchy of game genres exist, with top-level genres like "shooter game" and "action game" that broadly capture the game's main gameplay style, and several subgenres of specific implementation, such as within the shooter game first-person shooter and third-person shooter. Some cross-genre types also exist that fall until multiple top-level genres such as action-adventure game.

Mode

A video game's mode describes how many players can use the game at the same type. This is primarily distinguished by single-player video games and multiplayer video games. Within the latter category, multiplayer games can be played in a variety of ways, including locally at the same device, on separate devices connected through a local network such as LAN parties, or online via separate Internet connections. Most multiplayer games are based on competitive gameplay, but many offer cooperative and team-based options as well as asymmetric gameplay. Online games use server structures that can also enable massively multiplayer online games (MMOs) to support hundreds of players at the same time.

A LAN party at the 2004 DreamHack with hundreds of players

A small number of video games are zero-player games, in which the player has very limited interaction with the game itself. These are most commonly simulation games where the player may establish a starting state and then let the game proceed on its own, watching the results as a passive observer, such as with many computerized simulations of Conway's Game of Life.[61]

Types

Most video games are intended for entertainment purposes.[34] Different game types include:

Core games

Core or hard-core games refer to the typical perception of video games, developed for entertainment purposes. These games typically require a fair amount of time to learn and master, in contrast to casual games, and thus are most appealing to gamers rather than a broader audience. Most of the AAA video game industry is based around the delivery of core games.[62]

Casual games

In contrast to core games, casual games are designed for ease of accessibility, simple to understand gameplay and quick to grasp rule sets, and aimed at mass market audience. They frequently support the ability to jump in and out of play on demand, such as during commuting or lunch breaks. Numerous browser and mobile games fall into the casual game area, and casual games often are from genres with low intensity game elements such as match three, hidden object, time management, and puzzle games.[63] Causal games frequently use social-network game mechanics, where players can enlist the help of friends on their social media networks for extra turns or moves each day.[64] Popular casual games include *Tetris* and *Candy Crush Saga*. More recent, starting in the late 2010s, are hyper-casual games which use even more simplistic rules for short but infinitely replayable games, such as *Flappy Bird*.[65]

Educational games

Education software has been used in homes and classrooms to help teach children and students, and video games have been similarly adapted for these reasons, all designed to provide a form of interactivity and entertainment tied to game design elements. There are a variety of differences in their designs and how they educate the user. These are broadly split between edutainment games that tend to focus on the entertainment value and rote learning but are unlikely to engage in critical thinking, and educational video games that are geared towards problem solving through motivation and positive reinforcement while downplaying the entertainment value.[66] Examples of educational games include *The Oregon Trail* and the *Carmen Sandiego* series. Further, games not initially developed for educational purposes have found their way into the classroom after release, such as that feature open worlds or virtual sandboxes like *Minecraft*,[67] or offer critical thinking skills through puzzle video games like *SpaceChem*.[68]

Serious games

Further extending from educational games, serious games are those where the entertainment factor may be augmented, overshadowed, or even eliminated by other purposes for the game. Game design is used to reinforce the non-entertainment purpose of the game, such as using video game technology for the game's interactive world, or gamification for reinforcement training. Educational games are a form of serious games, but other types of games include fitness games that incorporate significant physical exercise to help keep the player fit (such as *Wii Fit*), simulator games that resemble flight simulators to pilot aircraft (such as *Microsoft Flight Simulator*), advergames that are built around the advertising of a product (such as *Pepsiman*), and newsgames aimed at conveying a specific advocacy message (such as *NarcoGuerra*).[69][70]

Microsoft Flight Simulator is an example of a simulation game.

Art games

Although video games have been considered an art form on their own, games may be developed to try to purposely communicate a story or message, using the medium as a work of art. These art or arthouse games are designed to generate emotion and empathy from the player by challenging societal norms and offering critique through the interactivity of the video game medium. They may not have any type of win condition and are designed to let the player explore through the game world and scenarios. Most art games are indie games in nature, designed based on personal experiences or stories through a single developer or small team. Examples of art games include *Passage*, *Flower*, and *That Dragon, Cancer*.[71][72][73]

Content rating

A typical ESRB rating label, listing the rating and specific content descriptors for *Rabbids Go Home*

Video games can be subject to national and international content rating requirements. Like with film content ratings, video game ratings typing identify the target age group that the national or regional ratings board believes is appropriate for the player, ranging from all-ages, to a teenager-or-older, to mature, to the infrequent adult-only games. Most content review is based on the level of violence, both in the type of violence and how graphic it may be represented, and sexual content, but other themes such as drug and alcohol use and gambling that can influence children may also be identified. A primary identifier based on a minimum age is used by nearly all systems, along with additional descriptors to identify specific content that players and parents should be aware of.

The regulations vary from country to country but generally are voluntary systems upheld by vendor practices, with penalty and fines issued by the ratings body on the video game publisher for misuse of the ratings. Among the major content rating systems include:

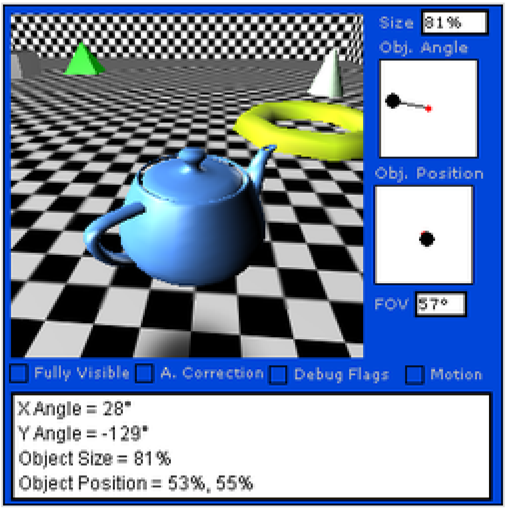
- Entertainment Software Rating Board (ESRB) that oversees games released in the United States. ESRB ratings are voluntary and rated along a E (Everyone), E10+ (Everyone 10 and older), T (Teen), M (Mature), and AO (Adults Only). Attempts to mandate video games ratings in the U.S. subsequently led to the landmark Supreme Court case, *Brown v. Entertainment Merchants Association* in 2011 which ruled video games were a protected form of art, a key victory for the video game industry.[74]
- Pan European Game Information (PEGI) covering the United Kingdom, most of the European Union and other European countries, replacing previous national-based systems. The PEGI system uses content rated based on minimum recommended ages, which include 3+, 8+, 12+, 16+, and 18+.
- Australian Classification Board (ACB) oversees the ratings of games and other works in Australia, using ratings of G (General), PG (Parental Guidance), M (Mature), MA15+ (Mature Accompanied), R18+ (Restricted), and X (Restricted for pornographic material). ACB can also deny to give a rating to game (RC – Refused Classification). The ACB's ratings are enforceable by law, and importantly, games cannot be imported or purchased digitally in Australia if they have failed to gain a rating or were given the RC rating, leading to a number of notable banned games.
- Computer Entertainment Rating Organization (CERO) rates games for Japan. Their ratings include A (all ages), B (12 and older), C (15 and over), D (17 and over), and Z (18 and over).
- Unterhaltungssoftware Selbstkontrolle (USK) rates games for Germany. Their ratings include 0, 6, 12, 16, and 18.

Additionally, the major content system provides have worked to create the International Age Rating Coalition (IARC), a means to streamline and align the content ratings system between different region, so that a publisher would only need to complete the content ratings review for one provider, and use the

IARC transition to affirm the content rating for all other regions.

Certain nations have even more restrictive rules related to political or ideological content. Within Germany, until 2018, the Unterhaltungssoftware Selbstkontrolle (*Entertainment Software Self-Regulation*) would refuse to classify, and thus allow sale, of any game depicting Nazi imagery, and thus often requiring developers to replace such imagery with fictional ones. This ruling was relaxed in 2018 to allow for such imagery for "social adequacy" purposes that applied to other works of art.[75] China's video game segment is mostly isolated from the rest of the world due to the government's censorship, and all games published there must adhere to strict government review, disallowing content such as smearing the image of the Chinese Communist Party. Foreign games published in China often require modification by developers and publishers to meet these requirements.[76]

Development



Developers use various tools to create video games. Here an editor is fine-tuning the virtual camera system.

Video game development and authorship, much like any other form of entertainment, is frequently a cross-disciplinary field. Video game developers, as employees within this industry are commonly referred to, primarily include programmers and graphic designers. Over the years, this has expanded to include almost every type of skill that one might see prevalent in the creation of any movie or television program, including sound designers, musicians, and other technicians; as well as skills that are specific to video games, such as the game designer. All of these are managed by producers.

In the early days of the industry, it was more common for a single person to manage all of the roles needed to create a video game. As platforms have become more complex and powerful in the type of material they can present, larger teams have been needed to generate all of the art, programming, cinematography, and more. This is not to say that the age of the "one-man shop" is gone, as this is still sometimes found in the casual gaming and handheld markets,[77] where smaller games are prevalent due to technical limitations such as limited RAM or lack of dedicated 3D graphics rendering capabilities on the target platform (e.g., some PDAs).[78]

Video games are programmed like any other piece of computer software. Prior to the mid-1970s, arcade and home consoles were programmed by assembling discrete electro-mechanical components on circuit boards, which limited games to relatively simple logic. By 1975, low-cost microprocessors were available at volume to be used for video game hardware, which allowed game developers to program more detailed games, widening the scope of what was possible.[79][80] Ongoing improvements in computer hardware technology have expanded what has become possible to create in video games, coupled with convergence of common hardware between console, computer, and arcade platforms to simplify the development process.[81] Today, game developers have a number of commercial and open source tools available for use to make games, often which are across multiple platforms to support portability, or may still opt to create their own for more specialized features and direct control of the game. Today, many games are built around a game engine that handles the bulk of the game's logic, gameplay, and rendering. These engines can be augmented with specialized engines for specific features, such as a physics engine that simulates the physics of objects in real-time. A variety of middleware exists to help developers access other features, such as playback of videos within games, network-oriented code for games that communicate via online services, matchmaking for online games, and similar features. These features can be used from a developer's programming language of choice, or they may opt to also use game development kits that minimize the amount of direct programming they have to do but can also limit the amount of customization they can add into a game. Like all software, video games usually undergo quality testing before release to assure there are no bugs or glitches in the product, though frequently developers will release patches and updates.

With the growth of the size of development teams in the industry, the problem of cost has increased. Development studios need the best talent, while publishers reduce costs to maintain profitability on their investment. Typically, a video game console development team ranges from 5 to 50 people, and some exceed 100. In May 2009, *Assassin's Creed II* was reported to have a development staff of 450.[82] The growth of team size combined with greater pressure to get completed projects into the market to begin recouping production costs has led to a greater occurrence of missed deadlines, rushed games, and the release of unfinished products.[83]

While amateur and hobbyist game programming had existed since the late 1970s with the introduction of home computers, a newer trend since the mid-2000s is indie game development. Indie games are made by small teams outside any direct publisher control, their games being smaller in scope than those from the larger "AAA" game studios, and are often experiments in gameplay and art style. Indie game development is aided by the larger availability of digital distribution, including the newer mobile gaming market, and readily-available and low-cost development tools for these platforms.[84]

Game theory and studies

Although departments of computer science have been studying the technical aspects of video games for years, theories that examine games as an artistic medium are a relatively recent development in the humanities. The two most visible schools in this emerging field are ludology and narratology. Narrativists approach video games in the context of what Janet Murray calls "Cyberdrama". That is to say, their major concern is with video games as a storytelling medium, one that arises out of interactive fiction. Murray puts video games in the context of the Holodeck, a fictional piece of technology from *Star Trek*, arguing for the video game as a medium in which the player is allowed to become another person, and to act out in another world.[85] This image of video games received early widespread popular support, and forms the basis of films such as *Tron*, *eXistenZ* and *The Last Starfighter*.

Ludologists break sharply and radically from this idea. They argue that a video game is first and foremost a game, which must be understood in terms of its rules, interface, and the concept of play that it deploys. Espen J. Aarseth argues that, although games certainly have plots, characters, and aspects of traditional narratives, these aspects are incidental to gameplay. For example, Aarseth is critical of the widespread attention that narrativists have given to the heroine of the game *Tomb Raider*, saying that "the dimensions of Lara Croft's body, already analyzed to death by film theorists, are irrelevant to me as a player, because a different-looking body would not make me play differently... When I play, I don't even see her body, but see through it and past it."[86] Simply put, ludologists reject traditional theories of art because they claim that the artistic and socially relevant qualities of a video game are primarily determined by the underlying set of rules, demands, and expectations imposed on the player.

While many games rely on emergent principles, video games commonly present simulated story worlds where emergent behavior occurs within the context of the game. The term "emergent narrative" has been used to describe how, in a simulated environment, storyline can be created simply by "what happens to the player."[87] However, emergent behavior is not limited to sophisticated games. In general, any place where event-driven instructions occur for AI in a game, emergent behavior will exist. For instance, take a racing game in which cars are programmed to avoid crashing, and they encounter an obstacle in the track: the cars might then maneuver to avoid the obstacle causing the cars behind them to slow or maneuver to accommodate the cars in front of them and the obstacle. The programmer never wrote code to specifically create a traffic jam, yet one now exists in the game.

Intellectual property for video games

Most commonly, video games are protected by copyright, though both patents and trademarks have been used as well.

Though local copyright regulations vary to the degree of protection, video games qualify as copyrighted visual-audio works, and enjoy cross-country protection under the Berne Convention.[88] This typically only applies to the underlying code, as well as to the artistic aspects of the game such as its writing, art assets, and music. Gameplay itself is generally not considered copyrightable; in the United States among other countries, video games are considered to fall into the idea–expression distinction in that it is how the game is presented and expressed to the player that can be copyrighted, but not the underlying principles of the game.[89]

Because gameplay is normally ineligible for copyright, gameplay ideas in popular games are often replicated and built upon in other games. At times, this repurposing of gameplay can be seen as beneficial and a fundamental part of how the industry has grown by building on the ideas of others.[90][91] For example *Doom* (1993) and *Grand Theft Auto III* (2001) introduced gameplay that created popular new game genres, the first-person shooter and the *Grand Theft Auto* clone, respectively, in the few years after their release.[92][93] However, at times and more frequently at the onset of the industry, developers would intentionally create video game clones of successful games and game hardware with few changes, which led to the flooded arcade and dedicated home console market around 1978.[90][94][91] Cloning is also a major issue with countries that do not have strong intellectual property protection laws, such as within China. The lax oversight by China's government and the difficulty for foreign companies to take Chinese entities to court had enabled China to support a large grey market of cloned hardware and software systems.[95] The industry remains challenged to distinguish between creating new games based on refinements of past successful games to create a new type of gameplay, and intentionally creating a clone of a game that may simply swap out art assets.[96]

Industry

History

The early history of the video game industry, following the first game hardware releases and through 1983, had little structure. Video games quickly took off during the golden age of arcade video games from the late 1970s to early 1980s, but the newfound industry was mainly composed of game developers with little business experience. This led to numerous companies forming simply to create clones of popular games to try to capitalize on the market.[97] Due to loss of publishing control and oversaturation of the market, the North American home video game market crashed in 1983, dropping from revenues of around $3 billion in 1983 to $100 million by 1985. Many of the North American companies created in the prior years closed down. Japan's growing game industry was briefly shocked by this crash but had sufficient longevity to withstand the short-term effects, and Nintendo helped to revitalize the industry with the release of the Nintendo Entertainment System in North America in 1985.[97] Along with it, Nintendo established a number of core industrial practices to prevent unlicensed game development and control game distribution on their platform, methods that continue to be used by console manufacturers today.[97]

E3 was one of the typical trade show events of the video game industry.

Attendees at Gamescom 2015 playing a video game

The industry remained more conservative following the 1983 crash, forming around the concept of publisher-developer dichotomies, and by the 2000s, leading to the industry centralizing around low-risk, triple-A games and studios with large development budgets of at least $10 million or more.[98] The advent of the Internet brought digital distribution as a viable means to distribute games, and contributed to the growth of more riskier, experimental independent game development as an alternative to triple-A games in the late 2000s and which has continued to grow as a significant portion of the video game industry.[99][84]

Industry roles

Video games have a large network effect that draw on many different sectors that tie into the larger video game industry. While video game developers are a significant portion of the industry, other key participants in the market include:[100]

- Publishers: Companies generally that oversee bringing the game from the developer to market. This often includes performing the marketing, public relations, and advertising of the game. Publishers frequently pay the developers ahead of time to make their games and will be involved in critical decisions about the direction of the game's progress, and then pay the developers additional royalties or bonuses based on sales performances. Other smaller, boutique publishers may simply offer to perform the publishing of a game for a small fee and a portion of the sales, and otherwise leave the developer with the creative freedom to proceed. A range of other publisher-developer relationships exist between these points.
- Distributors: Publishers often are able to produce their own game media and take the role of distributor, but there are also third-party distributors that can mass-produce game media and distribute to retailers. Digital storefronts like Steam and the iOS App Store also serve as distributors and retailers in the digital space.
- Retailers: Physical storefronts, which include large online retailers, department and electronic stores, and specialty video game stores, sell games, consoles, and other accessories to consumers. This has also including a trade-in market in certain regions,

allowing players to turn in used games for partial refunds or credit towards other games. However, with the uprising of digital marketplaces and e-commerce revolution, retailers have been performing worse than in the past.

- Hardware manufacturers: The video game console manufacturers produce console hardware, often through a value chain system that include numerous component suppliers and contract manufacturer that assemble the consoles. Further, these console manufacturers typically require a license to develop for their platform and may control the production of some games, such as Nintendo does with the use of game cartridges for its systems. In exchange, the manufacturers may help promote games for their system and may seek console exclusivity for certain games. For games on personal computers, a number of manufacturers are devoted to high-performance "gaming computer" hardware, particularly in the graphics card area; several of the same companies overlap with component supplies for consoles. A range of third-party manufacturers also exist to provide equipment and gear for consoles post-sale, such as additional controllers for console or carrying cases and gear for handheld devices.
- Journalism: While journalism around video games used to be primarily print-based, and focused more on post-release reviews and gameplay strategy, the Internet has brought a more proactive press that use web journalism, covering games in the months prior to release as well as beyond, helping to build excitement for games ahead of release.
- Influencers: With the rising importance of social media, video game companies have found that the opinions of influencers using streaming media to play through their games has had a significant impact on game sales, and have turned to use influencers alongside traditional journalism as a means to build up attention to their game before release.
- Esports: Esports is a major function of several multiplayer games with numerous professional leagues established since the 2000s, with large viewership numbers, particularly out of southeast Asia since the 2010s.
- Trade and advocacy groups: Trade groups like the Entertainment Software Association were established to provide a common voice for the industry in response to governmental and other advocacy concerns. They frequently set up the major trade events and conventions for the industry such as E3.
- Gamers: Proactive hobbyists who are players and consumers of video games. While their representation in the industry is primarily seen through game sales, many companies follow gamers' comments on social media or on user reviews and engage with them to work to improve their products in addition to other feedback from other parts of the industry. Demographics of the larger player community also impact parts of the market; while once dominated by younger men, the market shifted in the mid-2010s towards women and older players who generally preferred mobile and causal games, leading to further growth in those sectors.[101]

Major regional markets

The industry itself grew out from both the United States and Japan in the 1970s and 1980s before having a larger worldwide contribution. Today, the video game industry is predominantly led by major companies in North America (primarily the United States and Canada), Europe, and southeast Asia including Japan, South Korea, and China. Hardware production remains an area dominated by Asian companies either directly involved in hardware design or part of the production process, but digital distribution and indie game development of the late 2000s has allowed game developers to flourish nearly anywhere and diversify the field.[102]

Game sales

A retail display in Switzerland with a large selection of games for platforms popular in the early 2000s

According to the market research firm Newzoo, the global video game industry drew estimated revenues of over $159 billion in 2020. Mobile games accounted for the bulk of this, with a 48% share of the market, followed by console games at 28% and personal computer games at 23%.[1]

Sales of different types of games vary widely between countries due to local preferences. Japanese consumers tend to purchase much more handheld games than console games and especially PC games, with a strong preference for games catering to local tastes.[103][104] Another key difference is that, though having declined in the West, arcade games remain an important sector of the Japanese gaming industry.[105] In South Korea, computer games are generally preferred over console games, especially MMORPG games and real-time strategy games. Computer games are also popular in China.[106]

Effects on society

Culture


The Art of Video Games exhibit at the Smithsonian American Art Museum in 2012

Video game culture is a worldwide new media subculture formed around video games and game playing. As computer and video games have increased in popularity over time, they have had a significant influence on popular culture. Video game culture has also evolved over time hand in hand with internet culture as well as the increasing popularity of mobile games. Many people who play video games identify as gamers, which can mean anything from someone who enjoys games to someone who is passionate about it. As video games become more social with multiplayer and online capability, gamers find themselves in growing social networks. Gaming can both be entertainment as well as competition, as a new trend known as electronic sports is becoming more widely accepted. In the 2010s, video games and discussions of video game trends and topics can be seen in social media, politics, television, film and music. The COVID-19 pandemic during 2020–2021 gave further visibility to video games as a pastime to enjoy with friends and family online as a means of social distancing.[107][108]

Art

Since the mid-2000s there has been debate whether video games qualify as art, primarily as the form's interactivity interfered with the artistic intent of the work and that they are designed for commercial appeal. A significant debate on the matter came after film critic Roger Ebert published an essay "Video

Games can never be art",[109] which challenged the industry to prove him and other critics wrong.[110] The view that video games were an art form was cemented in 2011 when the U.S. Supreme Court ruled in the landmark case *Brown v. Entertainment Merchants Association* that video games were a protected form of speech with artistic merit.[111] Since then, video game developers have come to use the form more for artistic expression, including the development of art games,[112] and the cultural heritage of video games as works of arts, beyond their technical capabilities, have been part of major museum exhibits, including *The Art of Video Games* at the Smithsonian American Art Museum and toured at other museums from 2012 to 2016.

Video games will inspire sequels and other video games within the same franchise, but also have influenced works outside of the video game medium. Numerous television shows (both animated and live-action), films, comics and novels have been created based on existing video game franchises. Because video games are an interactive medium there has been trouble in converting them to these passive forms of media, and typically such works have been critically panned or treated as children's media. For example, until 2019, no video game film had ever been received a "Fresh" rating on Rotten Tomatoes, but the releases of *Detective Pikachu* (2019) and *Sonic the Hedgehog* (2020), both receiving "Fresh" ratings, shows signs of the film industry having found an approach to adapt video games for the large screen.[113][114] That said, some early video game-based films have been highly successful at the box office, such as 1995's *Mortal Kombat* and 2001's *Lara Croft: Tomb Raider*.[115]

More recently since the 2000s, there has also become a larger appreciation of video game music, which ranges from chiptunes composed for limited sound-output devices on early computers and consoles, to fully-scored compositions for most modern games. Such music has frequently served as a platform for covers and remixes, and concerts featuring video game soundtracks performed by bands or orchestras, such as *Video Games Live*, have also become popular.[115] Video games also frequently incorporate licensed music, particularly in the area of rhythm games, furthering the depth of which video games and music can work together.[115]

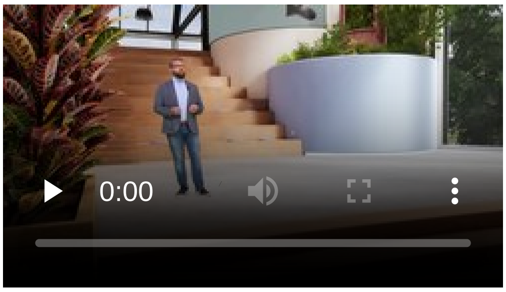
Further, video games can serve as a virtual environment under full control of a producer to create new works. With the capability to render 3D actors and settings in real-time, a new type of work machinima (short for "machine cinema") grew out from using video game engines to craft narratives.[116] As video game engines gain higher fidelity, they have also become part of the tools used in more traditional filmmaking. Unreal Engine has been used as a backbone by Industrial Light & Magic for their StageCraft technology for shows like *The Mandalorian*.[117]

Separately, video games are also frequently used as part of the promotion and marketing for other media, such as for films, anime, and comics. However, these licensed games in the 1990s and 2000s often had a reputation for poor quality, developed without any input from the intellectual property rights owners, and several of them are considered among lists of games with notably negative reception, such as *Superman 64*. More recently, with these licensed games being developed by triple-A studios or through studios directly connected to the licensed property owner, there has been a significant improvement in the quality of these games, with an early trendsetting example of *Batman: Arkham Asylum*.[118]

Beneficial uses

Besides their entertainment value, appropriately-designed video games have been seen to provide value in education across several ages and comprehension levels. Learning principles found in video games have been identified as possible techniques with which to reform the U.S. education system.[119] It has been noticed that gamers adopt an attitude while playing that is of such high concentration, they do not realize they are learning, and that if the same attitude could be adopted at school, education would enjoy significant benefits.[120] Students are found to be "learning by doing" while playing video games while fostering creative thinking.[121]

Video games are also believed to be beneficial to the mind and body. It has been shown that action video game players have better hand–eye coordination and visuo-motor skills, such as their resistance to distraction, their sensitivity to information in the peripheral vision and their ability to count briefly presented objects, than nonplayers.[122] Researchers found that such enhanced abilities could be acquired by training with action games, involving challenges that switch attention between different locations, but not with games requiring concentration on single objects. A 2018 systematic review found evidence that video gaming training had positive effects on cognitive and emotional skills in the adult population, especially with young adults.[123] A 2019 systematic review also added support for the claim that video games are beneficial to the brain, although the beneficial effects of video gaming on the brain differed by video games types.[124]



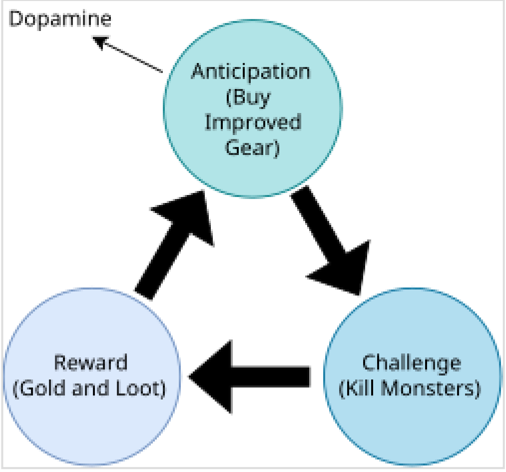
How can video games boost your mental health? – Max Birk (Eindhoven University of Technology)

Organisers of video gaming events, such as the organisers of the *D-Lux* video game festival in Dumfries, Scotland, have emphasised the positive aspects video games can have on mental health. Organisers, mental health workers and mental health nurses at the event emphasised the relationships and friendships that can be built around video games and how playing games can help people learn about others as a precursor to discussing the person's mental health.[125] A study in 2020 from Oxford University also suggested that playing video games can be a benefit to a person's mental health. The report of 3,274 gamers, all over the age of 18, focused on the games Animal Crossing: New Horizons and Plants vs Zombies: Battle for Neighborville and used actual play-time data. The report found that those that played more games tended to report greater "wellbeing".[126][127] Also in 2020, computer science professor Regan Mandryk of the University of Saskatchewan said her research also showed that video games can have health benefits such as reducing stress and improving mental health. The university's research studied all age groups – "from pre-literate children through to older adults living in long term care homes" – with a main focus on 18 to 55-year-olds.[128]

A study of gamers attitudes towards gaming which was reported about in 2018 found that millennials use video games as a key strategy for coping with stress. In the study of 1,000 gamers, 55% said that it "helps them to unwind and relieve stress ... and half said they see the value in gaming as a method of escapism to help them deal with daily work pressures".[129]

Controversies

Video games have caused controversy since the 1970s.[130] Parents and children's advocates regularly raise concerns that violent video games can influence young players into performing those violent acts in real life, and events such as the Columbine High School massacre in 1999 in which some claimed the perpetrators specifically alluded to using video games to plot out their attack, raised further fears. Medical experts and mental health professionals have also raised concerned that video games may be addictive, and the World Health Organization has included "gaming disorder" in the 11th revision of its International Statistical Classification of Diseases. Other health experts, including the American Psychiatric Association, have stated that there is insufficient evidence that video games can create violent tendencies or lead to addictive behavior,[131] though agree that video games typically use a compulsion loop in their core design that can create dopamine that can help reinforce the desire to continue to play through that compulsion loop and potentially lead into violent or addictive behavior.[132][133][134] Even with case law establishing that video games qualify as a protected art form, there has been pressure on the video game industry to keep their products in check to avoid over-excessive violence particularly for games aimed at younger children. The potential addictive behavior around games, coupled with increased used of post-sale monetization of video games, has also raised concern among parents, advocates, and government officials about gambling tendencies that may come from video games, such as controversy around the use of loot boxes in many high-profile games.



The compulsion loop for video games is believed to trigger dopamine release that can encourage addictive behavior.

Numerous other controversies around video games and its industry have arisen over the years, among the more notable incidents include the 1993 United States Congressional hearings on violent games like *Mortal Kombat* which lead to the formation of the ESRB ratings system, numerous legal actions taken by attorney Jack Thompson over violent games such as *Grand Theft Auto III* and *Manhunt* from 2003 to 2007, the outrage over the "No Russian" level from *Call of Duty: Modern Warfare 2* in 2009 which allowed the player to shoot a number of innocent non-player characters at an airport, and the Gamergate harassment campaign in 2014 that highlighted misogyny from a portion of the player demographic. The industry as a whole has also dealt with issues related to gender, racial, and LGBTQ+ discrimination and mischaracterization of these minority groups in video games. A further issue in the industry is related to working conditions, as development studios and publishers frequently use "crunch time", required extended working hours, in the weeks and months ahead of a game's release to assure on-time delivery.

Collecting and preservation

Players of video games often maintain collections of games. More recently there has been interest in retrogaming, focusing on games from the first decades. Games in retail packaging in good shape have become collectors items for the early days of the industry, with some rare publications having gone for over US$100,000 as of 2020. Separately, there is also concern about the preservation of video games, as both game media and the hardware to play them degrade over time. Further, many of the game developers and publishers from the first decades no longer exist, so records of their games have disappeared. Archivists and preservations have worked within the scope of copyright law to save these games as part of the cultural history of the industry.

There are many video game museums around the world, including the National Videogame Museum in Frisco, Texas,[135] which serves as the largest museum wholly dedicated to the display and preservation of the industry's most important artifacts.[136] Europe hosts video game museums such as the Computer

Games Museum in Berlin[137] and the Museum of Soviet Arcade Machines in Moscow and Saint-Petersburg.[138][139] The Museum of Art and Digital Entertainment in Oakland, California is a dedicated video game museum focusing on playable exhibits of console and computer games.[140] The Video Game Museum of Rome is also dedicated to preserving video games and their history.[141] The International Center for the History of Electronic Games at The Strong in Rochester, New York contains one of the largest collections of electronic games and game-related historical materials in the world, including a 5,000-square-foot (460 m^2) exhibit which allows guests to play their way through the history of video games.[142][143][144] The Smithsonian Institution in Washington, DC has three video games on permanent display: *Pac-Man*, *Dragon's Lair*, and *Pong*.[145]

The Museum of Modern Art has added a total of 20 video games and one video game console to its permanent Architecture and Design Collection since 2012.[146][147] In 2012, the Smithsonian American Art Museum ran an exhibition on "The Art of Video Games".[148] However, the reviews of the exhibit were mixed, including questioning whether video games belong in an art museum.[149][150]

See also

Video games portal

- Lists of video games
- List of accessories to video games by system
- Outline of video games

Notes

a. "Videogame" may also be used, though this is less frequent.

References

1. Hall, Stefan (15 May 2020). "How COVID-19 is taking gaming and esports to the next level" (https://www.weforum.org/agenda/2020/05/covid-19-taking-gaming-and-esports-next-level/). World Economic Forum. Archived (https://web.archive.org/web/20210505214601/https://www.weforum.org/agenda/2020/05/covid-19-taking-gaming-and-esports-next-level/) from the original on 5 May 2021. Retrieved 5 May 2021.
2. U.S. patent 2,455,992 (https://patents.google.com/patent/US2455992)
3. "Welcome to Pong-Story – Introduction" (http://www.pong-story.com/intro.htm). Pong-Story. Archived (https://web.archive.org/web/20100827183321/http://www.pong-story.com/intro.htm) from the original on 27 August 2010. Retrieved 24 October 2007.
4. "Welcome to... NIMROD!" (http://www.goodeveca.net/nimrod/). *goodeveca.net*. Archived (https://web.archive.org/web/20110723135752/http://www.goodeveca.net/nimrod/) from the original on 23 July 2011. Retrieved 24 October 2007.
5. Winter, David. "A.S.Douglas' 1952 Noughts and Crosses game" (http://www.pong-story.com/1952.htm). PONG-Story. Archived (https://web.archive.org/web/20151223140718/http://www.pong-story.com/1952.htm) from the original on 23 December 2015. Retrieved 3 July 2009.

6. Rabin, Steve (2005) [14 June 2005]. *Introduction to Game Development* (https://archive.org/details/introductiontoga0000unse). Massachusetts: Charles River Media. ISBN 978-1-58450-377-4.
7. Kent 2001, p. 25.
8. Orlando, Greg (15 May 2007). "Console Portraits: A 40-Year Pictorial History of Gaming" (https://www.wired.com/2007/06/gallery-game-history/). *Wired News*. Archived (https://web.archive.org/web/20080516011520/http://www.wired.com/gaming/gamingreviews/multimedia/2007/05/gallery_game_history) from the original on 16 May 2008. Retrieved 7 May 2022.
9. Yagoda, Marvin (2008). "1972 Nutting Associates Computer Space" (https://web.archive.org/web/20081228061939/http://www.marvin3m.com/arcade/cspace.htm). *Marvin's Marvelous Mechanical Museum*. Archived from the original (http://marvin3m.com/arcade/cspace.htm) on 28 December 2008.
10. "History of Gaming – Interactive Timeline of Game History" (https://www.pbs.org/kcts/videogamerevolution/history/timeline_flash.html). PBS. Archived (https://web.archive.org/web/20060218080821/http://www.pbs.org/kcts/videogamerevolution/history/timeline_flash.html) from the original on 18 February 2006. Retrieved 25 October 2007.
11. Miller, Michael (1 April 2005). *A History of Home Video Game Consoles* (https://web.archive.org/web/20071012152258/http://www.informit.com/articles/article.aspx?p=378141). InformIT. Archived from the original (https://www.informit.com/articles/article.aspx?p=378141) on 12 October 2007. Retrieved 7 May 2022.
12. Barton, Mat; Loguidice, Bill (9 January 2009). "The History Of Pong: Avoid Missing Game to Start Industry" (https://web.archive.org/web/20090112004852/http://www.gamasutra.com/view/feature/3900/the_history_of_pong_avoid_missing_.php). Game Developer. Archived from the original (https://www.gamedeveloper.com/business/the-history-of-i-pong-i-avoid-missing-game-to-start-industry) on 12 January 2009. Retrieved 7 May 2022.
13. Vendel, Curt; Goldberg, Marty (2012). *Atari Inc.: Business Is Fun* (https://archive.org/details/atariincbusiness0000gold/page/26). Syzygy Press. pp. 26 (https://archive.org/details/atariincbusiness0000gold/page/26). ISBN 978-0985597405.
14. Wolf 2007, pp. 3–7.
15. Wolf, Mark JP (2001). "Chapter 1: Video game as the medium". *The Medium of the Video Game*. University of Texas Press. pp. 13–33. ISBN 9780292791503.
16. Wolf, Mark; Perron, Bernard (2003). "Introduction: An Introduction To The Video Game Theory". *The Video Game Theory Reader*. Routledge, Taylor & Francis Group.
17. "Film, Television and Digital Games, Australia, 2015-16 financial year" (https://www.abs.gov.au/statistics/industry/technology-and-innovation/film-television-and-digital-games-australia/latest-release). *Australian Bureau of Statistics*. 15 June 2017. Retrieved 16 March 2023.
18. Wills, John (2019). *Gamer Nation: Video Games and American Culture*. Johns Hopkins Press. p. 5. ISBN 9781421428697. OCLC 1088892215 (https://search.worldcat.org/oclc/1088892215).
19. Newman, Michael Z. (2017). *Atari Age: The Emergence of Video Games in America*. Cambridge, MA: MIT Press. p. 45. ISBN 9780262035712. LCCN 2016028476 (https://lccn.loc.gov/2016028476).
20. Picard, Martin (December 2013). "The Foundation of Geemu: A Brief History of Early Japanese video games" (http://gamestudies.org/1302/articles/picard). *International Journal of Computer Game Research*. **13** (2). Archived (https://web.archive.org/web/20150624050100/http://gamestudies.org/1302/articles/picard) from the original on 24 June 2015. Retrieved 19 November 2016.
21. Akagi, Masumi (13 October 2006). *アーケードTVゲームリスト国内•海外編(1971–2005)* (https://archive.org/details/ArcadeGameList1971-2005) [*Arcade TV Game List: Domestic • Overseas Edition (1971–2005)*] (in Japanese). Japan: Amusement News Agency. ISBN 978-4990251215.

22. "A red-hot market for video games". *BusinessWeek*. 10 November 1973. p. 212.
23. Edwards, Benj (12 December 2007). "VC&G Interview: Nolan Bushnell, Founder of Atari" (http://www.vintagecomputing.com/index.php/archives/404/vcg-interview-nolan-bushnell-founder-of-atari). *Vintage Computing*. Archived (https://web.archive.org/web/20200723130632/http://www.vintagecomputing.com/index.php/archives/404/vcg-interview-nolan-bushnell-founder-of-atari) from the original on 23 July 2020. Retrieved 3 August 2020.
24. "The Golden Age Arcade Historian: The etymology of "video game" " (http://allincolorforaquarter.blogspot.com/2015/04/the-etymology-of-term-video-game.html?m=1). 3 April 2015.
25. Adlum, Eddie (November 1985). "The Replay Years: Reflections from Eddie Adlum" (https://archive.org/details/re-play-volume-11-issue-no.-2-november-1985-600DPI/RePlay%20-%20Volume%2011%2C%20Issue%20No.%202%20-%20November%201985/page/152). *RePlay*. Vol. 11, no. 2. pp. 134–175 (152).
26. Perron 2013, pp. 259–280.
27. Clark, Nicole (11 November 2017). "A brief history of the "walking simulator," gaming's most detested genre" (https://www.salon.com/2017/11/11/a-brief-history-of-the-walking-simulator-gamings-most-detested-genre/). *Salon*. Archived (https://web.archive.org/web/20200824042010/https://www.salon.com/2017/11/11/a-brief-history-of-the-walking-simulator-gamings-most-detested-genre/) from the original on 24 August 2020. Retrieved 12 September 2021.
28. Zimmermann, Felix; Huberts, Christian (2019). "From walking simulator to ambience action game". *Press Start*. **5** (2): 29–50.
29. Knoop, Joseph (10 September 2021). "Epic v Apple judge grapples with the big question: What is a videogame?" (https://www.pcgamer.com/videogame-definition-legal/). *PC Gamer*. Archived (https://web.archive.org/web/20210915140722/https://www.pcgamer.com/videogame-definition-legal/) from the original on 15 September 2021. Retrieved 10 September 2021.
30. Vargas, Jose Antonio (28 August 2006). "In Game World, Cheaters Proudly Prosper" (https://www.washingtonpost.com/wp-dyn/content/article/2006/08/27/AR2006082701059.html). *The Washington Post*. Archived (https://web.archive.org/web/20110820080450/http://www.washingtonpost.com/wp-dyn/content/article/2006/08/27/AR2006082701059.html) from the original on 20 August 2011. Retrieved 24 October 2007.
31. "Cracking the Code: The Konami Code" (https://web.archive.org/web/20110522201923/http://www.1up.com/features/cracking-code-konami-code). *1UP.com*. Archived from the original (http://www.1up.com/features/cracking-code-konami-code) on 22 May 2011. Retrieved 24 October 2007.
32. Wolf, Mark J.P. (2012). *Encyclopedia of Video Games: The Culture, Technology, and Art of Gaming*. ABC-CLIO. p. 177. ISBN 978-0-313-37936-9.
33. Björk, Staffan; Holopainen, Jussi (2005). *Patterns In Game Design* (https://books.google.com/books?id=IFQfyODK4wAC&pg=PA235) Archived (https://web.archive.org/web/20160505124122/https://books.google.com/books?id=IFQfyODK4wAC&pg=PA235#v=onepage&f=false) 5 May 2016 at the Wayback Machine. Charles River Media. p. 235. ISBN 978-1-58450-354-5. Retrieved 25 January 2013.
34. De Prato, Guiditta; Feijóo, Claudio; Nepelski, Daniel; Bogdanowicz, Marc; Simon, Jean Paul (2010). Born digital/grown digital: Assessing the future competitiveness of the EU video games software industry (https://publications.jrc.ec.europa.eu/repository/handle/JRC60711) (Report). JRC Scientific and Technical Reports.
35. "Platform" (http://dictionary.reference.com/browse/platform). *Dictionary.com*. Archived (https://web.archive.org/web/20071107070648/http://dictionary.reference.com/browse/platform) from the original on 7 November 2007. Retrieved 3 November 2007.
36. "The future for platform holders is cooperation, not just competition / Opinion" (https://www.gamesindustry.biz/the-future-for-platform-holders-is-cooperation-not-just-competition-opinion). 23 August 2019. Retrieved 18 March 2024.

37. "Platform holders and indies: Handy tips from the developers" (https://www.gamesindustry.biz/platform-holders-and-indies-handy-tips-from-developers). 25 July 2013. Retrieved 18 March 2024.
38. "Platform Holders" (https://flylib.com/books/en/2.594.1.28/1/). Retrieved 18 March 2024.
39. "Game development for console platforms" (https://unity.com/solutions/console). "New to console video game development? Start with resources to help you gain approval with platform holders and get started on developing your game."
40. "Technical Requirements Checklist" (https://research.ncl.ac.uk/game/mastersdegree/workshops/technicalrequirementschecklists/Technical%20Requirements%20Checklist%20Workshop.pdf) (PDF). Retrieved 18 March 2024.
41. Gamble, John (2007). "Competition in Video Game Consoles: Sony, Microsoft and Nintendo Battle for Supremacy". In Thompson, Arthur; Strickland III, A. J.; Gamble, John (eds.). *Crafting and Executing Strategy: The Quest for Competitive Advantage: Concepts and Cases*. McGraw-Hill. pp. C-198 – C211. ISBN 978-0073381244.
42. Grabarczyk, Pawel; Aarseth, Espen (August 2019). "Port or conversion? An ontological framework for classifying game versions" (http://www.digra.org/digital-library/publications/port-or-conversion-an-ontological-framework-for-classifying-game-versions/). *Proceedings of the 2019 DiGRA International Conference: Game, Play and the Emerging Ludo-Mix*. DiGRA Conference 2019. Archived (https://web.archive.org/web/20200729142448/http://www.digra.org/digital-library/publications/port-or-conversion-an-ontological-framework-for-classifying-game-versions/) from the original on 29 July 2020. Retrieved 5 May 2021.
43. Olle, David; Westcott, Jean Riescher (2018). *Video Game Addiction* (https://books.google.com/books?id=w-pJDwAAQBAJ&pg=PT16). Stylus Publishing, LLC. p. 16. ISBN 978-1-937585-84-6. Archived (https://web.archive.org/web/20180809090716/https://books.google.es/books?id=w-pJDwAAQBAJ&pg=PT16) from the original on 9 August 2018. Retrieved 9 August 2018.
44. Lane, Rick (13 December 2011). "Is PC Gaming Really More Expensive Than Consoles?" (http://www.ign.com/articles/2011/12/13/is-pc-gaming-really-more-expensive-than-consoles). Archived (https://web.archive.org/web/20160813150831/http://www.ign.com/articles/2011/12/13/is-pc-gaming-really-more-expensive-than-consoles) from the original on 13 August 2016. Retrieved 9 August 2016.
45. GameCentral staff (27 June 2013). "Xbox 360 beats Wii as the UK's best-selling console" (http://metro.co.uk/2013/06/27/xbox-360-beats-wii-as-the-uks-best-selling-console-3858990/). *Metro*. Archived (https://web.archive.org/web/20190419064841/https://metro.co.uk/2013/06/27/xbox-360-beats-wii-as-the-uks-best-selling-console-3858990/) from the original on 19 April 2019. Retrieved 31 October 2013.
46. Bashir, Dale (26 October 2020). "10 Gaming Innovations That The PlayStation 2 Brought to Us 20 Years Ago" (https://sea.ign.com/god-of-war/165395/news/10-gaming-innovations-that-the-playstation-2-brought-to-us-20-years-ago).
47. Gelbart, Bryn (13 May 2020). "PlayStation's Long, Complicated History with Backward Compatibility" (https://www.ign.com/articles/sony-playstation-history-with-backward-compatibility).
48. "History of Backwards Compatibility" (https://huguesjohnson.com/features/backwards/). 2024.
49. Hyman, Paul (9 April 2004). "Hollywood Reporter interviewing Doug Lombardi, Quote: "Mods absolutely helped us drive huge sales to 'Half-Life""" (https://web.archive.org/web/20080506004712/http://www.hollywoodreporter.com/hr/search/article_display.jsp?vnu_content_id=1000484956). *The Hollywood Reporter*. Archived from the original on 6 May 2008. Retrieved 10 August 2009..

50. Thomas, David; Orland, Kyle; Steinberg, Scott (2007). *The Videogame Style Guide and Reference Manual* (http://www.gamestyleguide.com/VideoGameStyleGuideeBook.pdf) (PDF). Power Play Publishing. p. 21. ISBN 9781430313052. Archived (https://web.archive.org/web/20110711072136/http://www.gamestyleguide.com/VideoGameStyleGuideeBook.pdf) (PDF) from the original on 11 July 2011. Retrieved 13 September 2021.
51. Apperley, Thomas H. (2006). "Genre and game studies" (https://web.archive.org/web/20131005003753/http://trac.assembla.com/CommanderAssembler/export/32/docs/Genre%20and%20game%20studies%20-%20tom-apperley.pdf) (PDF). *Simulation & Gaming*. **37** (1): 6–23. doi:10.1177/1046878105282278 (https://doi.org/10.1177%2F1046878105282278). ISSN 1046-8781 (https://search.worldcat.org/issn/1046-8781). S2CID 17373114 (https://api.semanticscholar.org/CorpusID:17373114). Archived from the original (http://trac.assembla.com/CommanderAssembler/export/32/docs/Genre%20and%20game%20studies%20-%20tom-apperley.pdf) (PDF) on 5 October 2013. Retrieved 19 April 2013.
52. Adams, Ernest (9 July 2009). "Background: The Origins of Game Genres" (https://web.archive.org/web/20141217155232/http://www.gamasutra.com/view/feature/4074/the_designers_notebook_sorting_.php). *Game Developer*. Archived from the original (http://www.gamasutra.com/view/feature/4074/the_designers_notebook_sorting_.php) on 17 December 2014. Retrieved 7 May 2022.
53. Wolf, Mark J.P. (2008). *The Video Game Explosion: A History from PONG to Playstation and Beyond* (https://books.google.com/books?id=XiM0ntMybNwC&q=video+game+genres). ABC-CLIO. p. 259. ISBN 978-0313338687. Archived (https://web.archive.org/web/20210423094107/https://books.google.com/books?id=XiM0ntMybNwC&q=video+game+genres) from the original on 23 April 2021. Retrieved 3 December 2014.
54. Adams & Rollings 2006, p. 67.
55. Harteveld 2011, p. 71.
56. Perron 2009, pp. 26–45.
57. Parish, Jeremy. "The Essential 50 Part 12 – Rogue" (https://web.archive.org/web/20130228092550/http://www.1up.com/features/essential-50-rogue). *1UP.com*. Archived from the original (http://www.1up.com/features/essential-50-rogue) on 28 February 2013. Retrieved 1 March 2009.
58. Lecky-Thompson 2008, p. 23.
59. Zavarise, Giada (6 December 2018). "How Battle Royale went from a manga to a Fortnite game mode" (https://www.rockpapershotgun.com/2018/12/06/how-battle-royale-went-from-a-manga-to-a-fortnite-game-mode/). *Rock, Paper, Shotgun*. Archived (https://web.archive.org/web/20200726154840/https://www.rockpapershotgun.com/2018/12/06/how-battle-royale-went-from-a-manga-to-a-fortnite-game-mode/) from the original on 26 July 2020. Retrieved 24 May 2020.
60. Arsenault 2009, pp. 149–176.
61. Phillips, Tom (15 April 2020). "Video game pioneer John H. Conway dies aged 82" (https://www.eurogamer.net/articles/2020-04-15-video-game-pioneer-john-h-conway-dies-aged-82). *Eurogamer*. Archived (https://web.archive.org/web/20200427101454/https://www.eurogamer.net/articles/2020-04-15-video-game-pioneer-john-h-conway-dies-aged-82) from the original on 27 April 2020. Retrieved 19 March 2021.
62. Keogh 2015, pp. 152–162.
63. Boyes, Emma (18 February 2008). "GDC '08: Are casual games the future?" (https://web.archive.org/web/20110711072428/http://uk.gamespot.com/news/6186207.html?tag=result%3Btitle%3B0). *CNET*. Archived from the original (https://www.cnet.com/tech/gaming/gdc-08-are-casual-games-the-future/) on 11 July 2011. Retrieved 7 May 2022.

64. Ricchetti, Matt (17 February 2012). "What Makes Social Games Social?" (https://web.archive.org/web/20200812010938/https://www.gamasutra.com/view/feature/6735/what_makes_social_games_social.php?print=1). *Gamasutra*. Archived from the original (https://www.gamasutra.com/view/feature/6735/what_makes_social_games_social.php?print=1) on 12 August 2020. Retrieved 13 August 2020.
65. Bradshaw, Tim (20 August 2020). "How 'hyper-casual' games are winning the mobile market" (https://www.ft.com/content/55f64cfe-125e-4529-b16e-ad77f8cd8710). *Financial Times*. Archived (https://web.archive.org/web/20210113104256/https://www.ft.com/content/55f64cfe-125e-4529-b16e-ad77f8cd8710) from the original on 13 January 2021. Retrieved 29 March 2021.
66. Dondlinger, Mary Jo (2007). "Educational Video Game Design: A Review of the Literature". *Journal of Applied Educational Technology*. **4** (1): 21–31.
67. Walton, Mark (25 November 2012). "Minecraft In Education: How Video Games Are Teaching Kids" (http://www.gamespot.com/articles/minecraft-in-education-how-video-games-are-teaching-kids/1100-6400549/). *GameSpot*. CBS Interactive. Archived (https://web.archive.org/web/20131010082122/http://www.gamespot.com/articles/minecraft-in-education-how-video-games-are-teaching-kids/1100-6400549/) from the original on 10 October 2013. Retrieved 9 December 2012.
68. Davidson, Pete (7 July 2011). "SpaceChem Used as Educational Tool in Schools" (https://web.archive.org/web/20111202190506/http://www.gamepro.com/article/news/220936/spacechem-used-as-educational-tool-in-schools/). GamePro. Archived from the original (http://www.gamepro.com/article/news/220936/spacechem-used-as-educational-tool-in-schools/) on 2 December 2011. Retrieved 18 July 2011.
69. Wiemeyer, Josef; Dörner, Ralf; Göbel, Stefan; Effelsberg, Wolfgang (2016). "1. Introduction". *Serious Games: Foundations, Concepts and Practice*. Springer International Publishing. ISBN 978-3319406121.
70. Djaouti, Damien; Alvarez, Julian; Jessel, Jean-Pierre; Rampnoux, Olivier (2011). "Origins of Serious Games" (https://normandie-univ.hal.science/hal-04024666). *Serious Games and Edutainment Applications*. Springer. pp. 25–43. doi:10.1007/978-1-4471-2161-9_3 (https://doi.org/10.1007%2F978-1-4471-2161-9_3). ISBN 978-1-4471-2160-2.
71. Schilling, Chris (23 July 2009). "Art house video games" (https://www.telegraph.co.uk/technology/video-games/5893530/Art-house-video-games.html). *The Daily Telegraph*. London. Archived (https://ghostarchive.org/archive/20220110/https://www.telegraph.co.uk/technology/video-games/5893530/Art-house-video-games.html) from the original on 10 January 2022.
72. Holmes, Tiffany. *Arcade Classics Span Art? Current Trends in the Art Game Genre (http://hypertext.rmit.edu.au/dac/papers/Holmes.pdf)*. Archived (https://web.archive.org/web/20130420092835/http://hypertext.rmit.edu.au/dac/papers/Holmes.pdf) 2013-04-20 at the Wayback Machine. Melbourne DAC 2003. 2003.
73. Gintere 2019, pp. 346–360.
74. McCauley, Dennis. "The Political Game: A Brief History of Video Game Legislation" (https://web.archive.org/web/20121224031138/http://www.joystiq.com/2006/08/18/the-political-game-a-brief-history-of-video-game-legislation). *joystiq.com*. Archived from the original (http://www.joystiq.com/2006/08/18/the-political-game-a-brief-history-of-video-game-legislation/) on 24 December 2012. Retrieved 20 January 2013.
75. Handrahan, Matthew (9 August 2018). "Germany relaxes stance on Nazi symbols in video games" (https://www.gamesindustry.biz/articles/2018-08-09-germany-relaxes-stance-on-nazi-symbols-in-video-games). *GamesIndustry.biz*. Archived (https://web.archive.org/web/20210713220333/https://www.gamesindustry.biz/articles/2018-08-09-germany-relaxes-stance-on-nazi-symbols-in-video-games) from the original on 13 July 2021. Retrieved 9 August 2018.

76. Jones, Ali (11 December 2018). "Fortnite, PUBG, and Paladins have reportedly been banned by the Chinese government" (https://www.pcgamesn.com/fortnite/fortnite-china-ban). *PCGamesN*. Archived (https://web.archive.org/web/20200806211701/https://www.pcgamesn.com/fortnite/fortnite-china-ban) from the original on 6 August 2020. Retrieved 11 December 2018.
77. Parkin, Simon (3 August 2009). "The Edge of Reason?" (http://www.eurogamer.net/articles/the-edge-of-reason). Eurogamer. Archived (https://web.archive.org/web/20091119022451/http://www.eurogamer.net/articles/the-edge-of-reason) from the original on 19 November 2009. Retrieved 16 November 2009.
78. Reimer, Jeremy (8 November 2005). "Cross-platform game development and the next generation of consoles" (https://arstechnica.com/features/2005/11/crossplatform/). *Ars Technica*. Archived (https://web.archive.org/web/20200603073211/https://arstechnica.com/features/2005/11/crossplatform/) from the original on 3 June 2020. Retrieved 3 June 2020.
79. Kent 2001, p. 65.
80. June, Laura (16 January 2013). "For Amusement Only: The Life and Death of the American Arcade" (https://www.theverge.com/2013/1/16/3740422/the-life-and-death-of-the-american-arcade-for-amusement-only). *The Verge*. Archived (https://web.archive.org/web/20141006081005/http://www.theverge.com/2013/1/16/3740422/the-life-and-death-of-the-american-arcade-for-amusement-only) from the original on 6 October 2014. Retrieved 8 March 2021.
81. Edwards, Benj (26 August 2016). "Son of PC: The History of x86 Game Consoles" (https://www.pcmag.com/news/son-of-pc-the-history-of-x86-game-consoles). *PC Magazine*. Archived (https://web.archive.org/web/20201205092602/https://www.pcmag.com/news/son-of-pc-the-history-of-x86-game-consoles) from the original on 5 December 2020. Retrieved 31 July 2020.
82. "Assassin's Creed II dev team triples in size" (http://www.nowgamer.com/news/513/assassins-creed-ii-triples-size-of-dev-team), Christopher Reynolds, 18 May 2009, NOW Gamer. Archived (https://arquivo.pt/wayback/20160515200949/http://www.nowgamer.com/assassins-creed-ii-dev-team-triples-in-size/) 15 May 2016 at the Portuguese Web Archive
83. Schreier, Jason. "The Messy, True Story Behind The Making of Destiny" (https://kotaku.com/the-messy-true-story-behind-the-making-of-destiny-1737556731). *Kotaku*. Archived (https://web.archive.org/web/20161115203558/http://kotaku.com/the-messy-true-story-behind-the-making-of-destiny-1737556731) from the original on 15 November 2016. Retrieved 15 November 2016.
84. Crogan, Patrick (2018). "Indie Dreams: Video Games, Creative Economy, and the Hyperindustrial Epoch" (http://eprints.uwe.ac.uk/34123/3/Crogan%20for%20Ludic%20Economies%20special%20issue%20v2.pdf) (PDF). *Games and Culture*. **13** (7): 671–689. doi:10.1177/1555412018756708 (https://doi.org/10.1177%2F1555412018756708). S2CID 148890661 (https://api.semanticscholar.org/CorpusID:148890661). Archived (https://web.archive.org/web/20220130011803/https://uwe-repository.worktribe.com/output/857444) from the original on 30 January 2022. Retrieved 27 April 2021.
85. Murray, Janet (1998). *Hamlet on the Holodeck* (https://archive.org/details/hamletonholodeck00murr). MIT Press. p. . ISBN 978-0-262-63187-7.
86. Aarseth, Espen J. (21 May 2004). "Genre Trouble" (http://www.electronicbookreview.com/thread/firstperson/vigilant). Electronic Book Review. Archived (https://web.archive.org/web/20060619063237/http://www.electronicbookreview.com/thread/firstperson/vigilant) from the original on 19 June 2006. Retrieved 14 June 2006.
87. "IGN: GDC 2004: Warren Spector Talks Games Narrative" (https://web.archive.org/web/20090411094042/http://xbox.ign.com/articles/502/502409p1.html). Xbox.ign.com. Archived from the original (http://xbox.ign.com/articles/502/502409p1.html) on 11 April 2009.

88. "Berne Convention for the Protection of Literary and Artistic Works" (http://www.wipo.int/treaties/en/ip/berne/). *www.wipo.int*. Archived (https://web.archive.org/web/20211101035910/https://www.wipo.int/treaties/en/ip/berne/) from the original on 1 November 2021. Retrieved 4 May 2017.
89. Lampros, Nicholas M. (2013). "Leveling Pains: Clone Gaming and the Changing Dynamics of an Industry". *Berkeley Technology Law Journal*. **28**: 743.
90. Chen, Brian X. (11 March 2012). "For Creators of Games, a Faint Line on Cloning" (https://www.nytimes.com/2012/03/12/technology/for-creators-of-video-games-a-faint-line-on-cloning.html?_r=1&pagewanted=all). *The New York Times*. Archived (https://web.archive.org/web/20211101063430/https://www.nytimes.com/2012/03/12/technology/for-creators-of-video-games-a-faint-line-on-cloning.html?_r=1&pagewanted=all) from the original on 1 November 2021. Retrieved 6 September 2019.
91. Webster, Andrew (6 December 2009). "Cloning or theft? Ars explores game design with Jenova Chen" (https://arstechnica.com/gaming/2009/12/cloning-or-theft-ars-explores-game-design-with-jenova-chen/). Ars Technica. Archived (https://web.archive.org/web/20210507202648/https://arstechnica.com/gaming/2009/12/cloning-or-theft-ars-explores-game-design-with-jenova-chen/) from the original on 7 May 2021. Retrieved 20 October 2015.
92. Turner, Benjamin & Bowen, Kevin, Bringin' in the DOOM Clones (https://web.archive.org/web/20120712042757/http://archive.gamespy.com/articles/december03/doom/clones/index.shtml), *GameSpy,* 11 December 2003, Accessed 19 February 2009
93. "Hunt for Grand Theft Auto pirates" (http://news.bbc.co.uk/1/hi/technology/3762350.stm). *BBC News*. 21 October 2004. Archived (https://web.archive.org/web/20090429070441/http://news.bbc.co.uk/1/hi/technology/3762350.stm) from the original on 29 April 2009. Retrieved 26 August 2008.
94. Kelly, Tadhg (5 January 2014). "Why all the Clones" (https://techcrunch.com/2014/01/05/why-all-the-clones/). TechCrunch. Archived (https://web.archive.org/web/20210305174532/https://techcrunch.com/2014/01/05/why-all-the-clones/) from the original on 5 March 2021. Retrieved 20 October 2015.
95. Snyder, Matt (17 May 2018). China's Digital Game Sector (https://www.uscc.gov/sites/default/files/Research/China%27s%20Digital%20Game%20Sector.pdf) (PDF) (Report). United States-China Economic and Security Review Commission. Archived (https://web.archive.org/web/20210321230756/https://www.uscc.gov/sites/default/files/Research/China%27s%20Digital%20Game%20Sector.pdf) (PDF) from the original on 21 March 2021. Retrieved 25 September 2019.
96. Parkin, Simon (23 December 2011). "Clone Wars: is plagiarism killing creativity in the games industry?" (https://www.theguardian.com/technology/gamesblog/2011/dec/21/clone-wars-games-industry-plagiarism). *The Guardian*. Archived (https://web.archive.org/web/20211101224913/https://www.theguardian.com/technology/gamesblog/2011/dec/21/clone-wars-games-industry-plagiarism) from the original on 1 November 2021. Retrieved 7 August 2012.
97. Ernkvist, Mirko (2008). "Down many times, but still playing the game: Creative destruction and industry crashes in the early video game industry 1971–1986". In Gratzer, Karl; Stiefel, Dieter (eds.). *History of Insolvancy and Bankruptcy*. Södertörns högskola. pp. 161–191. ISBN 978-91-89315-94-5.
98. Demaria, Rusel; Wilson, John (2002). *High Score!: The Illustrated History of Electronic Games* (1st ed.). McGraw-Hill Osborne Media. ISBN 0-07-222428-2.
99. Cobbett, Richard (22 September 2017). "From shareware superstars to the Steam gold rush: How indie conquered the PC" (http://www.pcgamer.com/from-shareware-superstars-to-the-steam-gold-rush-how-indie-conquered-the-pc/). *PC Gamer*. Archived (https://web.archive.org/web/20210909121952/https://www.pcgamer.com/from-shareware-superstars-to-the-steam-gold-rush-how-indie-conquered-the-pc/) from the original on 9 September 2021. Retrieved 25 September 2017.

100. Marchand, André; Hennig-Thurau, Thorsten (August 2013). "Value Creation in the Video Game Industry: Industry Economics, Consumer Benefits, and Research Opportunities". *Journal of Interactive Marketing*. **27** (3): 141–157. doi:10.1016/j.intmar.2013.05.001 (https://doi.org/10.1016%2Fj.intmar.2013.05.001).
101. Pulliam-Moore, Charles (21 August 2014). "Women significantly outnumber teenage boys in gamer demographics" (https://www.pbs.org/newshour/nation/female-adults-oust-teenage-boys-largest-gaming-demographic). *PBS News Hour*. Archived (https://web.archive.org/web/20210312225703/https://www.pbs.org/newshour/nation/female-adults-oust-teenage-boys-largest-gaming-demographic) from the original on 12 March 2021. Retrieved 2 April 2021.
102. Sotamaa, Olli (2009). "Studying Game Development Cultures". *Games and Culture*. **4**: 276. doi:10.1177/1555412009339732 (https://doi.org/10.1177%2F1555412009339732). S2CID 8568117 (https://api.semanticscholar.org/CorpusID:8568117).
103. Ashcraft, Brian (18 January 2013). "Why PC Gaming Is Still Niche in Japan" (https://kotaku.com/5977001/why-pc-gaming-is-still-niche-in-japan). *Kotaku*. Archived (https://web.archive.org/web/20180623222027/https://kotaku.com/5977001/why-pc-gaming-is-still-niche-in-japan) from the original on 23 June 2018. Retrieved 23 June 2018.
104. Byford, Sam (20 March 2014). "Japan used to rule video games, so what happened?" (https://www.theverge.com/2014/3/20/5522320/final-fight-can-japans-gaming-industry-be-saved). *The Verge*. Archived (https://web.archive.org/web/20180623222003/https://www.theverge.com/2014/3/20/5522320/final-fight-can-japans-gaming-industry-be-saved) from the original on 23 June 2018. Retrieved 23 June 2018.
105. Lewis, Leo (9 February 2017). "Game on: why Japan's arcades are still winning" (https://www.ft.com/japanarcades). *Financial Times*. Archived (https://web.archive.org/web/20171227122301/https://www.ft.com/japanarcades) from the original on 27 December 2017. Retrieved 26 December 2017.
106. Usher, William (11 March 2012). "PC Game Sales Top $18.6 Billion In 2011" (http://www.cinemablend.com/games/PC-Game-Sales-Top-18-6-Billion-2011-40337.html). *Gaming Blend*. Cinema Blend. Archived (https://web.archive.org/web/20120609070958/http://www.cinemablend.com/games/PC-Game-Sales-Top-18-6-Billion-2011-40337.html) from the original on 9 June 2012. Retrieved 17 March 2012.
107. Romero, Nick (19 March 2020). "Game (still) on: How coronavirus is impacting the gaming industry" (https://ew.com/gaming/coronavirus-videogames-industry-impact/). *Entertainment Weekly*. Archived (https://web.archive.org/web/20200408220631/https://ew.com/gaming/coronavirus-videogames-industry-impact/) from the original on 8 April 2020. Retrieved 27 March 2020.
108. Howley, Daniel (18 March 2020). "The world is turning to video games amid coronavirus outbreak" (https://finance.yahoo.com/news/coronavirus-world-turning-to-video-games-150704969.html). *Yahoo!*. Archived (https://web.archive.org/web/20200408220632/https://finance.yahoo.com/news/coronavirus-world-turning-to-video-games-150704969.html) from the original on 8 April 2020. Retrieved 27 March 2020.
109. Ebert, Roger (16 April 2010). "Video games can never be art" (https://web.archive.org/web/20111010001841/http://blogs.suntimes.com/ebert/2010/04/video_games_can_never_be_art.html). *Chicago Sun-Times*. Archived from the original (http://blogs.suntimes.com/ebert/2010/04/video_games_can_never_be_art.html) on 10 October 2011. Retrieved 31 August 2010.
110. Shackford, Scott (4 April 2013). "The Time Roger Ebert Dismissed Video Games and What Happened Next" (https://reason.com/2013/04/04/the-time-roger-ebert-dismissed-video-gam/). *Reason*. Archived (https://web.archive.org/web/20200914234541/https://reason.com/2013/04/04/the-time-roger-ebert-dismissed-video-gam/) from the original on 14 September 2020. Retrieved 13 August 2020.

111. Kuchera, Ben (27 June 2011). "Supreme Court strikes down video game law on first amendment grounds" (https://arstechnica.com/gaming/2011/06/us-supreme-court-strikes-down-video-game-law-on-first-amendment-grounds/). *Ars Technica*. Archived (https://web.archive.org/web/20200915001410/https://arstechnica.com/gaming/2011/06/us-supreme-court-strikes-down-video-game-law-on-first-amendment-grounds/) from the original on 15 September 2020. Retrieved 13 August 2020.
112. Steinberg, Scott (31 August 2010). "Who says video games aren't art?" (http://www.cnn.com/2010/TECH/gaming.gadgets/08/31/video.games.art.steinberg/). CNN. Archived (https://web.archive.org/web/20100903031954/http://www.cnn.com/2010/TECH/gaming.gadgets/08/31/video.games.art.steinberg/) from the original on 3 September 2010. Retrieved 31 August 2010.
113. Whitten, Sarah (14 February 2020). " 'Sonic the Hedgehog' is up against the stigma of video game flops at the box office" (https://www.cnbc.com/2020/02/14/sonic-the-hedgehog-faces-video-game-adaptation-box-office-curse.html). *CNBC*. Archived (https://web.archive.org/web/20210726194546/https://www.cnbc.com/2020/02/14/sonic-the-hedgehog-faces-video-game-adaptation-box-office-curse.html) from the original on 26 July 2021. Retrieved 16 February 2020.
114. Kroll, Justin (28 May 2020). " 'Sonic the Hedgehog' Sequel in the Works" (https://variety.com/2020/film/news/sonic-the-hedgehog-sequel-1234619356/). *Variety*. Archived (https://web.archive.org/web/20200529030235/https://variety.com/2020/film/news/sonic-the-hedgehog-sequel-1234619356/) from the original on 29 May 2020. Retrieved 28 May 2020.
115. "10.4 The Impact of Video Games on Culture". *Understanding Media and Culture*. University of Minnesota Press. 2010.
116. Lowood, Henry (2005). "Real-Time Performance: Machinima and Game Studies" (https://web.archive.org/web/20060101161233/http://www.idmaa.org/journal/pdf/iDMAa_Journal_Vol_2_No_1_screen.pdf) (PDF). *The International Digital Media & Arts Association Journal*. **2** (1): 10–17. ISSN 1554-0405 (https://search.worldcat.org/issn/1554-0405). Archived from the original (http://www.idmaa.org/journal/pdf/iDMAa_Journal_Vol_2_No_1_screen.pdf) (PDF) on 1 January 2006. Retrieved 22 March 2013.
117. Good, Owen (20 February 2020). "How Lucasfilm used Unreal Engine to make The Mandalorian" (https://www.polygon.com/tv/2020/2/20/21146152/the-mandalorian-making-of-video-unreal-engine-projection-screen). *Polygon*. Archived (https://web.archive.org/web/20201113105740/https://www.polygon.com/tv/2020/2/20/21146152/the-mandalorian-making-of-video-unreal-engine-projection-screen) from the original on 13 November 2020. Retrieved 1 August 2020.
118. Favis, Elise (5 March 2021). "From Star Wars to Marvel, licensed video games are becoming more ambitious. Here's why" (https://www.washingtonpost.com/video-games/2021/03/05/star-wars-indiana-jones-marvel-007-licensed-games/). *The Washington Post*. Archived (https://web.archive.org/web/20211101055942/https://www.washingtonpost.com/video-games/2021/03/05/star-wars-indiana-jones-marvel-007-licensed-games/) from the original on 1 November 2021. Retrieved 26 July 2021.
119. Gee, James Paul (2003). *What Video Games Have to Teach us About Literacy and Learning* (https://archive.org/details/whatvideogamesha0000geej). Palgrave Macmillan. ISBN 978-1-4039-6169-3.
120. James Paul Gee; et al. (2007). "Wired 11.05: View" (https://www.wired.com/wired/archive/11.05/view.html). Codenet, Inc. Archived (https://web.archive.org/web/20080517033243/http://www.wired.com/wired/archive/11.05/view.html) from the original on 17 May 2008. Retrieved 4 December 2007.
121. Glazer, S. (2006). "Video games". *CQ Researcher*. **16**: 937–960. cqresrre2006111000.

122. Green, C. Shawn; Bavelier, Daphne (2003). "Action video game modifies visual selective attention" (https://archive-ouverte.unige.ch/unige:103890). *Nature*. **423** (6939): 534–537. Bibcode:2003Natur.423..534G (https://ui.adsabs.harvard.edu/abs/2003Natur.423..534G). doi:10.1038/nature01647 (https://doi.org/10.1038%2Fnature01647). PMID 12774121 (https://pubmed.ncbi.nlm.nih.gov/12774121). S2CID 1521273 (https://api.semanticscholar.org/CorpusID:1521273). Green & Bavelier. Archived (https://web.archive.org/web/20200606033158/https://archive-ouverte.unige.ch/unige:103890) from the original on 6 June 2020. Retrieved 13 September 2020.
123. Pallavicini, Federica; Ferrari, Ambra; Mantovani, Fabrizia (7 November 2018). "Video Games for Well-Being: A Systematic Review on the Application of Computer Games for Cognitive and Emotional Training in the Adult Population" (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6234876). *Frontiers in Psychology*. **9**: 2127. doi:10.3389/fpsyg.2018.02127 (https://doi.org/10.3389%2Ffpsyg.2018.02127). ISSN 1664-1078 (https://search.worldcat.org/issn/1664-1078). PMC 6234876 (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6234876). PMID 30464753 (https://pubmed.ncbi.nlm.nih.gov/30464753).
124. Brilliant T., Denilson; Nouchi, Rui; Kawashima, Ryuta (25 September 2019). "Does Video Gaming Have Impacts on the Brain: Evidence from a Systematic Review" (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6826942). *Brain Sciences*. **9** (10): 251. doi:10.3390/brainsci9100251 (https://doi.org/10.3390%2Fbrainsci9100251). ISSN 2076-3425 (https://search.worldcat.org/issn/2076-3425). PMC 6826942 (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6826942). PMID 31557907 (https://pubmed.ncbi.nlm.nih.gov/31557907).
125. "D-Lux: The video game festival talking about mental health" (https://www.bbc.co.uk/news/av/uk-scotland-south-scotland-51402936). *BBC News*. 11 February 2020. Archived (https://web.archive.org/web/20210727161310/https://www.bbc.co.uk/news/av/uk-scotland-south-scotland-51402936) from the original on 27 July 2021. Retrieved 27 June 2021.
126. Hern, Alex (16 November 2020). "Video gaming can benefit mental health, find Oxford academics" (https://www.theguardian.com/games/2020/nov/16/video-gaming-can-benefit-mental-health-find-oxford-academics). *The Guardian*. Archived (https://web.archive.org/web/20210623054448/https://www.theguardian.com/games/2020/nov/16/video-gaming-can-benefit-mental-health-find-oxford-academics) from the original on 23 June 2021. Retrieved 27 June 2021.
127. Chilton, Louis (16 November 2020). "Video games can provide benefits to mental health, suggests new Oxford University study" (https://www.independent.co.uk/arts-entertainment/games/video-games-mental-health-study-oxford-b1723729.html). *The Independent*. Archived (https://web.archive.org/web/20211102181814/http://www.independent.co.uk/arts-entertainment/games/video-games-mental-health-study-oxford-b1723729.html) from the original on 2 November 2021. Retrieved 27 June 2021.
128. "U of S research finds video games can relieve stress, improve mental health" (https://www.cbc.ca/news/canada/saskatoon/u-of-s-research-finds-video-games-can-relieve-stress-improve-mental-health-1.5563824). *CBC News*. 10 May 2020. Archived (https://web.archive.org/web/20210804232447/https://www.cbc.ca/news/canada/saskatoon/u-of-s-research-finds-video-games-can-relieve-stress-improve-mental-health-1.5563824) from the original on 4 August 2021. Retrieved 5 August 2021.
129. Bailey, Grant (9 February 2018). "Playing video games is a key strategy for coping with stress, study finds" (https://www.independent.co.uk/life-style/video-games-stress-playing-strategy-key-gamers-study-a8202541.html). *The Independent*. Archived (https://web.archive.org/web/20210804232444/https://www.independent.co.uk/life-style/video-games-stress-playing-strategy-key-gamers-study-a8202541.html) from the original on 4 August 2021. Retrieved 5 August 2021.
130. "A Timeline of Video Game Controversies" (https://ncac.org/resource/a-timeline-of-video-game-controversies). *National Coalition Against Censorship*. Retrieved 28 June 2023.

131. Draper, Kevin (5 August 2019). "Video Games Aren't Why Shootings Happen. Politicians Still Blame Them" (https://www.nytimes.com/2019/08/05/sports/trump-violent-video-games-studies.html). *The New York Times*. Archived (https://web.archive.org/web/20200426031749/https://www.nytimes.com/2019/08/05/sports/trump-violent-video-games-studies.html) from the original on 26 April 2020. Retrieved 22 August 2019.
132. Barrus, Michael M.; Winstanley, Catharine A. (20 January 2016). "Dopamine D3 Receptors Modulate the Ability of Win-Paired Cues to Increase Risky Choice in a Rat Gambling Task" (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6602008). *The Journal of Neuroscience*. **36** (3): 785–794. doi:10.1523/JNEUROSCI.2225-15.2016 (https://doi.org/10.1523%2FJNEUROSCI.2225-15.2016). PMC 6602008 (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6602008). PMID 26791209 (https://pubmed.ncbi.nlm.nih.gov/26791209). S2CID 23617462 (https://api.semanticscholar.org/CorpusID:23617462).
133. Kim, Joseph (23 March 2014). "The Compulsion Loop Explained" (https://www.gamedeveloper.com/business/the-compulsion-loop-explained). *Gamasutra*. Archived (https://web.archive.org/web/20200814040040/https://www.gamasutra.com/blogs/JosephKim/20140323/213728/The_Compulsion_Loop_Explained.php) from the original on 14 August 2020. Retrieved 3 February 2020.
134. Parkin, Simon (8 August 2019). "No, Video Games Don't Cause Mass Shootings. But The Conversation Shouldn't End There" (https://time.com/5647659/no-video-games-dont-cause-mass-shootings-but-the-conversation-shouldnt-end-there/). *Time*. Archived (https://web.archive.org/web/20200903005730/https://time.com/5647659/no-video-games-dont-cause-mass-shootings-but-the-conversation-shouldnt-end-there/) from the original on 3 September 2020. Retrieved 3 February 2020.
135. "National Videogame Museum" (http://www.nvmusa.org). nvmusa.org. Archived (https://web.archive.org/web/20171202165057/http://www.nvmusa.org/) from the original on 2 December 2017.
136. "National Videogame Museum opens in Frisco" (http://www.fox4news.com/good-day/national-videogame-museum-opens-in-frisco). *Fox4news.com*. 12 April 2016. Archived (https://web.archive.org/web/20171107015007/http://www.fox4news.com/good-day/national-videogame-museum-opens-in-frisco) from the original on 7 November 2017.
137. "Computerspielemuseum – Berlin" (http://www.computerspielemuseum.de/index.php?lg=en). Computerspielemuseum.de. Archived (https://web.archive.org/web/20110501141756/http://www.computerspielemuseum.de/index.php?lg=en) from the original on 1 May 2011.
138. "Museum of Soviet arcade machines" (http://www.15kop.ru/en/). 15kop.ru. Archived (https://web.archive.org/web/20100128025438/http://15kop.ru/en/) from the original on 28 January 2010.
139. "Red Penguin: Review of the Museum of Soviet arcade machines" (https://web.archive.org/web/20130921225512/http://www.redpenguin.net/trans-siberian-part-7-the-museum-of-soviet-arcade-machines.php). redpenguin.net. Archived from the original (http://www.redpenguin.net/trans-siberian-part-7-the-museum-of-soviet-arcade-machines.php) on 21 September 2013. Retrieved 1 June 2013.
140. "About The MADE" (https://web.archive.org/web/20130530145047/http://www.themade.org/content/what-are-we). themade.org. Archived from the original (http://www.themade.org/content/what-are-we) on 30 May 2013.
141. "ViGaMus" (http://www.vigamus.com/en/). vigamus.com. Archived (https://web.archive.org/web/20121213050653/http://www.vigamus.com/en/) from the original on 13 December 2012.
142. Wolf, Mark J.P., ed. (2012). "International Center for the History of Electronic Games (ICHEG)". *Encyclopedia of Video Games: The Culture, Technology, and Art of Gaming*. p. 329.

143. Jacobs, Stephen (22 November 2010). "Strong's eGameRevolution Exhibit Gives Game History Its First Permanent Home" (https://web.archive.org/web/20130512142656/http://www.gamasutra.com/view/news/31666/Strongs_eGameRevolution_Exhibit_Gives_Game_History_Its_First_Permanent_Home.php). *Gamasutra*. Archived from the original (http://www.gamasutra.com/view/news/31666/Strongs_eGameRevolution_Exhibit_Gives_Game_History_Its_First_Permanent_Home.php) on 12 May 2013. Retrieved 26 May 2013.
144. "eGameRevolution" (https://web.archive.org/web/20130527093827/http://www.icheg.org/see-do/egamerevolution). International Center for the History of Electronic Games. Archived from the original (http://www.icheg.org/see-do/egamerevolution) on 27 May 2013. Retrieved 26 May 2013.
145. "History of Computing: Video games – Golden Age" (https://web.archive.org/web/20111226002728/http://www.thocp.net/software/games/golden_age.htm#PacMan). *thocp.net*. Archived from the original (http://www.thocp.net/software/games/golden_age.htm#PacMan) on 26 December 2011.
146. Anttonelli, Paola (29 November 2012). "Video Games: 14 in the Collection, for Starters" (http://www.moma.org/explore/inside_out/2012/11/29/video-games-14-in-the-collection-for-starters/). MoMA. Archived (https://web.archive.org/web/20150908154513/http://www.moma.org/explore/inside_out/2012/11/29/video-games-14-in-the-collection-for-starters/) from the original on 8 September 2015.
147. Galloway, Paul (23 June 2013). "Video Games: Seven More Building Blocks in MoMA's Collection" (http://www.moma.org/explore/inside_out/2013/06/28/video-games-seven-more-building-blocks-in-momas-collection). MoMA. Archived (https://web.archive.org/web/20150908155158/http://www.moma.org/explore/inside_out/2013/06/28/video-games-seven-more-building-blocks-in-momas-collection) from the original on 8 September 2015.
148. "Exhibitions: The Art of Video Games" (http://americanart.si.edu/exhibitions/archive/2012/games/). Americanart.si.edu. Archived (https://web.archive.org/web/20110110015838/http://americanart.si.edu/exhibitions/archive/2012/games/) from the original on 10 January 2011. Retrieved 8 July 2012.
149. Barron, Christina (29 April 2012). "Museum exhibit asks: Is it art if you push 'start'?" (https://web.archive.org/web/20130604201354/http://www.washingtonpost.com/gog/exhibits/the-art-of-video-games,1215143/critic-review.html). *The Washington Post*. Archived from the original (https://www.washingtonpost.com/gog/exhibits/the-art-of-video-games,1215143/critic-review.html) on 4 June 2013. Retrieved 12 February 2013.
150. Kennicott, Philip (18 March 2012). "The Art of Video Games" (https://web.archive.org/web/20130604201354/http://www.washingtonpost.com/gog/exhibits/the-art-of-video-games,1215143/critic-review.html). *The Washington Post*. Archived from the original (https://www.washingtonpost.com/gog/exhibits/the-art-of-video-games,1215143/critic-review.html) on 4 June 2013. Retrieved 12 February 2013.

Bibliography

- Kent, Steve L. (2001). *The Ultimate History of Video Games*. Prima. ISBN 0-7615-3643-4.
- Wolf, Mark (2007). "Chapter 1: What Is a Video Game?". In Wolf, Mark (ed.). *The Video Game Explosion*. Westport, CT: Greenwood Press. ISBN 978-0313338687.
- Perron, Bernard (2013). "From gamers to players and gameplayers: The example of interactive movies". In Wolf, Mark JP; Perron, Bernard (eds.). *The Video Game Theory Reader*. Routledge. ISBN 9781135205188.
- Adams, Ernest; Rollings, Andrew (2006). *Fundamentals of Game Design* (https://books.google.com/books?id=Lm1jAgAAQBAJ). Prentice Hall. p. 67. ISBN 978-0-13-343571-9. Archived (https://web.archive.org/web/20200915042311/https://books.google.com/books?id=Lm1jAgAAQBAJ) from the original on 15 September 2020. Retrieved 12 November 2015.

- Harteveld, Casper (2011). *Triadic Game Design: Balancing Reality, Meaning and Play* (https://books.google.com/books?id=sIcQYKcoH7IC&q=lives+game+design). Springer Science & Business Media. ISBN 978-1-84996-157-8. Archived (https://web.archive.org/web/20220130011802/https://books.google.com/books?id=sIcQYKcoH7IC&q=lives+game+design) from the original on 30 January 2022. Retrieved 12 November 2020.
- Perron, Bernard (2009). "Games of Fear: A Multi-Faceted Historical Account of the Horror Genre in Video Games". *Horror Video Games: Essays on the Fusion of Fear and Play*. McFarland & Company. ISBN 978-0786441976.
- Lecky-Thompson, Guy W. (1 January 2008). *Video Game Design Revealed* (https://books.google.com/books?id=3aQLAAAAQBAJ&q=video+game+genres). Cengage Learning. p. 23. ISBN 978-1584506072. Archived (https://web.archive.org/web/20210422130424/https://books.google.com/books?id=3aQLAAAAQBAJ&q=video+game+genres) from the original on 22 April 2021. Retrieved 3 December 2014.
- Arsenault, Dominic (2009). "Video Game Genre, Evolution and Innovation" (https://doi.org/10.7557%2F23.6003). *Eludamos. Journal for Computer Game Culture*. **3** (2): 149–176. doi:10.7557/23.6003 (https://doi.org/10.7557%2F23.6003). S2CID 62171492 (https://api.semanticscholar.org/CorpusID:62171492).
- Keogh, Brendan (2015). "Between Triple-A, indie, casual, and DIY: Sites of tension in the videogames cultural industries". *The Routledge Companion to the Cultural Industries*. Routledge.
- Gintere, Ieva (2019). *A New Digital Art Game: The Art of the Future*. Society. Integration. Education. Proceedings of the International Scientific Conference. Vol. 4.

Further reading

- Blodget, Henry (12 April 2005). "How to Solve China's Piracy Problem" (http://www.slate.com/articles/arts/go_east_young_man/2005/04/how_to_solve_chinas_piracy_problem.html). Slate.com. Archived (https://web.archive.org/web/20111108033525/http://www.slate.com/articles/arts/go_east_young_man/2005/04/how_to_solve_chinas_piracy_problem.html) from the original on 8 November 2011. Retrieved 12 February 2006.
- Costikyan, Greg (1994). "I Have No Words & I Must Design" (https://web.archive.org/web/20080812015347/http://www.costik.com/nowords.html). Archived from the original (http://www.costik.com/nowords.html) on 12 August 2008.
- Crawford, Chris (1982). *The Art of Computer Game Design* (https://web.archive.org/web/20150702204947/http://library.vancouver.wsu.edu/art-computer-game-design). Archived from the original (http://library.vancouver.wsu.edu/art-computer-game-design) on 2 July 2015. Retrieved 2 February 2011.
- Lieu, Tina (August 1997). "Where have all the PC games gone?" (https://web.archive.org/web/19980112155115/http://www.cjmag.co.jp/magazine/issues/1997/aug97/0897pcgames.html). Computing Japan. Archived from the original (http://www.cjmag.co.jp/magazine/issues/1997/aug97/0897pcgames.html) on 12 January 1998.
- Pursell, Carroll (2015). *From Playgrounds to PlayStation: The Interaction of Technology and Play*. Baltimore, MD: Johns Hopkins University Press.
- Salen, Katie; Eric Zimmerman (2005). *The Game Design Reader: A Rules of Play Anthology*. The MIT Press. ISBN 978-0-262-19536-2.
- Smuts, Aaron (2005). "Are Video Games Art?" (http://www.contempaesthetics.org/newvolume/pages/article.php?articleID=299). Archived (https://web.archive.org/web/20101214025334/http://contempaesthetics.org/newvolume/pages/article.php?articleID=299) from the original on 14 December 2010. Retrieved 17 August 2006.

- Winegarner, Beth (28 January 2005). "Game sales hit record highs" (http://www.gamespot.com/news/game-sales-hit-record-highs-6117438). *GameSpot*. Gamespot. Archived (https://web.archive.org/web/20120106224517/http://www.gamespot.com/news/game-sales-hit-record-highs-6117438) from the original on 6 January 2012. Retrieved 12 February 2006.
- John Wills (1 October 2002). "Digital Dinosaurs and Artificial Life: Exploring the Culture of Nature in Computer and Video Games". *Cultural Values (Journal for Cultural Research)*. **6** (4): 395–417. doi:10.1080/1362517022000047334 (https://doi.org/10.1080%2F1362517022000047334). S2CID 144132612 (https://api.semanticscholar.org/CorpusID:144132612).
- Williams, J.P.; Smith, J.H., eds. (2007). *The players' realm: studies on the culture of video games and gaming*. Jefferson, NC: McFarland & Co.
- *The Ultimate History of Video Games, Volume 2: Nintendo, Sony, Microsoft, and the Billion-Dollar Battle to Shape Modern Gaming* by Steven L. Kent, Crown, 2021, ISBN 1984825437

External links

- Video games bibliography (http://www.ludoscience.com/EN/ressources/bibliographie/index.html?sort=files%20ASC) by the French video game research association *Ludoscience*
- The Virtual Museum of Computing (VMoC) (https://web.archive.org/web/20141010105238/http://archives.icom.museum/vlmp/computing.html) (archived 10 October 2014)

WIKIPEDIA
The Free Encyclopedia

Web application

A **web application** (or **web app**) is application software that is created with web technologies and runs via a web browser.[1][2] Web applications emerged during the late 1990s and allowed for the server to dynamically build a response to the request, in contrast to static web pages.[3]

Screenshot from 2007 of Horde, a groupware and open-source web application

Web applications are commonly distributed via a web server. There are several different tier systems that web applications use to communicate between the web browsers, the client interface, and server data. Each system has its own uses as they function in different ways. However, there are many security risks that developers must be aware of during development; proper measures to protect user data are vital.

Web applications are often constructed with the use of a web application framework. Single-page applications (SPAs) and progressive web apps (PWAs) are two architectural approaches to creating web applications that provide a user experience similar to native apps, including features such as smooth navigation, offline support, and faster interactions.

History

The concept of a "web application" was first introduced in the Java language in the Servlet Specification version 2.2, which was released in 1999. At that time, both JavaScript and XML had already been developed, but the XMLHttpRequest object had only been recently introduced on Internet Explorer 5 as an ActiveX object.[*citation needed*] Beginning around the early 2000s, applications such as "Myspace (2003), Gmail (2004), Digg (2004), [and] Google Maps (2005)," started to make their client sides more and more interactive. A web page script is able to contact the server for storing/retrieving data without downloading an entire web page. The practice became known as Ajax in 2005.

In earlier computing models like client-server, the processing load for the application was shared between code on the server and code installed on each client locally. In other words, an application had its own pre-compiled client program which served as its user interface and had to be separately installed on each user's personal computer. An upgrade to the server-side code of the application would typically also require an upgrade to the client-side code installed on each user workstation, adding to the support cost and decreasing productivity. Additionally, both the client and server components of the application were bound tightly to a particular computer architecture and operating system, which made porting them to other systems prohibitively expensive for all but the largest applications.

Later, in 1995, Netscape introduced the client-side scripting language called JavaScript, which allowed programmers to add dynamic elements to the user interface that ran on the client side. Essentially, instead of sending data to the server in order to generate an entire web page, the embedded scripts of the downloaded page can perform various tasks such as input validation or showing/hiding parts of the page.

"Progressive web apps", the term coined by designer Frances Berriman and Google Chrome engineer Alex Russell in 2015, refers to apps taking advantage of new features supported by modern browsers, which initially run inside a web browser tab but later can run completely offline and can be launched without entering the app URL in the browser.

Structure

Traditional PC applications are typically single-tiered, residing solely on the client machine. In contrast, web applications inherently facilitate a multi-tiered architecture. Though many variations are possible, the most common structure is the three-tiered application. In its most common form, the three tiers are called *presentation*, *application* and *storage*. The first tier, presentation, refers to a web browser itself. The second tier refers to any engine using dynamic web content technology (such as ASP, CGI, ColdFusion, Dart, JSP/Java, Node.js, PHP, Python or Ruby on Rails). The third tier refers to a database that stores data and determines the structure of a user interface. Essentially, when using the three-tiered system, the web browser sends requests to the engine, which then services them by making queries and updates against the database and generates a user interface.

The 3-tier solution may fall short when dealing with more complex applications, and may need to be replaced with the n-tiered approach; the greatest benefit of which is how business logic (which resides on the application tier) is broken down into a more fine-grained model.[4] Another benefit would be to add an integration tier, which separates the data tier and provides an easy-to-use interface to access the data.[4] For example, the client data would be accessed by calling a "list_clients()" function instead of making an SQL query directly against the client table on the database. This allows the underlying database to be replaced without making any change to the other tiers.[4]

There are some who view a web application as a two-tier architecture. This can be a "smart" client that performs all the work and queries a "dumb" server, or a "dumb" client that relies on a "smart" server.[4] The client would handle the presentation tier, the server would have the database (storage tier), and the business logic (application tier) would be on one of them or on both.[4] While this increases the scalability of the applications and separates the display and the database, it still does not allow for true specialization of layers, so most applications will outgrow this model.[4]

Security

Security breaches on these kinds of applications are a major concern because it can involve both enterprise information and private customer data. Protecting these assets is an important part of any web application, and there are some key operational areas that must be included in the development process.[5] This includes processes for authentication, authorization, asset handling, input, and logging and auditing. Building security into the applications from the beginning is sometimes more effective and less disruptive in the long run.

Development

Writing web applications is simplified with the use of web application frameworks. These frameworks facilitate rapid application development by allowing a development team to focus on the parts of their application which are unique to their goals without having to resolve common development issues such as user management.[6]

In addition, there is potential for the development of applications on Internet operating systems, although currently there are not many viable platforms that fit this model.

See also

Internet portal

- Web API
- Software as a service (SaaS)
- Web 2.0
- Web engineering
- Web GIS
- Web services
- Web sciences
- Web widget

References

1. "Web app | Definition, History, Development, Examples, Uses, & Facts | Britannica" (https://www.britannica.com/topic/Web-application). *www.britannica.com*. Encyclopædia Britannica. Retrieved 4 November 2024.
2. "What is a Web App? - Web Application Explained - AWS" (https://aws.amazon.com/what-is/web-application/). *Amazon Web Services, Inc*. Retrieved 4 November 2024.
3. "Web applications" (https://web.archive.org/web/20150419035250/http://docforge.com/wiki/Web_application). *DocForge*. Archived from the original (http://docforge.com/wiki/Web_application) on 19 April 2015. Retrieved 9 November 2024.
4. Petersen, Jeremy (4 September 2008). "Benefits of using the n-tiered approach for web applications" (http://krunal-ajax-javascript.blogspot.bg/2008/09/benefits-of-using-n-tiered-approach-for.html). Archived (https://web.archive.org/web/20171201080946/http://krunal-ajax-javascript.blogspot.bg/2008/09/benefits-of-using-n-tiered-approach-for.html) from the original on 1 December 2017. Retrieved 24 November 2017.
5. "Top Tips for Secure App Development" (https://web.archive.org/web/20120522022522/http://content.dell.com/us/en/enterprise/d/large-business/secure-app-development.aspx). Dell.com. Archived from the original (http://content.dell.com/us/en/enterprise/d/large-business/secure-app-development.aspx) on 2012-05-22. Retrieved 2012-06-22.
6. Multiple (wiki). "Web application framework" (https://web.archive.org/web/20200620213429/http://www.docforge.com/wiki/Web_application_framework). *Docforge*. Archived from the original (http://docforge.com/wiki/Web_application_framework) on 2020-06-20. Retrieved 2010-03-06.

External links

- HTML5 Draft recommendation (https://www.whatwg.org/specs/web-apps/current-work/multipage/), changes to HTML and related APIs to ease authoring of web-based applications.
- Web Applications Working Group (http://www.w3.org/2008/webapps/) at the World Wide Web Consortium (http://www.w3.org/) (W3C)
- PWAs on Web.dev (https://web.dev/progressive-web-apps/) by Google Developers.

E-commerce

(Redirected from E-commerce software)

E-commerce (**electronic commerce**) refers to commercial activities including the electronic buying or selling products and services which are conducted on online platforms or over the Internet.[1] E-commerce draws on technologies such as mobile commerce, electronic funds transfer, supply chain management, Internet marketing, online transaction processing, electronic data interchange (EDI), inventory management systems, and automated data collection systems. E-commerce is the largest sector of the electronics industry and is in turn driven by the technological advances of the semiconductor industry.

Defining e-commerce

The term was coined and first employed by Robert Jacobson, Principal Consultant to the California State Assembly's Utilities & Commerce Committee, in the title and text of California's Electronic Commerce Act, carried by the late Committee Chairwoman Gwen Moore (D-L.A.) and enacted in 1984.

E-commerce typically uses the web for at least a part of a transaction's life cycle although it may also use other technologies such as e-mail. Typical e-commerce transactions include the purchase of products (such as books from Amazon) or services (such as music downloads in the form of digital distribution such as the iTunes Store).[2] There are three areas of e-commerce: online retailing, electronic markets, and online auctions. E-commerce is supported by electronic business.[3] The existence value of e-commerce is to allow consumers to shop online and pay online through the Internet, saving the time and space of customers and enterprises, greatly improving transaction efficiency, especially for busy office workers, and also saving a lot of valuable time.[4]

E-commerce businesses may also employ some or all of the following:

- Online shopping for retail sales direct to consumers via web sites and mobile apps, conversational commerce via live chat, chatbots, and voice assistants.[5]
- Providing or participating in online marketplaces, which process third-party business-to-consumer (B2C) or consumer-to-consumer (C2C) sales;
- Business-to-business (B2B) buying and selling.[6]
- Gathering and using demographic data through web contacts and social media.
- B2B electronic data interchange.
- Marketing to prospective and established customers by e-mail or fax (for example, with newsletters).
- Engaging in pretail for launching new products and services.
- Online financial exchanges for currency exchanges or trading purposes.

There are five essential categories of E-commerce:[7]

- Business to Business
- Business to Consumer

- Business to Government
- Consumer to Business
- Consumer to Consumer

Forms

Contemporary electronic commerce can be classified into two categories. The first category is business based on types of goods sold (involves everything from ordering "digital" content for immediate online consumption, to ordering conventional goods and services, to "meta" services to facilitate other types of electronic commerce). The second category is based on the nature of the participant (B2B, B2C, C2B and C2C).[8]

On the institutional level, big corporations and financial institutions use the internet to exchange financial data to facilitate domestic and international business. Data integrity and security are pressing issues for electronic commerce.

Aside from traditional e-commerce, the terms m-Commerce (mobile commerce) as well (around 2013) t-Commerce[9] have also been used.

Governmental regulation

In the United States, California's Electronic Commerce Act (1984), enacted by the Legislature, the more recent California Privacy Rights Act (2020), enacted through a popular election proposition and to control specifically how electronic commerce may be conducted in California. In the US in its entirety, electronic commerce activities are regulated more broadly by the Federal Trade Commission (FTC). These activities include the use of commercial e-mails, online advertising and consumer privacy. The CAN-SPAM Act of 2003 establishes national standards for direct marketing over e-mail. The Federal Trade Commission Act regulates all forms of advertising, including online advertising, and states that advertising must be truthful and non-deceptive.[10] Using its authority under Section 5 of the FTC Act, which prohibits unfair or deceptive practices, the FTC has brought a number of cases to enforce the promises in corporate privacy statements, including promises about the security of consumers' personal information.[11] As a result, any corporate privacy policy related to e-commerce activity may be subject to enforcement by the FTC.

The Ryan Haight Online Pharmacy Consumer Protection Act of 2008, which came into law in 2008, amends the Controlled Substances Act to address online pharmacies.[12]

Conflict of laws in cyberspace is a major hurdle for harmonization of legal framework for e-commerce around the world. In order to give a uniformity to e-commerce law around the world, many countries adopted the UNCITRAL Model Law on Electronic Commerce (1996).[13]

Internationally there is the International Consumer Protection and Enforcement Network (ICPEN), which was formed in 1991 from an informal network of government customer fair trade organisations. The purpose was stated as being to find ways of co-operating on tackling consumer problems connected with cross-border transactions in both goods and services, and to help ensure exchanges of information among the participants for mutual benefit and understanding. From this came Econsumer.gov, an ICPEN initiative since April 2001. It is a portal to report complaints about online and related transactions with foreign companies.

There is also **Asia Pacific Economic Cooperation**. APEC was established in 1989 with the vision of achieving stability, security and prosperity for the region through free and open trade and investment. APEC has an Electronic Commerce Steering Group as well as working on common privacy regulations throughout the APEC region.

In **Australia**, trade is covered under Australian Treasury Guidelines for electronic commerce and the Australian Competition & Consumer Commission[14] regulates and offers advice on how to deal with businesses online,[15] and offers specific advice on what happens if things go wrong.[16]

The **European Union** undertook an extensive enquiry into e-commerce in 2015–16 which observed significant growth in the development of e-commerce, along with some developments which raised concerns, such as increased use of selective distribution systems, which allow manufacturers to control routes to market, and "increased use of contractual restrictions to better control product distribution". The European Commission felt that some emerging practices might be justified if they could improve the quality of product distribution, but "others may unduly prevent consumers from benefiting from greater product choice and lower prices in e-commerce and therefore warrant Commission action" in order to promote compliance with EU competition rules.[17]

In **the United Kingdom**, the Financial Services Authority (FSA)[18] was formerly the regulating authority for most aspects of the EU's Payment Services Directive (PSD), until its replacement in 2013 by the Prudential Regulation Authority and the Financial Conduct Authority.[19] The UK implemented the PSD through the Payment Services Regulations 2009 (PSRs), which came into effect on 1 November 2009. The PSR affects firms providing payment services and their customers. These firms include banks, non-bank credit card issuers and non-bank merchant acquirers, e-money issuers, etc. The PSRs created a new class of regulated firms known as payment institutions (PIs), who are subject to prudential requirements. Article 87 of the PSD requires the European Commission to report on the implementation and impact of the PSD by 1 November 2012.[20]

In **India**, the Information Technology Act 2000 governs the basic applicability of e-commerce.

In **China**, the Telecommunications Regulations of the People's Republic of China (promulgated on 25 September 2000), stipulated the Ministry of Industry and Information Technology (MIIT) as the government department regulating all telecommunications related activities, including electronic commerce.[21] On the same day, the Administrative Measures on Internet Information Services were released, the first administrative regulations to address profit-generating activities conducted through the Internet, and lay the foundation for future regulations governing e-commerce in China.[22] On 28 August 2004, the eleventh session of the tenth NPC Standing Committee adopted an Electronic Signature Law, which regulates data message, electronic signature authentication and legal liability issues. It is considered the first law in China's e-commerce legislation. It was a milestone in the course of improving China's electronic commerce legislation, and also marks the entering of China's rapid development stage for electronic commerce legislation.[23]

Global trends

E-commerce has become an important tool for small and large businesses worldwide, not only to sell to customers, but also to engage them.[24][25]

Cross-border e-Commerce is also an essential field for e-Commerce businesses. It has responded to the trend of globalization. It shows that numerous firms have opened up new businesses, expanded new markets, and overcome trade barriers; more and more enterprises have started exploring the cross-border cooperation field. In addition, compared with traditional cross-border trade, the information on cross-border e-commerce is more concealed. In the era of globalization, cross-border e-commerce for inter-firm companies means the activities, interactions, or social relations of two or more e-commerce enterprises. However, the success of cross-border e-commerce promotes the development of small and medium-sized firms, and it has finally become a new transaction mode. It has helped the companies solve financial problems and realize the reasonable allocation of resources field. SMEs (small and medium enterprises) can also precisely match the demand and supply in the market, having the industrial chain majorization and creating more revenues for companies.[26]

In 2012, e-commerce sales topped $1 trillion for the first time in history.[27]

Mobile devices are playing an increasing role in the mix of e-commerce, this is also commonly called mobile commerce, or m-commerce. In 2014, one estimate saw purchases made on mobile devices making up 25% of the market by 2017.[28]

For traditional businesses, one research stated that information technology and cross-border e-commerce is a good opportunity for the rapid development and growth of enterprises. Many companies have invested an enormous volume of investment in mobile applications. The DeLone and McLean Model stated that three perspectives contribute to a successful e-business: information system quality, service quality and users' satisfaction.[29] There is no limit of time and space, there are more opportunities to reach out to customers around the world, and to cut down unnecessary intermediate links, thereby reducing the cost price, and can benefit from one on one large customer data analysis, to achieve a high degree of personal customization strategic plan, in order to fully enhance the core competitiveness of the products in the company.[30]

Modern 3D graphics technologies, such as Facebook 3D Posts, are considered by some social media marketers and advertisers as a preferable way to promote consumer goods than static photos, and some brands like Sony are already paving the way for augmented reality commerce. Wayfair now lets you inspect a 3D version of its furniture in a home setting before buying.[31]

China

Among emerging economies, China's e-commerce presence continues to expand every year. With 668 million Internet users as of 2014, China's online shopping sales reached $253 billion in the first half of 2015, accounting for 10% of total Chinese consumer retail sales in that period.[32] The Chinese retailers have been able to help consumers feel more comfortable shopping online.[33] e-commerce transactions between China and other countries increased 32% to 2.3 trillion yuan ($375.8 billion) in 2012 and accounted for 9.6% of China's total international trade.[34] In 2013, Alibaba had an e-commerce market share of 80% in China.[35] In 2014, Alibaba still dominated the B2B marketplace in China with a market share of 44.82%, followed by several other companies including Made-in-China.com at 3.21%, and GlobalSources.com at 2.98%, with the total transaction value of China's B2B market exceeding 4.5 billion yuan.[36]

China is also the largest e-commerce market in the world by value of sales, with an estimated US$899 billion in 2016.[37] It accounted for 42.4% of worldwide retail e-commerce in that year, the most of any country.[38]:110 Research shows that Chinese consumer motivations are different enough from Western audiences to require unique e-commerce app designs instead of simply porting Western apps into the Chinese market.[39]

The expansion of e-commerce in China has resulted in the development of Taobao villages, clusters of e-commerce businesses operating in rural areas.[38]:112 Because Taobao villages have increased the incomes or rural people and entrepreneurship in rural China, Taobao villages have become a component of rural revitalization strategies.[40]:278

In 2015, the State Council promoted the Internet Plus initiative, a five-year plan to integrate traditional manufacturing and service industries with big data, cloud computing, and Internet of things technology.[41]:44 The State Council provided support for Internet Plus through policy support in area including cross-border e-commerce and rural e-commerce.[41]:44

In 2019, the city of Hangzhou established a pilot program artificial intelligence-based Internet Court to adjudicate disputes related to e-commerce and internet-related intellectual property claims.[42]:124

Europe

In 2010, the United Kingdom had the highest per capita e-commerce spending in the world.[43] As of 2013, the Czech Republic was the European country where e-commerce delivers the biggest contribution to the enterprises' total revenue. Almost a quarter (24%) of the country's total turnover is generated via the online channel.[44]

Arab states

The rate of growth of the number of internet users in the Arab countries has been rapid – 13.1% in 2015. A significant portion of the e-commerce market in the Middle East comprises people in the 30–34 year age group. Egypt has the largest number of internet users in the region, followed by Saudi Arabia and Morocco; these constitute 3/4th of the region's share. Yet, internet penetration is low: 35% in Egypt and 65% in Saudi Arabia.[45]

The Gulf Cooperation Council countries have a rapidly growing market and are characterized by a population that becomes wealthier (Yuldashev). As such, retailers have launched Arabic-language websites as a means to target this population. Secondly, there are predictions of increased mobile purchases and an expanding internet audience (Yuldashev). The growth and development of the two aspects make the GCC countries become larger players in the electronic commerce market with time progress. Specifically, research shows that the e-commerce market is expected to grow to over $20 billion by 2020 among these GCC countries (Yuldashev). The e-commerce market has also gained much popularity among western countries, and in particular Europe and the U.S. These countries have been highly characterized by consumer-packaged goods (CPG) (Geisler, 34). However, trends show that there are future signs of a reverse. Similar to the GCC countries, there has been increased purchase of goods and services in online channels rather than offline channels. Activist investors are trying hard to consolidate and slash their overall cost and the governments in western countries continue to impose more regulation on CPG manufacturers (Geisler, 36). In these senses, CPG investors are being forced to adapt to e-commerce as it is effective as well as a means for them to thrive.

The future trends in the GCC countries will be similar to that of the western countries. Despite the forces that push business to adapt e-commerce as a means to sell goods and products, the manner in which customers make purchases is similar in countries from these two regions. For instance, there has been an increased usage of smartphones which comes in conjunction with an increase in the overall internet audience from the regions. Yuldashev writes that consumers are scaling up to more modern technology that allows for mobile marketing. However, the percentage of smartphone and internet users who make online purchases is expected to vary in the first few years. It will be independent on the willingness of the people to adopt this new trend (The Statistics Portal). For example, UAE has the greatest smartphone penetration of 73.8 per cent and has 91.9 per cent of its population has access to the internet. On the other hand, smartphone penetration in Europe has been reported to be at 64.7 per cent (The Statistics Portal). Regardless, the disparity in percentage between these regions is expected to level out in future because e-commerce technology is expected to grow to allow for more users.

The e-commerce business within these two regions will result in competition. Government bodies at the country level will enhance their measures and strategies to ensure sustainability and consumer protection (Krings, et al.). These increased measures will raise the environmental and social standards in the countries, factors that will determine the success of the e-commerce market in these countries. For example, an adoption of tough sanctions will make it difficult for companies to enter the e-commerce market while lenient sanctions will allow ease of companies. As such, the future trends between GCC countries and the Western countries will be independent of these sanctions (Krings, et al.). These countries need to make rational conclusions in coming up with effective sanctions.

India

India has an Internet user base of about 460 million as of December 2017.[46] Despite being the third largest user base in the world, the penetration of the Internet is low compared to markets like the United States, United Kingdom or France but is growing at a much faster rate, adding around six million new entrants every month. In India, cash on delivery is the most preferred payment method, accumulating 75% of the e-retail activities.[47] The India retail market is expected to rise from 2.5% in 2016 to 5% in 2020.[48]

Brazil

In 2013, Brazil's e-commerce was growing quickly with retail e-commerce sales expected to grow at a double-digit pace through 2014. By 2016, eMarketer expected retail e-commerce sales in Brazil to reach $17.3 billion.[49]

Logistics

Logistics in e-commerce mainly concerns fulfillment. Online markets and retailers have to find the best possible way to fill orders and deliver products. Small companies usually control their own logistic operation because they do not have the ability to hire an outside company. Most large companies hire a fulfillment service that takes care of a company's logistic needs.[50] The optimization of logistics processes that contains long-term investment in an efficient storage infrastructure system and adoption of inventory management strategies is crucial to prioritize customer satisfaction throughout the entire process, from order placement to final delivery.[51]

Impacts

Impact on markets and retailers

Store closing flags outside a Toys R Us in Deptford, New Jersey. Despite investments, the chain struggled to win market share in the age of digital commerce.

E-commerce markets are growing at noticeable rates. The online market is expected to grow by 56% in 2015–2020. In 2017, retail e-commerce sales worldwide amounted to 2.3 trillion US dollars and e-retail revenues are projected to grow to 4.891 trillion US dollars in 2021.[52] Traditional markets are only expected 2% growth during the same time. Brick and mortar retailers are struggling because of online retailer's ability to offer lower prices and higher efficiency. Many larger retailers are able to maintain a presence offline and online by linking physical and online offerings.[53]

E-commerce allows customers to overcome geographical barriers and allows them to purchase products anytime and from anywhere. Online and traditional markets have different strategies for conducting business. Traditional retailers offer fewer assortment of products because of shelf space where, online retailers often hold no inventory but send customer orders directly to the manufacturer. The pricing strategies are also different for traditional and online retailers. Traditional retailers base their prices on store traffic and the cost to keep inventory. Online retailers base prices on the speed of delivery.

There are two ways for marketers to conduct business through e-commerce: fully online or online along with a brick and mortar store. Online marketers can offer lower prices, greater product selection, and high efficiency rates. Many customers prefer online markets if the products can be delivered quickly at relatively low price. However, online retailers cannot offer the physical experience that traditional retailers can. It can be difficult to judge the quality of a product without the physical experience, which may cause customers to experience product or seller uncertainty. Another issue regarding the online market is concerns about the security of online transactions. Many customers remain loyal to well-known retailers because of this issue.[54]

Security is a primary problem for e-commerce in developed and developing countries. E-commerce security is protecting businesses' websites and customers from unauthorized access, use, alteration, or destruction. The type of threats include: malicious codes, unwanted programs (ad ware, spyware), phishing, hacking, and cyber vandalism. E-commerce websites use different tools to avert security threats. These tools include firewalls, encryption software, digital certificates, and passwords.

Impact on supply chain management

For a long time, companies had been troubled by the gap between the benefits which supply chain technology has and the solutions to deliver those benefits. However, the emergence of e-commerce has provided a more practical and effective way of delivering the benefits of the new supply chain technologies.[55]

E-commerce has the capability to integrate all inter-company and intra-company functions, meaning that the three flows (physical flow, financial flow and information flow) of the supply chain could be also affected by e-commerce. The affections on physical flows improved the way of product and inventory movement level for companies. For the information flows, e-commerce optimized the capacity of information processing than companies used to have, and for the financial flows, e-commerce allows companies to have more efficient payment and settlement solutions.[55]

In addition, e-commerce has a more sophisticated level of impact on supply chains: Firstly, the performance gap will be eliminated since companies can identify gaps between different levels of supply chains by electronic means of solutions; Secondly, as a result of e-commerce emergence, new capabilities such implementing ERP systems, like SAP ERP, Xero, or Megaventory, have helped companies to manage operations with customers and suppliers. Yet these new capabilities are still not fully exploited. Thirdly, technology companies would keep investing on new e-commerce software solutions as they are expecting investment return. Fourthly, e-commerce would help to solve many aspects of issues that companies may feel difficult to cope with, such as political barriers or cross-country changes. Finally, e-commerce provides companies a more efficient and effective way to collaborate with each other within the supply chain.[55]

Impact on employment

E-commerce helps create new job opportunities due to information related services, software app and digital products. It also causes job losses. The areas with the greatest predicted job-loss are retail, postal, and travel agencies. The development of e-commerce will create jobs that require highly skilled workers to manage large amounts of information, customer demands, and production processes. In contrast, people with poor technical skills cannot enjoy the wages welfare. On the other hand, because e-commerce requires sufficient stocks that could be delivered to customers in time, the warehouse becomes an important element. Warehouse needs more staff to manage, supervise and organize, thus the condition of warehouse environment will be concerned by employees.[56]

Impact on customers

E-commerce brings convenience for customers as they do not have to leave home and only need to browse websites online, especially for buying products which are not sold in nearby shops. It could help customers buy a wider range of products and save customers' time. Consumers also gain power through online shopping. They are able to research products and compare prices among retailers. Thanks to the practice of user-generated ratings and reviews from companies like Bazaarvoice, Trustpilot, and Yelp, customers can also see what other people think of a product, and decide before buying if they want to spend money on it.[57][58] Also, online shopping often provides sales promotion or discounts code, thus it is more price effective for customers. Moreover, e-commerce provides products' detailed information; even the in-store staff cannot offer such detailed explanation. Customers can also review and track the order history online.

E-commerce technologies cut transaction costs by allowing both manufactures and consumers to skip through the intermediaries. This is achieved through by extending the search area best price deals and by group purchase. The success of e-commerce in urban and regional levels depend on how the local firms and consumers have adopted to e-commerce.[59]

However, e-commerce lacks human interaction for customers, especially who prefer face-to-face connection. Customers are also concerned with the security of online transactions and tend to remain loyal to well-known retailers. In recent years, clothing retailers such as Tommy Hilfiger have started adding Virtual Fit platforms to their e-commerce sites to reduce the risk of customers buying the wrong sized clothes, although these vary greatly in their fit for purpose.[60] When the customer regret the purchase of a product, it involves returning goods and refunding process. This process is inconvenient as customers need to pack and post the goods. If the products are expensive, large or fragile, it refers to safety issues.[53]

Impact on the environment

In 2018, E-commerce generated 1.3 million short tons (1.2 megatonnes) of container cardboard in North America, an increase from 1.1 million (1.00)) in 2017. Only 35 percent of North American cardboard manufacturing capacity is from recycled content. The recycling rate in Europe is 80 percent and Asia is 93 percent. Amazon, the largest user of boxes, has a strategy to cut back on packing material and has reduced packaging material used by 19 percent by weight since 2016. Amazon is requiring retailers to manufacture their product packaging in a way that does not require additional shipping packaging. Amazon also has an 85-person team researching ways to reduce and improve their packaging and shipping materials.[61]

Accelerated movement of packages around the world includes accelerated movement of living things, with all its attendant risks.[62] Weeds, pests, and diseases all sometimes travel in packages of seeds.[62] Some of these packages are part of brushing manipulation of e-commerce reviews.[62]

Impact on traditional retail

E-commerce has been cited as a major force for the failure of major U.S. retailers in a trend frequently referred to as a "retail apocalypse."[63] The rise of e-commerce outlets like Amazon has made it harder for traditional retailers to attract customers to their stores and forced companies to change their sales strategies. Many companies have turned to sales promotions and increased digital efforts to lure shoppers while shutting down brick-and-mortar locations.[64] The trend has forced some traditional retailers to shutter its brick and mortar operations.[65]

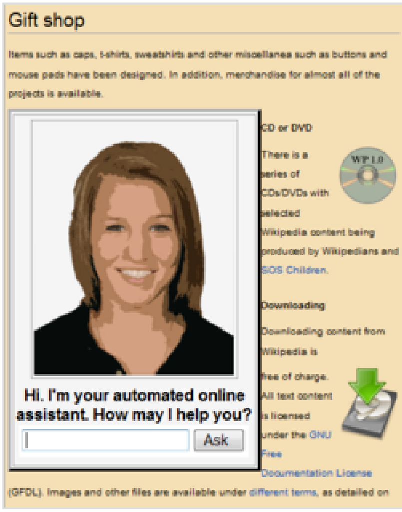
E-commerce during COVID-19

In March 2020, global retail website traffic hit 14.3 billion visits[66] signifying an unprecedented growth of e-commerce during the lockdown of 2020. Later studies show that online sales increased by 25% and online grocery shopping increased by over 100% during the crisis in the United States.[67] Meanwhile, as many as 29% of surveyed shoppers state that they will never go back to shopping in person again; in the UK, 43% of consumers state that they expect to keep on shopping the same way even after the lockdown is over.[68]

Retail sales of e-commerce shows that COVID-19 has a significant impact on e-commerce and its sales are expected to reach $6.5 trillion by 2023.[69]

Business application

Some common applications related to electronic commerce are:



An example of an older generation of avatar-style automated online assistant on a merchandising website

- B2B e-commerce (business-to-business)
- B2C e-commerce (business-to-consumer)
- Conversational commerce: e-commerce via chat
- Digital Wallet
- Document automation in supply chain and logistics
- Electronic tickets
- Enterprise content management
- Group buying
- Instant messaging
- Internet security
- Online auction
- Online banking
- Online office suites
- Online shopping and order tracking
- Online transaction processing
- Pretail
- Print on demand
- Shopping cart software
- Social networking
- Teleconference
- Usenet newsgroup
- Virtual assistant
- Domestic and international payment systems

Timeline

A timeline for the development of e-commerce:

- 1971 or 1972: The ARPANET is used to arrange a cannabis sale between students at the Stanford Artificial Intelligence Laboratory and the Massachusetts Institute of Technology, later described as "the seminal act of e-commerce" in John Markoff's book *What the Dormouse Said*.[70]
- 1979: Michael Aldrich demonstrates the first online shopping system.[71]
- 1981: Thomson Holidays UK is the first business-to-business (B2B) online shopping system to be installed.[72]
- 1982: Minitel was introduced nationwide in France by France Télécom and used for online ordering.
- 1983: California State Assembly holds first hearing on "electronic commerce" in Volcano, California.[73] Testifying are CPUC, MCI Mail, Prodigy, CompuServe, Volcano Telephone, and Pacific Telesis. (Not permitted to testify is Quantum Technology, later to become AOL.) California's Electronic Commerce Act was passed in 1984.

- 1983: Karen Earle Lile (AKA Karen Bean) and Kendall Ross Bean create e-commerce service in San Francisco Bay Area. Buyers and sellers of pianos connect through a database created by Piano Finders on a Kaypro personal computer using DOS interface. Pianos for sale are listed on a Bulletin board system. Buyers print list of pianos for sale by a dot matrix printer. Customer service happened through a Piano Advice Hotline listed in the San Francisco Chronicle classified ads and money transferred by a bank wire transfer when a sale was completed.[74][75]
- 1984: Gateshead SIS/Tesco is first B2C online shopping system[76] and Mrs Snowball, 72, is the first online home shopper[77]
- 1984: In April 1984, CompuServe launches the Electronic Mall in the US and Canada. It is the first comprehensive electronic commerce service.[78]
- 1989: In May 1989, Sequoia Data Corp. introduced Compumarket, the first internet based system for e-commerce. Sellers and buyers could post items for sale and buyers could search the database and make purchases with a credit card.
- 1990: Tim Berners-Lee writes the first web browser, WorldWideWeb, using a NeXT computer.[79]
- 1992: Book Stacks Unlimited in Cleveland opens a commercial sales website (www.books.com) selling books online with credit card processing.
- 1993: Paget Press releases edition No. 3[80] of the first[81] app store, The Electronic AppWrapper[82]
- 1994: Netscape releases the Navigator browser in October under the code name Mozilla. Netscape 1.0 is introduced in late 1994 with SSL encryption that made transactions secure.
- 1994: Ipswitch IMail Server becomes the first software available online for sale and immediate download via a partnership between Ipswitch, Inc. and OpenMarket.
- 1994: "Ten Summoner's Tales" by Sting becomes the first secure online purchase through NetMarket.[83]
- 1995: The US National Science Foundation lifts its former strict prohibition of commercial enterprise on the Internet.[84]
- 1995: Thursday 27 April 1995, the purchase of a book by Paul Stanfield, product manager for CompuServe UK, from W H Smith's shop within CompuServe's UK Shopping Centre is the UK's first national online shopping service secure transaction. The shopping service at launch featured W H Smith, Tesco, Virgin Megastores/Our Price, Great Universal Stores (GUS), Interflora, Dixons Retail, Past Times, PC World (retailer) and Innovations.
- 1995: Amazon is launched by Jeff Bezos.
- 1995: eBay is founded by computer programmer Pierre Omidyar as AuctionWeb. It is the first online auction site supporting person-to-person transactions.[85]
- 1995: The first commercial-free 24-hour, internet-only radio stations, Radio HK and NetRadio start broadcasting.
- 1996: The use of Excalibur BBS with replicated "storefronts" was an early implementation of electronic commerce started by a group of SysOps in Australia and replicated to global partner sites.
- 1998: Electronic postal stamps can be purchased and downloaded for printing from the Web.[86]
- 1999: Alibaba Group is established in China. Business.com sold for US$7.5 million to eCompanies, which was purchased in 1997 for US$149,000. The peer-to-peer filesharing software Napster launches. ATG Stores launches to sell decorative items for the home online.
- 1999: Global e-commerce reaches $150 billion[56]
- 2000: The dot-com bust.

- 2001: eBay has the largest userbase of any e-commerce site.[85]
- 2001: Alibaba.com achieved profitability in December 2001.
- 2002: eBay acquires PayPal for $1.5 billion.[87] Niche retail companies Wayfair and NetShops are founded with the concept of selling products through several targeted domains, rather than a central portal.
- 2003: Amazon posts first yearly profit.
- 2004: DHgate.com, China's first online B2B transaction platform, is established, forcing other B2B sites to move away from the "yellow pages" model.[88]
- 2007: Business.com acquired by R.H. Donnelley for $345 million.[89]
- 2014: US e-commerce and online retail sales projected to reach $294 billion, an increase of 12 percent over 2013 and 9% of all retail sales.[90] Alibaba Group has the largest Initial public offering ever, worth $25 billion.
- 2015: Amazon accounts for more than half of all e-commerce growth,[91] selling almost 500 Million SKU's in the US.
- 2016: The Government of India launches the BHIM UPI digital payment interface. In the year 2020 it has 2 billion digital payment transactions.[92][93]
- 2017: Retail e-commerce sales across the world reaches $2.304 trillion, which was a 24.8 percent increase than previous year.[94]
- 2017: Global e-commerce transactions generate $29.267 trillion, including $25.516 trillion for business-to-business (B2B) transactions and $3.851 trillion for business-to-consumer (B2C) sales.[95]

See also

- Comparison of free software e-commerce web application frameworks
- Comparison of shopping cart software
- Customer intelligence
- Digital economy
- E-commerce credit card payment system
- Electronic bill payment
- Electronic money
- Non-store retailing
- Online shopping
- Payments as a service
- South Dakota v. Wayfair, Inc.
- Types of e-commerce
- Timeline of e-commerce

References

1. "E-COMMERCE" (https://dictionary.cambridge.org/dictionary/english/e-commerce). *Cambridge Dictionary*.
2. "Retail e-commerce sales CAGR forecast in selected countries from 2016 to 2021" (https://web.archive.org/web/20171126014712/https://www.statista.com/statistics/220177/b2c-e-commerce-sales-cagr-forecast-for-selected-countries/). *Statista*. October 2016. Archived from the original (https://www.statista.com/statistics/220177/b2c-e-commerce-sales-cagr-forecast-for-selected-countries/) on 26 November 2017. Retrieved 4 May 2021.

3. Wienclaw, Ruth A. (2013). "B2B Business Models" (https://www.ebscohost.com/uploads/imported/thisTopic-dbTopic-1074.pdf) (PDF). *Research Starters: Business*. Archived (https://web.archive.org/web/20130718091809/http://www.ebscohost.com/uploads/imported/thisTopic-dbTopic-1074.pdf) (PDF) from the original on 18 July 2013. Retrieved 4 May 2021.
4. Subramani, Mani; Walden, Eric (June 2001). "The Impact of E-Commerce Announcements on the Market Value of Firms" (http://pubsonline.informs.org/doi/abs/10.1287/isre.12.2.135.9698). *Information Systems Research*. **12** (2): 135–154. doi:10.1287/isre.12.2.135.9698 (https://doi.org/10.1287%2Fisre.12.2.135.9698). ISSN 1047-7047 (https://search.worldcat.org/issn/1047-7047). Archived (https://web.archive.org/web/20220615223403/http://pubsonline.informs.org/doi/abs/10.1287/isre.12.2.135.9698) from the original on 15 June 2022. Retrieved 30 April 2022.
5. Bussey, Ed (6 March 2018). "How to prepare your products and brand for conversational commerce" (https://venturebeat.com/2018/03/06/how-to-prepare-your-products-and-brand-for-conversational-commerce/). *VentureBeat*. Archived (https://web.archive.org/web/20200929193753/https://venturebeat.com/2018/03/06/how-to-prepare-your-products-and-brand-for-conversational-commerce/) from the original on 29 September 2020. Retrieved 4 May 2021.
6. "The Ultimate Guide to eCommerce Marketing" (https://www.mayple.com/ecommerce-marketing-strategy). *Mayple*. Archived (https://web.archive.org/web/20210428042004/https://www.mayple.com/ecommerce-marketing-strategy) from the original on 28 April 2021. Retrieved 4 May 2021.
7. Simjanović, Dušan J.; Zdravković, Nemanja; Vesić, Nenad O. (March 2022). "On the Factors of Successful e-Commerce Platform Design during and after COVID-19 Pandemic Using Extended Fuzzy AHP Method" (https://doi.org/10.3390%2Faxioms11030105). *Axioms*. **11** (3): 105. doi:10.3390/axioms11030105 (https://doi.org/10.3390%2Faxioms11030105). ISSN 2075-1680 (https://search.worldcat.org/issn/2075-1680).
8. Khurana, Ajeet (25 November 2019). "Did You Know That There Are 4 Types of Ecommerce?" (https://www.thebalancesmb.com/ecommerce-businesses-understanding-types-1141595). *The Balance Small Business*. Dotdash. Archived (https://web.archive.org/web/20210122135823/https://www.thebalancesmb.com/ecommerce-businesses-understanding-types-1141595) from the original on 22 January 2021. Retrieved 4 May 2021.
9. Hacon, Tom (4 March 2013). "T-Commerce – What the tablet can do for brands and their consumers" (https://web.archive.org/web/20160607185627/https://www.governor.co.uk/news-plus-views/2013/march/t-commerce-what-the-tablet-can-do-for-brands-and-their-consumers/). *Governor Technology*. Archived from the original (http://www.governor.co.uk/news-plus-views/2013/march/t-commerce-what-the-tablet-can-do-for-brands-and-their-consumers/) on 7 June 2016. Retrieved 4 May 2021.
10. "Advertising and Marketing on the Internet: Rules of the Road" (https://www.ftc.gov/tips-advice/business-center/guidance/advertising-marketing-internet-rules-road). *Federal Trade Commission*. September 2000. Archived (https://web.archive.org/web/20200308111628/https://www.ftc.gov/tips-advice/business-center/guidance/advertising-marketing-internet-rules-road) from the original on 8 March 2020. Retrieved 4 May 2021.
11. "Privacy and Security" (https://www.ftc.gov/tips-advice/business-center/privacy-and-security). *Federal Trade Commission*. Archived (https://web.archive.org/web/20210504101605/https://www.ftc.gov/tips-advice/business-center/privacy-and-security) from the original on 4 May 2021. Retrieved 4 May 2021.
12. "H.R. 6353 (110th): Ryan Haight Online Pharmacy Consumer Protection Act of 2008" (https://www.govtrack.us/congress/bills/110/hr6353/summary). *GovTrack*. 2 October 2008. Archived (https://web.archive.org/web/20210319223858/https://www.govtrack.us/congress/bills/110/hr6353/summary) from the original on 19 March 2021. Retrieved 4 May 2021.

13. *UNCITRAL Model Law on Electronic Commerce* (https://uncitral.un.org/sites/uncitral.un.org/files/media-documents/uncitral/en/19-04970_ebook.pdf) (PDF). New York: United Nations Commission on International Trade Law. 1999. ISBN 92-1-133607-4. Archived (https://web.archive.org/web/20210225221820/https://uncitral.un.org/sites/uncitral.un.org/files/media-documents/uncitral/en/19-04970_ebook.pdf) (PDF) from the original on 25 February 2021. Retrieved 4 May 2021.
14. "Australian Competition and Consumer Commission" (https://www.accc.gov.au/). *Australian Competition & Consumer Commission*. Government of Australia. Archived (https://web.archive.org/web/20210503171522/https://www.accc.gov.au/) from the original on 3 May 2021. Retrieved 4 May 2021.
15. "Dealing with other businesses online" (https://web.archive.org/web/20130119002704/http://www.accc.gov.au/content/index.phtml/itemId/54056). *Australian Competition & Consumer Commission*. Government of Australia. Archived from the original (https://www.accc.gov.au/content/index.phtml/itemId/54056) on 19 January 2013. Retrieved 4 May 2021.
16. "What to do if thing go wrong in Australia" (https://web.archive.org/web/20130212103511/http://www.accc.gov.au/content/index.phtml/itemId/815341). *Australian Competition & Consumer Commission*. Australian Federal Government. Archived from the original (https://www.accc.gov.au/content/index.phtml/itemId/815341) on 12 February 2013. Retrieved 4 May 2021.
17. European Commission, Sector inquiry into e-commerce (https://competition-policy.ec.europa.eu/sectors/ict/sector-inquiry-e-commerce_en) Archived (https://web.archive.org/web/20230206060506/https://competition-policy.ec.europa.eu/sectors/ict/sector-inquiry-e-commerce_en) 6 February 2023 at the Wayback Machine, accessed 6 February 2023
18. "Financial Services Authority" (https://web.archive.org/web/20121228184712/http://www.fsa.gov.uk/portal/site/fsa). *Financial Services Authority*. Archived from the original (http://www.fsa.gov.uk/portal/site/fsa) on 28 December 2012. Retrieved 4 May 2021.
19. Parker, George; Masters, Brooke (16 June 2010). "Osborne abolishes FSA and boosts Bank" (https://www.ft.com/content/0203b99e-797f-11df-b063-00144feabdc0). *Financial Times*. Archived (https://web.archive.org/web/20210308132651/https://www.ft.com/content/0203b99e-797f-11df-b063-00144feabdc0) from the original on 8 March 2021. Retrieved 4 May 2021.
20. "The Payment Services Regulations 2009" (https://www.legislation.gov.uk/uksi/2009/209/contents/made). *legislation.gov.uk*. 9 February 2009. Archived (https://web.archive.org/web/20210312123339/https://www.legislation.gov.uk/uksi/2009/209/contents/made) from the original on 12 March 2021. Retrieved 4 May 2021.
21. "Telecommunications Regulations of the People's Republic of China" (http://www.china.org.cn/business/laws_regulations/2010-01/20/content_19273945.htm). *China Internet Information Center*. 25 September 2000. Archived (https://web.archive.org/web/20150405012152/http://www.china.org.cn/business/laws_regulations/2010-01/20/content_19273945.htm) from the original on 5 April 2015. Retrieved 4 May 2021.
22. "Administrative Measures on Internet Information Services" (http://www.china.org.cn/business/2010-01/20/content_19274704.htm). *China Internet Information Center*. 20 September 2000. Archived (https://web.archive.org/web/20150716184027/http://www.china.org.cn/business/2010-01/20/content_19274704.htm) from the original on 16 July 2015. Retrieved 4 May 2021.
23. Swan, Erin (30 October 2015). "The PRC Electronic Signature Law" (https://www.efilecabinet.com/the-prc-electronic-signature-law/). *eFileCabinet*. Archived (https://web.archive.org/web/20171107024110/https://www.efilecabinet.com/the-prc-electronic-signature-law/) from the original on 7 November 2017. Retrieved 4 May 2021.
24. Eisingerich, Andreas B.; Kretschmer, Tobias (March 2008). "In E-Commerce, More is More" (https://hbr.org/2008/03/in-e-commerce-more-is-more). *Harvard Business Review*. **86**: 20–21. Archived (https://web.archive.org/web/20201203204602/https://hbr.org/2008/03/in-e-commerce-more-is-more) from the original on 3 December 2020. Retrieved 4 May 2021.

25. Burgess, Stephen; Sellitto, Carmine; Karanasios, Stan (28 February 2009). *Effective Web Presence Solutions for Small Businesses: Strategies for Successful Implementation: Strategies for Successful Implementation* (https://books.google.com/books?id=n9KF8HtigrAC). IGI Global. ISBN 9781605662251. Archived (https://web.archive.org/web/20210504223600/https://books.google.com/books/?id=n9KF8HtigrAC) from the original on 4 May 2021. Retrieved 4 May 2021.
26. Chen, Si-Hua; Xiao, Hua; Huang, Wen-de; He, Wei (2 January 2022). "Cooperation of Cross-border E-commerce: A reputation and trust perspective" (https://doi.org/10.1080/1097198X.2021.2022396). *Journal of Global Information Technology Management*. **25** (1): 7–25. doi:10.1080/1097198X.2021.2022396 (https://doi.org/10.1080%2F1097198X.2021.2022396). ISSN 1097-198X (https://search.worldcat.org/issn/1097-198X). S2CID 246867732 (https://api.semanticscholar.org/CorpusID:246867732).
27. "Ecommerce Sales Topped $1 Trillion for First Time in 2012" (https://www.emarketer.com/Article/Ecommerce-Sales-Topped-1-Trillion-First-Time-2012/1009649). *eMarketer*. 5 February 2013. Archived (https://web.archive.org/web/20210330180629/https://www.emarketer.com/Article/Ecommerce-Sales-Topped-1-Trillion-First-Time-2012/1009649) from the original on 30 March 2021. Retrieved 4 May 2021.
28. Enright, Allison (25 April 2013). "U.S. e-commerce sales could top $434 billion in 2017" (https://www.digitalcommerce360.com/2013/04/25/us-e-commerce-sales-could-top-434-billion-2017/). *Digital Commerce 360*. Archived (https://web.archive.org/web/20201202043908/https://www.digitalcommerce360.com/2013/04/25/us-e-commerce-sales-could-top-434-billion-2017/) from the original on 2 December 2020. Retrieved 4 May 2021.
29. DeLone, William H.; McLean, Ephraim R. (8 December 2014). "Measuring e-Commerce Success: Applying the DeLone & McLean Information Systems Success Model" (https://www.tandfonline.com/doi/abs/10.1080/10864415.2004.11044317). *International Journal of Electronic Commerce*. **9** (1): 31–47. doi:10.1080/10864415.2004.11044317 (https://doi.org/10.1080%2F10864415.2004.11044317). S2CID 205751936 (https://api.semanticscholar.org/CorpusID:205751936). Archived (https://web.archive.org/web/20210323032755/https://www.tandfonline.com/doi/abs/10.1080/10864415.2004.11044317) from the original on 23 March 2021. Retrieved 4 May 2021 – via Taylor & Francis.
30. Bakos, Yannis (2001). "The Emerging Landscape for Retail E-Commerce" (https://www.aeaweb.org/articles?id=10.1257/jep.15.1.69). *Journal of Economic Perspectives*. **15** (1): 69–80. CiteSeerX 10.1.1.4.9128 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.4.9128). doi:10.1257/jep.15.1.69 (https://doi.org/10.1257%2Fjep.15.1.69). Archived (https://web.archive.org/web/20210118041827/https://www.aeaweb.org/articles?id=10.1257%2Fjep.15.1.69) from the original on 18 January 2021. Retrieved 4 May 2021.
31. Constine, Josh (20 February 2018). "Facebook's plan to unite AR, VR and News Feed with 3D posts" (https://techcrunch.com/2018/02/20/facebook-3d-posts/). *TechCrunch*. Archived (https://web.archive.org/web/20180504090211/https://techcrunch.com/2018/02/20/facebook-3d-posts/) from the original on 4 May 2018. Retrieved 4 May 2021.
32. Millward, Steven (18 August 2015). "China is making a huge shift to mobile" (https://www.techinasia.com/china-internet-social-media-ecommerce-stats-2015). *Tech in Asia* (Infographic). Archived (https://web.archive.org/web/20160306203048/https://www.techinasia.com/china-internet-social-media-ecommerce-stats-2015) from the original on 6 March 2016. Retrieved 4 May 2021.
33. Olsen, Robert (18 January 2010). "China's Migration To E-Commerce" (https://www.forbes.com/2010/01/18/china-internet-commerce-markets-equities-alibaba.html/). *Forbes*. Archived (https://web.archive.org/web/20170806145344/https://www.forbes.com/2010/01/18/china-internet-commerce-markets-equities-alibaba.html) from the original on 6 August 2017. Retrieved 4 May 2021.

34. Tong, Frank (16 September 2013). "China's cross-border e-commerce tops $375 billion in 2012" (https://www.digitalcommerce360.com/2013/09/16/chinas-cross-border-e-commerce-tops-375-billion-2012/). *Digital Commerce 360*. Vertical Web Media LLC. Archived (https://web.archive.org/web/20171018171240/https://www.digitalcommerce360.com/2013/09/16/chinas-cross-border-e-commerce-tops-375-billion-2012/) from the original on 18 October 2017. Retrieved 4 May 2021.
35. Millward, Steven (17 September 2014). "Here are all the must-see numbers on Alibaba ahead of record-breaking IPO" (http://www.techinasia.com/alibaba-numbers-ipo-breaks-records-2014/). *Tech in Asia*. Archived (https://web.archive.org/web/20140920190816/http://www.techinasia.com/alibaba-numbers-ipo-breaks-records-2014/) from the original on 20 September 2014. Retrieved 4 May 2021.
36. PYMNTS.com (22 October 2014). "China B2B Passes 4.5B Yuan" (https://www.pymnts.com/news/2014/china-b2b-passes-4-5b-yuan/). *PYMNTS.com*. Archived (https://web.archive.org/web/20230308085620/https://www.pymnts.com/news/2014/china-b2b-passes-4-5b-yuan/) from the original on 8 March 2023. Retrieved 8 March 2023.
37. Millward, Steven (18 August 2016). "Asia's ecommerce spending to hit record $1 trillion this year – but most of that is China" (https://www.techinasia.com/asia-ecommerce-spending-1-trillion-dollars-2016). *Tech in Asia*. Archived (https://web.archive.org/web/20160819043228/https://www.techinasia.com/asia-ecommerce-spending-1-trillion-dollars-2016) from the original on 19 August 2016. Retrieved 4 May 2021.
38. Hu, Richard (2023). *Reinventing the Chinese City*. New York: Columbia University Press. ISBN 978-0-231-21101-7.
39. Parker, Christopher J.; Wenyu, Lu (13 May 2019). "What influences Chinese fashion retail? Shopping motivations, demographics and spending" (https://www.emerald.com/insight/content/doi/10.1108/JFMM-09-2017-0093/full/html). *Journal of Fashion Marketing and Management*. **23** (2): 158–175. doi:10.1108/jfmm-09-2017-0093 (https://doi.org/10.1108%2Fjfmm-09-2017-0093). ISSN 1361-2026 (https://search.worldcat.org/issn/1361-2026). S2CID 170031856 (https://api.semanticscholar.org/CorpusID:170031856). Archived (https://web.archive.org/web/20210308110006/https://www.emerald.com/insight/content/doi/10.1108/JFMM-09-2017-0093/full/html) from the original on 8 March 2021. Retrieved 16 April 2021.
40. Yang, Saidi; Wang, Puqing; Zhou, Deyi (2021). "Transformation and Development of Taobao Village in China Based on "Zhijiang Mode" ". *Proceedings of the 2020 3rd International Seminar on Education Research and Social Science (ISERSS 2020)*. Paris, France: Atlantis Press. doi:10.2991/assehr.k.210120.053 (https://doi.org/10.2991%2Fassehr.k.210120.053). ISBN 978-94-6239-316-5.
41. Zhang, Angela Huyue (2024). *High Wire: How China Regulates Big Tech and Governs Its Economy*. Oxford University Press. ISBN 9780197682258.
42. Šimalčík, Matej (2023). "Rule by Law". In Kironska, Kristina; Turscanyi, Richard Q. (eds.). *Contemporary China: a New Superpower?*. Routledge. ISBN 978-1-03-239508-1.
43. Robinson, James (28 October 2010). "UK's internet industry worth £100bn" (https://www.theguardian.com/technology/2010/oct/28/net-worth-100bn-uk). *The Guardian* (report). London. Archived (https://web.archive.org/web/20180219090350/https://www.theguardian.com/technology/2010/oct/28/net-worth-100bn-uk) from the original on 19 February 2018. Retrieved 4 May 2021.
44. "Ecommerce contribution in Europe" (https://ecommercenews.eu/infographic-ecommerce-contribution-in-europe/). *Ecommerce News* (infographic). 18 June 2013. Archived (https://web.archive.org/web/20210125131628/https://ecommercenews.eu/infographic-ecommerce-contribution-in-europe/) from the original on 25 January 2021. Retrieved 4 May 2021.
45. "Ecommerce in the Middle East – What are the demographics?" (https://www.embitel.com/blog/ecommerce-blog/ecommerce-in-the-middle-east-what-are-the-demographics). *Embitel*. 17 June 2016. Archived (https://web.archive.org/web/20170827010001/https://www.embitel.com/blog/ecommerce-blog/ecommerce-in-the-middle-east-what-are-the-demographics) from the original on 27 August 2017. Retrieved 4 May 2021.

46. Keelery, Sandhya (7 July 2020). "Internet usage in India – statistics & facts" (https://www.statista.com/topics/2157/internet-usage-in-india/). *Statista*. Archived (https://web.archive.org/web/20171230114442/https://www.statista.com/topics/2157/internet-usage-in-india/) from the original on 30 December 2017. Retrieved 4 May 2021.
47. Pasumarthy, Phani Bhaskar (December 2016). "AFFECT OF DEMONETIZATION ON E-COMMERCE" (https://www.researchgate.net/publication/311518414). *ResearchGate*. Archived (https://web.archive.org/web/20201026152116/https://www.researchgate.net/publication/311518414_AFFECT_OF_DEMONETIZATION_ON_E-COMMERCE) from the original on 26 October 2020. Retrieved 4 May 2021.
48. "Fulfilled!:India's e-commerce retail logistics growth story" (https://assets.kpmg/content/dam/kpmg/in/pdf/2016/08/E-commerce-retail-logistics-India.pdf) (PDF). *KPMG*. August 2016. Archived (https://web.archive.org/web/20200508153115/https://assets.kpmg/content/dam/kpmg/in/pdf/2016/08/E-commerce-retail-logistics-India.pdf) (PDF) from the original on 8 May 2020. Retrieved 4 May 2021.
49. "More Buyers Join Brazil's Robust Ecommerce Market" (https://www.emarketer.com/Article/More-Buyers-Join-Brazils-Robust-Ecommerce-Market/1009725). *eMarketer*. 13 March 2013. Archived (https://web.archive.org/web/20200804174125/https://www.emarketer.com/Article/More-Buyers-Join-Brazils-Robust-Ecommerce-Market/1009725) from the original on 4 August 2020. Retrieved 4 May 2021.
50. Kawa, Arkadiusz (2017). "Fulfillment Service in E-Commerce Logistics" (https://www.logforum.net/pdf/13_4_4_17.pdf) (PDF). *LogForum*. **13** (4): 429–438. doi:10.17270/J.LOG.2017.4.4 (https://doi.org/10.17270%2FJ.LOG.2017.4.4). eISSN 1734-459X (https://search.worldcat.org/issn/1734-459X). ISSN 1895-2038 (https://search.worldcat.org/issn/1895-2038). Archived (https://web.archive.org/web/20210504223600/https://www.logforum.net/pdf/13_4_4_17.pdf) (PDF) from the original on 4 May 2021. Retrieved 4 May 2021.
51. Gomes, Alysson Cáceres; Junior, Francisco Bezerra de Lima; Soliani, Rodrigo Duarte; Oliveira, Pollyana Rufino de Souza; Oliveira, Dion Alves de; Siqueira, Reinaldo Maia; Nora, Leonardo Augusto Rodrigues da Silva; Macêdo, Jailson Juracy Souza de (10 May 2023). "Logistics management in e-commerce: challenges and opportunities" (https://ojs.revistagesec.org.br/secretariado/article/view/2119). *Revista de Gestão e Secretariado (Management and Administrative Professional Review)*. **14** (5): 7252–7272. doi:10.7769/gesec.v14i5.2119 (https://doi.org/10.7769%2Fgesec.v14i5.2119). ISSN 2178-9010 (https://search.worldcat.org/issn/2178-9010).
52. Sabanoglu, Tugba (26 March 2021). "Retail e-commerce sales worldwide from 2014 to 2024" (https://www.statista.com/statistics/379046/worldwide-retail-e-commerce-sales/). *Statista*. Archived (https://web.archive.org/web/20181122172313/https://www.statista.com/statistics/379046/worldwide-retail-e-commerce-sales/) from the original on 22 November 2018. Retrieved 4 May 2021.
53. "Electronic money and electronic commerce" (https://www.bbc.co.uk/bitesize/guides/zkrgscw/revision/3). BBC News. Archived (https://web.archive.org/web/20210504223617/https://www.bbc.co.uk/bitesize/guides/zkrgscw/revision/3) from the original on 4 May 2021. Retrieved 4 May 2021.
54. Dimoka, Angelika; Hong, Yili; Pavlou, Paul A. (June 2012). "On Product Uncertainty in Online Markets: Theory and Evidence" (https://web.archive.org/web/20180106174640/http://yilihong.github.io/paper/Product%20Uncertainty%20in%20Online%20Markets_Dimoka_Hong_Pavlou2012.pdf) (PDF). *Management Information Systems Quarterly*. **36** (2): 395–426. doi:10.2307/41703461 (https://doi.org/10.2307%2F41703461). JSTOR 41703461 (https://www.jstor.org/stable/41703461). S2CID 8963257 (https://api.semanticscholar.org/CorpusID:8963257). Archived from the original (https://yilihong.github.io/paper/Product%20Uncertainty%20in%20Online%20Markets_Dimoka_Hong_Pavlou2012.pdf) (PDF) on 6 January 2018. Retrieved 4 May 2021.

55. Marincas, Delia Adriana (2008). "Information system for the supply chain management" (https://ideas.repec.org/a/aes/amfeco/v10y2008i24p236-253.html). *The AMFITEATRU ECONOMIC Journal*. **10** (24): 236–253. Archived (https://web.archive.org/web/20150518065928/https://ideas.repec.org/a/aes/amfeco/v10y2008i24p236-253.html) from the original on 18 May 2015. Retrieved 8 May 2015.
56. Terzi, Nuray (2011). "The impact of e-commerce on international trade and employment". *Procedia - Social and Behavioral Sciences*. **24**: 745–753. doi:10.1016/j.sbspro.2011.09.010 (https://doi.org/10.1016%2Fj.sbspro.2011.09.010). hdl:11424/223579 (https://hdl.handle.net/11424%2F223579).
57. "Consumers trump marketers in battle for purchasing influence" (https://www.hometextilestoday.com/brands-amp-marketing/consumers-trump-marketers-in-battle-for-purchasing-influence/). *Home Textiles Today*. 13 May 2022. Archived (https://web.archive.org/web/20221107222125/https://www.hometextilestoday.com/brands-amp-marketing/consumers-trump-marketers-in-battle-for-purchasing-influence/) from the original on 7 November 2022. Retrieved 7 November 2022.
58. "Ecommerce Rating and Review Tools Market – A Comprehensive Study by Key Players" (https://thedailyvale.com/2022/05/22/ecommerce-rating-and-review-tools-market-a-comprehensive-study-by-key-playerstrustpilot-feefo-kiyoh-bazaarvoice-ekomi-trustspot-reevoo-reziew-yelp-reviews-co-uk-yotpo-powerreview/). *The Daily Vale*. 22 May 2022. Archived (https://web.archive.org/web/20221103214341/https://thedailyvale.com/2022/05/22/ecommerce-rating-and-review-tools-market-a-comprehensive-study-by-key-playerstrustpilot-feefo-kiyoh-bazaarvoice-ekomi-trustspot-reevoo-reziew-yelp-reviews-co-uk-yotpo-powerreview/) from the original on 3 November 2022. Retrieved 7 November 2022.
59. Evans, Richard (1 May 2002). "E-commerce, Competitiveness and Local and Regional Governance in Greater Manchester and Merseyside: A Preliminary Assessment". *Urban Studies*. **39** (5–6). SAGE Publishing: 947–975. Bibcode:2002UrbSt..39..947E (https://ui.adsabs.harvard.edu/abs/2002UrbSt..39..947E). doi:10.1080/00420980220128390 (https://doi.org/10.1080%2F00420980220128390). JSTOR 43084757 (https://www.jstor.org/stable/43084757). S2CID 154155858 (https://api.semanticscholar.org/CorpusID:154155858).
60. Januszkiewicz, Monika (October 2017). "Online Virtual Fit Is Not Yet Fit For Purpose: An Analysis of Fashion e-Commerce Interfaces" (https://www.3dbody.tech/cap/papers/2017/17210januszkiewicz.pdf) (PDF). *Proceedings of 3DBODY.TECH 2017*: 210–217. doi:10.15221/17.210 (https://doi.org/10.15221%2F17.210). ISBN 9783033064362. Archived (https://web.archive.org/web/20180722014228/http://www.3dbody.tech/cap/papers/2017/17210januszkiewicz.pdf) (PDF) from the original on 22 July 2018. Retrieved 4 May 2021.
61. DePillis, Lynda (16 July 2019). "Amazon's incredible, vanishing cardboard box" (https://edition.cnn.com/2019/07/16/business/amazon-cardboard-box-prime-day/). *CNN Business*. Archived (https://web.archive.org/web/20190716130808/https://www.cnn.com/2019/07/16/business/amazon-cardboard-box-prime-day/index.html) from the original on 16 July 2019. Retrieved 4 May 2021.
62. Newman, Jesse; Bunge, Jacob (2020). "U.S. Postal Service Is Urged to Stop Delivering Mysterious Seeds" (https://www.wsj.com/articles/u-s-postal-service-is-urged-to-stop-delivering-mysterious-seeds-11596555249). *The Wall Street Journal*. Archived (https://web.archive.org/web/20220920192435/https://www.wsj.com/articles/u-s-postal-service-is-urged-to-stop-delivering-mysterious-seeds-11596555249) from the original on 20 September 2022. Retrieved 17 September 2022.
63. Barrabi, Thomas; Carter, Shawn M. (14 July 2017). "Retail Apocalypse: Pier 1 and the other retailers closing, filing for bankruptcy" (https://www.foxbusiness.com/retail/features-retail-apocalypse-bankruptcy-stores-closing). *Fox Business*. Archived (https://web.archive.org/web/20190812203525/https://www.foxbusiness.com/retail/features-retail-apocalypse-bankruptcy-stores-closing) from the original on 12 August 2019. Retrieved 4 May 2021.

64. Forte, Daniela (11 March 2019). "Store Closures, Failures Continue to Mount as Retailers Seek to Pivot Faster" (https://multichannelmerchant.com/ecommerce/store-closures-failures-continue-to-mount-as-retailers-seek-to-pivot-faster/). *Multichannel Merchant*. Access Intelligence, LLC. Archived (https://web.archive.org/web/20190813025717/https://multichannelmerchant.com/ecommerce/store-closures-failures-continue-to-mount-as-retailers-seek-to-pivot-faster/) from the original on 13 August 2019. Retrieved 4 May 2021.
65. "The retail apocalypse is shutting down flagship stores" (https://www.usatoday.com/story/money/2019/08/01/retail-flagship-stores-close-rent-online-shopping-skyrockets/1888004001/). *USA Today*. 1 August 2019. Archived (https://web.archive.org/web/20190813033959/https://www.usatoday.com/story/money/2019/08/01/retail-flagship-stores-close-rent-online-shopping-skyrockets/1888004001/) from the original on 13 August 2019. Retrieved 12 August 2019.
66. Clement, J. (12 February 2021). "Most popular online retail websites worldwide in 2020, by average monthly traffic" (https://www.statista.com/statistics/274708/online-retail-and-auction-ranked-by-worldwide-audiences/). *Statista*. Archived (https://web.archive.org/web/20200521233438/https://www.statista.com/statistics/274708/online-retail-and-auction-ranked-by-worldwide-audiences/) from the original on 21 May 2020. Retrieved 4 May 2021.
67. Song, Zhouying (January 2022). "The geography of online shopping in China and its key drivers" (http://journals.sagepub.com/doi/10.1177/23998083211002189). *Environment and Planning B: Urban Analytics and City Science*. **49** (1): 259–274. doi:10.1177/23998083211002189 (https://doi.org/10.1177%2F23998083211002189). ISSN 2399-8083 (https://search.worldcat.org/issn/2399-8083). S2CID 233623855 (https://api.semanticscholar.org/CorpusID:233623855). Archived (https://web.archive.org/web/20230210000327/https://journals.sagepub.com/doi/10.1177/23998083211002189) from the original on 10 February 2023. Retrieved 15 April 2022.
68. Kuhuk, Jane (19 May 2020). "COVID-19 shopping behavior: what products would customers rather buy online?" (https://competera.net/resources/articles/ecommerce-online-shopping-behavior-retail-infographic). *Competera* (Infographic). Archived (https://web.archive.org/web/20200521233436/https://competera.net/resources/articles/ecommerce-online-shopping-behavior-retail-infographic) from the original on 21 May 2020. Retrieved 4 May 2021.
69. Anam, Bhatti; Akram, Hamza; Basit, Hafiz Muhammad; Khan, Ahmed Usman; Naqvi, Syeda Mahwish Raza; Bilal, Muhammad (2020). "E-commerce trends during COVID-19 Pandemic" (https://lovacky.eu/ws/media-library/8291b8bb61d0458d9bec753432dc4842/qualitativepaper.pdf) (PDF). *International Journal of Future Generation Communication and Networking*. **13** (2): 1449–1452. ISSN 2233-7857 (https://search.worldcat.org/issn/2233-7857). Archived (https://web.archive.org/web/20201230160442/https://www.lovacky.eu/ws/media-library/8291b8bb61d0458d9bec753432dc4842/qualitativepaper.pdf) (PDF) from the original on 30 December 2020. Retrieved 4 May 2021.
70. Power, Michael 'Mike' (19 April 2013). "Online highs are old as the net: the first e-commerce was a drugs deal" (https://www.theguardian.com/science/2013/apr/19/online-high-net-drugs-deal). *The Guardian*. Archived (https://web.archive.org/web/20161130044158/https://www.theguardian.com/science/2013/apr/19/online-high-net-drugs-deal) from the original on 30 November 2016. Retrieved 4 May 2021.
71. Tkacz, Ewaryst; Kapczynski, Adrian (2009). *Internet – Technical Development and Applications* (https://books.google.com/books?id=a9_NJIBC87gC). Springer Science+Business Media. p. 255. ISBN 9783642050190. Archived (https://web.archive.org/web/20210504223615/https://books.google.com/books?id=a9_NJIBC87gC) from the original on 4 May 2021. Retrieved 4 May 2021. "The first pilot system was installing in Tesco in the UK (first demonstrated in 1979 by Michael Aldrich)."

72. Palmer, Colin (December 1988). "Using IT for competitive advantage at Thomson holidays" (https://www.sciencedirect.com/science/article/abs/pii/0024630188901550). *Long Range Planning*. **21** (6): 26–29. doi:10.1016/0024-6301(88)90155-0 (https://doi.org/10.1016%2F0024-6301%2888%2990155-0). Archived (https://web.archive.org/web/20210504223609/https://www.sciencedirect.com/science/article/abs/pii/0024630188901550) from the original on 4 May 2021. Retrieved 4 May 2021 – via Elsevier Science Direct.
73. "E Commerce" (https://www.studymode.com/essays/e-Commerce-1554293.html). *StudyMode*. 31 March 2013. Archived (https://web.archive.org/web/20200805215334/https://www.studymode.com/essays/e-Commerce-1554293.html) from the original on 5 August 2020. Retrieved 4 May 2021.
74. "Piano Entrepreneurs". *Newspaper*. Contra Costa Newspapers. Contra Costa Times. 11 February 1985.
75. "In Tune With The Times - Piano Business Thrives in Slump". *Newspaper*. Contra Costa Newspapers. The Daily Review. 26 June 1986.
76. "Online shopping: The pensioner who pioneered a home shopping revolution" (https://www.bbc.co.uk/news/magazine-24091393). BBC News. 16 September 2013. Archived (https://web.archive.org/web/20180717063557/https://www.bbc.co.uk/news/magazine-24091393) from the original on 17 July 2018. Retrieved 21 June 2018.
77. Aldrich, Michael (March 2009). "Finding Mrs Snowball" (https://web.archive.org/web/20201207045137/http://www.aldricharchive.com/snowball.html). *Michael Aldrich Archive*. Archived from the original (http://www.aldricharchive.com/snowball.html) on 7 December 2020. Retrieved 4 May 2021.
78. "The Electronic Mall" (https://web.archive.org/web/20160326123900/http://gsbrown.org/compuserve/electronic-mall-1984-04/). GS Brown. 30 April 2010. Archived from the original on 26 March 2016. Retrieved 4 May 2021.
79. Berners-Lee, Tim. "The WorldWideWeb browser" (https://www.w3.org/People/Berners-Lee/WorldWideWeb.html). *World Wide Web Consortium*. Archived (https://web.archive.org/web/20210501222006/https://www.w3.org/People/Berners-Lee/WorldWideWeb.html) from the original on 1 May 2021. Retrieved 4 May 2021.
80. Geiger, Conrad (15 September 1992). "NeXT Nugget News Digest" (https://web.archive.org/web/20160310211403/http://next.68k.org/ftp.peak.org/next/documents/newsletters/NuggetNewsText/vol.04.11.txt). *NeXT*. Archived from the original (http://next.68k.org/ftp.peak.org/next/documents/newsletters/NuggetNewsText/vol.04.11.txt) on 10 March 2016. Retrieved 4 May 2021.
81. Tayler, Jesse (11 April 2016). "Jesse Tayler talks App Store and NeXTSTEP with AppStorey" (https://appstorey.com/2016/04/11/jesse-tayler-talks-appstore-and-nextstep-with-appstorey/). *AppStorey*. Archived (https://web.archive.org/web/20180321063209/https://appstorey.com/2016/04/11/jesse-tayler-talks-appstore-and-nextstep-with-appstorey/) from the original on 21 March 2018. Retrieved 4 May 2021.
82. "PRESS RELEASE: AppWrapper Volume1 Issue 3 Ships" (https://groups.google.com/g/comp.sys.next.announce/c/0mE2n2zaeVE/m/znflGBPPqlYJ). *Google Groups* (Press Release). 28 September 1993. Archived (https://web.archive.org/web/20210504223607/https://groups.google.com/g/comp.sys.next.announce/c/0mE2n2zaeVE/m/znflGBPPqlYJ) from the original on 4 May 2021. Retrieved 4 May 2021.
83. Lewis, Peter H. (12 August 1994). "Attention Shoppers: Internet Is Open" (https://www.nytimes.com/1994/08/12/business/attention-shoppers-internet-is-open.html). *The New York Times*. Archived (https://web.archive.org/web/20170903095512/http://www.nytimes.com/1994/08/12/business/attention-shoppers-internet-is-open.html) from the original on 3 September 2017. Retrieved 4 May 2021.

84. Kelly, Kevin (August 2005). "We Are the Web" (https://web.archive.org/web/20130824064357/http://www.wired.com/wired/archive/13.08/tech.html?pg=2). *Wired*. Archived from the original (https://www.wired.com/wired/archive/13.08/tech.html?pg=2) on 24 August 2013. Retrieved 4 May 2021.
85. Bunnell, David (16 May 2001). "The eBay Business Model" (https://books.google.com/books?id=pnict8L-_iYC&pg=PA71). *The ebay Phenomenon: Business Secrets Behind the World's Hottest Internet Company* (https://books.google.com/books?id=pnict8L-_iYC). John Wiley & Sons. pp. 71–81. ISBN 9780471436799. Archived (https://web.archive.org/web/20210504223601/https://books.google.com/books?id=pnict8L-_iYC) from the original on 4 May 2021. Retrieved 5 September 2019.
86. "First Electronic Stamps Being Put to Test" (https://news.google.com/newspapers?id=6EVWAAAAIBAJ&pg=5537%2C1474649). *Sunday Business*. 6 April 1998. p. 16. Archived (https://web.archive.org/web/20210504223612/https://news.google.com/newspapers?id=6EVWAAAAIBAJ&pg=5537%2C1474649) from the original on 4 May 2021. Retrieved 4 May 2021.
87. "eBay acquires PayPal" (https://web.archive.org/web/20141006214543/http://investor.ebay.com/releasedetail.cfm?releaseid=84142). *eBay*. 8 July 2002. Archived from the original (http://investor.ebay.com/releasedetail.cfm?ReleaseID=84142) on 6 October 2014. Retrieved 4 May 2021.
88. "Diane Wang: Rounding up the "Ant" Heroes" (https://web.archive.org/web/20120223015245/http://www.zwgl.com.cn/article_info.asp?nid=4243). Sino Foreign Management. Archived from the original (http://www.zwgl.com.cn/article_info.asp?nid=4243) on 23 February 2012. Retrieved 3 September 2011.
89. Allemann, Andrew (26 July 2007). "R.H. Donnelley Acquires Business.com for $345M" (https://domainnamewire.com/2007/07/26/rh-donnelley-acquires-businesscom-for-345m/). *Domain Name Wire*. Brainstorm Labs, LLC. Archived (https://web.archive.org/web/20210408054344/https://domainnamewire.com/2007/07/26/rh-donnelley-acquires-businesscom-for-345m/) from the original on 8 April 2021. Retrieved 4 May 2021.
90. Kodali, Sucharita (12 May 2014). "US eCommerce Forecast: 2013 to 2018" (https://www.forrester.com/report/US+eCommerce+Forecast+2013+To+2018/-/E-RES115513). *Forrester Research*. Archived (https://web.archive.org/web/20210123105902/https://www.forrester.com/report/US+eCommerce+Forecast+2013+To+2018/-/E-RES115513) from the original on 23 January 2021. Retrieved 4 May 2021.
91. Garcia, Tonya (22 December 2015). "Amazon will account for more than half of 2015 e-commerce growth, says Macquarie" (https://www.marketwatch.com/story/amazon-will-account-for-more-than-half-of-2015-e-commerce-growth-says-macquarie-2015-12-22). *MarketWatch*. Archived (https://web.archive.org/web/20210128162404/https://www.marketwatch.com/story/amazon-will-account-for-more-than-half-of-2015-e-commerce-growth-says-macquarie-2015-12-22) from the original on 28 January 2021. Retrieved 4 May 2021.
92. "UPI crosses 2 billion transactions milestone in October, up 80% from year-ago; value nears Rs 4 lakh cr" (https://www.financialexpress.com/industry/banking-finance/upi-crosses-2-billion-transactions-milestone-in-october-up-80-from-year-ago-value-nears-rs-4-lakh-cr/2118690/). *Financialexpress*. November 2020. Archived (https://web.archive.org/web/20220926011703/https://www.financialexpress.com/industry/banking-finance/upi-crosses-2-billion-transactions-milestone-in-october-up-80-from-year-ago-value-nears-rs-4-lakh-cr/2118690/) from the original on 26 September 2022. Retrieved 24 March 2022.
93. "India: number of BHIM transactions 2022" (https://www.statista.com/statistics/1055414/india-number-of-transactions-through-bhim/). *Statista*. Archived (https://web.archive.org/web/20220324044607/https://www.statista.com/statistics/1055414/india-number-of-transactions-through-bhim/) from the original on 24 March 2022. Retrieved 24 March 2022.

94. Mcnair, Corey (29 January 2018). "Worldwide Retail and Ecommerce Sales: eMarketer's Updated Forecast and New Mcommerce Estimates for 2016—2021" (https://www.emarketer.com/Report/Worldwide-Retail-Ecommerce-Sales-eMarketers-Updated-Forecast-New-Mcommerce-Estimates-20162021/2002182). *eMarketer*. Insider Intelligence Inc. Archived (https://web.archive.org/web/20181127152046/https://www.emarketer.com/Report/Worldwide-Retail-Ecommerce-Sales-eMarketers-Updated-Forecast-New-Mcommerce-Estimates-20162021/2002182) from the original on 27 November 2018. Retrieved 4 May 2021.
95. "Global e-Commerce sales surged to $29 trillion" (https://unctad.org/press-material/global-e-commerce-sales-surged-29-trillion) (Press Release). United Nations Conference on Trade and Development. 29 March 2019. Archived (https://web.archive.org/web/20210502055911/https://unctad.org/press-material/global-e-commerce-sales-surged-29-trillion) from the original on 2 May 2021. Retrieved 4 May 2021.

Further reading

- Laudon, Kenneth C.; Traver, Carol Guercio (2014). *E-commerce: Business, Technology, Society* (https://books.google.com/books?id=-cgDnwEACAAJ) (10th ed.). Pearson plc. ISBN 9781292009094. Archived (https://web.archive.org/web/20210505004149/https://books.google.com/books?id=-cgDnwEACAAJ) from the original on 5 May 2021. Retrieved 5 May 2021.
- Chaudhury, Abijit; Kuilboer, Jean-Pierre (2002). *E-business and E-commerce Infrastructure: Technologies Supporting the E-business Initiative* (https://books.google.com/books?id=zHoeAQAAIAAJ). McGraw Hill Education. ISBN 9780071123136. Archived (https://web.archive.org/web/20210505004143/https://books.google.com/books?id=zHoeAQAAIAAJ) from the original on 5 May 2021. Retrieved 5 May 2021.
- Frieden, Jonathan D.; Roche, Sean Patrick (2006). "E-Commerce: Legal Issues of the Online Retailer in Virginia" (http://jolt.richmond.edu/jolt-archive/v13i2/article5.pdf) (PDF). *Richmond Journal of Law and Technology*. **13** (2). Archived (https://web.archive.org/web/20200922172841/http://jolt.richmond.edu/jolt-archive/v13i2/article5.pdf) (PDF) from the original on 22 September 2020. Retrieved 5 May 2021.
- Graham, Mark (2008). "Warped Geographies of Development: The Internet and Theories of Economic Development" (https://web.archive.org/web/20161126085903/http://geospace.co.uk/files/compass.pdf) (PDF). *Geography Compass*. **2** (3). Blackwell publishing: 771–789. Bibcode:2008GComp...2..771G (https://ui.adsabs.harvard.edu/abs/2008GComp...2..771G). doi:10.1111/j.1749-8198.2008.00093.x (https://doi.org/10.1111%2Fj.1749-8198.2008.00093.x). S2CID 16190907 (https://api.semanticscholar.org/CorpusID:16190907). Archived from the original (http://geospace.co.uk/files/compass.pdf) (PDF) on 26 November 2016. Retrieved 5 May 2021 – via Wiley Online Library.
- Humeau, Philippe; Jung, Matthieu (21 June 2013). *In depth benchmark of 12 ecommerce solutions* (http://docshare01.docshare.tips/files/18925/189257997.pdf) (PDF). Archived (https://web.archive.org/web/20210505004106/http://docshare01.docshare.tips/files/18925/189257997.pdf) (PDF) from the original on 5 May 2021. Retrieved 5 May 2021.
- Kessler, Michelle (22 December 2003), *More shoppers proceed to checkout online* (https://usatoday30.usatoday.com/tech/news/2003-12-22-shoppers_x.htm), archived (https://web.archive.org/web/20201231093120/https://usatoday30.usatoday.com/tech/news/2003-12-22-shoppers_x.htm) from the original on 31 December 2020, retrieved 5 May 2021

- Lowry, Paul Benjamin; Wells, Taylor Michael; Moody, Greg; Humpherys, Sean; Kettles, Degan (3 February 2006), *Online Payment Gateways Used to Facilitate E-Commerce Transactions and Improve Risk Management* (https://poseidon01.ssrn.com/delivery.php?ID=00606712403100201210207300507110202812100306701604900512510102912309111610700610308803300018125109112124079012078102065108100010041074039018102125087094069029023054054032101027123099084006096096098113025008086066011113125031124019127031001124116068), vol. 17 (published January 2006), pp. 1–48, SSRN 879797 (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=879797), archived (https://web.archive.org/web/20210505004134/https://papers.ssrn.com/sol3/papers.cfm?abstract_id=879797) from the original on 5 May 2021, retrieved 5 May 2021
- Kotler, Philip (2009). *Marketing Management* (https://archive.org/details/frameworkformark00kotl_2) (4th ed.). Upper Saddle River, New Jersey: Prentice Hall. ISBN 9780136026600. OCLC 1149204899 (https://search.worldcat.org/oclc/1149204899). Retrieved 5 May 2021.
- Miller, Roger LeRoy; Cross, Frank B. (2002). *The Legal and E-Commerce Environment Today: Business in Its Ethical, Regulatory, and International Setting* (https://www.google.com/books/?id=4wQ9AQAAIAAJ) (3rd ed.). South-Western. ISBN 9780324061888. Archived (https://web.archive.org/web/20210505004111/https://books.google.com/books/?id=4wQ9AQAAIAAJ) from the original on 5 May 2021. Retrieved 5 May 2021.
- Nissanoff, Daniel (2006). *FutureShop: How the New Auction Culture Will Revolutionize the Way We Buy, Sell and Get the Things We Really Want* (https://archive.org/details/futureshophownew00niss). New York City: The Penguin Press. ISBN 978-1-59420-077-9. OCLC 1149173925 (https://search.worldcat.org/oclc/1149173925). Retrieved 5 May 2021.
- Seybold, Patricia B. (2001). *The Customer Revolution* (https://archive.org/details/customerrevoluti00seyb) (1st ed.). New York City: Crown Business. ISBN 978-0-609-60772-5. OCLC 1148801120 (https://search.worldcat.org/oclc/1148801120). Retrieved 5 May 2021.

External links

- *E-Commerce Resources* (https://web.archive.org/web/20170521010809/https://www.sba.gov/managing-business/running-business/using-technology/ecommerce-resources), Small Business Administration, archived from the original (http://www.sba.gov/content/ecommerce-resources) on 21 May 2017

WIKIPEDIA
The Free Encyclopedia

Business models for open-source software

(Redirected from Commercial open-source software)

Software companies focusing on the development of open-source software (OSS) employ a variety of business models to solve the challenge of making profits from software that is under an open-source license. Each of these business strategies rest on the premise that users of open-source technologies are willing to purchase additional software features under proprietary licenses, or purchase other services or elements of value that complement the open-source software that is core to the business. This additional value can be, but not limited to, enterprise-grade features and up-time guarantees (often via a service-level agreement) to satisfy business or compliance requirements, performance and efficiency gains by features not yet available in the open source version, legal protection (e.g., indemnification from copyright or patent infringement), or professional support/training/consulting that are typical of proprietary software applications.

Historically, these business models started in the late 1990s and early 2000s as "dual-licensing" models (for example MySQL[1]), and they have matured over time, giving rise to multiple variations as described in the sections below. Pure dual licensing models are not uncommon, as a more nuanced business approach to open source software businesses has developed. Many of these variations are referred to as "open core" model, where the companies develop both open source software elements and other elements of value for a combined product.

A variety of open-source compatible business approaches have gained prominence in recent years, as illustrated and tracked by the Commercial Open Source Software Index (COSSI),[2] a list of commercial open source companies that have reached at least US$100 million in revenue. Notable examples include open core (sometimes referred to as dual licensing or multi-licensing), software as a service (not charging for the software but for the tooling and platform to consume the software as a service often via subscription), freemium, donation-based funding, crowdfunding, and crowdsourcing.

There are several different types of business models for making profit using OSS or funding the creation and ongoing development and maintenance. The list below shows a series of current existing and legal commercial business models approaches in the context of open-source software and open-source licenses.[3] The acceptance of these approaches has been varied; some of these approaches are recommended (like open core and selling services), others are accepted, while still others are considered controversial or even unethical by the open-source community. The underlying objective of these business models is to harness the size and international scope of the open-source community. Depending on the project the funding options and their success differs for a sustainable commercial venture.[4] The vast majority of commercial open-source companies experience a conversion ratio (as measured by the percentage of downloaders who buy something) well below 1%, so low-cost and highly-scalable marketing and sales functions are key to these firms' profitability.[5]

Not selling code

Professional services

Open-source software can also be commercialized from selling services, such as training, technical support, or consulting, rather than the software itself.[6][7]

Another possibility is offering open-source software in source code form only, while providing executable binaries to paying customers only, offering the commercial service of compiling and packaging of the software. Also, providing goods like physical installation media (e.g., DVDs) can be a commercial service.

Open-source companies using this business model successfully are, for instance RedHat,[8] IBM, SUSE, Hortonworks (for Apache Hadoop), Chef, and Percona (for open-source database software).

Branded merchandise

Some open-source organizations such as the Mozilla Foundation[9] and the Wikimedia Foundation[10] sell branded merchandise articles like t-shirts and coffee mugs. This can be also seen as an additional service provided to the user community.

Software as a service

Selling subscriptions for online accounts and server access to customers is one way of adding value to open-source software. Another way is combining desktop software with a service, called software plus services. Most open core companies that use this approach also provide the software in a fashion suitable for on-premises, do-it-yourself deployment. To some customers, however, there is significant value in a "plug and play" hosted product. Open source businesses that use this model often cater to small and medium enterprises who do not have the technology resources to run the software. Providing cloud computing services or software as a service (SaaS) without the release of the open-source software is not an open source deployment. With a SaaS approach, businesses no longer need to write new code from scratch, but instead can use the software they need by paying a subscription. Serverless technology allows businesses to completely transfer infrastructure management to the provider, which means that teams can create scalable applications more efficiently, cheaper, easier, and more reliably.[11]

The FSF called the server-side use-case without release of the source-code the "ASP loophole in the GPLv2" and encourage therefore the use of the GNU Affero General Public License which plugged this hole in 2002.[12][13]

Voluntary donations

There were experiments by Independent developers to fund development of open-source software donation-driven directly by the users, e.g. with the Illumination Software Creator in 2012.[14] Since 2011, SourceForge allows users to donate to hosted projects that opted to accept donations, which is enabled via PayPal.[15]

Larger donation campaigns also exist. In 2004 the Mozilla Foundation carried out a fundraising campaign to support the launch of the Firefox 1.0 web browser. It placed a two-page ad in the December 16 edition of *The New York Times* listing the names of the thousands who had donated.[16][17]

In May 2019, GitHub, a Git-based software repository hosting, management and collaboration platform owned by Microsoft, launched a Sponsors program that allows people who support certain open source projects hosted on GitHub to donate money to developers who contribute and maintain the project.[18]

Crowdsourcing

Crowdsourcing is a type of participative online activity in which an individual, an institution, a nonprofit organization, or company proposes to a group of individuals of varying knowledge, heterogeneity, and number, the voluntary undertaking of a task via a flexible open call. The undertaking of the task, of variable complexity and modularity, and in which the crowd should participate, bringing their work, money, knowledge and/or experience, always entails mutual benefit. The user will receive the satisfaction of a given type of need, be it economic, social recognition, self-esteem, or the development of individual skills, while the crowdsourcer will obtain and use to their advantage that which the user has brought to the venture, whose form will depend on the type of activity undertaken. Caveats in pursuing a Crowdsourcing strategy are to induce a substantial market model or incentive, and care has to be taken that the whole thing doesn't end up in an open source anarchy of adware and spyware plagiates, with a lot of broken solutions, started by people who just wanted to try it out, then gave up early, and a few winners. Popular examples for Crowdsourcing are Linux, Google Android, the Pirate Party movement, and Wikipedia.

Training and certification

Offering training programs and certification courses related to the open-source software, catering to individuals or organizations, like Red Hat Certification Program or Linux Professional Institute Certification Programs.

Selling users

Partnership with funding organizations

Other financial situations include partnerships with other companies. Governments, universities, companies, and non-governmental organizations may develop internally or hire a contractor for custom in-house modifications, then release that code under an open-source license. Some organizations support the development of open-source software by grants or stipends, like Google's Summer of Code initiative founded in 2005.[19]

Advertising-supported software

In order to commercialize FOSS (free and open-source software), many companies (including Google, Mozilla, and Canonical) have moved towards an economic model of advertising-supported software. For instance, the open-source application AdBlock Plus gets paid by Google for letting whitelisted *Acceptable*

Ads bypassing the browser ad remover.[20] As another example is SourceForge, an open-source project service provider, has the revenue model of advertising banner sales on their website. In 2006, SourceForge reported quarterly takings of $6.5 million[21] and $23 million in 2009.[22]

Pre-selling code

Bounty driven development

The users of a particular software artifact may come together and pool money into an open-source bounty for the implementation of a desired feature or functionality. Offering bounties as funding has existed for some time. For instance, Bountysource was a web platform which has been offering this funding model for open source software since 2003.

Another bounty source is companies or foundations that set up bounty programs for implemented features or bugfixes in open-source software relevant to them. For instance, Mozilla has been paying and funding freelance open-source programmers for security bug hunting and fixing since 2004.[23][24][25]

Pre-order/crowdfunding/reverse-bounty model

A newer funding opportunity for open-source software projects is crowdfunding, which shares similarities with the pre-order or Praenumeration business model, as well as the reverse bounty model, typically organized over web platforms like Kickstarter,[26] Indiegogo,[27] or Bountysource[28] (see also comparison of crowd funding services). One example is the successfully funded Indiegogo campaign in 2013 by Australian programmer Timothy Arceri, who offered to implement an OpenGL 4.3 extension for the Mesa library in two weeks for $2,500.[27] Arceri delivered the OpenGL extension code which was promptly merged upstream, and he later continued his efforts on Mesa with successive crowdfunding campaigns.[29] Later, he found work as an employee in this domain with Collabora and in 2017 with Valve.[30] Another example is the June 2013 crowdfunding on Kickstarter[31][32] of the open source video game Cataclysm: Dark Days Ahead which raised the payment of a full-time developer for 3.5 months. Patreon funding has also become an effective option, as the service gives the option to pay out each month to creators, many of whom intend to develop free and open-source software.[33]

Selling licensing deals

Dual-licensing or Open Core

In a dual licensing model, the vendor develops software and offers it under an open-source license but also under separate proprietary license terms. The proprietary version can be licensed to finance the continued development of the free open-source version.[34] Customers may prefer a no-cost and open-source edition for testing, evaluation, proof of concept development, and small scale deployment. If the customer wishes to deploy the software at scale, or in proprietary distributed products, the customer then negotiates for a commercial license to an enterprise edition. Further, customers will learn of open-source software in a company's portfolio and offerings but generate business in other proprietary products and solutions, including commercial technical support contracts and services. A popular example is Oracle's

MySQL database which is dual-licensed under a commercial proprietary license and also under the GPLv2.[35] Another example is the Sleepycat License. Flask developer Armin Ronacher stated that the AGPLv3 was a "terrible success" as "vehicle for dual commercial licensing" and noted that MongoDB, RethinkDB, OpenERP, SugarCRM as well as WURFL utilizing the license for this purpose.[36]

Dual license products are generally sold as a "community version" and an "enterprise version." In a pure dual licensing model, as was common before 2010, these versions are identical but available under a choice of licensing terms. Added proprietary software may help customers analyze data, or more efficiently deploy the software on their infrastructure or platform. Examples include the IBM proprietary Linux software, where IBM contributes to the Linux open-source ecosystem, but it builds and delivers (to IBM's paying customers) database software, middleware, and other software that runs on top of the open-source core. Other examples of proprietary products built on open-source software include Red Hat Enterprise Linux and Cloudera's Apache Hadoop-based software.

Selling certificates and use of trademark

Another financing approach is innovated by Moodle, an open source learning management system and community platform.[37][38] The business model revolves around a network of commercial partners[39] who are certified and therefore authorised to use the Moodle name and logo,[40] and in turn provide a proportion of revenue to the Moodle Trust, which funds core development.[41]

Re-licensing under a proprietary license

If a software product uses only own software and open-source software under a permissive free software licence, a company can re-license the resulting software product under a proprietary license and sell the product without the source code or software freedoms.[42] For instance, Apple Inc. is an avid user of this approach by using source code and software from open-source projects. For example, the BSD Unix operating system kernel (under the BSD license) was used in Apple's Mac PCs that were sold as proprietary products.[43] Another variant is to re-license the software under a license that allows modification, but restricts commercial use.[44][45]

Selling proprietary additives

Selling optional proprietary extensions

Some companies sell proprietary but optional extensions, modules, plugins or add-ons to an open-source software product. This approach is a variant of the freemium business model. The proprietary software may be intended to let customers get more value out of their data, infrastructure, or platform, e.g., operate their infrastructure/platform more effectively and efficiently, manage it better, or secure it better. Examples include the IBM proprietary Linux software, where IBM contributes to the Linux open-source ecosystem, but it builds and delivers (to IBM's paying customers) database software, middleware, and other software that runs on top of the open-source core. Other examples of proprietary products built on open-source software include Red Hat Enterprise Linux and Cloudera's Apache Hadoop-based software. Some companies appear to re-invest a portion of their financial profits from the sale of proprietary software back into the open source infrastructure.[46]

The approach can be problematic with many open source licenses ("not license conform") if not carried out with sufficient care. For instance, mixing proprietary code and open-source licensed code in statically linked libraries[47] or compiling all source code together in a software product might violate open-source licenses, while keeping them separated by interfaces and dynamic-link libraries would adhere to license conform.

Selling required proprietary parts of a software product

A variant of the approach above is the keeping of required data content (for instance a video game's audio, graphic, and other art assets) of a software product proprietary while making the software's source code open-source. While this approach is completely legitimate and compatible with most open-source licenses, customers have to buy the content to have a complete and working software product.[48] Restrictive licenses can then be applied on the content, which prevents the redistribution or re-selling of the complete software product. Examples for open-source developed software are Kot-in-Action Creative Artel video game *Steel Storm*, engine GPLv2 licensed while the artwork is CC BY-NC-SA 3.0 licensed,[49] and *Frogatto & Friends* with an own developed open-source engine[50] and commercialization via the copyrighted game assets[51] for iPhone, BlackBerry and MacOS.[52]

Other examples are *Arx Fatalis* (by Arkane Studios)[53] and *Catacomb 3-D* (by Flat Rock Software)[54] with source code opened to the public delayed after release, while copyrighted assets and binaries are still sold on gog.com as digital distribution.[55]

Richard Stallman stated that freedom for works for art or entertainment are not required.[56]

The similar product bundling of an open-source software product with hardware which prevents users from running modified versions of the software is called tivoization and is legal with most open-source licenses except GPLv3, which explicitly prohibits this use-case.[57]

Selling proprietary update systems

Another variant of the approach above, mainly use for data-intensive, data-centric software programs, is the keeping of all versions of the software under a free and open-source software license, but refraining from providing update scripts from a n to an n+1 version. Users can still deploy and run the open source software. However, any update to the next version requires either exporting the data, reinstalling the new version, then reimporting the data to the new version, or subscribing to the proprietary update system, or studying the two versions and recreating the scripts from scratch.

This practice does not conform with the free software principles as espoused by the FSF. Richard Stallman condemns this practice and names it "diachronically trapped software".[58]

Selling without proprietary license

All of the above methods follows from the traditional approach in the selling software, where Software is licensed for installation and execution on a user- or customer-supplied infrastructure. In the classic software product business, revenues typically originate from selling software upgrades to the customer. However, it's also practicing selling exactly the same programs or add-ons but without proprietary licensing. For example, applications like ardour,[59] radium[60] or fritzing[61] it's completely free software on GPL license but there is a fee to get the official binary, often bundled with tech support or the privileges of attracting developers' attention to adding new functionalities to the program. It is also practiced to sell both source code and binaries, as Red Hat did.[62]

This practice does conform with the free software principles as espoused by the FSF.[63]

Other common business models

Obfuscation of source code

An approach to allow commercialization under some open-source licenses while still protecting crucial business secrets, intellectual property and technical know-how is obfuscation of source code. This approach was used in several cases, for instance by Nvidia in their open-source graphic card device drivers.[64] This practice is used to get the open-source-friendly propaganda without bearing the inconveniences. There has been debate in the free-software/open-source community on whether it is illegal to skirt copyleft software licenses by releasing source code in obfuscated form, such as in cases in which the author is less willing to make the source code available. The general consensus was that while unethical, it was not considered a violation.

The Free Software Foundation is against this practice.[65] The GNU General Public License since version 2 has defined "source code" as "the preferred form of the work for making modifications to it." This is intended to prevent the release of obfuscated source code.[66]

Delayed open-sourcing

Some companies provide the latest version available only to paying customers. A vendor forks a non-copyleft software project then adds closed-source additions to it and sells the resulting software. After a fixed time period the patches are released back upstream under the same license as the rest of the codebase. This business model is called version lagging or time delaying.[46][67]

For instance, 2016 the MariaDB Corporation created for business compatible "delayed open-sourcing" the source-available *Business source license* (BSL) which automatically relicenses after three years to the FOSS GPL.[68][69] This approach guarantees licensees that they have source code access (e.g. for code audits), are not locked into a closed platform, or suffer from planned obsolescence, while for the software developer a time-limited exclusive commercialization is possible.[68] In 2017 followed version 1.1, revised with feedback also from Bruce Perens.[70][71]

However, this approach works only with own software or permissive licensed code parts, as there is no copyleft FOSS license available which allows the time delayed opening of the source code after distributing or selling of a software product.

Open sourcing on end-of-life

An extreme variant of "delayed open-sourcing" is a business practice popularized by id Software[72][73] and 3D Realms,[74][75] which released several software products under a free software license after a long proprietary commercialization time period and the return of investment was achieved. The motivation of companies following this practice of releasing the source code when a software reaches the commercial end-of-life, is to prevent that their software becomes unsupported Abandonware or even get lost due to digital obsolescence.[76] This gives the user communities the chance to continue development and support of the software product themselves as an open-source software project.[77] Many examples from the video game domain are in the list of commercial video games with later released source code.

Popular non-game software examples are the Netscape Communicator which was open-sourced in 1998[78][79] and Sun Microsystems's office suite, StarOffice, which was released in October 2000 at its commercial end of life.[80] Both releases made foundational contributions to now prominent open-source projects, namely Mozilla Firefox and OpenOffice.org/LibreOffice.

Funding

Unlike proprietary off-the-shelf software that come with restrictive licenses, open-source software is distributed freely, through the web and in physical media. Because creators cannot require each user to pay a license fee to fund development this way, a number of alternative development funding models have emerged.

An example of those funding models is when bespoke software is developed as a consulting project for one or more customers who request it. These customers pay developers to have this software developed according to their own needs and they could also closely direct the developers' work. If both parties agree, the resulting software could then be publicly released with an open-source license in order to allow subsequent adoption by other parties. That agreement could reduce the costs paid by the clients while the original developers (or independent consultants) can then charge for training, installation, technical support, or further customization if and when more interested customers would choose to use it after the initial release.

There also exist stipends to support the development of open source software, such as Google's Summer of Code[19] and Outreachy.[81]

Another approach to funding is to provide the software freely, but sell licenses to proprietary add-ons such as data libraries. For instance, an open-source CAD program may require parts libraries which are sold on a subscription or flat-fee basis. Open-source software can also promote the sale of specialized hardware that it interoperates with, some example cases being the Asterisk telephony software developed by PC-telephony hardware manufacturer Digium and the Robot Operating System (ROS) robotics platform by Willow Garage and Stanford AI Labs. Many open source software projects have begun as research projects within universities, as personal projects of students or professors, or as tools to aid scientific research. The influence of universities and research institutions on open-source shows in the number of projects named after their host institutions, such as BSD Unix, CMU Common Lisp, or the NCSA HTTPd which evolved into Apache.

Companies may employ developers to work on open-source projects that are useful to the company's infrastructure: in this case, it is developed not as a product to be sold but as a sort of shared public utility. A local bug-fix or solution to a software problem, written by a developer either at a company's request or to make his/her own job easier, can be released as an open-source contribution without costing the company anything.[82] A larger project such as the Linux kernel may have contributors from dozens of companies which use and depend upon it, as well as hobbyist and research developers.

A new funding approach for open-source projects is crowdfunding, organized over web platforms like Kickstarter, Indiegogo, or Bountysource.[28] Liberapay is a crowdfunding platform, primarily for open-source projects, that is itself open-source.[83]

Challenges

Open-source software can be sold and used in general commercially. Also, commercial open-source applications have been a part of the software industry for some time.[84][85] While commercialization or funding of open-source software projects is possible, it is considered challenging.[86]

Since several open-source licenses stipulate that authors of derivative works must distribute them under an open-source (copyleft) license, ISVs and VARs have to develop new legal and technical mechanisms to foster their commercial goals,[3] as many traditional mechanisms are not directly applicable anymore.

Traditional business wisdom suggests that a company's methods, assets, and intellectual properties should remain concealed from market competitors (trade secret) as long as possible to maximize the profitable commercialization time of a new product.[87] Open-source software development minimizes the effectiveness of this tactic; development of the product is usually performed in view of the public, allowing competing projects or clones to incorporate new features or improvements as soon as the public code repository is updated, as permitted by most open-source licenses. Also in the computer hardware domain, a hardware producer who provides free and open software drivers reveals the knowledge about hardware implementation details to competitors, who might use this knowledge to catch up.

Therefore, there is considerable debate about whether vendors can make a sustainable business from an open-source strategy. In terms of a traditional software company, this is probably the wrong question to ask. Looking at the landscape of open source applications, many of the larger ones are sponsored (and largely written) by system companies such as IBM who may not have an objective of software license revenues. Other software companies, such as Oracle and Google, have sponsored or delivered significant open-source code bases. These firms' motivation tends to be more strategic, in the sense that they are trying to change the rules of a marketplace and reduce the influence of vendors such as Microsoft. Smaller vendors doing open-source work may be less concerned with immediate revenue growth than developing a large and loyal community, which may be the basis of a corporate valuation at merger time.

FOSS and economy

According to Yochai Benkler, the Berkman Professor for Entrepreneurial Legal Studies at Harvard Law School, free software is the most visible part of a new economy of commons-based peer production of information, knowledge, and culture. As examples, he cites a variety of FOSS projects, including both free software and open source.[88]

This new economy is already under development. In order to commercialize FOSS, many companies, Google being the most successful, are moving towards an economic model of advertising-supported software. In such a model, the only way to increase revenue is to make the advertising more valuable. Facebook has recently come under fire for using novel user tracking methods to accomplish this.[89]

This new economy is not without alternatives. Apple's App Stores have proven very popular with both users and developers. The Free Software Foundation considers Apple's App Stores to be incompatible with its GPL and complained that Apple was infringing on the GPL with its iTunes terms of use.[90] Rather than change those terms to comply with the GPL, Apple removed the GPL-licensed products from its App Stores.[91] The authors of VLC, one of the GPL-licensed programs at the center of those complaints, recently began the process to switch from the GPL to the LGPL and MPL.[92][93]

Examples

Much of the Internet runs on open-source software tools and utilities such as Linux, Apache, MySQL, and PHP, known as the LAMP stack for web servers. Using open source appeals to software developers for three main reasons: low or no cost, access to source code they can tailor themselves, and a shared community that ensures a generally robust code base, with quick fixes for new issues.

Despite doing much business in proprietary software, some companies like Oracle Corporation and IBM participated in developing free and open-source software to deter from monopolies and take a portion of market share for themselves. See Commercial open-source applications for the list of current commercial open-source offerings. Netscape's actions were an example of this, and thus Mozilla Firefox has become more popular, getting market share from Internet Explorer.

- Active Agenda is offered for free, but requires all extensions to be shared back with the world community. The project sells a "Non-Reciprocal Private License" to anyone interested in keeping module extensions private.
- Adobe Systems offers Flex for free, while selling the Flash Builder IDE.
- Apple Inc. offers Darwin for free, while selling Mac OS X.
- Asterisk, digital electronics hardware controlled by open-source software
- Codeweavers sells CrossOver commercially, deriving it from the free Wine project they also back.
- Canonical Ltd. offers Ubuntu for free, while they sell commercial technical support contracts.
- Cloudera's Apache Hadoop-based software.
- Francisco Burzi offers PHP-Nuke for free, but the latest version is offered commercially.
- IBM proprietary Linux software, where IBM delivers database software, middleware and other software.
- Ingres is offered for free, but services and support are offered as a subscription. The Ingres Icebreaker Appliance is also offered as a commercial database appliance.
- id Software releases their legacy game engines under the GPL, while retaining proprietary ownership on their latest incarnation.
- Mozilla Foundation have a partnership with Google and other companies which provides revenue for inclusion of search engines in Mozilla Firefox.
- MySQL is offered for free, but with the enterprise version includes support and additional features.
- SUSE offers openSUSE for free through the openSUSE Project, while selling SUSE Linux Enterprise (SLE).
- OpenSearchServer offers its community edition on SourceForge and an enterprise edition with professional services to enterprises with a paid license
- Oracle - VirtualBox is free and open to anyone, but the VirtualBox extension pack can only be used for free at home, thus requiring payment from business users

- OWASP Foundation is a professional community of open-source developers focused on raising visibility for software security.
- Red Hat sells support subscriptions for Red Hat Enterprise Linux (RHEL) which is an enterprise distribution periodically forked from the community-developed Fedora.
- Sourcefire offers Snort for free, while selling Sourcefire 3D.
- Sun Microsystems (acquired by Oracle in 2010) once offered OpenOffice.org for free, while selling StarOffice
- Untangle provides its Lite Package for free, while selling its Standard and Premium Packages by subscription
- Zend Technologies offers Zend Server CE and Laminas for free, but sells Zend Server with support and additional features.

See also

Business and Economics portal

- Commercial use of copyleft works
- Free software business model
- Open Source Development Labs
- Open business
- Open innovation
- Software monetization

References

1. "MySQL :: Commercial License for OEMs, ISVs and VARs" (https://www.mysql.com/about/legal/licensing/oem/). *www.mysql.com*. Retrieved 2019-09-11.
2. "COSSI: $100M+ Revenue Commercial Open-Source Software Company Index" (https://docs.google.com/spreadsheets/d/17nKMpi_Dh5slCqzLSFBoWMxNvWiwt2R-t4e_l7LPLhU/edit?usp=embed_facebook). *Google Docs*. Retrieved 2019-08-28.
3. Popp, Dr. Karl Michael (2015). *Best Practices for commercial use of open source software*. Norderstedt, Germany: Books on Demand. ISBN 978-3738619096.
4. Augmenta. "Open Source Monetization: Challenges and Strategies" (https://www.license-token.com/wiki/open-source-monetization-challenges-and-strategies). *license-Token wiki*. Retrieved 26 March 2025.
5. Riehle, Dirk. "The Single-Vendor Commercial Open Source Business Model" (https://www.researchgate.net/publication/220385101). *Researchgate*. Retrieved 19 September 2021.
6. Germain, Jack M. (5 November 2013). "FOSS in the Enterprise: To Pay or Not to Pay?" (http://www.linuxinsider.com/story/79341.html). *LinuxInsider*. ECT News Network, Inc. Retrieved 18 June 2016.
7. Rubens, Paul (13 February 2013). "6 Reasons to Pay for Open Source Software" (http://www.cio.com/article/2388344/open-source-tools/6-reasons-to-pay-for-open-source-software.html). *CIO*. CXO Media, Inc. Retrieved 18 June 2016. "Open source software is free to download, modify and use, but that doesn't mean it's not worth paying for sometimes. If you're using open source software in a commercial, enterprise capacity, here are six reasons why you should pay for free software."

8. McMillan, Robert (28 March 2012). "Red Hat Becomes Open Source's First $1 Billion Baby" (https://www.wired.com/2012/03/red-hat/). *Wired*. Retrieved 18 June 2016. "Other companies have made big money selling Linux — Intel, IBM, Dell, and others have used it as a way to sell hardware and support services — but Red Hat has managed the tricky business of building a software platform that big businesses will pay for."
9. Markham, Gervase (16 March 2004). "Mozilla Foundation Open Letter Orders Unofficial Mozilla Merchandise Sellers to Stop, Legal Action Hinted" (http://www.mozillazine.org/articles/article4484.html). *MozillaZine*. Retrieved 18 June 2016.
10. "Wikipedia Store" (https://store.wikimedia.org/). Wikimedia Foundation. 2016. Retrieved 18 June 2016.
11. Ballhausen, Miriam (2014). "OpenSaaS: Using Free and Open Source Software as Software-as-a-Service" (https://www.jolts.world/index.php/jolts/article/view/103). *International Free and Open Source Software Law Review*. **6**: 61–68. ISSN 2666-8106 (https://search.worldcat.org/issn/2666-8106).
12. "Licenses" (https://www.gnu.org/licenses/index.html). *GNU Project*. Free Software Foundation. The GNU Affero General Public License. Retrieved 18 June 2016. "We recommend that people consider using the GNU AGPL for any software which will commonly be run over a network."
13. Tiemann, Michael (7 June 2007). "GNU Affero GPL version 3 and the "ASP loophole" " (https://opensource.org/blog/gnu-affero-gpl-version-3-and-the-asp-loophole). Open Source Initiative. Retrieved 18 June 2016.
14. Sneddon, Joey-Elijah (2012-06-01). "Will You Help Change The Way Open-Source Apps are Funded?" (http://www.omgubuntu.co.uk/2012/06/help-linux-tycoon-more-go-open-source). OMGUbuntu. Retrieved 2013-08-08. "Lunduke is pledging to open-source and distribute his portfolio of hitherto paid software – which includes the Linux distro management simulator Linux Tycoon - for free, under the GPL, if he can reach a donation-driven funding goal of $4000/m. Reaching this goal, Lunduke says, 'will provide proof for others, who would also like to move their software businesses to be open source, that it is doable.' "
15. Naramore, Elizabeth (4 March 2011). "SourceForge.net Donation System" (https://sourceforge.net/p/forge/documentation/Donations/?version=1). *SourceForge*. Slashdot Media. Retrieved 16 October 2017.
16. Mozilla Foundation (December 15, 2004). "Mozilla Foundation Places Two-Page Advocacy Ad in *The New York Times*" (https://www-archive.mozilla.org/press/mozilla-2004-12-15.html). Retrieved June 15, 2010.
17. Marson, Ingrid (2004-12-16). "New York Times runs Firefox ad" (http://news.cnet.com/New-York-Times-runs-Firefox-ad/2100-1032_3-5493774.html). cnet.com. Retrieved 2013-08-12. "Fans of the Mozilla Foundation's Firefox browser who funded an advertisement in The New York Times will finally get to see their names in print on Thursday."
18. Tung, Liam. "GitHub will now let you back your favourite open source developers" (https://www.zdnet.com/article/github-will-now-let-you-back-your-favourite-open-source-developers/). *ZDNet*. Retrieved 2019-08-26.
19. Byfield, Bruce (21 September 2005). "Google's Summer of Code concludes" (https://www.linux.com/news/googles-summer-code-concludes). linux.com. Retrieved 18 June 2016. "DiBona said that the SOC was designed to benefit everyone involved in it. Students had the chance to work on real projects, rather than academic ones, and to get paid while gaining experience and making contacts. FOSS projects benefited from getting new code and having the chance to recruit new developers."
20. Callaham, John (2013-06-06). "Report: Google paying AdBlock Plus to not block Google's ads" (https://www.neowin.net/news/report-google-paying-adblock-plus-to-not-block-google039s-ads). neowin.com. Retrieved 2013-08-13. "Google is paying money to Eyeo, the company behind AdBlock Plus, so that its ads get through the browser ad remover."

21. Hunt, Katherine (2007-05-24). "Sourceforge quarterly profit surges as revenue rises" (http://www.marketwatch.com/story/sourceforge-quarterly-profit-surges-as-revenue-rises). marketwatch.com. Retrieved 2013-08-13. "Software Corp., late Thursday reported third-quarter net earnings of $6.49 million, or 9 cents a share, up from $997,000, or 2 cents a share, during the year-ago period. Pro forma earnings from continuing operations were $2.1 million, or 3 cents a share, compared with $1.2 million, or 2 cents a share, last year. The Fremont, Calif.-based maker of computer servers and storage systems said revenue for the three months ended April 30 rose to $10.3 million from $7.9 million. Analysts, on average, had forecast a per-share profit of 2 cents on revenue of $12 million."
22. "SourceForge Reports Second Quarter Fiscal 2009 Financial Results" (https://web.archive.org/web/20150603170625/http://ir.corp.sourceforge.com/phoenix.zhtml?c=82629&p=irol-newsArticle&ID=1260642&highlight=). Archived from the original (http://ir.corp.sourceforge.com/phoenix.zhtml?c=82629&p=irol-newsArticle&ID=1260642&highlight=) on 2015-06-03.
23. Leyden, John (2004-08-03), *Mozilla to pay bounty on bugs* (http://www.securityfocus.com/news/9255), The Register, retrieved 2013-08-10
24. Evers, Joris (July 25, 2005). "Offering a bounty for security bugs" (http://news.cnet.com/2100-7350_3-5802411.html). *CNET*. CBS Interactive. Retrieved 12 August 2007.
25. "Mozilla Foundation Announces Security Bug Bounty Program" (https://blog.mozilla.org/press/2004/08/mozilla-foundation-announces-security-bug-bounty-program/). *Mozilla Foundation*. Mountain View, California. August 2, 2004. Retrieved 2013-08-10.
26. Lunduke, Bryan (2013-08-07). "Open source gets its own crowd-funding site, with bounties included - Bountysource is the crowd-funding site the open source community has been waiting for" (http://www.networkworld.com/community/blog/open-source-gets-its-own-crowd-funding-site-bounties-included). networkworld.com. Retrieved 2013-08-10. "Many open source projects (from phones to programming tools) have taken to crowd-funding sites (such as Kickstarter and indiegogo) in order to raise the cash needed for large-scale development. And, in some cases, this has worked out quite well."
27. Arceri, Timothy (2013-07-26). "Help improve OpenGL support for the Linux Graphics Drivers" (http://www.indiegogo.com/projects/help-improve-opengl-support-for-the-linux-graphics-drivers). Indiegogo. Retrieved 2013-08-11. "Helping fund the time for me to become a Mesa contributor and document the experience to make it easier for others to understand where to start with the Mesa codebase. Many people have brought up the idea of crowd sourcing open source driver development. This is a small scale experiment to see if it could actually work."
28. "Bountysource Raises $1.1 Million for the First Crowdfunding Platform for Open-Source Software Projects" (https://finance.yahoo.com/news/bountysource-raises-1-1-million-130000440.html). *Yahoo Finance*. Marketwired. 16 July 2013. Retrieved 18 June 2016.
29. Larabel, Michael (12 November 2013). "Crowd-Funding Is Back For Another Mesa Extension" (https://www.phoronix.com/scan.php?page=news_item&px=MTUxMTQ). *Phoronix*.
30. Larabel, Michael (14 February 2017). "Valve Has Another Linux Graphics Driver Developer Working On Open-Source AMD" (https://www.phoronix.com/scan.php?page=news_item&px=Arceri-Joined-Valve). *Phoronix*.
31. "Cataclysm: Dark Days Ahead - Dedicated Developer" (http://www.kickstarter.com/projects/568375735/cataclysm-dark-days-ahead-dedicated-developer/). *Kickstarter*. 22 June 2013.
32. "Multipocalyptic Roguelike Cataclysm: Dark Days Ahead Turns To Kickstarter" (https://web.archive.org/web/20140401001904/http://indiestatik.com/2013/06/22/cataclysm-dda-kickstarter/). Archived from the original (http://indiestatik.com/2013/06/22/cataclysm-dda-kickstarter/) on 2014-04-01.
33. Marchant, Julie. "Julie Marchant is creating libre video games" (https://www.patreon.com/onpon4). *Patreon*.

34. Solatan, Jean (2011). *Advances in software economics: A reader on business models and Partner Ecosystems in the software industry*. Norderstedt, Germany: BOD. ISBN 978-3-8448-0405-8.
35. "Commercial License for OEMs, ISVs and VARs" (http://www.mysql.com/about/legal/licensing/oem/#5). *MySQL.com*. Oracle. July 2010. Q4: What is Oracle's dual license model for MySQL software?. Retrieved 18 June 2016. "Oracle makes its MySQL database server and MySQL Client Libraries available under both the GPL and a commercial license. As a result, developers who use or distribute open source applications under the GPL can use the GPL-licensed MySQL software, and OEMs, ISVs and VARs that do not want to combine or distribute the MySQL software with their own commercial software under a GPL license can purchase a commercial license."
36. Ronacher, Armin (23 July 2013). "Licensing in a Post Copyright World" (http://lucumr.pocoo.org/2013/7/23/licensing/). *Armin Ronacher's Thoughts and Writings*. What Changed in 2007. Retrieved 18 June 2016. "The AGPLv3 was a terrible success, especially among the startup community that found the perfect base license to make dual licensing with a commercial license feasible. MongoDB, RethinkDB, OpenERP, SugarCRM as well as WURFL all now utilize the AGPLv3 as a vehicle for dual commercial licensing. The AGPLv3 makes that generally easy to accomplish as the original copyright author has the rights to make a commercial license possible but nobody who receives the sourcecode itself through the APLv3 inherits that right. I am not sure if that was the intended use of the license, but that's at least what it's definitely being used for now."
37. Gartner, Samantha (6 October 2014). "Moodle will always be an open source project" (https://opensource.com/education/14/10/open-access-learning-moodle). *opensource.com*. Retrieved 18 June 2016.
38. Dougiamas, Martin (22 January 2014). "Moodle: a case study in sustainability" (http://oss-watch.ac.uk/resources/cs-moodle). *OSS Watch*. University of Oxford (published 5 June 2007). Retrieved 18 June 2016.
39. "How do the Moodle Partners work?" (https://web.archive.org/web/20140722023159/http://moodle.com/partners/about/). *Moodle*. 2012. Archived from the original (http://moodle.com/partners/about/) on 22 July 2014. Retrieved 18 June 2016.
40. "The Moodle Trademark" (https://moodle.com/trademarks/). *Moodle*. 2016. Retrieved 18 June 2016.
41. Kolowich, Steve (27 March 2012). "Blackboard's Open-Source Pivot" (http://www.insidehighered.com/news/2012/03/27/blackboard-buys-moodlerooms-creates-open-source-division). *Inside Higher Ed*. Retrieved 18 June 2016.
42. Montague, Bruce (2013-11-13). "Why you should use a BSD style license for your Open Source Project - GPL Advantages and Disadvantages" (http://www.freebsd.org/doc/en_US.ISO8859-1/articles/bsdl-gpl/article.html). FreeBSD. Retrieved 2015-11-28. "In contrast to the GPL, which is designed to prevent the proprietary commercialization of Open Source code, the BSD license places minimal restrictions on future behavior. This allows BSD code to remain Open Source or become integrated into commercial solutions, as a project's or company's needs change. In other words, the BSD license does not become a legal time-bomb at any point in the development process. In addition, since the BSD license does not come with the legal complexity of the GPL or LGPL licenses, it allows developers and companies to spend their time creating and promoting good code rather than worrying if that code violates licensing."
43. Oram, Andy (2011-08-26). "How Free Software Contributed to the Success of Steve Jobs and Apple" (http://radar.oreilly.com/2011/08/how-free-software-contributed.html). radar.oreilly.com. Retrieved 2013-08-10. "the BSD license allowed Apple to keep its changes proprietary"
44. Oberhauser, Jan; Malac, Kenneth. "Fair-code" (https://faircode.io/). Retrieved 2025-03-27.
45. "Open Compensation Token License Homepage" (https://license-token.com/). iunera.com. Retrieved 2025-03-27.

46. Olson, Mike (13 November 2013). *Opportunities Abound in the Big Data Space* (http://ecorner.stanford.edu/videos/3223/Opportunities-Abound-in-the-Big-Data-Space-Entire-Talk). *Stanford eCorner*. Stanford University.
47. Hustvedt, Eskild (2009-02-08). "Our new way to meet the LGPL" (https://web.archive.org/web/20090220234542/http://blog.linuxgamepublishing.com/2009/02/08/our-new-way-to-meet-the-lgpl/). Archived from the original (http://blog.linuxgamepublishing.com/2009/02/08/our-new-way-to-meet-the-lgpl/) on 2009-02-20. Retrieved 2011-03-09. "You can use a special keyword $ORIGIN to say 'relative to the actual location of the executable'. Suddenly we found we could use -rpath $ORIGIN/lib and it worked. The game was loading the correct libraries, and so was stable and portable, but was also now completely in the spirit of the LGPL as well as the letter!"
48. "TTimo/doom3.gpl" (https://github.com/TTimo/doom3.gpl). GitHub. 2012-04-07. Retrieved 2013-08-10. "Doom 3 GPL source release [...] This source release does not contain any game data, the game data is still covered by the original EULA and must be obeyed as usual."
49. "STEEL STORM EPISODE 1 LIMITED USER SOFTWARE LICENSE AGREEMENT" (http://www.steel-storm.com/ss_license.html). steel-storm.com. Retrieved 2013-08-10. "For the purpose of this Agreement, the Art Assets include pk3 archive inside of 'steelstorm/gamedata/' folder that contain two-dimensional and three-dimensional works of graphic art, photographs, prints and art reproductions, maps, charts, diagrams, models, and technical drawings, sound effects and musical arrangements, documentation and tutorial videos, and are licensed under Attribution-NonCommercial-ShareAlike 3.0 Unported license. The Engine, which includes Windows, Linux and Mac binaries, and the Engine's source code, are licensed under GNU GPL v2 license."
50. Simpson, Kristina (2015-04-26). "LICENCE" (https://github.com/anura-engine/anura/blob/trunk/LICENSE). *anura-engine - GitHub*. Retrieved 2015-10-10.
51. frogatto (15 April 2020). "License" (https://github.com/frogatto/frogatto/blob/master/LICENSE). *GitHub*. "CC-BY 3.0 LICENSE [...] assets under copyright"
52. Gabovitch, Iwan (22 April 2011). "Humble Indie Bundle's Source Releases" (http://freegamer.blogspot.de/2011/04/humble-indie-bundles-source-releases.html). "Another game which is commercial (on iDevices) and has FOSS code and closed art [...] is Frogatto."
53. Nick (2011-01-14). "Arx Fatalis source code, patch released!" (http://bethblog.com/index.php/2011/01/14/arx-fatalis-source-code-patch-released/). bethblog.com. Retrieved 2011-08-10.
54. Larabel, Michael (6 June 2014). "id Software's Softdisk Open-Sources Some Really Old Games" (https://www.phoronix.com/scan.php?page=news_item&px=MTcxMjM). Phoronix. Retrieved 6 June 2014.
55. Ohle, Tom (4 December 2008). "Straight out of the Dungeon, Arx Fatalis invades GOG.com" (http://www.develop-online.net/press-releases/straight-out-of-the-dungeon-arx-fatalis-invades-gog-com/0129540). *Develop-Online.net* (Press release). Warsaw, Poland.
56. Stallman, Richard (2012). "On-line education is using a flawed Creative Commons license" (http://stallman.org/articles/online-education.html). stallman.org. Retrieved 2013-08-10. "In my view, nonfree licenses that permit sharing are ok for works of art/entertainment, or that present some party's viewpoint (such as this article itself). Those works aren't meant for doing a practical job, so the argument about the users' control does not apply. Thus, I do not object if they are published with the CC BY-NC-ND license, which allows only noncommercial redistribution of exact copies."
57. "Activities - FSFE" (https://fsfe.org/activities/activities.html). *FSFE - Free Software Foundation Europe*.
58. "gnu.org" (https://www.gnu.org/philosophy/when-free-depends-on-nonfree.en.html). *www.gnu.org*. Retrieved 2017-11-10.
59. "ardour" (https://ardour.org/download.html).

60. "radium" (http://users.notam02.no/~kjetism/radium/download.php).
61. "fritzing" (https://fritzing.org/download/).
62. McGrath, Mike. "Furthering the evolution of CentOS Stream" (https://www.redhat.com/en/blog/furthering-evolution-centos-stream). *redhat.com*. Retrieved 21 June 2023.
63. "Frequently Asked Questions about the GNU Licenses" (https://www.gnu.org/licenses/gpl-faq.en.html#DoesTheGPLAllowMoney).
64. Larabel, Michael (26 March 2010). "NVIDIA Drops Their Open-Source Driver, Refers Users To VESA Driver" (https://www.phoronix.com/scan.php?page=article&item=nvidia_kills_nv&num=1). *Phoronix*. "The xf86-video-nv driver has been around that provides very basic 2D acceleration and a crippled set of features besides that (no proper RandR 1.2/1.3, KMS, power management, etc.) while the code has also been obfuscated to try to protect their intellectual property."
65. "What is free software?" (https://www.gnu.org/philosophy/free-sw.html). *Free Software Foundation*. "Obfuscated "source code" is not real source code and does not count as source code."
66. "Reasoning behind the "preferred form" language in the GPL" (https://lwn.net/Articles/431651/). lwn.net. 2011-03-07. Retrieved 2013-08-19.
67. Sprewell (29 April 2010). "Towards A Real Business Model For Open-Source Software" (https://www.phoronix.com/scan.php?page=article&item=sprewell_licensing&num=1). *Phoronix*.
68. Martin, Alexander J (24 August 2016). "MySQL daddy Widenius: Open-source religion won't feed MariaDB" (https://www.theregister.co.uk/2016/08/24/monty_interview/). *The Register*.
69. Phipps, Simon (19 August 2016). "Uproar: MariaDB Corp. veers away from open source" (https://www.infoworld.com/article/3109213/open-source-tools/open-source-uproar-as-mariadb-goes-commercial.html). *InfoWorld*.
70. sl-1-1 (https://perens.com/2017/02/14/bsl-1-1/) on perens.com (2017-02-14)
71. releasing-bsl-11 (https://mariadb.com/resources/blog/releasing-bsl-11) on mariadb.com by Kaj Arnö (2017)
72. "id Software releases Doom 3 source code" (http://www.h-online.com/open/news/item/id-Software-releases-Doom-3-source-code-1383572.html). *The H Open*. 23 November 2011. Archived (https://web.archive.org/web/20131208041324/http://www.h-online.com/open/news/item/id-Software-releases-Doom-3-source-code-1383572.html) from the original on 8 December 2013.
73. Spencer, Spanner (24 March 2009). "id Software makes iPhone Wolfenstein open source" (http://www.pocketgamer.co.uk/r/iPhone/Wolfenstein+3D+Classic/news.asp?c=12324). *PocketGamer.co.uk*.
74. Siegler, Joe (1 April 2005). "Shadow Warrior Source Code Released" (http://legacy.3drealms.com/news/2005/04/shadow_warrior_12.html). 3D Realms.
75. "Games" (http://legacy.3drealms.com/games.html). 3D Realms. "Selected games have had their source code released by us. These games are: Duke Nukem 3D, Shadow Warrior, Rise of the Triad, Word Whiz, Beyond the Titanic, Supernova, & Kroz. You can obtain these from our downloads page."
76. Andersen, John (2011-01-27). "Where Games Go To Sleep: The Game Preservation Crisis, Part 1" (https://www.gamedeveloper.com/business/where-games-go-to-sleep-the-game-preservation-crisis-part-1). Gamasutra. Retrieved 2013-01-10. "The existence of decaying technology, disorganization, and poor storage could in theory put a video game to sleep permanently – never to be played again. Troubling admissions have surfaced over the years concerning video game preservation. When questions concerning re-releases of certain game titles are brought up during interviews with developers, for example, these developers would reveal issues of game production material being lost or destroyed. Certain game titles could not see a re-release due to various issues. One story began to circulate of source code being lost altogether for a well-known RPG, preventing its re-release on a new console."

77. Bell, John (2009-10-01). "Opening the Source of Art" (https://web.archive.org/web/20140330084636/http://timreview.ca/article/294). Technology Innovation Management Review. Archived from the original (http://timreview.ca/article/294) on 2014-03-30. Retrieved 2013-08-09. "[...]that no further patches to the title would be forthcoming. The community was predictably upset. Instead of giving up on the game, users decided that if Activision wasn't going to fix the bugs, they would. They wanted to save the game by getting Activision to open the source so it could be kept alive beyond the point where Activision lost interest. With some help from members of the development team that were active on fan forums, they were eventually able to convince Activision to release Call to Power II's source code in October of 2003."
78. "NETSCAPE ANNOUNCES PLANS TO MAKE NEXT-GENERATION COMMUNICATOR SOURCE CODE AVAILABLE FREE ON THE NET" (https://web.archive.org/web/20070401072854/http://wp.netscape.com/newsref/pr/newsrelease558.html). Netscape Communications Corporation. 1998-01-22. Archived from the original (http://wp.netscape.com/newsref/pr/newsrelease558.html) on 2007-04-01. Retrieved 2013-08-08. "BOLD MOVE TO HARNESS CREATIVE POWER OF THOUSANDS OF INTERNET DEVELOPERS; COMPANY MAKES NETSCAPE NAVIGATOR AND COMMUNICATOR 4.0 IMMEDIATELY FREE FOR ALL USERS, SEEDING MARKET FOR ENTERPRISE AND NETCENTER BUSINESSES"
79. "MOUNTAIN VIEW, Calif., April 1 /PRNewswire/ -- Netscape Communications and open source developers are celebrating the first anniversary, March 31, 1999, of the release of Netscape's browser source code to mozilla.org" (http://www.prnewswire.com/news-releases/netscape-celebrates-first-anniversary-of-open-source-software-release-to-mozillaorg-73806207.html). Netscape Communications. 1999-03-31. Retrieved 2013-01-10. "[...] The organization that manages open source developers working on the next generation of Netscape's browser and communication software. This event marked a historical milestone for the Internet as Netscape became the first major commercial software company to open its source code, a trend that has since been followed by several other corporations. Since the code was first published on the Internet, thousands of individuals and organizations have downloaded it and made hundreds of contributions to the software. Mozilla.org is now celebrating this one-year anniversary with a party Thursday night in San Francisco."
80. Proffitt, Brian (2000-10-13). "StarOffice Code Released in Largest Open Source Project" (https://web.archive.org/web/20131016014106/http://www.linuxtoday.com/developer/2000101300221NWDTSW). linuxtoday.com. Archived from the original (http://www.linuxtoday.com/developer/2000101300221NWDTSW) on 2013-10-16. Retrieved 2013-01-10. "Sun's joint effort with CollabNet kicked into high gear on the OpenOffice Web site at 5 a.m. PST this morning with the release of much of the source code for the upcoming 6.0 version of StarOffice. According to Sun, this release of 9 million lines of code under GPL is the beginning of the largest open source software project ever."
81. "Outreachy | Internships Supporting Diversity in Tech" (https://www.outreachy.org/). *www.outreachy.org*. Retrieved 2019-11-28.
82. Holtgrewe, Ursula (March 2004). "Articulating the Speed(s) of the Internet: The Case of Open Source/Free Software" (http://www.ssoar.info/ssoar/handle/document/12259). *Time & Society*. **13**: 129–146. doi:10.1177/0961463X04040750 (https://doi.org/10.1177%2F0961463X04040750). S2CID 61327593 (https://api.semanticscholar.org/CorpusID:61327593).
83. "Après un an d'activité, Liberapay veut être l'alternative libre pour les dons récurrents" (https://web.archive.org/web/20170916020622/https://www.nextinpact.com/news/104027-apres-an-dactivite-liberapay-veut-etre-alternative-libre-pour-dons-recurrents.htm) [A year after its launch, Liberapay wants to be the open source alternative for recurrent donations] (in French). 2017-04-13. Archived from the original (https://www.nextinpact.com/news/104027-apres-an-dactivite-liberapay-veut-etre-alternative-libre-pour-dons-recurrents.htm) on 2017-09-16.

84. Popp, Dr. Karl Michael; Meyer, Ralf (2010). *Profit from Software Ecosystems: Business Models, Ecosystems and Partnerships in the Software Industry* (https://books.google.com/books?id=i1VGDLCMyKAC). Norderstedt, Germany: Books on Demand. ISBN 9783839169834.
85. Wheeler, David A. (February 2009). "F/LOSS is Commercial Software" (http://timreview.ca/article/229). *Technology Innovation Management Review*. Talent First Network. Retrieved 18 June 2016.
86. Stallman, Richard (11 March 2012). *Richard Stallman (S20E10)* (https://www.youtube.com/watch?v=radmjL5OIaA&t=0h53m46s) (Podcast). The Linux Action Show. Jupiter Broadcasting. Event occurs at 0:53:46. Retrieved 18 June 2016. "I'm not going to claim that I got a way to make it easier to raise money to pay people who write free software. We all know, that to some extent there are ways to do that, but we all know that they are limited, they are not as broad as we would like."
87. Donovan, S. (6 August 2002). "Patent, copyright and trade secret protection for software". *IEEE Potentials*. **13** (3) (published 1994): 20. doi:10.1109/45.310923 (https://doi.org/10.1109%2F45.310923). ISSN 0278-6648 (https://search.worldcat.org/issn/0278-6648). S2CID 19873766 (https://api.semanticscholar.org/CorpusID:19873766). "Essentially there are only three ways to protect computer software under the law: patent it, register a copyright for it, or keep it as a trade secret."
88. Benkler, Yochai (April 2003). "Freedom in the Commons: Towards a Political Economy of Information" (https://web.archive.org/web/20110306041013/https://www.law.duke.edu/shell/cite.pl?52+Duke+L.+J.+1245+pdf). *Duke Law Journal*. **52** (6). Archived from the original (http://www.law.duke.edu/shell/cite.pl?52+Duke+L.+J.+1245+pdf) on 2011-03-06. Retrieved 2013-09-16.
89. ElBoghdady, Dina; Tsukayama, Hayley (2011-09-30). "Facebook tracking prompts calls for FTC investigation" (https://www.washingtonpost.com/business/economy/facebook-tracking-prompts-calls-for-ftc-investigation/2011/09/29/gIQAVdsP8K_story.html). *The Washington Post*. Retrieved 23 October 2011.
90. Cheng, Jacqui (10 January 2011). "VLC for iOS vanishes 2 months after eruption of GPL dispute" (https://arstechnica.com/gadgets/2011/01/vlc-for-ios-vanishes-2-months-after-eruption-of-gpl-dispute/). *Ars Technica*.
91. Vaughan-Nichols, Steven. "No GPL Apps for Apple's App Store" (https://web.archive.org/web/20110109000122/http://www.zdnet.com/blog/open-source/no-gpl-apps-for-apples-app-store/8046). *ZDNet*. Archived from the original (https://www.zdnet.com/blog/open-source/no-gpl-apps-for-apples-app-store/8046) on January 9, 2011. Retrieved 23 October 2011.
92. "Changing the VLC engine license to LGPL" (http://www.videolan.org/press/lgpl.html). Retrieved 23 October 2011.
93. Johnston, Casey (18 July 2013). "VLC media player returns to the iOS App Store after 30-month hiatus" (https://arstechnica.com/gadgets/2013/07/vlc-media-player-returns-to-the-ios-app-store-after-30-month-hiatus/). *Ars Technica*. Retrieved 10 October 2013.

Further reading

- Koenig, John (2004). "Seven Open Source Business Strategies for Commercial Advantage" (https://web.archive.org/web/20170112232640/http://riseforth.com/pdf/seven_open_source_business_strategies.pdf) (PDF). Riseforth. Archived from the original (http://riseforth.com/pdf/seven_open_source_business_strategies.pdf) (PDF) on 2017-01-12. Retrieved 2014-10-18.
- "Selling Free Software" (https://www.gnu.org/philosophy/selling.html). *Free Software Foundation*. 18 November 2016.

- Chang, Victor; Mills, Hugo; Newhouse, Steven (27 March 2014). "From Open Source to long-term sustainability: Review of Business Models and Case studies" (http://eprints.soton.ac.uk/263925/). *University of Southampton* (published 20 June 2007).
- Fink, Martin (2003). *The Business and Economics of Linux and Open Source* (https://archive.org/details/businesseconomic00fink). Prentice Hall Professional. ISBN 9780130476777.
- Dornan, Andy (2 January 2008). "The Five Open Source Business Models" (http://www.informationweek.com/applications/the-five-open-source-business-models/d/d-id/1062798?,). *InformationWeek*.
- Golden, Bernard (2005). "2: Open Source Business Models" (https://books.google.com/books?id=S4d9SzBjGIgC). *Succeeding with Open Source* (https://archive.org/details/succeedingwithop00bern). Boston: Addison-Wesley Professional. ISBN 9780321268532.
- Perens, Bruce (3 October 2005). "The emerging economic paradigm of Open Source" (http://firstmonday.org/ojs/index.php/fm/article/viewArticle/1470/1385). *First Monday* (2). doi:10.5210/fm.v0i0.1470 (https://doi.org/10.5210%2Ffm.v0i0.1470).
- Popp, Karl Michael (2015). "Open Source Business Models" (http://opensourcebestpractices.net/OpenSourceBusinessModels.html). *Open Source Best Practices*.
- Raya, Amadeu Albós; Martínez, Lluís Bru; Monsalve, Irene Fernández (September 2010). "Economic aspects and business models of Free Software" (http://ftacademy.org/sites/ftacademy.org/files/materials/fta-m5-economic_models.pdf) (PDF). Archived (https://web.archive.org/web/20160527144253/http://ftacademy.org/sites/ftacademy.org/files/materials/fta-m5-economic_models.pdf) (PDF) from the original on 27 May 2016.
- Tapscott, Don; Williams, Anthony D. (December 2006). *Wikinomics: How Mass Collaboration Changes Everything*. Portfolio. ISBN 9781591841388.

Free software movement

The **free software movement** is a social movement with the goal of obtaining and guaranteeing certain freedoms for software users, namely the freedoms to run, study, modify, and share copies of software.[1][2] Software which meets these requirements, The Four Essential Freedoms of Free Software, is termed free software.

Although drawing on traditions and philosophies among members of the 1970s hacker culture and academia, Richard Stallman formally founded the movement[3] in 1983 by launching the GNU Project.[4] Stallman later established the Free Software Foundation in 1985 to support the movement.

Philosophy

The philosophy of the Free Software Movement is based on promoting collaboration between programmers and computer users. This process necessitates the rejection of proprietary software and the promotion of free software.[5] Stallman notes that this action would not hinder the progression of technology, as he states, "Wasteful duplication of system programming effort will be avoided. This effort can go instead into advancing the state of the art."[6]

Members of the Free Software Movement believe that all software users should have the freedoms listed in The Free Software Definition. Members hold the belief that it is immoral to prohibit or prevent people from exercising these freedoms, and that they are required in creating a community where software users can help each other and have control over their technology.[7] Regarding proprietary software, some believe that it is not strictly immoral, citing increased profitability in the business models available for proprietary software, along with technical features and convenience.[8]

The Free Software Foundation espouses the principle that all software needs free documentation, as programmers should have the ability to update manuals to reflect modifications made to the software.[9] Within the movement, the FLOSS Manuals foundation specializes in providing such documentation.

Actions

Writing and spreading free software

The core work of the free software movement is focused on software development. The free software movement also rejects proprietary software, refusing to install software that does not give them the freedoms of free software. According to Stallman, "The only thing in the software field that is worse than an unauthorised copy of a proprietary program, is an authorised copy of the proprietary program because this does the same harm to its whole community of users, and in addition, usually the developer, the perpetrator of this evil, profits from it."[10]

Building awareness

GNU and Tux mascots around free software supporters at FISL 16

Some supporters of the free software movement take up public speaking, or host a stall at software-related conferences to raise awareness of software freedom. This is seen as important since people who receive free software, but who are not aware that it is free software, will later accept a non-free replacement or will add software that is not free software.[11]

Organisations

Asia

- Free Software Movement of India
- International Centre for Free and Open Source Software (ICFOSS)

Africa

- Free Software and Open Source Foundation for Africa

North America

- Free Software Foundation
- Software Freedom Law Center

South America

- Free Software Foundation Latin America
- Software Livre Brasil

Europe

- Free Software Foundation Europe
- Framasoft
- Irish Free Software Organisation

Australia

- Free Software Australia

Legislation and government

A lot of lobbying work has been done against software patents and expansions of copyright law. Other lobbying focuses directly on the use of free software by government agencies and government-funded projects.

Asia

China

In June 1997, the Society for Study, Application, and Development of Free Software was established under the China Software Industry Association in Beijing. Through this organization, the website freesoft.cei.gov.cn was developed, though the website is currently inaccessible on IP addresses located in the United States. The use of open-source software Linux in China has moved beyond government and educational institutions and has extended to other organizations such as financial institutions, telecommunications, and public security. Several Chinese researchers and scholars have claimed that the existence of FOSS in China has been important in challenging the presence of Microsoft, which Guangnan Ni, a member of the Chinese Academy of Engineering stated, "The monopoly of (Microsoft Windows) is even more powerful in China than other places in the world".[12] Yi Zhou, a professor of mathematics at Fudan University, has also alleged that, "Government procurement of FLOSS for a number of years in China has compelled Microsoft to cut its prices of Office software substantially" [12]

India

Government of India had issued Policy on Adoption of Open Source Software for Government of India in 2015 to drive uptake within the government. With the vision to transform India as a Software Product Nation, National Policy on Software Products-2019 was approved by the Government.[13]

Pakistan

Free and Open Source Software (Foss) is crucial for countries such as Pakistan which is set up by Union of Information Technology. For the case of Pakistan, Pakistan Software Export Board (PSEB) aids in the creation and advocate of FOSS usage in various government departments in addition to curbing illegality of copying that is software piracy. Promotion of adoption of FOSS is essential however it comes with problems of proprietary anti competition software practices including indulging in bribing and corruption by government departments. Pakistan works on the introduction of usage of open type basis of source Solutions in the curricula in schools and colleges. This is because of FOSS uniqueness in terms of political, democratic and social varieties of aspect regarding information communication and technology.[14]

North America

United States

In the United States, there have been efforts to pass legislation at the state level encouraging the use of free software by state government agencies.[15]

On January 11, 2022, two bills were shown on the New Hampshire legislating floor. The first bill called "HB 1273" was introduced by Democratic New Hampshire representative Eric Gallager, the bill prioritized "replacing proprietary software used by state agencies with free software." Gallager stated that to an extent, the proposed legislation will help distinguish "free software" and "open-source software", this will also put these two into state regulation. The second bill called "HB 1581" was proposed by Grafton Republican representative Lex Berezhny. The bill would've restored a requisite forcing "state agencies to use proprietary software" and as Lex put it, "when it is the most effective solution." He also said that requisite was happening between 2012 and 2018. According to the Concord Monitor, the state of New Hampshire had an already "thriving open source software community" with a view of "live free or die" but they had difficulty getting that notion with the state.[16]

South America

Peru

Congressmen Edgar David Villanueva and Jacques Rodrich Ackerman have been instrumental in introducing free software in Peru, with bill 1609 on "Free Software in Public Administration".[17] The incident invited the attention of Microsoft, Peru, whose general manager wrote a letter to Villanueva. His response received worldwide attention and is seen as a classic piece of argumentation favouring use of free software in governments.[18]

Uruguay

Uruguay has a sanctioned law requiring that the state give priority to free software. It also requires that information be exchanged in open formats.[19]

Venezuela

The Government of Venezuela implemented a free software law in January 2006. Decree No. 3,390 mandated all government agencies to migrate to free software over a two-year period.[20]

Europe

Publiccode.eu is a campaign launched demanding a legislation requiring that publicly financed software developed for the public sector be made publicly available under a Free and Open Source Software licence. If it is public money, it should be public code as well.[21]

France

The French Gendarmerie and the French National Assembly utilize the open source operating system Linux.[22]

United Kingdom

Gov.uk keeps a list of "key components, tools and services that have gone into the construction of GOV.UK".[23][title needed]

Events

Free Software events happening all around the world connects people to increase visibility for Free software projects and foster collaborations.

Economics

The free software movement has been extensively analyzed using economic methodologies, including perspectives from heterodox economics. Of particular interest to economists is the willingness of programmers in the free software movement to work, often producing higher-quality than proprietary programmers, without financial compensation.

In his 1998 article "The High-Tech Gift Economy", Richard Barbrook suggested that the then-nascent free software movement represented a return to the gift economy building on hobbyism and the absence of economic scarcity on the Internet.[24]

Gabriella Coleman has emphasized the importance of accreditation, respect, and honour within the free software community as a form of compensation for contributions to projects, over and against financial motivations.[25]

The Swedish Marxian economist Johan Söderberg has argued that the free software movement represents a complete alternative to capitalism that may be expanded to create a post-work society. He argues that the combination of a manipulation of intellectual property law and private property to make goods available to the public and a thorough blend between labor and fun make the free software movement a communist economy.[26]

Subgroups and schisms

Since its inception, there is an ongoing contention between the many FLOSS organizations (FSF, OSI, Debian, Mozilla Foundation, Apache Foundation, etc.) within the free software movement, with the main conflicts centered around the organization's needs for compromise and pragmatism rather than adhering to founding values and philosophies.[27]

Open source

The Open Source Initiative (OSI) was founded in February 1998 by Eric Raymond and Bruce Perens to promote the term "open-source software" as an alternative term for free software. The OSI aimed to address the perceived shortcomings and ambiguity of the term "free software", as well as shifting the focus of free software from a social and ethical issue to instead emphasize open source as a superior model for software development.[28][29][30][31] The latter became the view of Eric Raymond and Linus Torvalds, while Bruce Perens argued that open source was meant to popularize free software under a new brand and called for a return to basic ethical principles.[32]

Some free software advocates use the terms "Free and Open-Source Software" (FOSS) or "Free/Libre and Open-Source Software" (FLOSS) as a form of inclusive compromise, which brings free and open-source software advocates together to work on projects cohesively. Some users believe this is an ideal solution in order to promote both the user's freedom with the software and the pragmatic efficiency of an open-source development model. This view is reinforced by fact that majority of OSI-approved licenses and self-avowed open-source programs are also compatible with the free software formalisms and vice versa.[33]

While free and open source software are often linked together, they offer two separate ideas and values. Richard Stallman has referred to open source as "*a non-movement*", as it "*does not campaign for anything*".[34]

"Open source" addresses software being open as a practical question rather than an ethical dilemma – non-free software is not the best solution but nonetheless a solution. The free software movement views free software as a moral imperative: that proprietary software should be rejected, and that only free software should be developed and taught in order to make computing technology beneficial to the general public.[35]

Although the movements have differing values and goals, collaborations between the Free Software Movement and Open Source Initiative have taken place when it comes to practical projects.[36] By 2005, Richard Glass considered the differences to be a "serious fracture" but "vitally important to those on both sides of the fracture" and "of little importance to anyone else studying the movement from a software engineering perspective" since they have had "little effect on the field".[37]

Criticism and controversy

Principle compromises

Eric Raymond criticises the speed at which the free software movement is progressing, suggesting that temporary compromises should be made for long-term gains. Raymond argues that this could raise awareness of the software and thus increase the free software movement's influence on relevant standards and legislation.[38]

Richard Stallman, on the other hand, sees the current level of compromise as a greater cause for worry.[27][5][39]

Programmer income

Stallman said that this is where people get the misconception of "free": there is no wrong in programmers' requesting payment for a proposed project, or charging for copies of free software.[40] Restricting and controlling the user's decisions on use is the actual violation of freedom. Stallman defends that in some cases, monetary incentive is not necessary for motivation since the pleasure in expressing creativity is a reward in itself.[6] Conversely, Stallman admits that it is not easy to raise money for free software projects.[41]

"Viral" copyleft licensing

The free software movement champions copyleft licensing schema (often pejoratively called "viral licenses"). In its strongest form, copyleft mandates that any works *derived* from copyleft-licensed software must also carry a copyleft license, so the license spreads from work to work like a computer virus might spread from machine to machine. Stallman has previously stated his opposition to describing the GNU GPL as "viral". These licensing terms can only be enforced through asserting copyrights.[42]

Critics of copyleft licensing challenge the idea that restricting modifications is in line with the free software movement's emphasis on various "freedoms", especially when alternatives like MIT, BSD, and Apache licenses are more permissive.[43][44] Proponents enjoy the assurance that copylefted work cannot usually be incorporated into non-free software projects.[45] They emphasize that copyleft licenses may not attach for all uses and that in any case, developers can simply choose not to use copyleft-licensed software.[46][47]

License proliferation and compatibility

FLOSS license proliferation is a serious concern in the FLOSS domain due to increased complexity of license compatibility considerations which limits and complicates source code reuse between FLOSS projects.[48] The OSI and the FSF maintain their own lists of dozens of existing and acceptable FLOSS licenses.[49] There is an agreement among most that the creation of new licenses should be minimized and those created should be made compatible with the major existing FLOSS licenses. Therefore, there was a strong controversy around the update of the GNU GPLv2 to the GNU GPLv3 in 2007,[50][51] as the updated license is not compatible with the previous version.[52][53][54] Several projects (mostly of the open source faction[51] like the Linux kernel[55][56]) decided to not adopt the GPLv3 while almost all of the GNU project's packages adopted it.

See also

Free and open-source software portal

- GNU Manifesto
- History of free software
- Linux adoption
- Open-source movement
- Free-culture movement
- Free Software Foundation
- Open Source Initiative
- Software Freedom Conservancy
- Free Software Movement of India

- Free Software Foundation of India
- Free Software Foundation Europe
- Free Software Movement Karnataka
- Free Software Foundation Tamil Nadu
- Swecha
- Gift economy

References

1. "What is Free Software?" (https://www.gnu.org/philosophy/free-sw.html). *(gnu.org)*. Archived (https://web.archive.org/web/20131014132149/https://www.gnu.org/philosophy/free-sw.html) from the original on 2013-10-14. Retrieved 2020-03-18.
2. Richard Stallman on the nature of the Free software movement (http://lists.gnu.org/archive/html/emacs-devel/2008-03/msg00635.html) Archived (https://web.archive.org/web/20170513005715/http://lists.gnu.org/archive/html/emacs-devel/2008-03/msg00635.html) 2017-05-13 at the Wayback Machine in 2008 on emacs-devel mailing list.
3. Corrado, Edward M.; Moualison Sandy, Heather; Mitchell, Erik T. (2018-07-03). "Nullis in Verba: The Free Software Movement as a model for Openness and Transparency" (https://www.tandfonline.com/doi/full/10.1080/07317131.2018.1456849). *Technical Services Quarterly*. **35** (3): 269–279. doi:10.1080/07317131.2018.1456849 (https://doi.org/10.1080%2F07317131.2018.1456849). ISSN 0731-7131 (https://search.worldcat.org/issn/0731-7131). S2CID 196159979 (https://api.semanticscholar.org/CorpusID:196159979). Archived (https://web.archive.org/web/20230710173206/https://www.tandfonline.com/doi/full/10.1080/07317131.2018.1456849) from the original on 2023-07-10. Retrieved 2022-05-03.
4. Stallman, Richard (September 27, 1983). "Initial Announcement" (https://www.gnu.org/gnu/initial-announcement.html). *GNU Project*. Archived (https://web.archive.org/web/20090305002259/http://www.gnu.org/gnu/initial-announcement.html) from the original on 2009-03-05. Retrieved 2014-04-18.
5. Stallman, Richard. "The Free Software Community After 20 Years" (https://www.gnu.org/philosophy/use-free-software.html). *GNU*. Archived (https://web.archive.org/web/20210425024735/http://www.gnu.org/philosophy/use-free-software.html) from the original on 2021-04-25. Retrieved 2021-04-04.
6. "The GNU Manifesto" (https://www.gnu.org/gnu/manifesto.html). GNU. Archived (https://web.archive.org/web/20200511025028/http://www.gnu.org/gnu/manifesto.html) from the original on 2020-05-11. Retrieved 2014-04-18.
7. "Why free software?" (https://www.gnu.org/philosophy/why-free.html). GNU. Archived (https://web.archive.org/web/20210504231916/http://www.gnu.org/philosophy/why-free.html) from the original on 2021-05-04. Retrieved 2014-04-18.
8. "Copyleft: Pragmatic Idealism" (http://www.fsf.org/licensing/essays/pragmatic.html). GNU. Archived (https://web.archive.org/web/20120919112405/http://www.gnu.org/philosophy/pragmatic.html) from the original on 2012-09-19. Retrieved 2007-04-09.
9. "Free Software and Free Manuals" (https://www.gnu.org/philosophy/free-doc.html). GNU. Archived (https://web.archive.org/web/20210815064923/https://www.gnu.org/philosophy/free-doc.html) from the original on 2021-08-15. Retrieved 2014-04-18.
10. "Transcript of Stallman on Free Software" (http://fsfeurope.org/documents/rms-fs-2006-03-09.en.html). FSFE. 2006-03-09. Archived (https://web.archive.org/web/20090312070656/http://fsfeurope.org/documents/rms-fs-2006-03-09.en.html) from the original on 2009-03-12. Retrieved 2007-01-09.

11. "Transcript of Stallman speaking at WSIS" (https://web.archive.org/web/20081221131332/http://fsfe.org/en/fellows/ciaran/ciaran_s_free_software_notes/transcript_of_rms_at_wsis_on_is_free_open_source_software_the_answer#wsis--importance-of-awareness). Ciarán O'Riordan. Archived from the original (http://fsfe.org/en/fellows/ciaran/ciaran_s_free_software_notes/transcript_of_rms_at_wsis_on_is_free_open_source_software_the_answer#wsis--importance-of-awareness) on 2008-12-21. Retrieved 2006-10-07.
12. Zhou, Yi (2011). "Against Intellectual Monopoly: Free Software in China" (https://www.jstor.org/stable/41917739). *World Review of Political Economy*. **2** (2): 290–306. JSTOR 41917739 (https://www.jstor.org/stable/41917739). Archived (https://web.archive.org/web/20230407192053/https://www.jstor.org/stable/41917739) from the original on 2023-04-07. Retrieved 2023-04-07.
13. "FOSS4GOV Innovation Challenge" (https://innovateindia.mygov.in/foss4gov-innovation-challenge/). Archived (https://web.archive.org/web/20210827062916/https://innovateindia.mygov.in/foss4gov-innovation-challenge/) from the original on 2021-08-27. Retrieved 2021-08-27.
14. Rehman, Ata (July 2017). "Free and Open Source Software Movement in LIS Profession in Pakistan" (https://digitalcommons.unl.edu/libphilprac/852/). *University of Nebraska-Lincoln's Digital Commons*. Archived (https://web.archive.org/web/20230408150248/https://digitalcommons.unl.edu/libphilprac/852/) from the original on 2023-04-08. Retrieved 2023-04-08.
15. "Open source's new weapon: The law?" (https://www.cnet.com/tech/tech-industry/open-sources-new-weapon-the-law/). *CNET*. Archived (https://web.archive.org/web/20230316221116/https://www.cnet.com/tech/tech-industry/open-sources-new-weapon-the-law/) from the original on 2023-03-16. Retrieved 2023-03-16.
16. "Lots of recycled arguments when the legislature debates open source software" (https://www.concordmonitor.com/open-software-nh-new-hampshire-44505163). *Concord Monitor*. Archived (https://web.archive.org/web/20230409184759/https://www.concordmonitor.com/open-software-nh-new-hampshire-44505163) from the original on 2023-04-09. Retrieved 2023-04-09.
17. "An English translation of the Free Software bill proposed in Peru" (https://en.wikisource.org/wiki/Free_Software_in_Public_Agencies). Archived (https://web.archive.org/web/20101204201527/http://en.wikisource.org/wiki/Free_Software_in_Public_Agencies) from the original on 2010-12-04. Retrieved 2007-05-10.
18. "Peruvian Congressman Edgar Villanueva writing to Microsoft about free software" (https://web.archive.org/web/20070829215908/http://www.gnu.org.pe/resmseng.html). Archived from the original (http://www.gnu.org.pe/resmseng.html) on 2007-08-29.
19. "Group:Free Software in Government" (https://libreplanet.org/wiki/Group:Free_Software_in_Government#Uruguay). *libreplanet.org*. Archived (https://web.archive.org/web/20201023002248/https://libreplanet.org/wiki/Group:Free_Software_in_Government#Uruguay) from the original on 2020-10-23. Retrieved 2019-05-09.
20. "Free software liberates Venezuela" (https://web.archive.org/web/20160303214841/http://www.freesoftwaremagazine.com/articles/professional_services_venezuela/). Free Software Magazine n°10. 2006-02-08. Archived from the original (http://www.freesoftwaremagazine.com/articles/professional_services_venezuela/) on 2016-03-03. Retrieved 2007-03-16.
21. "Public Money? Public Code!" (https://publiccode.eu/). Archived (https://web.archive.org/web/20200219184118/https://publiccode.eu/) from the original on 2020-02-19. Retrieved 2021-08-27.
22. "AFP: French police deal blow to Microsoft" (https://web.archive.org/web/20140227230123/http://www.google.com/hostednews/afp/article/ALeqM5iU4Lq7tOR_WVOJLZ3IeRaIH03x6w?hl=en). 2014-02-27. Archived from the original (http://www.google.com/hostednews/afp/article/ALeqM5iU4Lq7tOR_WVOJLZ3IeRaIH03x6w?hl=en) on 2014-02-27. Retrieved 2019-05-12.

23. "Colophon for GOV.UK at launch | Government Digital Service" (https://archive.today/20141116114020/https://gds.blog.gov.uk/govuk-launch-colophon/). Archived from the original (https://gds.blog.gov.uk/govuk-launch-colophon/) on 2014-11-16. Retrieved 2014-11-16.
24. Barbrook, Richard (1998). "The High-Tech Gift Economy" (http://firstmonday.org/ojs/index.php/fm/article/view/631/552). *First Monday*. **13** (12). Archived (https://web.archive.org/web/20180722184805/http://firstmonday.org/ojs/index.php/fm/article/view/631/552) from the original on July 22, 2018. Retrieved July 22, 2018.
25. Coleman (2013), p. 116-7.
26. Söderberg (2007), p. 153-4.
27. Pragmatism in the History of GNU, Linux and Free/Open Source Software (https://web.archive.org/web/20160217000741/http://thevarguy.com/open-source-application-software-companies/060915/pragmatism-history-gnu-linux-and-freeopen-source-software) Jun 9, 2015 Christopher Tozzi
28. Eric S. Raymond. "Goodbye, "free software"; hello, "open source" " (http://www.catb.org/~esr/open-source.html). Archived (https://web.archive.org/web/20200102094841/http://www.catb.org/~esr/open-source.html) from the original on 2020-01-02. Retrieved 2016-03-22. "The problem with it is twofold. First, ... the term "free" is very ambiguous ... Second, the term makes a lot of corporate types nervous."
29. Kelty, Christpher M. (2008). "The Cultural Significance of free Software – Two Bits" (http://twobits.net/pub/Kelty-TwoBits.pdf) (PDF). Durham and London: Duke University Press. p. 99. Archived (https://web.archive.org/web/20080827175442/http://twobits.net/pub/Kelty-TwoBits.pdf) (PDF) from the original on 2008-08-27. Retrieved 2016-03-22. "Prior to 1998, Free Software referred either to the Free Software Foundation (and the watchful, micromanaging eye of Stallman) or to one of thousands of different commercial, avocational, or university-research projects, processes, licenses, and ideologies that had a variety of names: sourceware, freeware, shareware, open software, public domain software, and so on. The term Open Source, by contrast, sought to encompass them all in one movement."
30. Shea, Tom (1983-06-23). "Free software – Free software is a junkyard of software spare parts" (https://books.google.com/books?id=yy8EAAAAMBAJ&q=us%20government%20public%20domain%20software&pg=PA31). *InfoWorld*. Retrieved 2016-02-10. "In contrast to commercial software is a large and growing body of free software that exists in the public domain. Public-domain software is written by microcomputer hobbyists (also known as 'hackers') many of whom are professional programmers in their work life. [...] Since everybody has access to source code, many routines have not only been used but dramatically improved by other programmers."
31. "Open Source misses the point" (https://www.gnu.org/philosophy/open-source-misses-the-point.html). GNU. Archived (https://web.archive.org/web/20110804231811/http://www.gnu.org/philosophy/open-source-misses-the-point.html) from the original on 2011-08-04. Retrieved 2014-04-18.
32. Bruce Perens (17 February 1999). "It's Time to Talk About Free Software Again" (https://web.archive.org/web/20140716055445/https://lists.debian.org/debian-devel/1999/02/msg01641.html). Archived from the original (https://lists.debian.org/debian-devel/1999/02/msg01641.html) on 16 July 2014. Retrieved 2 April 2015.
33. Stallman, Richard. "Why Open Source Misses the Point of Free Software" (https://www.gnu.org/philosophy/open-source-misses-the-point.html). *GNU Operating System*. Free Software Foundation. Archived (https://web.archive.org/web/20110804231811/http://www.gnu.org/philosophy/open-source-misses-the-point.html) from the original on 4 August 2011. Retrieved 11 February 2013.
34. Gillin, Paul (2016-04-28). "GNU founder Stallman: 'Open source is not free software' " (https://siliconangle.com/blog/2016/04/28/gnu-founder-stallman-open-source-is-not-free-software/). *SiliconANGLE*. Archived (https://web.archive.org/web/20171017145802/https://siliconangle.com/blog/2016/04/28/gnu-founder-stallman-open-source-is-not-free-software/) from the original on 2017-10-17. Retrieved 2017-10-17.

35. Stallman, Richard. "Why 'Open Source' Misses the Point of Free Software | June 2009 | Communications of the ACM" (https://cacm.acm.org/magazines/2009/6/28491-why-open-source-misses-the-point-of-free-software). *cacm.acm.org*. Retrieved 2017-10-17.
36. "Why "Free Software" is better than "Open Source" " (https://www.gnu.org/philosophy/free-software-for-freedom.html). GNU. Archived (https://web.archive.org/web/20210327080246/https://www.gnu.org/philosophy/free-software-for-freedom.html) from the original on 2021-03-27. Retrieved 2014-04-18.
37. Richard Glass (2005), "Standing in Front of the Open Source Steamroller", in Joseph Feller; Brian Fitzgerald; Scott A. Hissam; Karim R. Lakahani (eds.), *Perspectives on Free and Open Source Software*, MIT Press, p. 89, ISBN 0262062461
38. Eric S. Raymond (2006-07-01). "ESR's "World Domination 201", on the need for more compromise by the free software movement" (http://www.catb.org/~esr/writings/world-domination/world-domination-201.html). catb.org. Archived (https://web.archive.org/web/20190905031807/http://catb.org/~esr/writings/world-domination/world-domination-201.html) from the original on 2019-09-05. Retrieved 2016-02-11.
39. "Richard Stallman on "World Domination 201" " (https://web.archive.org/web/20130603124006/http://www.libervis.com/article/richard_stallman_on_world_domination_201). Archived from the original (http://www.libervis.com/article/richard_stallman_on_world_domination_201) on 2013-06-03. Retrieved 2008-01-31. "I cannot agree to that compromise, and my experience teaches me that it won't be temporary. ... What our community needs most is more spine in rejection of non-free software. It has far too much willingness to compromise. ... To "argue" in favor of adding non-free software in GNU/Linux distros is almost superfluous, since that's what nearly all of them have already done."
40. "Selling Free Software" (https://www.gnu.org/philosophy/selling.html). *GNU*. Archived (https://web.archive.org/web/20180207222030/http://www.gnu.org/philosophy/selling.html) from the original on 2018-02-07. Retrieved 2021-04-04.
41. "Interview with Richard Stallman" (https://www.youtube.com/watch?v=radmjL5OIaA&t=0h53m46s). *GNU/LAS s20e10*. Linux action show. 2012-03-11. Archived (https://ghostarchive.org/varchive/youtube/20211211/radmjL5OIaA) from the original on 2021-12-11. Retrieved 2014-08-22. "RMS: *I'm not gone to claim that I got a way to make it easier to raise money to pay people who write free software. We all know, that to some extent there are ways to do that, but we all know that they are limited, they are not as broad as we would like.*"
42. 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*, MIT Press, p. 382, ISBN 0-262-06246-1
43. "Open Source Licensing Guide" (http://www.newmediarights.org/open_source/new_media_rights_open_source_licensing_guide). *New Media Rights*. Archived (https://web.archive.org/web/20150213052820/http://www.newmediarights.org/open_source/new_media_rights_open_source_licensing_guide) from the original on 13 February 2015. Retrieved 13 February 2015.
44. Newbart, Dave (2001-06-01). "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 2001-06-15.(Internet archive link)
45. 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.
46. Byfield, Bruce (2006-08-29). "IT Manager's Journal: 10 Common Misunderstandings About the GPL" (http://www.linuxtoday.com/developer/2006082902126OSHLLL). Archived (https://web.archive.org/web/20200216180035/https://www.linuxtoday.com/developer/2006082902126oshlll) from the original on 2020-02-16. Retrieved 2008-08-23.

47. 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.
48. OSI and License Proliferation (https://fossbazaar.org/content/osi-and-license-proliferation/) Archived (https://web.archive.org/web/20160220081208/https://fossbazaar.org/content/osi-and-license-proliferation/) 2016-02-20 at the Wayback Machine on fossbazar.com by Martin Michlmayr *"Too many different licenses makes it difficult for licensors to choose: it's difficult to choose a good license for a project because there are so many. Some licenses do not play well together: some open source licenses do not inter-operate well with other open source licenses, making it hard to incorporate code from other projects. Too many licenses makes it difficult to understand what you are agreeing to in a multi-license distribution: since a FLOSS application typically contains code with different licenses and people use many applications which each contain one or several licenses, it's difficult to see what your obligations are."* (on August 21st, 2008)
49. "Various Licenses and Comments about Them" (https://www.gnu.org/licenses/license-list.html). *GNU*. Archived (https://web.archive.org/web/20100724023833/https://www.gnu.org/licenses/license-list.html) from the original on 2010-07-24. Retrieved 2021-04-04.
50. Mark (2008-05-08). "The Curse of Open Source License Proliferation" (https://web.archive.org/web/20151208112000/http://socializedsoftware.com/2008/05/08/the-curse-of-open-source-license-proliferation/). socializedsoftware.com. Archived from the original (http://socializedsoftware.com/2008/05/08/the-curse-of-open-source-license-proliferation/) on 2015-12-08. Retrieved 2015-11-30. "Currently the decision to move from GPL v2 to GPL v3 is being hotly debated by many open source projects. According to Palamida, a provider of IP compliance software, there have been roughly 2489 open source projects that have moved from GPL v2 to later versions."
51. McDougall, Paul (2007-07-10). "Linux Creator Calls GPLv3 Authors 'Hypocrites' As Open Source Debate Turns Nasty" (https://web.archive.org/web/20080413091038/http://www.informationweek.com/blog/main/archives/2007/07/linux_creator_c.html). informationweek.com. Archived from the original (http://www.informationweek.com/blog/main/archives/2007/07/linux_creator_c.html) on 2008-04-13. Retrieved 2015-02-12. "[...]the latest sign of a growing schism in the open source community between business-minded developers like Torvalds and free software purists."
52. "Frequently Asked Questions about the GNU Licenses – Is GPLv3 compatible with GPLv2?" (https://www.gnu.org/licenses/gpl-faq.html#v2v3Compatibility). GNU. Archived (https://web.archive.org/web/20200330051354/http://www.gnu.org/licenses/gpl-faq.html#v2v3Compatibility) from the original on 30 March 2020. Retrieved 3 June 2014. "No. Some of the requirements in GPLv3, such as the requirement to provide Installation Information, do not exist in GPLv2. As a result, the licenses are not compatible: if you tried to combine code released under both these licenses, you would violate section 6 of GPLv2. However, if code is released under GPL "version 2 or later," that is compatible with GPLv3 because GPLv3 is one of the options it permits."
53. Larabel, Michael (24 January 2013). "FSF Wastes Away Another "High Priority" Project" (https://web.archive.org/web/20161109200145/https://www.phoronix.com/scan.php?page=news_item&px=MTI4Mjc). Phoronix. Archived from the original (https://www.phoronix.com/scan.php?page=news_item&px=MTI4Mjc) on 9 November 2016. Retrieved 22 August 2013. "*Both LibreCAD and FreeCAD both want to use LibreDWG and have patches available for supporting the DWG file format library, but can't integrate them. The programs have dependencies on the popular GPLv2 license while the Free Software Foundation will only let LibreDWG be licensed for GPLv3 use, not GPLv2.*"
54. Chisnall, David (2009-08-31). *The Failure of the GPL* (http://www.informit.com/articles/article.aspx?p=1390172&seqNum=3). informit.com. Archived (https://web.archive.org/web/20160124132254/http://www.informit.com/articles/article.aspx?p=1390172&seqNum=3) from the original on 2016-01-24. Retrieved 2016-01-24.

55. Kerner, Sean Michael (2008-01-08). "Torvalds Still Keen On GPLv2" (http://www.internetnews.com/dev-news/article.php/3720371/Torvalds+Still+Keen+On+GPLv2.htm). internetnews.com. Archived (https://web.archive.org/web/20150212130610/http://www.internetnews.com/dev-news/article.php/3720371/Torvalds+Still+Keen+On+GPLv2.htm) from the original on 2015-02-12. Retrieved 2015-02-12. "*"In some ways, Linux was the project that really made the split clear between what the FSF is pushing which is very different from what open source and Linux has always been about, which is more of a technical superiority instead of a -- this religious belief in freedom," Torvalds told Zemlin. So, the GPL Version 3 reflects the FSF's goals and the GPL Version 2 pretty closely matches what I think a license should do and so right now, Version 2 is where the kernel is."*"
56. corbet (2006-10-01). "Busy busy busybox" (https://lwn.net/Articles/202106/). lwn.net. Archived (https://web.archive.org/web/20160107184000/https://lwn.net/Articles/202106/) from the original on 2016-01-07. Retrieved 2015-11-21. "*Since BusyBox can be found in so many embedded systems, it finds itself at the core of the GPLv3 anti-DRM debate. [...]The real outcomes, however, are this: BusyBox will be GPLv2 only starting with the next release. It is generally accepted that stripping out the "or any later version" is legally defensible, and that the merging of other GPLv2-only code will force that issue in any case*"

Further reading

- Coleman, E. Gabriella (2013). *Coding Freedom: The Ethics and Aesthetics of Hacking*. Princeton: Princeton University Press. ISBN 978-0691144610.
- David M. Berry, *Copy, Rip, Burn: The Politics of Copyleft and Open Source*, Pluto Press, 2008, ISBN 0-7453-2414-2
- Johan Söderberg, *Hacking Capitalism: The Free and Open Source Software Movement*, Routledge, 2007, ISBN 0-415-95543-2

External links

- The Free Software Movement and the Future of Freedom (https://archive.org/details/Stallman_Free_Software_Movement_and_Future_of_Freedom), a 2006 lecture by Richard Stallman
- Free Software Movement intro by FSF (https://www.gnu.org/philosophy/free-software-intro.html)
- The GNU Project Philosophy Directory (https://www.gnu.org/philosophy/), containing many defining documents of the free software movement
- An interview with Stallman, "Free Software as a social movement" (http://arquivo.pt/wayback/20090701040338/http://www.zmag.org/znet/viewArticle/4795)
- Christian Imhorst, *Anarchy and Source Code – What does the Free Software Movement have to do with Anarchism?* (http://www.imhorst.net/translations/anarchy-and-source-code/), 2005

History of free and open-source software

(Redirected from History of free software)

The **history of free and open-source software** begins at the advent of computer software in the early half of the 20th century. In the 1950s and 1960s, computer operating software and compilers were delivered as a part of hardware purchases without separate fees. At the time, source code—the human-readable form of software—was generally distributed with the software, providing the ability to fix bugs or add new functions. Universities were early adopters of computing technology. Many of the modifications developed by universities were openly shared, in keeping with the academic principles of sharing knowledge, and organizations sprung up to facilitate sharing.

As large-scale operating systems matured, fewer organizations allowed modifications to the operating software, and eventually such operating systems were closed to modification. However, utilities and other added-function applications are still shared and new organizations have been formed to promote the sharing of software.

Sharing techniques before software

The concept of free sharing of technological information existed long before computers. For example, in the early years of automobile development, one enterprise owned the rights to a 2-cycle gasoline engine patent originally filed by George B. Selden.[1] By controlling this patent, they were able to monopolize the industry and force car manufacturers to adhere to their demands, or risk a lawsuit. In 1911, independent automaker Henry Ford won a challenge to the Selden patent. The result was that the Selden patent became virtually worthless and a new association (which would eventually become the Motor Vehicle Manufacturers Association) was formed.[1] The new association instituted a cross-licensing agreement among all US auto manufacturers: although each company would develop technology and file patents, these patents were shared openly and without the exchange of money between all the manufacturers.[1] By the time the US entered World War 2, 92 Ford patents and 515 patents from other companies were being shared between these manufacturers, without any exchange of money (or lawsuits).[1]

Free software before the 1980s

Computer software was created in the early half of the 20th century.[2][3][4] In the 1950s and into the 1960s, almost all softwares were produced by academics and corporate researchers working in collaboration,[5] often shared as public-domain software. As such, it was generally distributed under the principles of openness and cooperation long established in the fields of academia, and was not seen as a commodity in itself. Such communal behavior later became a central element of the so-called hacking culture (a term with a positive connotation among free software programmers).

Computer operating software and compilers were delivered as a part of hardware purchases without separate fees. At this time, source code, the human-readable form of software, was generally distributed with the software machine code because users frequently modified the software themselves, because it would not run on different hardware or OS without modification, and also to fix bugs or add new functions.[6][7][8] The first example of free and open-source software is believed to be the A-2 system, developed at the UNIVAC division of Remington Rand in 1953,[9] which was released to customers with its source code. They were invited to send their improvements back to UNIVAC.[10] Later, almost all IBM mainframe software was also distributed with source code included. User groups such as that of the IBM 701, called SHARE, and that of Digital Equipment Corporation (DEC), called DECUS, were formed to facilitate the exchange of software. The SHARE Operating System, originally developed by General Motors, was distributed by SHARE for the IBM 709 and 7090 computers. Some university computer labs even had a policy requiring that all programs installed on the computer had to come with published source-code files.[11]

In 1969 the Advanced Research Projects Agency Network (ARPANET), a transcontinental, high-speed computer network was constructed. The network (later succeeded by the Internet) simplified the exchange of software code.[6]

Some free software which was developed in the 1970s continues to be developed and used, such as TeX (developed by Donald Knuth)[12] and SPICE.[13]

Initial decline of free software

By the late 1960s change was coming: as operating systems and programming language compilers evolved, software production costs were dramatically increasing relative to hardware. A growing software industry was competing with the hardware manufacturers' bundled software products (the cost of bundled products was included in the hardware cost), leased machines required software support while providing no revenue for software, and some customers, able to better meet their own needs,[14] did not want the costs of the manufacturer's software to be bundled with hardware product costs. In the *United States vs. IBM* antitrust suit, filed 17 January 1969, the U.S. government charged that bundled software was anticompetitive.[15] While some software continued to come at no cost, there was a growing amount of software that was for sale only under restrictive licenses.

In the early 1970s AT&T distributed early versions of Unix at no cost to the government and academic researchers, but these versions did not come with permission to redistribute or to distribute modified versions, and were thus not free software in the modern meaning of the phrase. After Unix became more widespread in the early 1980s, AT&T stopped the free distribution and charged for system patches. As it is quite difficult to switch to another architecture, most researchers paid for a commercial license.

Software was not considered copyrightable before the 1974 US Commission on New Technological Uses of Copyrighted Works (CONTU) decided that "computer programs, to the extent that they embody an author's original creation, are proper subject matter of copyright".[16][17] Therefore, software had no licenses attached and was shared as public-domain software, typically with source code. The CONTU decision plus later court decisions such as *Apple v. Franklin* in 1983 for object code, gave computer programs the copyright status of literary works and started the licensing of software and the shrink-wrap closed source software business model.[18]

In the late 1970s and early 1980s, computer vendors and software-only companies began routinely charging for software licenses, marketing software as "Program Products" and imposing legal restrictions on new software developments, now seen as assets, through copyrights, trademarks, and leasing contracts. In 1976 Bill Gates wrote an essay entitled "Open Letter to Hobbyists", in which he expressed dismay at the widespread sharing of Microsoft's product Altair BASIC by hobbyists without paying its licensing fee. In 1979, AT&T began to enforce its licenses when the company decided it might profit by selling the Unix system.[19] In an announcement letter dated 8 February 1983 IBM inaugurated a policy of no longer distributing sources with purchased software.[20][21]

To increase revenues, a general trend began to no longer distribute source code (easily readable by programmers), and only distribute the executable machine code that was compiled from the source code. One person especially distressed by this new practice was Richard Stallman. He was concerned that he could no longer study or further modify programs initially written by others. Stallman viewed this practice as ethically wrong. In response, he founded the GNU Project in 1983 so that people could use computers using only free software.[8] He established a non-profit organization, the Free Software Foundation, in 1985, to more formally organize the project. He invented copyleft, a legal mechanism to preserve the "free" status of a work subject to copyright, and implemented this in the GNU General Public License. Copyleft licenses allow authors to grant a number of rights to users (including rights to use a work without further charges, and rights to obtain, study and modify the program's complete corresponding source code) but requires derivatives to remain under the same license or one without any additional restrictions. Since derivatives include combinations with other original programs, downstream authors are prevented from turning the initial work into proprietary software, and invited to contribute to the copyleft commons.[6] Later, variations of such licenses were developed by others.

1980s and 1990s

Informal software sharing continues

However, there were still those who wished to share their source code with other programmers and/or with users on a free basis, then called "hobbyists" and "hackers".[22] Before the introduction and widespread public use of the internet, there were several alternative ways available to do this, including listings in computer magazines (like Dr. Dobb's Journal, *Creative Computing*, *SoftSide*, *Compute!*, *Byte*, etc.) and in computer programming books, like the bestseller *BASIC Computer Games*.[23] Though still copyrighted, annotated source code for key components of the system software for Atari 8-bit computers was published in mass market books, including *The Atari BASIC Source Book*[24] (full source for Atari BASIC) and *Inside Atari DOS* (full source for Atari DOS).[25]

SHARE program library

The SHARE users group, founded in 1955, began collecting and distributing free software. The first documented distribution from SHARE was dated 17 October 1955.[26] The "SHARE Program Library Agency" (SPLA) distributed information and software, notably on magnetic tape.

DECUS tapes

In the early 1980s, the so-called *DECUS tapes*[27] were a worldwide system for the transmission of free software for users of DEC equipment. Operating systems were usually proprietary software, but many tools like the *TECO* editor, *Runoff* text formatter, or *List* file listing utility, etc. were developed to make users' lives easier, and distributed on the DECUS tapes. These utility packages benefited DEC, which sometimes incorporated them into new releases of their proprietary operating system. Even compilers could be distributed and for example Ratfor (and Ratfiv) helped researchers to move from Fortran coding to structured programming (suppressing the GO TO statement). The 1981 Decus tape was probably the most innovative by bringing the Lawrence Berkeley Laboratory *Software Tools Virtual Operating System* which permitted users to use a Unix-like system on DEC 16-bit PDP-11s and 32-bit VAXes running under the VMS operating system. It was similar to the current cygwin system for Windows. Binaries and libraries were often distributed, but users usually preferred to compile from source code.

Online software sharing communities in the 1980s

In the 1980s, parallel to the free software movement, software with source code was shared on BBS networks. This was sometimes a necessity; software written in BASIC and other interpreted languages could only be distributed as source code, and much of it was freeware. When users began gathering such source code, and setting up boards specifically to discuss its modification, this was a de facto open-source system.

One of the most obvious examples of this is one of the most-used BBS systems and networks, WWIV, developed initially in BASIC by Wayne Bell. A culture of "modding" his software, and distributing the mods, grew up so extensively that when the software was ported to first Pascal, then C++, its source code continued to be distributed to registered users, who would share mods and compile their own versions of the software. This may have contributed to it being a dominant system and network, despite being outside the Fidonet umbrella that was shared by so many other BBS makers.

Meanwhile, the advent of Usenet and UUCPNet in the early 1980s further connected the programming community and provided a simpler way for programmers to share their software and contribute to software others had written.[28]

Launch of the free software movement

In 1983, Richard Stallman launched the GNU Project to write a complete operating system free from constraints on use of its source code. Particular incidents that motivated this include a case where an annoying printer couldn't be fixed because the source code was withheld from users.[29] Stallman also published the GNU Manifesto in 1985 to outline the GNU Project's purpose and explain the importance of free software. Another probable inspiration for the GNU project and its manifesto was a disagreement between Stallman and Symbolics, Inc. over MIT's access to updates Symbolics had made to its Lisp machine, which was based on MIT code.[30] Soon after the launch, he[22] used the existing term "free software" and founded the Free Software Foundation to promote the concept. *The Free Software Definition* was published in February 1986.

In 1989, the first version of the GNU General Public License was published.[31] A slightly updated version 2 was published in 1991. In 1989, some GNU developers formed the company Cygnus Solutions.[32] The GNU project's kernel, later called "GNU Hurd", was continually delayed, but most other components were completed by 1991. Some of these, especially the GNU Compiler Collection, had become market leaders in their own right. The GNU Debugger and GNU Emacs were also notable successes.

Linux (1991–present)

The Linux kernel, started by Linus Torvalds, was released as freely modifiable source code in 1991. The license was not a free software license, but with version 0.12 in February 1992, Torvalds relicensed the project under the GNU General Public License.[33] Much like Unix, Torvalds' kernel attracted attention from volunteer programmers.

Until this point, the GNU project's lack of a kernel meant that no complete free software operating systems existed. The development of Torvalds' kernel closed that last gap. The combination of the almost-finished GNU operating system and the Linux kernel made the first complete free software operating system.

Among Linux distributions, Debian GNU/Linux, begun by Ian Murdock in 1993, is noteworthy for being explicitly committed to the GNU and FSF principles of free software. The Debian developers' principles are expressed in the Debian Social Contract. Since its inception, the Debian project has been closely linked with the FSF, and in fact was sponsored by the FSF for a year in 1994–1995. In 1997, former Debian project leader Bruce Perens also helped found Software in the Public Interest, a non-profit funding and support organization for various free software projects.[34]

Since 1996, the Linux kernel has included proprietary licensed components, so that it was no longer entirely free software.[35] Therefore, the Free Software Foundation Latin America released in 2008 a modified version of the Linux-kernel called Linux-libre, where all proprietary and non-free components were removed.

Many businesses offer customized Linux-based products, or distributions, with commercial support. The naming remains controversial. Referring to the complete system as simply "Linux" is common usage. However, the Free Software Foundation, and many others, advocate the use of the term "GNU/Linux", saying that it is a more accurate name for the whole operating system.[36]

Linux adoption grew among businesses and governments in the 1990s and 2000s. In the English-speaking world at least, Ubuntu and its derivatives became a relatively popular group of Linux distributions.

The free BSDs (1993–present)

When the *USL v. BSDi* lawsuit was settled out of court in 1993, FreeBSD and NetBSD (both derived from 386BSD) were released as free software. In 1995, OpenBSD forked from NetBSD. In 2004, Dragonfly BSD forked from FreeBSD.

The dot-com years (late 1990s)

In the mid to late 90s, when many website-based companies were starting up, free software became a popular choice for web servers. The Apache HTTP Server became the most-used web-server software, a title that still holds as of 2015.[37] Systems based on a common "stack" of software with the Linux kernel at the base, Apache providing web services, the MySQL database engine for data storage, and the PHP programming language for providing dynamic pages, came to be termed LAMP systems. In actuality, the programming language that predated PHP and dominated the web in the mid and late 1990s was Perl. Web forms were processed on the server side through Common Gateway Interface scripts written in Perl.

The term "open source," as related to free software, was in common use by 1995.[38] Other recollection have it in use during the 1980s.[39]

The launch of Open Source

In 1997, Eric S. Raymond published "The Cathedral and the Bazaar", a reflective analysis of the hacker community and free software principles. The paper received significant attention in early 1998 and was one factor in motivating Netscape Communications Corporation to release their popular Netscape Communicator Internet suite as free software.[40]

Netscape's act prompted Raymond and others to look into how to bring free software principles and benefits to the commercial-software industry. They concluded that FSF's social activism was not appealing to companies like Netscape, and looked for a way to rebrand the free software movement to emphasize the business potential of the sharing of source code.[41]

The label "open source" was adopted by some people in the free software movement at a strategy session[42] held at Palo Alto, California, in reaction to Netscape's January 1998 announcement of a source code release for Navigator. The group of individuals at the session included Christine Peterson who suggested "open source",[8] Todd Anderson, Larry Augustin, Jon Hall, Sam Ockman, Michael Tiemann, and Eric S. Raymond. Over the next week, Raymond and others worked on spreading the word. Linus Torvalds gave an all-important sanction the following day. Phil Hughes offered a pulpit in *Linux Journal*. Richard Stallman, pioneer of the free software movement, flirted with adopting the term, but changed his mind.[42] Those people who adopted the term used the opportunity before the release of Navigator's source code to free themselves of the ideological and confrontational connotations of the term "free software". Netscape released its source code under the Netscape Public License and later under the Mozilla Public License.[43]

The term was given a big boost at an event organized in April 1998 by technology publisher Tim O'Reilly. Originally titled the "Freeware Summit" and later named the "Open Source Summit",[44] the event brought together the leaders of many of the most important free and open-source projects, including Linus Torvalds, Larry Wall, Brian Behlendorf, Eric Allman, Guido van Rossum, Michael Tiemann, Paul Vixie, Jamie Zawinski of Netscape, and Eric Raymond. At that meeting, the confusion caused by the name free software was brought up. Tiemann argued for "sourceware" as a new term, while Raymond argued for "open source". The assembled developers took a vote, and the winner was announced at a press conference that evening. Five days later, Raymond made the first public call to the free software community to adopt the new term.[45] The Open Source Initiative was formed shortly thereafter.[8][42] According to the OSI Richard Stallman initially flirted with the idea of adopting the open source term.[46] But as the enormous success of the open source term buried Stallman's free software term and his message on social values and computer users' freedom,[47][48][49] later Stallman and his FSF strongly objected to the OSI's approach and terminology.[50] Due to Stallman's rejection of the term "open-source software", the FOSS ecosystem is divided in its terminology; see also Alternative terms for free software.

For example, a 2002 FOSS developer survey revealed that 32.6% associated themselves with OSS, 48% with free software, and 19.4% in between or undecided.[51] Stallman still maintained, however, that users of each term were allies in the fight against proprietary software.

On 13 October 2000, Sun Microsystems released[52] the StarOffice office suite as free software under the GNU Lesser General Public License. The free software version was renamed OpenOffice.org, and coexisted with StarOffice.

By the end of the 1990s, the term "open source" gained much traction in public media[53] and acceptance in software industry in context of the dotcom bubble and the open-source software driven Web 2.0.

Desktop (1984–present)

The X Window System was created in 1984, and became the de facto standard window system in desktop free software operating systems by the mid-1990s. X runs as a server, and is responsible for communicating with graphics hardware on behalf of clients (which are individual software applications). It provides useful services such as having multiple virtual desktops for the same monitor, and transmitting visual data across the network so a desktop can be accessed remotely.

A historical example of graphical user interface and applications common to the MIT X Consortium's distribution running under the twm window manager: X Terminal, Xbiff, xload and a graphical manual page browser

Initially, users or system administrators assembled their own environments from X and available window managers (which add standard controls to application windows; X itself does not do this), pagers, docks and other software. While X can be operated without a window manager, having one greatly increases convenience and ease of use.

Two key "heavyweight" desktop environments for free software operating systems emerged in the 1990s that were widely adopted: KDE and GNOME. KDE was founded in 1996 by Matthias Ettrich. At the time, he was troubled by the inconsistencies in the user interfaces of UNIX applications. He proposed a new desktop environment. He also wanted to make this desktop easy to use. His initial Usenet post spurred a lot of interest.[54]

Ettrich chose to use the Qt toolkit for the KDE project. At the time, Qt did not use a free software license. Members of the GNU project became concerned with the use of such a toolkit for building a free software desktop environment. In August 1997, two projects were started in response to KDE: the Harmony toolkit (a free replacement for the Qt libraries) and GNOME (a different desktop without Qt and built entirely on top of free software).[55] GTK+ was chosen as the base of GNOME in place of the Qt toolkit.

In November 1998, the Qt toolkit was licensed under the free/open source Q Public License (QPL) but debate continued about compatibility with the GNU General Public License (GPL). In September 2000, Trolltech made the Unix version of the Qt libraries available under the GPL, in addition to the QPL, which has eliminated the concerns of the Free Software Foundation. KDE has since been split into KDE Plasma Workspaces, a desktop environment, and KDE Software Compilation, a much broader set of software that includes the desktop environment.

Both KDE and GNOME now participate in freedesktop.org, an effort launched in 2000 to standardize Unix desktop interoperability, although there is still competition between them.[56]

Since 2000, software written for X almost always uses some widget toolkit written on top of X, like Qt or GTK.

In 2010, Canonical released the first version of Unity, a replacement for the prior default desktop environment for Ubuntu, GNOME. This change to a new, under-development desktop environment and user interface was initially somewhat controversial among Ubuntu users.

In 2011, GNOME 3 was introduced, which largely discarded the desktop metaphor in favor of a more mobile-oriented interface. The ensuing controversy led Debian to consider making the Xfce environment default on Debian 7. Several independent projects were begun to keep maintaining the GNOME 2 code.

Fedora did not adopt Unity, retaining its existing offering of a choice of GNOME, KDE and LXDE with GNOME being the default, and hence Red Hat Enterprise Linux (for which Fedora acts as the "initial testing ground") did not adopt Unity either. A fork of Ubuntu was made by interested third-party developers that kept GNOME and discarded Unity. In March 2017, Ubuntu announced that it will be abandoning Unity in favour of GNOME 3 in future versions, and ceasing its efforts in developing Unity-based smartphones and tablets.[57][58]

When Google built the Linux-based Android operating system, mostly for phone and tablet devices, it replaced X with the purpose-built SurfaceFlinger.

Open-source developers also criticized X as obsolete, carrying many unused or overly complicated elements in its protocol and libraries, while missing modern functionality, e.g., compositing, screen savers, and functions provided by window managers.[59] Several attempts have been made or are underway to replace X for these reasons, including:

- The Y Window System, which had ceased development by 2006.[60]
- The Wayland project, started in 2008.
- The Mir project, started in 2013 by Canonical Ltd. to produce a replacement windowing system for Ubuntu.

Microsoft, SCO and other attacks (1998–2014)

As free software became more popular, industry incumbents such as Microsoft started to see it as a serious threat. This was shown in a leaked 1998 document, confirmed by Microsoft as genuine, which came to be called the first of the Halloween Documents.

Steve Ballmer once compared the GPL to "a cancer", but has since stopped using this analogy. Indeed, Microsoft has softened its public stance towards open source in general, with open source since becoming an important part of the Microsoft Windows ecosystem.[61] However, at the same time, behind the scenes, Microsoft's actions have been less favorable toward the open-source community.

SCO v. IBM and related bad publicity (2003–present)

In 2003, a proprietary Unix vendor and former Linux distribution vendor called SCO alleged that Unix intellectual property had been inappropriately copied into the Linux kernel, and sued IBM, claiming that it bore responsibility for this. Several related lawsuits and countersuits followed, some originating from SCO, some from others suing SCO. However, SCO's allegations lacked specificity, and while some in the media reported them as credible, many critics of SCO believed the allegations to be highly dubious at best.

Over the course of the *SCO v. IBM* case, it emerged that not only had SCO been distributing the Linux kernel for years under the GPL, and continued to do so (thus rendering any claims hard to sustain legally), but that SCO did not even own the copyrights to much of the Unix code that it asserted copyright over, and had no right to sue over them on behalf of the presumed owner, Novell.

This was despite SCO's CEO, Darl McBride, having made many wild and damaging claims of inappropriate appropriation to the media, many of which were later shown to be false, or legally irrelevant even if true.

The blog Groklaw was one of the most forensic examiners of SCO's claims and related events, and gained its popularity from covering this material for many years.

SCO suffered defeat after defeat in *SCO v. IBM* and its various other court cases, and filed for Chapter 11 bankruptcy in 2007. However, despite the courts finding that SCO did not own the copyrights (see above), and SCO's lawsuit-happy CEO Darl McBride no longer running the company, the bankruptcy trustee in charge of SCO-in-bankruptcy decided to press on with some portions he claimed remained relevant in the *SCO v. IBM* lawsuit. He could apparently afford to do this because SCO's main law firm in *SCO v. IBM* had signed an agreement at the outset to represent SCO for a fixed amount of money no matter how long the case took to complete.

In 2004, the Alexis de Tocqueville Institution (ADTI) announced its intent to publish a book, *Samizdat: And Other Issues Regarding the 'Source' of Open Source Code*, showing that the Linux kernel was based on code stolen from Unix, in essence using the argument that it was impossible to believe that Linus Torvalds could produce something as sophisticated as the Linux kernel. The book was never published, after it was widely criticised and ridiculed, including by people supposedly interviewed for the book. It emerged that some of the people were never interviewed, and that ADTI had not tried to contact Linus Torvalds, or ever put the allegations to him to allow a response. Microsoft attempted to draw a line under this incident, stating that it was a "distraction".

Many suspected that some or all of these legal and *fear, uncertainty and doubt* (FUD) attacks against the Linux kernel were covertly arranged by Microsoft, although this has never been proven. Both ADTI and SCO, however, received funding from Microsoft.

European Commission v. Microsoft (2004–2007)

In 2004 the European Commission found Microsoft guilty of anti-competitive behaviour with respect to interoperability in the workgroup software market. Microsoft had formerly settled *United States v. Microsoft* in 2001, in a case which charged that it illegally abused its monopoly power to force computer manufacturers to preinstall Internet Explorer.

The Commission demanded that Microsoft produce full documentation of its workgroup protocols to allow competitors to interoperate with its workgroup software, and imposed fines of 1.5 million euros per day for Microsoft's failure to comply. The commission had jurisdiction because Microsoft sells the software in question in Europe.

Microsoft, after a failed attempt to appeal the decision through the Court of Justice of the European Union, eventually complied with the demand, producing volumes of detailed documentation.

The Samba project, as Microsoft's sole remaining competitor in the workgroup software market, was the key beneficiary of this documentation.

ISO OOXML controversy (2008–present)

In 2008 the International Organization for Standardization published Microsoft's Office Open XML as an international standard, which crucially meant that it, and therefore Microsoft Office, could be used in projects where the use of open standards were mandated by law or by policy. Critics of the standardisation process, including some members of ISO national committees involved in the process itself, alleged irregularities and procedural violations in the process, and argued that the ISO should not have approved OOXML as a standard because it made reference to undocumented Microsoft Office behaviour.

As of 2012, no correct open-source implementation of OOXML exists, which validates the critics' remarks about OOXML being difficult to implement and underspecified. Presently, Google cannot yet convert Office documents into its own proprietary Google Docs format correctly. This suggests that OOXML is not a true open standard, but rather a partial document describing what Microsoft Office does, and only involving certain file formats.

Microsoft's contributions to open source and acquisition of related projects

In 2006 Microsoft launched its CodePlex open source code hosting site, to provide hosting for open-source developers targeting Microsoft platforms. In July 2009 Microsoft open sourced some Hyper-V-supporting patches to the Linux kernel, because they were required to do so by the GNU General Public License,[62][63] and contributed them to the mainline kernel. Note that Hyper-V itself is not open source. Microsoft's F# compiler, created in 2002, has also been released as open source under the Apache license. The F# compiler is a commercial product, as it has been incorporated into Microsoft Visual Studio, which is not open source.

Microsoft representatives have made regular appearances at various open source and Linux conferences for many years.

In 2012, Microsoft launched a subsidiary named Microsoft Open Technologies Inc., with the aim of bridging the gap between proprietary Microsoft technologies and non-Microsoft technologies by engaging with open-source standards.[64] This subsidiary was subsequently folded back into Microsoft as Microsoft's position on open source and non-Windows platforms became more favourable.

In January 2016 Microsoft released Chakra as open source under the MIT License; the code is available on GitHub.[65]

Microsoft's stance on open source has shifted as the company began endorsing more open-source software. In 2016, Steve Balmer, former CEO of Microsoft, has retracted his statement that Linux is a *malignant cancer*.[66] In 2017, the company became a platinum supporter of the Linux Foundation. By 2018, shortly before acquiring GitHub, Microsoft led the charts in the number of paid staff contributing to open-source projects there.[67] While Microsoft may or may not endorse the original philosophy of free software, data shows that it does endorse open source strategically.

Critics have noted that, in March 2019, Microsoft sued Foxconn's subsidiary over a 2013 patent contract;[68] in 2013, Microsoft had announced a patent agreement with Foxconn related to Foxconn's use of the Linux-based Android and ChromeOS.[69]

Open source and programming languages

The vast majority of programming languages in use today have a free software implementation available.

Since the 1990s, the release of major new programming languages in the form of open-source compilers and/or interpreters has been the norm, rather than the exception. Examples include Python in 1991, Ruby in 1995, and Scala in 2003. In recent times, the most notable exceptions have been Java, ActionScript, C#, and Apple's Swift until version 2.2 was proprietary. Partly compatible open-source implementations have been developed for most, and in the case of Java, the main open-source implementation is by now very close to the commercial version.

Java

Since its first public release in 1996, the Java platform had not been open source, although the Java source code portion of the Java runtime was included in Java Development Kits (JDKs), on a purportedly "confidential" basis, despite it being freely downloadable by the general public in most countries. Sun later expanded this "confidential" source code access to include the full source code of the Java Runtime Environment via a separate program which was open to members of the public, and later made the source of the Java compiler *javac* available also. Sun also made the JDK source code available confidentially to the Blackdown Java project, which was a collection of volunteers who ported early versions of the JDK to Linux, or improved on Sun's Linux ports of the JDK. However, none of this was open source, because modification and redistribution without Sun's permission were forbidden in all cases. Sun stated at the time that they were concerned about preventing forking of the Java platform.

However, several independent *partial* reimplementations of the Java platform had been created, many of them by the open-source community, such as the GNU Compiler for Java (GCJ). Sun never filed lawsuits against any of the open source *clone projects*. GCJ notably caused a bad user experience for Java on free software supporting distributions such as Fedora and Ubuntu which shipped GCJ at the time as their Java implementation. How to replace GCJ with the Sun JDK was a frequently asked question by users, because GCJ was an incomplete implementation, incompatible and buggy.

In 2006 Jonathan I. Schwartz became CEO of Sun Microsystems, and signalled his commitment to open source. On 8 May 2007, Sun Microsystems released the Java Development Kit as OpenJDK under the GNU General Public License. Part of the class library (4%) could not be released as open source due to them being licensed from other parties and were included as binary plugs. Because of this, in June 2007, Red Hat launched IcedTea to resolve the encumbered components with the equivalents from GNU Classpath implementation. Since the release, most of the encumbrances have been solved, leaving only the audio engine code and colour management system (the latter is to be resolved using Little CMS).

Distributed version control (2001–present)

The first open-source distributed revision control system (DVCS) was 'tla' in 2001 (since renamed to GNU arch); however, it and its successors 'baz' and 'bzr' (Bazaar) never became very popular, and GNU arch was discontinued, although Bazaar still continues and is used by Canonical.

However, other DVCS projects sprung up, and some started to get significant adoption.

Git (2005–present)

Git, the most popular DVCS, was created in 2005.[70] Some developers of the Linux kernel started to use a proprietary DVCS called BitKeeper, notably Linux founder Linus Torvalds, although some other kernel developers never used it due to its proprietary nature. The unusual situation whereby Linux kernel development involved the use by some of proprietary software "came to a head" when Andrew Tridgell started to reverse-engineer BitKeeper with the aim of producing an open-source tool which could provide some of the same functionality as the commercial version. BitMover, the company that developed BitKeeper, in response, in 2005 revoked the special free of-charge license it had granted to certain kernel developers.

As a result of the removal of the BitKeeper license, Linus Torvalds decided to write his own DVCS, called git, because he thought none of the existing open-source DVCSs were suitable for his particular needs as a kernel maintainer (which was why he had adopted BitKeeper in the first place). A number of other developers quickly jumped in and helped him, and git over time grew from a relatively simple "stupid content tracker" (on which some developers developed "porcelain" extensions) into the sophisticated and powerful DVCS that it is today. Torvalds no longer maintains git himself, however; it has been maintained by Junio Hamano for many years, and has continued receiving contributions from many developers.

The increasing popularity of open-source DVCSs such as git, and then, later, DVCS hosting sites, the most popular of which is GitHub (founded 2008), incrementally reduced the barriers to participation in free software projects still further. With sites like GitHub, no longer did potential contributors have to do things like hunt for the URL for the source code repository (which could be in different places on each website, or sometimes tucked away in a README file or developer documentation), or work out how to generate a patch, and if necessary subscribe to the right mailing list so that their patch email would get to the right people. Contributors can simply fork their own copy of a repository with one click, and issue a pull request from the appropriate branch when their changes are ready. GitHub has become the most popular hosting site in the world for open-source software, and this, together with the ease of forking and the visibility of forks has made it a popular way for contributors to make changes, large and small.

Recent developments

While copyright is the primary legal mechanism that FOSS authors use to ensure license compliance for their software, other mechanisms such as legislation, software patents, and trademarks have uses also. In response to legal issues with patents and the DMCA, the Free Software Foundation released version 3 of its GNU General Public License in 2007 that explicitly addressed the DMCA's digital rights management (DRM) provisions and patent rights.

After the development of the GNU GPLv3, as copyright holder of many pieces of the GNU system, such as the GNU Compiler Collection (GCC) software, the FSF updated most of the GNU programs' licenses from GPLv2 to GPLv3. Apple, a user of GCC and a heavy user of both DRM and patents, decided to switch the compiler in its Xcode IDE from GCC to Clang, another FOSS compiler,[71] but which is under a permissive license.[72] LWN speculated that Apple was motivated partly by a desire to avoid GPLv3.[71] The Samba project also switched to GPLv3, which Apple replaced in their software suite with a closed-source, proprietary software alternative.[73]

Recent mergers have affected major open-source software. Sun Microsystems (Sun) acquired MySQL AB, owner of the popular open-source MySQL database, in 2008.[74]

Oracle in turn purchased Sun in January 2010, acquiring their copyrights, patents, and trademarks. This made Oracle the owner of both the most popular proprietary database and the most popular open-source database (MySQL). Oracle's attempts to commercialize the open-source MySQL database have raised concerns in the FOSS community.[75] Partly in response to uncertainty about the future of MySQL, the FOSS community forked the project into new database systems outside of Oracle's control. These include MariaDB, Percona, and Drizzle.[76] All of these have distinct names; they are distinct projects and cannot use the trademarked name MySQL.[77]

Android (2008–present)

In September 2008, Google released the first version of Android, a new smartphone operating system, as open source (some Google applications that are sometimes but not always bundled with Android are not open source). Initially, the operating system was given away for free by Google, and was eagerly adopted by many handset makers; Google later bought Motorola Mobility and produced its own "vanilla" Android phones and tablets, while continuing to allow other manufacturers to use Android. Android is now the world's most popular mobile platform.[78]

Because Android is based on the Linux kernel, this means that Linux is now the dominant kernel on both mobile platforms (via Android), and supercomputers,[79] and a key player in server operating systems too.

Oracle v. Google

In August 2010, Oracle sued Google claiming that its use of Java in Android infringed on Oracle's copyrights and patents. The initial *Oracle v. Google* trial ended in May 2012, with the finding that Google did not infringe on Oracle's patents, and the trial judge ruled that the structure of the Java application programming interfaces (APIs) used by Google was not copyrightable. The jury found that Google made a trivial ("de minimis") copyright infringement, but the parties stipulated that Google would pay no damages, because it was so trivial.[80] However, Oracle appealed to the Federal Circuit, and Google filed a cross-appeal on the literal copying claim.[81] The Federal Circuit ruled that the small copyright infringement acknowledged by Google was not *de minimis*, and sent the fair use issue back to the trial judge for reconsideration. In 2016, the case was retried and a jury found for Google, on the grounds of fair use.

ChromiumOS (2009–present)

By 2013 Google's Chromebooks, running ChromeOS captured 20–25% of the sub-$300 US laptop market.[82] ChromeOS is built from the open-source ChromiumOS, which is based on Linux, in much the same way that versions of Android shipped on commercially available phones are built from the open source version of Android.

See also

Free and open-source software portal

- Open-source video game § History
- History of software
- History of software engineering
- Timeline of free and open-source software

References

1. James J. Flink (1977). *The Car Culture*. MIT Press. ISBN 978-0-262-56015-3.
2. Hally, Mike (2005). *Electronic brains/Stories from the dawn of the computer age*. London: British Broadcasting Corporation and Granta Books. p. 79. ISBN 1-86207-663-4.
3. Evans, Claire L. (2018). *Broad Band: The Untold Story of the Women Who Made the Internet* (https://books.google.com/books?id=C8ouDwAAQBAJ&q=9780735211759&pg=PP1). New York: Portfolio/Penguin. ISBN 978-0-7352-1175-9.
4. Light, Jennifer S. (1999). "When Computers Were Women". *Technology and Culture*. **40** (3): 455–483. doi:10.1353/tech.1999.0128 (https://doi.org/10.1353%2Ftech.1999.0128). JSTOR 25147356 (https://www.jstor.org/stable/25147356). S2CID 108407884 (https://api.semanticscholar.org/CorpusID:108407884).
5. Maracke, Catharina (25 February 2019). "Free and Open Source Software and FRAND-based patent licenses: How to mediate between Standard Essential Patent and Free and Open Source Software" (https://doi.org/10.1111%2Fjwip.12114). *The Journal of World Intellectual Property*. **22** (3–4): 78–102. doi:10.1111/jwip.12114 (https://doi.org/10.1111%2Fjwip.12114). ISSN 1422-2213 (https://search.worldcat.org/issn/1422-2213). S2CID 159111696 (https://api.semanticscholar.org/CorpusID:159111696).
6. Hippel, Eric von; Krogh, Georg von (1 April 2003). "Open Source Software and the "Private-Collective" Innovation Model: Issues for Organization Science" (https://dspace.mit.edu/bitstream/1721.1/66145/1/SSRN-id1410789.pdf) (PDF). *Organization Science*. **14** (2): 209–223. doi:10.1287/orsc.14.2.209.14992 (https://doi.org/10.1287%2Forsc.14.2.209.14992). hdl:1721.1/66145 (https://hdl.handle.net/1721.1%2F66145). ISSN 1047-7039 (https://search.worldcat.org/issn/1047-7039). S2CID 11947692 (https://api.semanticscholar.org/CorpusID:11947692).

7. "IBM 7090/7094 Page" (https://web.archive.org/web/20150827134534/http://www.cozx.com/~dpitts/ibm7090.html). Archived from the original (http://www.cozx.com/~dpitts/ibm7090.html) on 27 August 2015. Retrieved 11 August 2015.
8. VM Brasseur (2018). *Forge your Future with Open Source* (https://archive.org/details/isbn_9781680503012). Pragmatic Programmers. ISBN 978-1-68050-301-2.
9. Ceruzzi, Paul (1998). *A History of Modern Computing* (https://archive.org/details/historyofmoderncc00ceru). The MIT Press. ISBN 9780262032551.
10. "Heresy & Heretical Open Source: A Heretic's Perspective" (http://www.infoq.com/presentations/Heretical-Open-Source). Archived (https://web.archive.org/web/20121014192441/http://www.infoq.com/presentations/Heretical-Open-Source) from the original on 14 October 2012. Retrieved 16 November 2012.
11. Sam Williams. *Free as in Freedom: Richard Stallman's Crusade for Free Software*. "Chapter 1: For Want of a Printer" (http://www.oreilly.com/openbook/freedom/ch01.html) Archived (https://web.archive.org/web/20150910145624/http://www.oreilly.com/openbook/freedom/ch01.html) 10 September 2015 at the Wayback Machine. 2002.
12. Gaudeul, Alexia (2007). "Do Open Source Developers Respond to Competition? The LATEX Case Study" (https://doi.org/10.2202%2F1446-9022.1119). *Review of Network Economics*. **6** (2). doi:10.2202/1446-9022.1119 (https://doi.org/10.2202%2F1446-9022.1119). ISSN 1446-9022 (https://search.worldcat.org/issn/1446-9022). S2CID 201097782 (https://api.semanticscholar.org/CorpusID:201097782).
13. "A brief history of spice" (https://web.archive.org/web/20070523080617/http://www.ecircuitcenter.com/SpiceTopics/History.htm). Archived from the original (http://www.ecircuitcenter.com/SpiceTopics/History.htm) on 23 May 2007. Retrieved 18 June 2007.
14. Fisher, Franklin M.; McKie, James W.; Mancke, Richard B. (1983). *IBM and the U.S. Data Processing Industry: An Economic History*. Praeger. ISBN 978-0-03-063059-0.page 176
15. Fisher. *op.cit*.
16. Apple Computer, Inc. v. Franklin Computer Corporation Puts the Byte Back into Copyright Protection for Computer Programs (http://digitalcommons.law.ggu.edu/cgi/viewcontent.cgi?article=1344&context=ggulrev) Archived (https://web.archive.org/web/20170507231059/http://digitalcommons.law.ggu.edu/cgi/viewcontent.cgi?article=1344&context=ggulrev) 7 May 2017 at the Wayback Machine in Golden Gate University Law Review Volume 14, Issue 2, Article 3 by Jan L. Nussbaum (January 1984)
17. Lemley, Menell, Merges and Samuelson. *Software and Internet Law*, p. 34.
18. Landley, Rob (23 May 2009). "notes-2009" (http://landley.net/notes-2009.html). landley.net. Archived (https://web.archive.org/web/20170904111501/http://landley.net/notes-2009.html) from the original on 4 September 2017. Retrieved 2 December 2015. "*So if open source used to be the norm back in the 1960s and 70s, how did this _change_? Where did proprietary software come from, and when, and how? How did Richard Stallman's little utopia at the MIT AI lab crumble and force him out into the wilderness to try to rebuild it? Two things changed in the early 80s: the exponentially growing installed base of microcomputer hardware reached critical mass around 1980, and a legal decision altered copyright law to cover binaries in 1983.*"
19. Weber, Steven (2004). *The Success of Open Source* (https://web.archive.org/web/20090927011950/http://polisci.berkeley.edu/faculty/bio/permanent/Weber,S/). Cambridge, MA: Harvard University Press. pp. 38–44. ISBN 978-0-674-01858-7. Archived from the original (http://www.polisci.berkeley.edu/Faculty/bio/permanent/Weber,S/) on 27 September 2009.
20. IBM Corporation (8 February 1983). "DISTRIBUTION OF IBM LICENSED PROGRAMS AND LICENSED PROGRAM MATERIALS AND MODIFIED AGREEMENT FOR IBM LICENSED PROGRAMS" (http://landley.net/history/mirror/ibm/oco.html). Archived (https://web.archive.org/web/20170904075121/http://landley.net/history/mirror/ibm/oco.html) from the original on 4 September 2017. Retrieved 24 March 2017.

21. Gallant, John (18 March 1985). "IBM policy draws fire – Users say source code rules hamper change" (https://books.google.com/books?id=4Wgmey4obagC&q=1983object-only+model+IBM&pg=PA8). Computerworld. Retrieved 27 December 2015. "While IBM's policy of withholding source code for selected software products has already marked its second anniversary, users are only now beginning to cope with the impact of that decision. But whether or not the advent of object-code-only products has affected their day-to-day DP operations, some users remain angry about IBM's decision. Announced in February 1983, IBM's object-code-only policy has been applied to a growing list of Big Blue system software products"
22. Shea, Tom (23 June 1983). "Free software – Free software is a junkyard of software spare parts" (https://books.google.com/books?id=yy8EAAAAMBAJ&q=us+government+public+domain+software&pg=PA31). *InfoWorld*. Retrieved 10 February 2016.
23. Ahl, David. "David H. Ahl biography from Who's Who in America" (http://www.swapmeetdave.com/Ahl/DHAbio.htm). Archived (https://web.archive.org/web/20150924131314/http://www.swapmeetdave.com/Ahl/DHAbio.htm) from the original on 24 September 2015. Retrieved 23 November 2009.
24. Wilkinson, Bill (1983). *The Atari BASIC Source Book* (https://archive.org/details/ataribooks-the-atari-basic-source-book). COMPUTE! Books. ISBN 9780942386158. Archived (https://web.archive.org/web/20130119010453/http://archive.org/details/ataribooks-the-atari-basic-source-book) from the original on 19 January 2013. Retrieved 2 October 2017.
25. Wilkinson, Bill (1982). "Inside Atari DOS" (http://atariarchives.org/iad/). COMPUTE! Books. Archived (https://web.archive.org/web/20171002120523/http://atariarchives.org/iad/) from the original on 2 October 2017. Retrieved 2 October 2017.
26. Norman, Jeremy. "SHARE, The First Computer Users' Group, is Founded (1955)" (http://www.historyofinformation.com/expanded.php?id=4805). *HistofyofInformation.com*. Archived (https://web.archive.org/web/20170411125236/http://historyofinformation.com/expanded.php?id=4805) from the original on 11 April 2017. Retrieved 24 March 2017.
27. "The DECUS tapes" (http://www.ibiblio.org/pub/academic/computer-science/history/pdp-11/rsx/decus/). Archived (https://web.archive.org/web/20121228220557/http://www.ibiblio.org/pub/academic/computer-science/history/pdp-11/rsx/decus/) from the original on 28 December 2012. Retrieved 9 May 2013.
28. DiBona, C., et al. *Open Sources 2.0*. O'Reilly, ISBN 0-596-00802-3.
29. "Talk transcript where Stallman tells the printer story" (https://www.gnu.org/events/rms-nyu-2001-transcript.txt). Archived (https://web.archive.org/web/20140318074756/http://www.gnu.org/events/rms-nyu-2001-transcript.txt) from the original on 18 March 2014. Retrieved 18 April 2014.
30. "Transcript of Richard Stallman's Speech, 28 Oct 2002, at the International Lisp Conference" (https://www.gnu.org/gnu/rms-lisp.html). GNU Project. 28 October 2002. Archived (https://web.archive.org/web/20140416180457/https://www.gnu.org/gnu/rms-lisp.html) from the original on 16 April 2014. Retrieved 21 December 2008.
31. "GNU General Public License v1.0" (https://www.gnu.org/licenses/old-licenses/gpl-1.0.txt). Archived (https://web.archive.org/web/20140221032723/http://www.gnu.org/licenses/old-licenses/gpl-1.0.txt) from the original on 21 February 2014. Retrieved 18 April 2014.
32. Michael Tiemann (29 March 1999). "Future of Cygnus Solutions, An Entrepreneur's Account" (http://www.oreilly.com/catalog/opensources/book/tiemans.html). Archived (https://web.archive.org/web/20070610003451/http://www.oreilly.com/catalog/opensources/book/tiemans.html) from the original on 10 June 2007. Retrieved 18 June 2007.
33. "Release notes for Linux kernel 0.12" (https://www.kernel.org/pub/linux/kernel/Historic/old-versions/RELNOTES-0.12). Archived (https://web.archive.org/web/20070819045030/http://www.kernel.org/pub/linux/kernel/Historic/old-versions/RELNOTES-0.12) from the original on 19 August 2007. Retrieved 25 July 2016.

34. "A Brief History of Debian" (http://www.debian.org/doc/manuals/project-history/ch-detailed.en.html). Archived (https://web.archive.org/web/20120728073053/http://www.debian.org/doc/manuals/project-history/ch-detailed.en.html) from the original on 28 July 2012. Retrieved 11 February 2008.
35. Take your freedom back, with Linux-2.6.33-libre (http://www.fsfla.org/svnwiki/anuncio/2010-03-Linux-2.6.33-libre.en) Archived (https://web.archive.org/web/20121219081431/http://www.fsfla.org/svnwiki/anuncio/2010-03-Linux-2.6.33-libre.en) 19 December 2012 at the Wayback Machine FSFLA, 2010.
36. "Linux and GNU – GNU Project – Free Software Foundation" (https://www.gnu.org/gnu/linux-and-gnu.html). Gnu.org. 20 May 2013. Archived (https://web.archive.org/web/20170319145123/http://www.gnu.org/gnu/linux-and-gnu.html) from the original on 19 March 2017. Retrieved 2 October 2013.
37. "January 2015 Web Server Survey" (https://web.archive.org/web/20160119190846/http://news.netcraft.com/archives/2015/01/15/january-2015-web-server-survey.html). *Netcraft*. Archived from the original (http://news.netcraft.com/archives/2015/01/15/january-2015-web-server-survey.html) on 19 January 2016. Retrieved 3 February 2016.
38. "On the term "Open Source" " (https://web.archive.org/web/20180315075903/http://hyperlogos.org/blog/drink/term-Open-Source). *The Hyperlogos*. 15 March 2018. Archived from the original (http://hyperlogos.org/blog/drink/term-Open-Source) on 15 March 2018.
39. "Why do Tech Journalists so often get Computer History... totally wrong?" (https://www.youtube.com/watch?v=OMoIEG-I6JQ). 10 April 2022 – via www.youtube.com.
40. Kelty, Christpher M. (2008). "The Cultural Significance of free Software – Two Bits" (https://web.archive.org/web/20160304070704/http://twobits.net/pub/Kelty-TwoBits.pdf) (PDF). Duke University press – durham and london. p. 99. Archived from the original (http://twobits.net/pub/Kelty-TwoBits.pdf) (PDF) on 4 March 2016. Retrieved 17 February 2016.
41. Karl Fogel (2016). "Producing Open Source Software – How to Run a Successful Free Software Project" (http://producingoss.com/en/introduction.html#free-vs-open-source). O'Reilly Media. Archived (https://web.archive.org/web/20160420192933/http://producingoss.com/en/introduction.html#free-vs-open-source) from the original on 20 April 2016. Retrieved 11 April 2016.
42. Tiemann, Michael (19 September 2006). "History of the OSI" (https://web.archive.org/web/20021001164015/http://www.opensource.org/docs/history.php). Open Source Initiative. Archived from the original (http://www.opensource.org/history) on 1 October 2002. Retrieved 23 August 2008.
43. Muffatto, Moreno (2006). *Open Source: A Multidisciplinary Approach*. Imperial College Press. ISBN 978-1-86094-665-3.
44. Open Source Summit (http://linuxgazette.net/issue28/rossum.html) Archived (https://web.archive.org/web/20131229053622/http://linuxgazette.net/issue28/rossum.html) 29 December 2013 at the Wayback Machine Linux Gazette. 1998.
45. Eric S. Raymond. "Goodbye, 'free software'; hello, 'open source' " (http://www.catb.org/~esr/open-source.html). catb.org. Archived (https://web.archive.org/web/20110811021422/http://www.catb.org/~esr/open-source.html) from the original on 11 August 2011. Retrieved 11 August 2015.
46. Tiemann, Michael (19 September 2006). "History of the OSI" (https://web.archive.org/web/20021001164015/http://www.opensource.org/docs/history.php). Open Source Initiative. Archived from the original (http://www.opensource.org/docs/history.php) on 1 October 2002. Retrieved 23 August 2008.
47. Leander Kahney (5 March 1999). "Linux's Forgotten Man – You have to feel for Richard Stallman" (https://web.archive.org/web/20010622041610/http://www.wired.com/news/technology/0%2C1282%2C18291%2C00.html). *Wired*. Archived from the original (https://www.wired.com/news/technology/0,1282,18291,00.html) on 22 June 2001.

48. "Toronto Star: Freedom's Forgotten Prophet (Richard Stallman)" (http://www.linuxtoday.com/infrastructure/2000101000421OSCY). *Linux Today*. 10 October 2000. Archived (https://web.archive.org/web/20160417104602/http://www.linuxtoday.com/infrastructure/2000101000421OSCY) from the original on 17 April 2016. Retrieved 25 March 2016.
49. Nikolai Bezroukov (1 November 2014). "Portraits of Open Source Pioneers – Part IV. Prophet" (http://www.softpanorama.org/People/Stallman/prophet.shtml#Linux_schism). Archived (https://web.archive.org/web/20160410122520/http://softpanorama.org/People/Stallman/prophet.shtml#Linux_schism) from the original on 10 April 2016. Retrieved 25 March 2016.
50. Richard Stallman. "Why Open Source Misses the Point" (https://www.gnu.org/philosophy/open-source-misses-the-point.html). Archived (https://web.archive.org/web/20110804231811/http://www.gnu.org/philosophy/open-source-misses-the-point.html) from the original on 4 August 2011. Retrieved 18 April 2014.
51. Rishab Aiyer Ghosh et al (2002)."Free/Libre and Open Source Software: Survey and Study FLOSS Deliverable D18: FINAL REPORT – Part IV: Survey of Developers" (https://web.archive.org/web/20090913045502/http://www.infonomics.nl/FLOSS/report/Final4.htm). Archived from the original on 13 September 2009.
52. released (https://www.openoffice.org/about_us/milestones.html)
53. Brian Fitzgerald, Pär J. Ågerfalk (2005). The Mysteries of Open Source Software: Black and White and Red All Over (https://www.computer.org/csdl/proceedings/hicss/2005/2268/07/22680196a.pdf) Archived (https://web.archive.org/web/20160405134050/https://www.computer.org/csdl/proceedings/hicss/2005/2268/07/22680196a.pdf) 5 April 2016 at the Wayback Machine University of Limerick, Ireland. "Open Source software (OSS) has attracted enormous media and research attention since the term was coined in February 1998."
54. Ettrich, Matthias (14 October 1996). "New Project: Kool Desktop Environment (KDE)" (https://groups.google.com/group/de.comp.os.linux.misc/msg/cb4b2d67ffc3ffce). Newsgroup: de.comp.os.linux.misc (news:de.comp.os.linux.misc). Usenet: 53tkvv$b4j@newsserv.zdv.uni-tuebingen.de (news:53tkvv$b4j@newsserv.zdv.uni-tuebingen.de). Archived (https://web.archive.org/web/20130530054434/http://groups.google.com/group/de.comp.os.linux.misc/msg/cb4b2d67ffc3ffce) from the original on 30 May 2013. Retrieved 29 December 2006.
55. Richard Stallman (5 September 2000). "Stallman on Qt, the GPL, KDE, and GNOME" (http://linuxtoday.com/news_story.php3?ltsn=2000-09-05-001-21-OP-LF-KE). Archived (https://web.archive.org/web/20090122070916/http://www.linuxtoday.com/news_story.php3?ltsn=2000-09-05-001-21-OP-LF-KE) from the original on 22 January 2009. Retrieved 9 September 2005.
56. "A tale of two desktops" (http://www.pcauthority.com.au/Feature/79378,a-tale-of-two-desktops.aspx). *PC & Tech Authority*. Archived (https://web.archive.org/web/20150704093709/http://www.pcauthority.com.au/Feature/79378,a-tale-of-two-desktops.aspx) from the original on 4 July 2015. Retrieved 11 August 2015.
57. Shuttleworth, Mark (5 April 2017). "Growing Ubuntu for cloud and IoT, rather than phone and convergence" (https://web.archive.org/web/20170507225729/https://insights.ubuntu.com/2017/04/05/growing-ubuntu-for-cloud-and-iot-rather-than-phone-and-convergence). *Canonical Ltd.* Archived from the original (https://insights.ubuntu.com/2017/04/05/growing-ubuntu-for-cloud-and-iot-rather-than-phone-and-convergence/) on 7 May 2017. Retrieved 5 April 2017.
58. "Ubuntu To Abandon Unity 8, Switch Back To GNOME" (http://phoronix.com/scan.php?page=news_item&px=Ubuntu-Dropping-Unity). Phoronix.com. Archived (https://web.archive.org/web/20170516161705/http://www.phoronix.com/scan.php?page=news_item&px=Ubuntu-Dropping-Unity) from the original on 16 May 2017. Retrieved 6 April 2017.
59. "The Wayland Situation: Facts About X vs. Wayland – Phoronix" (https://www.phoronix.com/scan.php?page=article&item=x_wayland_situation). Archived (https://web.archive.org/web/20150924091609/http://www.phoronix.com/scan.php?page=article&item=x_wayland_situation) from the original on 24 September 2015. Retrieved 11 August 2015.

60. "Post on y-devel by Brandon Black" (https://web.archive.org/web/20060924165154/http://www.y-windows.org/pipermail/y-devel/2006-January/001977.html). 3 January 2006. Archived from the original (http://www.y-windows.org/pipermail/y-devel/2006-January/001977.html) on 24 September 2006. Retrieved 14 May 2017.
61. "When Open Source Came to Microsoft" (https://www.codemag.com/Article/2009041/When-Open-Source-Came-to-Microsoft). *www.codemag.com*. Retrieved 29 December 2023.
62. Gavin Clarke (23 July 2009). "Microsoft opened Linux-driver code after 'violating' GPL" (https://www.theregister.co.uk/2009/07/23/microsoft_hyperv_gpl_violation/). The Register. Archived (https://web.archive.org/web/20131004230535/http://www.theregister.co.uk/2009/07/23/microsoft_hyperv_gpl_violation/) from the original on 4 October 2013. Retrieved 6 September 2013.
63. "GNU General Public License" (https://www.gnu.org/licenses/gpl.html). Archived (https://web.archive.org/web/20140418021611/http://www.gnu.org/licenses/gpl.html) from the original on 18 April 2014. Retrieved 18 April 2014.
64. Ovide, Shira (16 April 2012). "Microsoft Dips Further into Open-Source Software" (https://www.wsj.com/articles/SB10001424052702304432704577347783238850756). *The Wall Street Journal*. Archived (https://web.archive.org/web/20150210214357/http://www.wsj.com/articles/SB10001424052702304432704577347783238850756) from the original on 10 February 2015. Retrieved 17 April 2012.
65. "ChakraCore GitHub repository is now open" (https://blogs.windows.com/msedgedev/2016/01/13/chakracore-now-open/). 13 January 2016. Archived (https://web.archive.org/web/20160130180037/https://blogs.windows.com/msedgedev/2016/01/13/chakracore-now-open/) from the original on 30 January 2016. Retrieved 22 January 2016.
66. "Ballmer: I may have called Linux a cancer but now I love it" (https://www.zdnet.com/article/ballmer-i-may-have-called-linux-a-cancer-but-now-i-love-it/). *ZDNet*.
67. Asay, Matt (7 February 2018). "Who really contributes to open source" (https://www.infoworld.com/article/3253948/who-really-contributes-to-open-source.html). *InfoWorld*. Retrieved 8 June 2020.
68. "Foxconn rejects Microsoft patent lawsuit, says never had to pay royalties" (https://www.reuters.com/article/us-foxconn-microsoft-idUSKBN1QT0D9). *Reuters*. 19 March 2019. Retrieved 8 June 2020.
69. "Microsoft and Foxconn Parent Hon Hai Sign Patent Agreement For Android and Chrome Devices - Stories" (https://news.microsoft.com/2013/04/17/microsoft-and-foxconn-parent-hon-hai-sign-patent-agreement-for-android-and-chrome-devices/). *microsoft.com*. 16 April 2013. Archived (https://web.archive.org/web/20190330124348/https://news.microsoft.com/2013/04/17/microsoft-and-foxconn-parent-hon-hai-sign-patent-agreement-for-android-and-chrome-devices/) from the original on 30 March 2019. Retrieved 8 June 2020.
70. "SCM Ranking, Q3 2013" (https://web.archive.org/web/20130928180246/http://beta.gitgear.com/why_git/SCM_Ranking_2013Q3_F1.pdf) (PDF). Switch-Gears. Archived from the original (http://beta.gitgear.com/why_git/SCM_Ranking_2013Q3_F1.pdf) (PDF) on 28 September 2013. Retrieved 22 September 2013.
71. Brockmeier, Joe. "Apple's Selective Contributions to GCC" (https://lwn.net/Articles/405417/). Archived (https://web.archive.org/web/20111117002647/http://lwn.net/Articles/405417/) from the original on 17 November 2011. Retrieved 23 October 2011.
72. "LLVM Developer Policy" (https://web.archive.org/web/20121113204817/http://www.llvm.org/docs/DeveloperPolicy.html#license). LLVM. Archived from the original (http://llvm.org/docs/DeveloperPolicy.html#license) on 13 November 2012. Retrieved 19 November 2012.
73. Holwerda, Thom. "Apple Ditches SAMBA in Favour of Homegrown Replacement" (http://www.osnews.com/story/24572/). Archived (https://web.archive.org/web/20120114142006/http://www.osnews.com/story/24572/) from the original on 14 January 2012. Retrieved 23 October 2011.

74. "Sun to Acquire MySQL" (https://web.archive.org/web/20080117192218/http://www.mysql.com/news-and-events/sun-to-acquire-mysql.html). MySQL AB. Archived from the original (http://mysql.com/news-and-events/sun-to-acquire-mysql.html) on 17 January 2008. Retrieved 16 January 2008.
75. Thomson, Iain. "Oracle offers commercial extensions to MySQL" (https://www.theregister.co.uk/2011/09/16/oracle_commercial_extensions_mysql/). *The Register*. Archived (https://web.archive.org/web/20111023204950/http://www.theregister.co.uk/2011/09/16/oracle_commercial_extensions_mysql/) from the original on 23 October 2011. Retrieved 23 October 2011.
76. Samson, Ted. "Non-Oracle MySQL fork deemed ready for prime time" (https://web.archive.org/web/20111109094334/http://www.infoworld.com/d/open-source-software/non-oracle-mysql-fork-deemed-ready-prime-time-853). Archived from the original (http://www.infoworld.com/d/open-source-software/non-oracle-mysql-fork-deemed-ready-prime-time-853) on 9 November 2011. Retrieved 23 October 2011.
77. Nelson, Russell. "Open Source, MySQL, and trademarks" (http://www.opensource.org/node/496). Archived (https://web.archive.org/web/20111021095017/http://opensource.org/node/496) from the original on 21 October 2011. Retrieved 23 October 2011.
78. "Android, the world's most popular mobile platform" (https://developer.android.com/about/index.html). Archived (https://web.archive.org/web/20130922035131/http://developer.android.com/about/index.html) from the original on 22 September 2013. Retrieved 6 September 2013.
79. Steven J. Vaughan-Nichols (29 July 2013). "20 great years of Linux and supercomputers" (https://www.zdnet.com/article/20-great-years-of-linux-and-supercomputers/). ZDNet. Archived (https://web.archive.org/web/20130823182003/http://www.zdnet.com/20-great-years-of-linux-and-supercomputers-7000018681/) from the original on 23 August 2013. Retrieved 6 September 2013.
80. Niccolai, James (20 June 2012). "Oracle agrees to 'zero' damages in Google lawsuit, eyes appeal" (http://www.computerworld.com/s/article/9228298/Oracle_agrees_to_zero_damages_in_Google_lawsuit_eyes_appeal). Archived (https://web.archive.org/web/20121117001109/http://www.computerworld.com/s/article/9228298/Oracle_agrees_to_zero_damages_in_Google_lawsuit_eyes_appeal) from the original on 17 November 2012. Retrieved 23 June 2012.
81. Jones, Pamela (5 October 2012). "Oracle and Google File Appeals" (https://web.archive.org/web/20121201130542/http://www.groklaw.net/articlebasic.php?story=20121005082638280). Groklaw. Archived from the original (http://www.groklaw.net/articlebasic.php?story=20121005082638280) on 1 December 2012. Retrieved 17 November 2012.
82. Williams, Rhiannon (11 July 2013). "Google Chromebook sales soar in face of PC decline" (https://www.telegraph.co.uk/technology/google/10173494/Google-Chromebook-sales-soar-in-face-of-PC-decline.html). *The Daily Telegraph*. Archived (https://web.archive.org/web/20130925104825/http://www.telegraph.co.uk/technology/google/10173494/Google-Chromebook-sales-soar-in-face-of-PC-decline.html) from the original on 25 September 2013. Retrieved 3 September 2013.

External links

- Elmer-Dewitt, Philip (30 July 1984). Software Is for Sharing (https://web.archive.org/web/20070930122050/http://www.time.com/time/magazine/article/0,9171,926733,00.html), *Time*.
- Richard Stallman speaking about free software and the GNU project in 1986 (https://www.gnu.org/philosophy/stallman-kth.html), Sweden
- David A. Wheeler on the history of free software (http://www.dwheeler.com/oss_fs_why.html#history), from his "Look at the numbers!" paper

- The Daemon, the GNU, and the Penguin (http://www.groklaw.net/staticpages/index.php?page=20051013231901859), by Peter Salus
- Documents about the BSD lawsuit that lead to 386BSD and then FreeBSD (http://cm.bell-labs.com/cm/cs/who/dmr/bsdi/bsdisuit.html)
- *Open Sources: Voices from the Open Source Revolution* (http://www.oreilly.com/catalog/opensources/book/toc.html) (January 1999)
- The history of Cygnus solutions (https://web.archive.org/web/20171022105307/http://www.toad.com/gnu/cygnus/index.html), the largest free software company of the early 90s
- LWN.net's 1998–2008 timeline part 1 (https://lwn.net/Articles/264402/) (part 2 (https://lwn.net/Articles/264980/), 3 (https://lwn.net/Articles/265577/), 4 (https://lwn.net/Articles/266358/), 5 (https://lwn.net/Articles/268407/), 6 (https://lwn.net/Articles/269276/))
- A Brief History of FreeBSD (http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/history.html), by Jordan Hubbard
- UNESCO Free Software Portal (http://www.unesco.org/webworld/portal_freesoft/open_history.shtml)
- Infinite Hands (http://infinite-hands.draketo.de/), a free licensed folk song about the history of free software.

Wikipedia
The Free Encyclopedia

Open-source software movement

(Redirected from Open-source-software movement)

The **open-source software movement** is a social movement that supports the use of open-source licenses for some or all software, as part of the broader notion of open collaboration.[1] The open-source movement was started to spread the concept/idea of open-source software.

Programmers who support the open-source-movement philosophy contribute to the open-source community by voluntarily writing and exchanging programming code for software development.[2] The term *open source* requires that no one can discriminate against a group in not sharing the edited code or hinder others from editing their already-edited work. This approach to software development allows anyone to obtain and modify open-source code. These modifications are distributed back to the developers within the open-source community of people who are working with the software. In this way, the identities of all individuals participating in code modification are disclosed and the transformation of the code is documented over time.[3] This method makes it difficult to establish ownership of a particular bit of code but is in keeping with the open-source-movement philosophy. These goals promote the production of high-quality programs as well as working cooperatively with other similarly-minded people to improve open-source technology.[2]

Brief history

The label *open source* was created and adopted by a group of people in the free software movement at a strategy session[4] held at Palo Alto, California, in reaction to Netscape's January 1998 announcement of a source-code release for Navigator. One of the reasons behind using the term was that "the advantage of using the term open source is that the business world usually tries to keep free technologies from being installed."[5] Those people who adopted the term used the opportunity before the release of Navigator's source code to free themselves of the ideological and confrontational connotations of the term "free software". Later in February 1998, Bruce Perens and Eric S. Raymond founded an organization called Open Source Initiative (OSI) "as an educational, advocacy, and stewardship organization at a cusp moment in the history of that culture."[6]

Evolution

a difference between hardware and software did not exist. The user and programmer of a computer were one and the same. When the first commercial electronic computer was introduced by IBM in 1952, the machine was hard to maintain and expensive. Putting the price of the machine aside, it was the software that caused the problem when owning one of these computers. Then in 1952, a collaboration of all the owners of the computer got together and created a set of tools. The collaboration of people were in a group called PACT (The Project for the Advancement of Coding techniques). After passing this hurdle, in 1956, the Eisenhower administration decided to put restrictions on the types of sales AT&T could make. This did not stop the inventors from developing new ideas of how to bring the computer to the mass population. The next step was making the computer more affordable which slowly developed through different companies. Then they had to develop software that would host multiple users. MIT computation center developed one of the first systems, CTSS (Compatible Time-Sharing System). This laid the foundation for many more systems, and what we now call the open-source software movement.[7]

The open-source movement is branched from the free software movement which began in the late 80s with the launching of the GNU project by Richard Stallman.[8] Stallman is regarded within the open-source community as sharing a key role in the conceptualization of freely-shared source code for software development.[3] The term "free software" in the free software movement is meant to imply freedom of software exchange and modification. The term does not refer to any monetary freedom.[3] Both the free-software movement and the open-source movement share this view of free exchange of programming code, and this is often why both of the movements are sometimes referenced in literature as part of the FOSS or "Free and Open Software" or FLOSS "Free/Libre Open-Source" communities.

These movements share fundamental differences in the view on open software. The main, factionalizing difference between the groups is the relationship between open-source and proprietary software. Often, makers of proprietary software, such as Microsoft, may make efforts to support open-source software to remain competitive.[9] Members of the open-source community are willing to coexist with the makers of proprietary software[3] and feel that the issue of whether software is open source is a matter of practicality.[10]

In contrast, members of the free-software community maintain the vision that all software is a part of freedom of speech[3] and that proprietary software is unethical and unjust.[3] The free-software movement openly champions this belief through talks that denounce proprietary software. As a whole, the community refuses to support proprietary software. Further there are external motivations for these developers. One motivation is that, when a programmer fixes a bug or makes a program it benefits others in an open-source environment. Another motivation is that a programmer can work on multiple projects that they find interesting and enjoyable. Programming in the open-source world can also lead to commercial job offers or entrance into the venture capital community. These are just a few reasons why open-source programmers continue to create and advance software.[11]

While cognizant of the fact that both the free-software movement and the open-source movement share similarities in practical recommendations regarding open source, the free-software movement fervently continues to distinguish themselves from the open-source movement entirely.[12] The free-software movement maintains that it has fundamentally different attitudes towards the relationship between open-source and proprietary software. The free-software community does not view the open-source community as their target grievance, however. Their target grievance is proprietary software itself.[3]

Legal issues

The open-source movement has faced a number of legal challenges. Companies that manage open-source products have some difficulty securing their trademarks. For example, the scope of "implied license" conjecture remains unclear and can compromise an enterprise's ability to patent productions made with open-source software. Another example is the case of companies offering add-ons for purchase; licensees who make additions to the open-source code that are similar to those for purchase may have immunity from patent suits.

In the court case "Jacobsen v. Katzer", the plaintiff sued the defendant for failing to put the required attribution notices in his modified version of the software, thereby violating license. The defendant claimed Artistic License in not adhering to the conditions of the software's use, but the wording of the attribution notice decided that this was not the case. "Jacobsen v Katzer" established open-source software's equality to proprietary software in the eyes of the law.

In a court case accusing Microsoft of being a monopoly, Linux and open-source software was introduced in court to prove that Microsoft had valid competitors and was grouped in with Apple.

There are resources available for those involved open-source projects in need of legal advice. The Software Freedom Law Center features a primer on open-source legal issues. International Free and Open Source Software Law Review offers peer-reviewed information for lawyers on free-software issues.

Formalization

The Open Source Initiative (OSI) was instrumental in the formalization of the open-source movement. The OSI was founded by Eric Raymond and Bruce Perens in February 1998 with the purpose of providing general education and advocacy of the open-source label through the creation of the Open Source Definition that was based on the Debian Free Software Guidelines. The OSI has become one of the main supporters and advocators of the open-source movement.[6]

In February 1998, the open-source movement was adopted, formalized, and spearheaded by the Open Source Initiative (OSI), an organization formed to market software "as something more amenable to commercial business use"[3] The OSI applied to register "Open Source" with the US Patent and Trademark Office, but was denied due to the term being generic and/or descriptive. Consequently, the OSI does not own the trademark "Open Source" in a national or international sense, although it does assert common-law trademark rights in the term.[2] The main tool they adopted for this was The Open Source Definition.[13]

The open-source label was conceived at a strategy session that was held on February 3, 1998 in Palo Alto, California and on April 8 of the same year, the attendees of Tim O'Reilly's Free Software Summit voted to promote the use of the term *open source*.[6]

Overall, the software developments that have come out of the open-source movement have not been unique to the computer-science field, but they have been successful in developing alternatives to propriety software. Members of the open-source community improve upon code and write programs that can rival much of the propriety software that is already available.[3]

The rhetorical discourse used in open-source movements is now being broadened to include a larger group of non-expert users as well as advocacy organizations. Several organized groups such as the Creative Commons and global development agencies have also adopted the open-source concepts according to their own aims and for their own purposes.[14]

The factors affecting the open-source movement's legal formalization are primarily based on recent political discussion over copyright, appropriation, and intellectual property.[15]

Social structure of open source contribution teams

Historically, researchers have characterized open-source contributors as a centralized, onion-shaped group.[16] The center of the onion consists of the core contributors who drive the project forward through large amounts of code and software design choices. The second-most layer are contributors who respond to pull requests and bug reports. The third-most layer out are contributors who mainly submit bug reports. The farthest out layer are those who watch the repository and users of the software that's generated. This model has been used in research to understand the lifecycle of open-source software, understand contributors to open-source software projects, how tools such as can help contributors at the various levels of involvement in the project, and further understand how the distributed nature of open source software may affect the productivity of developers.[17][18][19]

Some researchers have disagreed with this model. Crowston et al.'s work has found that some teams are much less centralized and follow a more distributed workflow pattern.[17] The authors report that there's a weak correlation between project size and centralization, with smaller projects being more centralized and larger projects showing less centralization. However, the authors only looked at bug reporting and fixing, so it remains unclear whether this pattern is only associated with bug finding and fixing or if centralization does become more distributed with size for every aspect of the open-source paradigm.

An understanding of a team's centralization versus distributed nature is important as it may inform tool design and aid new developers in understanding a team's dynamic. One concern with open-source development is the high turnover rate of developers, even among core contributors (those at the center of the "onion").[20] In order to continue an open-source project, new developers must continually join but must also have the necessary skill-set to contribute quality code to the project. Through a study of GitHub contribution on open-source projects, Middleton et al. found that the largest predictor of contributors becoming full-fledged members of an open-source team (moving to the "core" of the "onion") was whether they submitted and commented on pull requests. The authors then suggest that GitHub, as a tool, can aid in this process by supporting "checkbox" features on a team's open-source project that urge contributors to take part in these activities.[19]

Youth engagement

The open-source community has long recognized the importance of engaging younger generations to ensure the sustainability and innovation of open-source projects. However, concerns have been raised about the aging demographic of open source contributors and the challenges of attracting younger developers. In 2010, James Bottomley, a prominent Linux kernel maintainer, observed the "graying" of the Linux kernel community, a trend that continues today. David Nalley, president of the Apache Software Foundation (ASF), emphasized that maintaining legacy code is often unappealing to younger developers, who prefer to work on new and innovative projects.[21]

While contributing to open source projects can provide valuable experience in development, documentation, internationalization, and other areas, barriers to entry often make it difficult for newcomers, particularly younger individuals, to get involved. These challenges include technical, psychological, and motivational factors.[22]

To address these challenges, initiatives like the Linux Kernel Mentorship Program (https://wiki.linuxfoundation.org/lkmp) aim to recruit and train new developers. The LFX Mentorship Program also seeks to sponsor and mentor the next generation of open source developers and leaders across various projects.[23]

Motivations of programmers

With the growth and attention on the open-source movement, the reasons and motivations of programmers for creating code for free has been under investigation. In a paper from the 15th Annual Congress of the European Economic Association on the open-source movement, the incentives of programmers on an individual level as well as on a company or network level were analyzed. What is essentially the intellectual gift giving of talented programmers challenges the "self-interested-economic-agent paradigm",[24] and has made both the public and economists search for an understanding of what the benefits are for programmers.

- **Altruism**: The argument for altruism is limited as an explanation because though some exists, the programmers do not focus their kindness on more charitable causes. If the generosity of working for free was a viable motivation for such a prevalent movement, it is curious why such a trend has not been seen in industries such as biotechnology that would have a much bigger impact on the public good.[24]
- **Community sharing and improvement**: The online community is an environment that promotes continual improvements, modifications, and contributions to each other's work. A programmer can easily benefit from open-source software because by making it public, other testers and subprograms can remove bugs, tailor code to other purposes, and find problems. This kind of peer-editing feature of open-source software promotes better programs and a higher standard of code.[24]
- **Recognition**: Though a project may not be associated with a specific individual, the contributors are often recognized and marked on a project's server or awarded social reputation. This allows for programmers to receive public recognition for their skills, promoting career opportunities and exposure. In fact, the founders of Sun Microsystems and Netscape began as open-source programmers.[24]
- **Ego**: "If they are somehow assigned to a trivial problem and that is their only possible task, they may spend six months coming up with a bewildering architecture...merely to show their friends and colleagues what a tough nut they are trying to crack."[25] Ego-gratification has been cited as a relevant motivation of programmers because of their competitive community.[25] An OSS (open-source software) community has no clear distinction between developers and users, because all users are potential developers. There is a large community of programmers trying to essentially outshine or impress their colleagues.[26] They enjoy having other programmers admire their works and accomplishments, contributing to why OSS projects have a recruiting advantage for unknown talent than a closed-source company.[25]
- **Creative expression**: Personal satisfaction also comes from the act of writing software as an equivalent to creative self-expression – it is almost equivalent to creating a work of art. The rediscovery of creativity, which has been lost through the mass production of commercial software products can be a relevant motivation.[27]

Gender diversity of programmers

The vast majority of programmers in open-source communities are male. In a 2006 study for the European Union on free and open-source software communities, researchers found that only 1.5% of all contributors are female.[28] Although women are generally underrepresented in computing, the percentage of women in tech professions is actually much higher, close to 25%.[29] This discrepancy suggests that female programmers are overall less likely than male programmers to participate in open-source projects.

Some research and interviews with members of open-source projects have described a male-dominated culture within open-source communities that can be unwelcoming or hostile towards females.[30] There are initiatives such as Outreachy that aim to support more women and other underrepresented gender identities to participate in open-source software. However, within the discussion forums of open-source projects the topic of gender diversity can be highly controversial and even inflammatory.[30] A central vision in open-source software is that because the software is built and maintained on the merit of individual code contributions, open-source communities should act as a meritocracy.[31] In a meritocracy, the importance of an individual in the community depends on the quality of their individual contributions and not demographic factors such as age, race, religion, or gender. Thus proposing changes to the community based on gender, for example, to make the community more inviting towards females, go against the ideal of a meritocracy by targeting certain programmers by gender and not based on their skill alone.[30]

There is evidence that gender does impact a programmer's perceived merit in the community. A 2016 study identified the gender of over one million programmers on GitHub, by linking the programmer's GitHub account to their other social media accounts.[32] Between male and female programmers, the researchers found that female programmers were actually *more likely* to have their pull requests accepted into the project than male programmers, however only when the female had a gender-neutral profile. When females had profiles with a name or image that identified them as female, they were less likely than male programmers to have their pull requests accepted. Another study in 2015 found that of open-source projects on GitHub, gender diversity was a significant positive predictor of a team's productivity, meaning that open-source teams with a more even mix of different genders tended to be more highly productive.[31]

Many projects have adopted the Contributor Covenant code of conduct in an attempt to address concerns of harassment of minority developers. Anyone found breaking the code of conduct can be disciplined and ultimately removed from the project.

In order to avoid offense to minorities many software projects have started to mandate the use of inclusive language and terminology.[33]

Evidence of open-source adoption

Libraries are using open-source software to develop information as well as library services. The purpose of open source is to provide a software that is cheaper, reliable and has better quality. The one feature that makes this software so sought after is that it is free. Libraries in particular benefit from this movement because of the resources it provides. They also promote the same ideas of learning and understanding new information through the resources of other people. Open source allows a sense of community. It is an invitation for anyone to provide information about various topics. The open-source tools even allow libraries to create web-based catalogs. According to the IT source there are various library programs that benefit from this.[34]

Government agencies and infrastructure software — Government Agencies are utilizing open-source infrastructure software, like the Linux operating system and the Apache Web-server into software, to manage information.[35] In 2005, a new government lobby was launched under the name National Center for Open Source Policy and Research (NCOSPR) "a non-profit organization promoting the use of open source software solutions within government IT enterprises."[36]

Open-source movement in the military — Open-source movement has potential to help in the military. The open-source software allows anyone to make changes that will improve it. This is a form of invitation for people to put their minds together to grow a software in a cost efficient manner. The reason the military is so interested is because it is possible that this software can increase speed and flexibility. Although there are security setbacks to this idea due to the fact that anyone has access to change the software, the advantages can outweigh the disadvantages. The fact that the open-source programs can be modified quickly is crucial. A support group was formed to test these theories. The Military Open Source Software Working Group was organized in 2009 and held over 120 military members. Their purpose was to bring together software developers and contractors from the military to discover new ideas for reuse and collaboration. Overall, open-source software in the military is an intriguing idea that has potential drawbacks but they are not enough to offset the advantages.[37]

Open source in education — Colleges and organizations use software predominantly online to educate their students. Open-source technology is being adopted by many institutions because it can save these institutions from paying companies to provide them with these administrative software systems. One of the first major colleges to adopt an open-source system was Colorado State University in 2009 with many others following after that. Colorado State Universities system was produced by the Kuali Foundation who has become a major player in open-source administrative systems. The Kuali Foundation defines itself as a group of organizations that aims to "build and sustain open-source software for higher education, by higher education."[38] There are many other examples of open-source instruments being used in education other than the Kuali Foundation as well.

"For educators, The Open Source Movement allowed access to software that could be used in teaching students how to apply the theories they were learning".[39] With open networks and software, teachers are able to share lessons, lectures, and other course materials within a community. OpenTechComm is a program that is dedicated to "open to access, open to use, and open to edit — textbook or pedagogical resource that teachers of technical and professional communication courses at every level can rely on to craft free offerings to their students."[40] As stated earlier, access to programs like this would be much more cost efficient for educational departments.

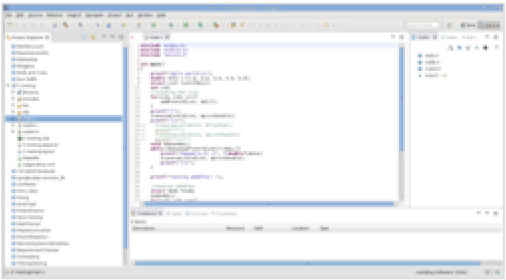
Open source in healthcare — Created in June 2009 by the nonprofit eHealthNigeria, the open-source software OpenMRS is used to document health care in Nigeria. The use of this software began in Kaduna, Nigeria to serve the purpose of public health. OpenMRS manages features such as alerting health care workers when patients show warning signs for conditions and records births and deaths daily, among other features. The success of this software is caused by its ease of use for those first being introduced to the technology, compared to more complex proprietary healthcare software available in first world countries. This software is community-developed and can be used freely by anyone, characteristic of open-source applications. So far, OpenMRS is being used in Rwanda, Mozambique, Haiti, India, China, and the Philippines.[41] The impact of open source in healthcare is also observed by Apelon Inc, the "leading provider of terminology and data interoperability solutions". Recently, its Distributed Terminology System (Open DTS) began supporting the open-source MySQL database system. This essentially allows for open-source software to be used in healthcare, lessening the dependence on expensive proprietary healthcare software. Due to open-source software, the healthcare industry has available a free open-source solution to implement healthcare standards. Not only does open source benefit healthcare economically, but the lesser dependence on proprietary software allows for easier integration of various systems, regardless of the developer.[42]

Companies

IBM

Originally, IBM was not the company that branched out to any means of open source software. They upheld into believing that intellectual property along with other privatized means of software around the 1990s.[43] From a citation, it wasn't until IBM was challenged by the evolving competitive market, specifically from Microsoft, that they decided to invest their resources more into open source software. Since then, their focus shifted more on customer service and a more robust software support.[43] IBM has been a leading proponent of the Open Source Initiative, and began supporting Linux in 1998.[44]

As another example, IBM had decided to make the Eclipse IDE(integrated development environment) open-source causing other companies to release their other IDEs due to Eclipse's popularity and outreach to the market.[45]



The Eclipse IDE, that IBM released as open source

Microsoft

Before summer of 2008, Microsoft has generally been known as an enemy of the open-source community. The company's anti-open-source sentiment was enforced by former CEO Steve Ballmer, who referred to Linux, a widely used open-source software, as a "cancer that attaches itself ... to everything it touches."[46] Microsoft also threatened Linux that they would charge royalties for violating 235 of their patents.

In 2004, Microsoft lost a European Union court case,[47] and lost the appeal in 2007,[48] and their further appeal in 2012:[49] being convicted of abusing its dominant position. Specifically they had withheld inter-operability information with the open-source Samba (software) project, which can be run on many platforms and aims to "removing barriers to interoperability".

In 2008, however, Sam Ramji, the then head of open-source-software strategy in Microsoft, began working closely with Bill Gates to develop a pro-open-source attitude within the software industry as well as Microsoft itself. Ramji, before leaving the company in 2009, built Microsoft's familiarity and involvement with open source, which is evident in Microsoft's contributions of open-source code to

Microsoft Azure among other projects. These contributions would have been previously unimaginable by Microsoft.[50] Microsoft's change in attitude about open source and efforts to build a stronger open-source community is evidence of the growing adoption and adaptation of open source.[51]

See also

Free and open-source software portal

- Digital rights
- Diversity in open-source software
- List of free and open-source software packages
- List of open-source hardware projects
- Mass collaboration
- Open-design movement
- Open-source model
- Open-source appropriate technology
- Open-source hardware
- Open-source governance
- Sharing economy
- P2P economic system
- Peer production
- *The Cathedral and the Bazaar*, a book by Eric S. Raymond on software engineering methods
- *The Virtual Revolution*

References

1. Levine, Sheen S.; Prietula, M. J. (2013). "Open Collaboration for Innovation: Principles and Performance". *Organization Science*. **25** (5): 1414–1433. arXiv:1406.7541 (https://arxiv.org/abs/1406.7541). doi:10.1287/orsc.2013.0872 (https://doi.org/10.1287%2Forsc.2013.0872). S2CID 6583883 (https://api.semanticscholar.org/CorpusID:6583883). SSRN 1096442 (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1096442).
2. Wyllys, R.E. (2000). Overview of the Open-Source Movement (http://www.ischool.utexas.edu/~l38613dw/readings/OpenSourceOverview.html) Archived (https://web.archive.org/web/20160316131053/https://www.ischool.utexas.edu/~l38613dw/readings/OpenSourceOverview.html) 2016-03-16 at the Wayback Machine. Retrieved November 22, 2009, from The University of Texas at Austin Graduate School of Library & Information Science
3. Warger, T. (2002)The Open Source Movement (http://net.educause.edu/ir/library/pdf/eqm0233.pdf) Archived (https://web.archive.org/web/20110717100415/http://net.educause.edu/ir/library/pdf/eqm0233.pdf) 2011-07-17 at the Wayback Machine. Retrieved November 22, 2009, from Education Resources Information Center
4. Tiemann, Michael (September 19, 2006). "History of the OSI" (http://www.opensource.org/history). Open Source Initiative. Archived (https://web.archive.org/web/20021001164015/http://www.opensource.org/docs/history.php) from the original on October 1, 2002. Retrieved August 23, 2008.

5. A Brief History of the Open-Source Movement (http://sloanreview.mit.edu/the-magazine/2000-fall/4211-3/a-brief-history-of-the-opensource-movement/) Archived (https://web.archive.org/web/20110411022933/http://sloanreview.mit.edu/the-magazine/2000-fall/4211-3/a-brief-history-of-the-opensource-movement/) 2011-04-11 at the Wayback Machine. Sloanreview.mit.edu (2011-11-18). Retrieved on 2011-11-30.
6. History of the OSI | Open Source Initiative (http://www.opensource.org/history) Archived (https://web.archive.org/web/20021001164015/http://www.opensource.org/docs/history.php) 2002-10-01 at the Wayback Machine. Opensource.org. Retrieved on 2011-11-30.
7. Weber, Steven. The Success of Open Source. The President and Fellows of Harvard College. 2004. Print pg.20–28. ISBN 978-0-674-01858-7 This whole paragraph is referenced to Steven Weber
8. Tennant, D. (2008, August 11). Standing on Principle. Computerworld, p. 4. Retrieved from Business Source Premier database.
9. Taft, D. K. (2009, November 3). Microsoft Recommits to $100k Apache Contribution at ApacheCon (https://www.eweek.com/servers/microsoft-recommits-to-100k-apache-contribution-at-apachecon) Archived (https://web.archive.org/web/20231120202137/https://www.eweek.com/servers/microsoft-recommits-to-100k-apache-contribution-at-apachecon/) 2023-11-20 at the Wayback Machine. Retrieved May 8, 2020 from eWeek
10. Elliott, M. S.; Scacchi, Walt (2008). "Mobilization of software developers: The free software movement". *Information Technology & People*. **21** (1): 4. doi:10.1108/09593840810860315 (https://doi.org/10.1108%2F09593840810860315).
11. Lerner, Josh; Tirole, Jean (March 2000). "The simple Economics of Open Source" (http://www.people.hbs.edu/jlerner/simple.pdf) (PDF). Cambridge, MA.: National Bureau of Economic Research. CiteSeerX 10.1.1.145.3577 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.145.3577). Archived (https://web.archive.org/web/20171215003553/http://www.people.hbs.edu/jlerner/simple.pdf) (PDF) from the original on 2017-12-15. Retrieved 2015-06-10.
`{{cite journal}}`: Cite journal requires `|journal=` (help)
12. Stallman, R. M. (2007). Why "Free Software" is better than "Open Source" (https://www.gnu.org/philosophy/free-software-for-freedom.html#relationship) Archived (https://web.archive.org/web/20210327080246/https://www.gnu.org/philosophy/free-software-for-freedom.html#relationship) 2021-03-27 at the Wayback Machine. Retrieved November 22, 2009, from GNU.org
13. The Open Source Definition | Open Source Initiative (http://www.opensource.org/docs/osd) Archived (https://web.archive.org/web/20070607054201/http://www.opensource.org/docs/osd) 2007-06-07 at the Wayback Machine. Opensource.org. Retrieved on 2011-11-30.
14. Sullivan, John L. (2011). "Free, Open Source Software Advocacy as a Social Justice Movement: The Expansion of F/OSS Movement Discourse in the 21st Century". *Journal of Information Technology & Politics*. **8** (3): 223–239. doi:10.1080/19331681.2011.592080 (https://doi.org/10.1080%2F19331681.2011.592080). S2CID 144013228 (https://api.semanticscholar.org/CorpusID:144013228).
15. Ceraso, A.; Pruchnic, J. (2011). "Introduction: Open source culture and aesthetics" (https://web.archive.org/web/20170922004457/http://digitalcommons.wayne.edu/cgi/viewcontent.cgi?article=1798&context=criticism). *Criticism*. **53** (3): 337. doi:10.1353/crt.2011.0026 (https://doi.org/10.1353%2Fcrt.2011.0026). S2CID 31800586 (https://api.semanticscholar.org/CorpusID:31800586). Archived from the original (https://digitalcommons.wayne.edu/cgi/viewcontent.cgi?article=1798&context=criticism) on 2017-09-22.
16. Nakakoji, Kumiyo; Yamamoto, Yasuhiro; Nishinaka, Yoshiyuki; Kishida, Kouichi; Ye, Yunwen (2002). "Evolution patterns of open-source software systems and communities". *Proceedings of the international workshop on Principles of software evolution - IWPSE '02*. p. 76. doi:10.1145/512035.512055 (https://doi.org/10.1145%2F512035.512055). ISBN 978-1-58113-545-9. S2CID 15341686 (https://api.semanticscholar.org/CorpusID:15341686).

17. Crowston, Kevin; Howison, James (7 February 2005). "The social structure of free and open source software development" (https://doi.org/10.5210%2Ffm.v10i2.1207). *First Monday*. doi:10.5210/fm.v10i2.1207 (https://doi.org/10.5210%2Ffm.v10i2.1207).
18. Sheoran, Jyoti; Blincoe, Kelly; Kalliamvakou, Eirini; Damian, Daniela; Ell, Jordan (2014). "Understanding 'watchers' on GitHub". *Proceedings of the 11th Working Conference on Mining Software Repositories - MSR 2014*. pp. 336–339. doi:10.1145/2597073.2597114 (https://doi.org/10.1145%2F2597073.2597114). ISBN 978-1-4503-2863-0. S2CID 11496776 (https://api.semanticscholar.org/CorpusID:11496776).
19. Middleton, Justin; Murphy-Hill, Emerson; Green, Demetrius; Meade, Adam; Mayer, Roger; White, David; McDonald, Steve (2018). "Which contributions predict whether developers are accepted into github teams". *Proceedings of the 15th International Conference on Mining Software Repositories*. pp. 403–413. doi:10.1145/3196398.3196429 (https://doi.org/10.1145%2F3196398.3196429). ISBN 978-1-4503-5716-6. S2CID 13695100 (https://api.semanticscholar.org/CorpusID:13695100).
20. Robles, G; J. M. Gonzalez-Barahona; I. Herraiz (2009). "Evolution of the core team of developers in libre software projects". *Proceedings of the 6th International Conference on Mining Software Repositories*: 167–170.
21. Vaughan-Nichols, Steven J. (2024-07-15). "The graying open source community needs fresh blood" (https://www.theregister.com/2024/07/15/opinion_open_source_attract_devs/). *The Register*. Retrieved 2025-02-17.
22. Proven, Liam (2025-02-14). "Why are there so many graybeards in FOSS?" (https://www.theregister.com/2025/02/14/youngsters_in_foss/). *The Register*. Retrieved 2025-02-17.
23. "LFX Mentorship" (https://lfx.linuxfoundation.org/tools/mentorship/). *Linux Foundation*. Retrieved 2025-02-17.
24. Lerner, Josh; Jean Tirole (9 May 2001). "The open source movement: Key research questions". *European Economic Review*. **45** (4–6): 819–826. doi:10.1016/S0014-2921(01)00124-6 (https://doi.org/10.1016%2FS0014-2921%2801%2900124-6).
25. Greenspun, Philip. "Managing Software Engineers" (http://philip.greenspun.com/ancient-history/managing-software-engineers). Archived (https://web.archive.org/web/20211009053631/https://philip.greenspun.com/ancient-history/managing-software-engineers) from the original on 9 October 2021. Retrieved 7 February 2012.
26. Yunwen Ye; Kishida, K. (2003). "Toward an understanding of the motivation of open source software developers". *25th International Conference on Software Engineering, 2003. Proceedings*. pp. 419–429. doi:10.1109/ICSE.2003.1201220 (https://doi.org/10.1109%2FICSE.2003.1201220). ISBN 0-7695-1877-X. S2CID 1476378 (https://api.semanticscholar.org/CorpusID:1476378).
27. Bonaccorsi, Andrea; Cristina Rossi (2003). "Why Open Source software can succeed" (http://www.lem.sssup.it/WPLem/files/2002-15.pdf) (PDF). *Open Source Software Development*. **32** (7): 1243–1258. doi:10.1016/S0048-7333(03)00051-9 (https://doi.org/10.1016%2FS0048-7333%2803%2900051-9). hdl:10419/89290 (https://hdl.handle.net/10419%2F89290). Archived (https://web.archive.org/web/20200919230218/http://www.lem.sssup.it/WPLem/files/2002-15.pdf) (PDF) from the original on 2020-09-19. Retrieved 2019-12-14.
28. Nafus, Dawn, James Leach, and Bernhard Krieger. "Gender: Integrated report of findings." *FLOSSPOLS, Deliverable D* 16 (2006).
29. "Women in tech: The facts" (https://web.archive.org/web/20210514211516/https://www.ncwit.org/sites/default/files/resources/womenintech_facts_fullreport_05132016.pdf) (PDF). Archived from the original on 14 May 2021. Retrieved 19 April 2021.
30. Nafus, Dawn (1 June 2012). " 'Patches don't have gender': What is not open in open source software". *New Media & Society*. **14** (4): 669–683. doi:10.1177/1461444811422887 (https://doi.org/10.1177%2F1461444811422887). S2CID 206727320 (https://api.semanticscholar.org/CorpusID:206727320).

31. Vasilescu, Bogdan; Posnett, Daryl; Ray, Baishakhi; Van Den Brand, Mark G.J.; Serebrenik, Alexander; Devanbu, Premkumar; Filkov, Vladimir (2015). "Gender and Tenure Diversity in GitHub Teams". *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. pp. 3789–3798. doi:10.1145/2702123.2702549 (https://doi.org/10.1145%2F2702123.2702549). ISBN 978-1-4503-3145-6. S2CID 11705263 (https://api.semanticscholar.org/CorpusID:11705263).
32. Terrell, Josh; Kofink, Andrew; Middleton, Justin; Rainear, Clarissa; Murphy-Hill, Emerson; Parnin, Chris; Stallings, Jon (1 May 2017). "Gender differences and bias in open source: pull request acceptance of women versus men" (https://doi.org/10.7717%2Fpeerj-cs.111). *PeerJ Computer Science*. **3**: e111. doi:10.7717/peerj-cs.111 (https://doi.org/10.7717%2Fpeerj-cs.111).
33. "Inclusive Naming Initiative" (https://inclusivenaming.org/). Archived (https://web.archive.org/web/20220203010436/https://inclusivenaming.org/) from the original on 3 February 2022. Retrieved 22 October 2021.
34. Poynder, Richard (2001). "The Open Source Movement" (http://www.infotoday.com/it/oct01/poynder.htm). *Information Today*. **8** (9). Archived (https://web.archive.org/web/20110527021708/http://www.infotoday.com/it/oct01/poynder.htm) from the original on 2011-05-27. Retrieved 2011-01-20.
35. Moore, John (14 August 2008). "A starring role for open source? -" (https://fcw.com/articles/2008/08/14/a-starring-role-for-open-source.aspx). *FCW*. Archived (https://web.archive.org/web/20210419144951/https://fcw.com/articles/2008/08/14/a-starring-role-for-open-source.aspx) from the original on 19 April 2021. Retrieved 19 April 2021.
36. Preimesberger, Chris (14 October 2005). "Open Source Movement Gets a Lobby" (https://www.eweek.com/servers/open-source-movement-gets-a-lobby/). *eWEEK*. Archived (https://web.archive.org/web/20231120202141/https://www.eweek.com/servers/open-source-movement-gets-a-lobby/) from the original on 20 November 2023. Retrieved 19 April 2021.
37. Toon, John (2009). "Open Source Movement May Accelerate Military Software Development" (http://www.gtri.gatech.edu/casestudy/open-source-mil-oss-military-software). Georgia Tech Research Institute. Archived (https://web.archive.org/web/20170711140410/https://www.gtri.gatech.edu/casestudy/open-source-mil-oss-military-software) from the original on 2017-07-11. Retrieved 2011-12-21.
38. "About the Kuali Community | Kuali Foundation" (http://web.archive.org/web/20100612222235/https://www.kuali.org/about). 2010-06-12. Archived from the original (https://www.kuali.org/about) on 2010-06-12. Retrieved 2024-09-14.
39. St.Amant & Ballentine 2011 p.343
40. Still, Brian (2010). "A Dozen Years after Open Source's 1998 Birth, It's Time for OpenTechComm" (https://doi.org/10.2190/TW.40.2.g). *Journal of Technical Writing and Communication*. **40** (2). Retrieved 2025-01-29.
41. eHealthNigeria. (2012). eHealthNigeria: FAQs (http://ehealthnigeria.org/home/faq/) Archived (https://web.archive.org/web/20120104064547/http://ehealthnigeria.org/home/faq/) 2012-01-04 at the Wayback Machine. Retrieved Feb 4, 2012
42. "Apelon Announces Availability of a Completely Open Source Terminology Management Solution" (https://web.archive.org/web/20220302204400/https://www.apelon.com/news/item/apelon-releases-an-open-source-terminology-solution) (Press release). Apelon. 17 January 2012. Archived from the original (https://www.apelon.com/news/item/apelon-releases-an-open-source-terminology-solution) on 2 March 2022. Retrieved 19 April 2021.

43. Munga, Neeshal; Fogwill, Thomas; Williams, Quentin (2009-10-12). "The adoption of open source software in business models: A Red Hat and IBM case study" (https://dl.acm.org/doi/abs/10.1145/1632149.1632165). *Proceedings of the 2009 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists*. SAICSIT '09. New York, NY, USA: Association for Computing Machinery. pp. 112–121. doi:10.1145/1632149.1632165 (https://doi.org/10.1145%2F1632149.1632165). ISBN 978-1-60558-643-4.
44. "IBM launches biggest Linux lineup ever" (https://web.archive.org/web/19991110114228/http://www.ibm.com/news/1999/03/02.phtml). IBM. March 2, 1999. Archived from the original (http://www.ibm.com/news/1999/03/02.phtml) on November 10, 1999.
45. Fitzgerald, Brian (2006). "The Transformation of Open Source Software" (https://www.jstor.org/stable/25148740). *MIS Quarterly*. **30** (3): 587–598. doi:10.2307/25148740 (https://doi.org/10.2307%2F25148740). ISSN 0276-7783 (https://search.worldcat.org/issn/0276-7783). JSTOR 25148740 (https://www.jstor.org/stable/25148740).
46. "Microsoft CEO takes launch break with the Sun-Times" (http://www.suntimes.com/output/tech/cst-fin-micro01.html). *Chicago Sun-Times*. June 1, 2001. Archived (https://web.archive.org/web/20011108013601/http://www.suntimes.com/output/tech/cst-fin-micro01.html) from the original on 2001-11-08. Retrieved June 27, 2021.
47. "Groklaw.net – The EU Microsoft Decision – December 2004" (https://web.archive.org/web/20141022024339/http://www.groklaw.net/articlebasic.php?story=20041222205501273). Archived from the original (http://www.groklaw.net/articlebasic.php?story=20041222205501273) on 22 October 2014. Retrieved 13 October 2014.
48. "Groklaw.net – EU Ct. of 1st Instance: Microsoft Abused its Dominant Position – Updated – September 2007" (http://www.groklaw.net/article.php?story=20070917053717322). Archived (https://web.archive.org/web/20150924023918/http://www.groklaw.net/article.php?story=20070917053717322) from the original on 24 September 2015. Retrieved 13 October 2014.
49. "Groklaw.net Microsoft Loses Its EU Appeal" (https://web.archive.org/web/20141022024358/http://www.groklaw.net/articlebasic.php?story=20120627094954981). Archived from the original (http://www.groklaw.net/articlebasic.php?story=20120627094954981) on 22 October 2014. Retrieved 13 October 2014.
50. Metz, Cade (30 January 2012). "Meet Bill Gates, the Man Who Changed Open Source Software" (https://www.wired.com/2012/01/meet-bill-gates/). *Wired*. Archived (https://web.archive.org/web/20210419144949/https://www.wired.com/2012/01/meet-bill-gates/) from the original on 19 April 2021. Retrieved 19 April 2021.
51. Metz, Cade (4 November 2011). "How Microsoft Learned to Stop Worrying and (Almost) Love Open Source" (https://www.wired.com/2011/11/how-microsoft-learned-to-stop-worrying-and-almost-love-open-source/). *Wired*. Archived (https://web.archive.org/web/20211025170156/https://www.wired.com/2011/11/how-microsoft-learned-to-stop-worrying-and-almost-love-open-source/) from the original on 25 October 2021. Retrieved 19 April 2021.

Further reading

- Ljungberg, J (2000). "Open source movements as a model for organising". *European Journal of Information Systems*. **9** (4): 208–216. doi:10.1057/palgrave.ejis.3000373 (https://doi.org/10.1057%2Fpalgrave.ejis.3000373).
- Meeker, Heather (2008). *The Open Source Alternative: Understanding Risks and Leveraging Opportunities*. John Wiley & Sons. ISBN 978-0-470-19495-9.
- Schrape, Jan-Felix (2019). "Open-source projects as incubators of innovation. From niche phenomenon to integral part of the industry". *Convergence*. **25** (3): 409–427. doi:10.1177/1354856517735795 (https://doi.org/10.1177%2F1354856517735795). S2CID 149165772 (https://api.semanticscholar.org/CorpusID:149165772).

- Software Freedom Law Center (3 March 2008). "A Legal Issues Primer for Open Source and Free Software Projects" (http://www.softwarefreedom.org/resources/2008/foss-primer.html).
- The Open Source Movement (24 October 2010). "The Open Source Movement" (https://web.archive.org/web/20130729110903/http://www.theopensourcemovement.com/). Archived from the original (http://www.theopensourcemovement.com) on 29 July 2013. Retrieved 25 April 2020.
- Rosen, Lawrence (July 2009). "Bad facts make good law: The Jacobsen case and Open Source" (http://www.ifosslr.org/ifosslr/article/view/5/9). *International Free and Open Source Software Law Review*. **1** (1). Software Freedom Law Center, Inc: 27–32.
- Goettsch, Kerry D. (2003). "SCO Group v. IBM: The Future of Open Source Software" (http://heinonline.org/HOL/Page?handle=hein.journals/jltp2003&div=26&g_sent=1&collection=journals#588). *Journal of Law, Technology & Policy*: 581.
- Wayner, P. (2000). *Free for all: how Linux and the free software movement undercut the high-tech titans* (https://archive.org/details/freeforall00pete). Harperbusiness. ISBN 978-0-06-662050-3.
- Kirk St. Amant (2011). "Open Source Software, Access, and Content creation in the global economy" (https://www.researchgate.net/publication/274346645).
- James McIntyre (January 31, 2012). "The Online Revolution" (https://web.archive.org/web/20130718231856/http://ssy.org.uk/2012/01/the-online-revolution/). Scottish Socialist Youth. Archived from the original (http://ssy.org.uk/2012/01/the-online-revolution/) on July 18, 2013.
- Weber, Steven (2005). *The Success of Open Source*. Cambridge: Harvard University Press. ISBN 9780674018587.
- Carillo, Kevin; Okoli, Chitu (2008). "The Open Source Movement: A Revolution in Software Development". *Journal of Computer Information Systems*. **49** (2). doi:10.1080/08874417.2009.11646043 (https://doi.org/10.1080%2F08874417.2009.11646043) (inactive 1 November 2024).
- Tozzi, Christopher (2017). *For Fun and Profit: A History of the Free and Open Source Software Revolution* (https://direct.mit.edu/books/monograph/3528/For-Fun-and-ProfitA-History-of-the-Free-and-Open). MIT Press. ISBN 9780262341172.

External links

- LFX Mentorship Program (https://lfx.linuxfoundation.org/tools/mentorship/) by the Linux Foundation

Free Software Movement of India

Free Software Movement of India (**FSMI**) is a national coalition of various regional and sectoral free software movements operating in different parts of India. The formation of FSMI was announced in the valedictory function of the National Free Software Conference - 2010 held in Bangalore during 20–21 March 2010.[1] FSMI is a pan Indian level initiative to propagate the ideology of free software and to popularize the usage of the free software. One of the declared aims of the movement is to take Free Software and its ideological implications to computer users "across the digital divide", to under-privileged sections of society.[2][3]

Free Software Movement of India

Free Software is the future, Future is ours

Abbreviation	FSMI
Formation	March 21, 2010
Type	Coalition of Organisations
Purpose	To take free software and its ideology to all computer users and to all sections of society.
Region served	Indian Union
President	Prabir Purkayastha
General Secretary	Kiran Chandra Yarlagada
Main organ	General Council
Website	fsmi.in (https://www.fsmi.in/)

Member organisations

FSMI differentiates itself from other organisations, forums or user groups in the free software domain by the method of movement building which is primarily grass root and mass movement.[4][5]

- **Regional movements**
  - Democratic Alliance of Knowledge Front, Kerala (abbrv. DAKF)
  - Free Software Movement of Delhi/NCR
  - Free Software Movement of Karnataka (abbrv. FSMK)
  - Free Software Movement of Maharashtra (abbrv. FSMM)
  - Free Software Movement of Rajasthan
  - Free Software Foundation Tamilnadu (abbrv. FSFTN)
  - Free Software Movement of West Bengal
  - Swadhin, Odisha
  - Swecha, Telangana & Andhra Pradesh
- **Sectoral movements**
  - Appropriate Technology Promotion Society
  - Knowledge Commons
  - National Consultative Committee of Computer Teachers Association(abbrv. NCCCTA)
  - Open Source Geospatial Foundation India (abbrv. OSGEO India)

Sectoral movements such as Knowledge Commons, Academics Initiative, OSGEO India and the National Consultative Committee of Computer Teachers (NCCCTA) joined the national coalition at the very initial stage itself.

Governance

FSMI elects the General Council, Executive Committee, Office Bearers including General Secretary & President in FSMI National Conferences.

The founding conference of FSMI elected a General Council having 69 members, an Executive Committee with 28 members with Joseph Thomas as the founding President and Kiran Chandra Yarlagadda as founding General Secretary. The second national conference named as 4Ccon (https://fsftn.gitlab.io/4ccon/) was held in B.S. Abdur Rahman University, Vandalur, Chennai. Prabir Purkayastha was elected as the President and Kiran Chandra Yarlagadda was re-elected as the General Secretary at the second conference.

Objectives

FSMI is a pan Indian level initiative to propagate the ideology of free software and to popularize the usage of the free software.[2][6] One of the declared aims of the movement is to take Free Software and its ideological implications to computer users "across the digital divide", to under-privileged sections of society.[2][3]

Activities

- The All India Council for Technical Education (AICTE) had made mandatory the use of the proprietary Office 365[7] in all engineering colleges in the country. This mandatory mandate was rescinded after intervention by several groups led by FSMI.[8]
- The Free Software Movement of India (FSMI) has alleged that the new Guidelines for Examination of Computer Related Inventions are illogical. It argues that they violate the spirit and law contained in the amended Patents Act of 1970 and could pose a grave threat to innovation in our country. It cautioned that permitting software patents through the backdoor would severely restrict the innovative capabilities and would adversely impact the creative and entrepreneurial spirit of the youth in the India.[9]
- FSMI has strongly criticized the Indian Telecom Giant Airtel for tampering user's online communications on its 3G network, and maliciously inserting advertisements into its user's data. In a statement issued they also stated that Airtel acted in violation of the privacy rights of at least thousands if not millions of Indian citizens and has rendered their online communications unsafe.[10] FSMI appealed to all groups working on net neutrality to oppose Airtel and Flash Networks illegal and unethical actions and share the code that exposes Airtel widely, thus defeating corporate attempts to muzzle citizens right to free speech (which are only intended to cover up acts of corporate malfeasance).
- A call to Protest on 9 June 2012 for Internet Freedom was given by FSMI to which there was a good response across the country. In Hyderabad police scuttled a planned protest by activists of Free Software Movement of India (FSMI). A good number of software engineers, students and other campaigners assembled near People's Plaza on the Necklace Road.A

senior police official threatened FSMI activists with arrest and confiscation of all protest material and T-shirts. The activists contended that theirs was a token protest and they had no plans to organise a rally. The police initially took Y. Kiran Chandra, General Secretary of FSMI in to policy custody along with fellow activists. After hour-long arguments, the police spoke to higher officials who directed them to be shifted to Indira Park. The protestors left the venue with a police patrol team piloting the protest for.[11]
- National Convention for Academics and Research was organised by FSMI in December 2011. The conference was inaugurated by former President of India Dr A P J Abdul Kalam. The concept of Free Software, wherein knowledge is created by the community for the community, without being driven by commercial interests, must be extended to research to solve problems in health care, agriculture, energy and safe drinking water, the former President A.P.J. Abdul Kalam, said.[12][13]

Controversy

A report in the Malayalam daily *Mathrubhumi* stated that the new organisation was under the tutelage of certain political parties and that it had driven a wedge in the free software movement in India by sidetracking the Free Software Foundation of India (FSFI).[14] The report in *Mathrubhumi* was criticized by FSMI in another newspaper for not upholding basic journalistic ethics and code of conduct of contacting the FSMI leadership before making allegations.[15] A rejoinder to the report has been published by FSMI.[16] The *Mathrubhumi* article stated that the general secretary of FSMI, Kiran Chandra was a representative of Novell while the FSFI website showed that he is on the Board of Directors.[17]

See also

Free and open-source software portal

- Free Software Movement
- History of free and open-source software
- List of free software project directories
- List of free and open-source software organizations
- Comparison of free and open-source software licences
- Software Freedom Law Center

References

1. "National Free Software Coalition Formed" (https://web.archive.org/web/20100325191812/http://www.hindu.com/2010/03/22/stories/2010032263622000.htm). *The Hindu*. Chennai, India. 22 March 2010. Archived from the original (http://www.hindu.com/2010/03/22/stories/2010032263622000.htm) on 25 March 2010. Retrieved 5 April 2010.
2. "Free Software Movement has arrived on the National Scene!" (https://web.archive.org/web/20100327082545/http://www.nc2010.fsmk.org/content/free-software-movement-has-arrived-national-scene). Archived from the original (http://www.nc2010.fsmk.org/content/free-software-movement-has-arrived-national-scene) on 2010-03-27. Retrieved 23 March 2010.

3. "സ്വതന്ത്ര സോഫ്റ്റ് വെയർ പ്രസ്ഥാനം ഉയരങ്ങളിലേക്ക്" (https://web.archive.org/web/20160130185457/http://www.deshabhimani.com/Profile.php?user=146982). *www.deshabhimani.com* (in Malayalam). Archived from the original (http://www.deshabhimani.com/Profile.php?user=146982) on 2016-01-30. Retrieved 2010-03-23.
4. "Freedom movement" (https://web.archive.org/web/20100728062637/http://www.flonnet.com/fl2709/stories/20100507270909400.htm). Frontline. Archived from the original (http://www.flonnet.com/fl2709/stories/20100507270909400.htm) on 28 July 2010. Retrieved 21 April 2010.
5. "Broad front" (http://www.flonnet.com/stories/20100507270909600.htm). Retrieved 21 April 2010.
6. Deepa Kurup (22 March 2010). "National Free Software coalition formed" (https://web.archive.org/web/20100325191812/http://www.hindu.com/2010/03/22/stories/2010032263622000.htm). *The Hindu*. Chennai, India. Archived from the original (http://www.hindu.com/2010/03/22/stories/2010032263622000.htm) on 25 March 2010. Retrieved 23 March 2010.
7. Hindu, The (9 May 2013). "News Paper" (http://www.thehindu.com/features/education/college-and-university/aicte-rescinds-microsoft-office-365-mandate/article4698205.ece). *The Hindu*. Kasturi and Sons.
8. Mukunth, Vasudevan; Srivas, Anuj (10 May 2013). "AICTE rescinds Microsoft Office 365 mandate" (http://www.thehindu.com/todays-paper/tp-national/aicte-rescinds-microsoft-office-365-mandate/article4701358.ece). *The Hindu*. Chennai, India. Retrieved 20 July 2013.
9. viwanathsai (7 September 2015). "Free Software activists against changes to patent norms" (http://www.thehindubusinessline.com/info-tech/free-software-activists-against-changes-to-patent-norms/article7625717.ece). *The Hindu Business Line*. India. Retrieved September 8, 2015.
10. "Airtel 3G Script Injection" (https://web.archive.org/web/20150615003902/http://fsmi.in/content/press-release-airtel-3g-script-injection) (Press release). June 11, 2015. Archived from the original (https://fsmi.in/content/press-release-airtel-3g-script-injection) on June 15, 2015. Retrieved September 17, 2015.
11. http://www.thehindubusinessline.com/economy/curbs-on-web-freedom-police-scuttle-anonymous-protests/article3509520.ece (http://Internet%20Freedom)
12. "Dr A P J Abdul Kalam Inaugural Address" (https://web.archive.org/web/20130620102753/http://abdulkalam.com/kalam/jsp/display_content_front.jsp?menuid=28&menuname=Speeches%20%2F%20Lectures&linkid=68&linkname=Recent&content=1716&columnno=0&starts=0&menu_image=-). Archived from the original (http://abdulkalam.com/kalam/jsp/display_content_front.jsp?menuid=28&menuname=Speeches%20/%20Lectures&linkid=68&linkname=Recent&content=1716&columnno=0&starts=0&menu_image=-) on 2013-06-20. Retrieved 27 April 2012.
13. Kurup, Deepa (22 December 2010). "News Report in Hindu" (http://www.thehindu.com/sci-tech/article967827.ece). *The Hindu*. Chennai, India. Retrieved 27 April 2012.
14. "സ്വതന്ത്ര സോഫ്റ്റ്‌വെയർ പ്രസ്ഥാനത്തെ പിളർത്തി; സി.പി.എം. പിന്തുണയോടെ പുതിയ സംഘടന" (https://web.archive.org/web/20100325082510/http://www.mathrubhumi.com/online/malayalam/news/story/221531/2010-03-22/kerala) (in Malayalam). Mathrubhumi. Archived from the original (http://www.mathrubhumi.com/online/malayalam/news/story/221531/2010-03-22/kerala) on 25 March 2010. Retrieved 23 March 2010.
15. Thomas, Joseph. "സ്വതന്ത്ര സോഫ്റ്റ് വെയർ പ്രസ്ഥാനം ഉയരങ്ങളിലേക്ക്" (https://web.archive.org/web/20160130185457/http://www.deshabhimani.com/Profile.php?user=146982) (in Malayalam). Deshabhimani. Archived from the original (http://www.deshabhimani.com/Profile.php?user=146982) on 2016-01-30. Retrieved 2010-03-23.
16. "Mathrubumi fabricated report" (https://web.archive.org/web/20100327082603/http://www.nc2010.fsmk.org/content/rejoinder-mathrubumis-fabricated-report). Archived from the original (http://www.nc2010.fsmk.org/content/rejoinder-mathrubumis-fabricated-report) on 2010-03-27. Retrieved 2010-03-24.

17. "Kiran Chandra, director in FSF-I board" (https://web.archive.org/web/20161218073900/http://civic.gnu.org.in/about/whos-who). Archived from the original (http://civic.gnu.org.in/about/whos-who) on 2016-12-18. Retrieved 2017-01-31.

Free Software Foundation

The **Free Software Foundation** (**FSF**) is a 501(c)(3) non-profit organization founded by Richard Stallman[6] on October 4, 1985. The organisation supports the free software movement, with the organization's preference for software being distributed under copyleft ("share alike") terms,[7] such as with its own GNU General Public License.[8] The FSF was incorporated in Boston[9] where it is also based.[10]

From its founding until the mid-1990s, FSF's funds were mostly used to employ software developers to write free software for the GNU Project[11] and its employees and volunteers have mostly worked on legal and structural issues for the free software movement and the free software community. Consistent with its goals, the FSF aims to use only free software on its own computers.[12]

The FSF holds the copyrights on many pieces of the GNU system, such as GNU Compiler Collection. As the holder of these copyrights, it has authority to enforce the copyleft requirements of the GNU General Public License (GPL) when copyright infringement occurs. The FSF is also the steward of several free software licenses, meaning it publishes them and has the ability to make revisions as needed.[13]

Free Software Foundation

Free Software Foundation logo

Abbreviation	FSF
Formation	October 4, 1985[1]
Founder	Richard Stallman
Type	501(c)(3) non-profit organization
Legal status	501(c)(3)
Purpose	Computer User Freedom (see Free software movement)
Headquarters	Remote work[2]
Region served	Worldwide
Membership	Individuals
President	Geoffrey Knauth
Executive director	Zoë Kooyman[3]
Revenue	$1,149,602[4] (2020)
Expenses	$1,809,358[4] (2020)
Staff	13[5]
Website	www.fsf.org (https://www.fsf.org)

History

The Free Software Foundation was founded in 1985 as a non-profit corporation supporting free software development. It continued existing GNU projects, such as the sale of manuals and tapes, and employed developers of the free software system.[14] Since then, it has continued these activities, as well as advocating for the free software movement. From 1991 until 2001, General Public License (GPL) enforcement was done informally, usually by Stallman himself, often with assistance from FSF's lawyer, Eben Moglen. Typically, GPL violations during this time were cleared up by short email exchanges between Stallman and the violator. In late 2001, Bradley M. Kuhn (then executive director), with the assistance of Moglen, David Turner, and Peter T. Brown, formalized these efforts into FSF's GPL Compliance Labs.[15][16][17] In the interest of promoting copyleft assertiveness by software companies to the level that the FSF was already doing, in 2004 Harald Welte launched gpl-violations.org.

From 2002–2004, high-profile GPL enforcement cases, such as those against Linksys and OpenTV, became frequent.[15][16][17] GPL enforcement and educational campaigns on GPL compliance was a major focus of the FSF's efforts during this period.[18][19] In March 2003, SCO filed suit against IBM alleging that IBM's contributions to various free software, including FSF's GNU, violated SCO's rights. While FSF was never a party to the lawsuit, FSF was subpoenaed on November 5, 2003.[20] During 2003 and 2004, FSF put substantial advocacy effort into responding to the lawsuit and quelling its negative impact on the adoption and promotion of free software.[21][22] From 2003 to 2005, FSF held legal seminars to explain the GPL and the surrounding law.[23] Usually taught by Bradley M. Kuhn and Daniel Ravicher, these seminars offered CLE credit and were the first effort to give formal legal education on the GPL.[22][24][25] In 2007, the FSF published the third version of the GNU General Public License after significant outside input.[26][27]

In December 2008, FSF filed a lawsuit against Cisco for using GPL-licensed components shipped with Linksys products. Cisco was notified of the licensing issue in 2003 but Cisco repeatedly disregarded its obligations under the GPL.[28] In May 2009, Cisco and FSF reached settlement under which Cisco agreed to make a monetary donation to the FSF and appoint a Free Software Director to conduct continuous reviews of the company's license compliance practices.[29]

In September 2019, Richard Stallman resigned as president of the FSF after pressure from journalists and members of the open source community in response to him making controversial comments in defense of Marvin Minsky on Jeffrey Epstein's sex trafficking scandal.[30] Nevertheless, Stallman remained head of the GNU Project and in 2021, he returned to the FSF board of directors.[31][32][33][34]

Current and ongoing activities

The GNU Project

The original purpose of the FSF was to promote the ideals of free software. The organization envisaged the GNU operating system as an example of this.

GNU licenses

The GNU General Public License (GPL) is a widely used license for free software projects. The current version (version 3) was released in June 2007. The FSF has also published the GNU Lesser General Public License (LGPL), the GNU Free Documentation License (GFDL), and the GNU Affero General Public License (AGPL).

GNU Press

The FSF's publishing department, responsible for "publishing affordable books on computer science using freely distributable licenses."[35][36]

The Free Software Directory

This is a list of software packages that have been verified as free software. Each package entry contains up to 47 pieces of information such as the project's homepage, developers, programming language, etc. The goals are to provide a search engine for free software, and to provide a cross-reference for users to check if a package has been verified as being free software. The FSF has received a small amount of funding from UNESCO for this project.

Maintaining the Free Software Definition

FSF maintains many of the documents that define the free software movement.

Project hosting

FSF hosts software development projects on its Savannah website.

h-node

An abbreviation for "Hardware-Node", the *h-node* website lists hardware and device drivers that have been verified as compatible with free software. It is user-edited and volunteer supported with hardware entries tested by users before publication.[37][38][39]

Advocacy

FSF sponsors a number of campaigns against what it perceives as dangers to software freedom, including software patents, digital rights management (which the FSF and others[40] have re-termed "digital restrictions management", as part of its effort to highlight technologies that are "designed to take away and limit your rights",[41]) and user interface copyright. Since 2012, Defective by Design is an FSF-initiated campaign against DRM.[42] It also has a campaign to promote Ogg+Vorbis, a free alternative to proprietary formats like AAC and MQA. FSF also sponsors free software projects it deems "high-priority".

Annual awards

"Outstanding new Free Software contributor", "Award for the Advancement of Free Software" and "Free Software Award for Projects of Social Benefit"

LibrePlanet wiki

The LibrePlanet wiki organizes FSF members into regional groups in order to promote free software activism against digital restrictions management and other issues promoted by the FSF.

High priority projects

The FSF maintains a list of "high-priority projects" to which the Foundation claims that "there is a vital need to draw the free software community's attention".[43] The FSF considers these projects "important because computer users are continually being seduced into using non-free software, because there is no adequate free replacement."[43]

Parabola GNU/Linux-libre is a distribution officially endorsed by the *FSF*.

As of 2021, high-priority tasks include reverse engineering proprietary firmware, reversible debugging in GNU Debugger; developing automatic transcription and video editing software, Coreboot, drivers for network routers, a free smartphone operating system and creating replacements for Skype and Siri.[43]

Previous projects highlighted as needing work included the Free Java implementations, GNU Classpath, and GNU Compiler for Java, which ensure compatibility for the Java part of OpenOffice.org, and the GNOME desktop environment (see Java: Licensing).[44]

The effort has been criticized by Michael Larabel for either not instigating active development or for being slow at the work being done, even after certain projects were added to the list.[45][46]

Endorsements

Operating systems

The FSF maintains a list of approved Linux operating systems that maintain free software by default:[47]

- Dragora GNU/Linux-Libre
- dyne:bolic
- GNU Guix System
- Hyperbola GNU/Linux-libre
- Parabola GNU/Linux-libre
- PureOS
- Trisquel
- Ututo
- LibreCMC
- ProteanOS

The project also maintains a list of operating systems that are not versions of the GNU system:

- Replicant

Discontinued operating systems

The following are previously endorsed operating systems that are no longer actively maintained:

- gNewSense
- BLAG Linux and GNU
- Musix GNU+Linux

Hardware endorsements (RYF)

Since 2012, the FSF maintains a "Respects Your Freedom" (RYF) hardware certification program. To be granted certification, a product must use 100% Free Software, allow user installation of modified software, be free of backdoors and conform with several other requirements.[48][49]

Structure

Board

The FSF's board of directors includes professors at leading universities, senior engineers, and founders. Current board members are:[50]

- Geoffrey Knauth, senior software engineer at SFA, Inc. (served since October 23, 1997)
- Christina Haralanova, founding member of the Free Software Association, Bulgaria. Board member of Koumbit, member of FACIL – for the adoption of free software in Quebec (FACiL, pour l'appropriation collective de l'informatique libre)
- Gerald Jay Sussman, professor of computer science at the Massachusetts Institute of Technology (served since inception)
- Henry Poole, founder of CivicActions, a government digital services firm (served since December 12, 2002)
- Ian Kelling, Senior Systems Administrator at the FSF and the staff representative on the board.
- John Gilmore, co-founder of the Electronic Frontier Foundation, and co-designed the DHCP protocol.
- Maria Chiara Pievatolo is a professor of political philosophy at the University of Pisa.
- Richard Stallman, founder, launched the GNU project, author of the GNU General Public License.

Previous board members include:

- Alexander Oliva, Vice President (served since August 28, 2019)[51]
- Hal Abelson, founding member,[52] professor of computer science at the Massachusetts Institute of Technology (served from inception until March 5, 1998, and rejoined c. 2005)
- Robert J. Chassell, founding treasurer,[52] as well as a founding director (served from inception until June 3, 1997)
- Miguel de Icaza (served from August 1999[note 1] until February 25, 2002[53])
- Benjamin Mako Hill, assistant professor at the University of Washington (served from July 25, 2007 until October 2019)
- Matthew Garrett, software developer (served since October 16, 2014)[54]
- Bradley Kuhn, executive director of the Software Freedom Conservancy and FSF's former executive director (served from March 25, 2010[55] to Oct 13, 2019[56])
- Lawrence Lessig, professor of law at Stanford University (served from March 28, 2004 until 2008)
- Eben Moglen (served from July 28, 2000[note 2] until 2007[57])
- Len Tower Jr., founding member,[52] (served until September 2, 1997)

- Kat Walsh is a copyright and technology attorney, free culture and free software advocate, and former chair of the Wikimedia Foundation. She joined the board in 2015.[58] She voted against the readmittance of Richard Stallman to the board and, on March 25, 2021, resigned saying "It's a decision that has been a long time coming for me".[59]
- Odile Bénassy, research engineer at the Paris-sud university computer science research [60][61]

Executive directors

Executive directors include:

- Zoë Kooyman (2022–present)[62]
- John Sullivan (2011–2022)
- Peter T. Brown (2005–2010)
- Bradley M. Kuhn (2001–2005)

Voting

The FSF Articles of Organization state that the board of directors are elected.[63]

The bylaws say who can vote for them.[64]

The board can grant powers to the Voting Membership.[65]

Employment

At any given time, there are usually around a dozen employees.[66] Most, but not all, worked at the FSF headquarters in Boston, Massachusetts until August 2024 when the FSF closed its offices[67] and switched to remote work.[68]

Membership

On November 25, 2002, the FSF launched the FSF Associate Membership program for individuals.[69] Bradley M. Kuhn (FSF executive director, 2001–2005) launched the program and also signed up as the first Associate Member[70]

Associate members are primarily an honorary and funding support role.[65] In 2023, associate members gained the ability to make board nominations, along with FSF staff and FSF voting members. There is also an annual meeting of FSF members, usually during lunch at LibrePlanet, in which feedback for FSF is solicited.

Legal

Eben Moglen and Dan Ravicher previously served individually as pro bono legal counsel to the FSF. After forming the Software Freedom Law Center, Eben Moglen continued to serve as the FSF's general counsel until 2016.[71]

Financial

Most of the FSF funding comes from patrons and members.[72] Revenue streams also come from free-software-related compliance labs, job postings, published works, and a web store. FSF offers speakers and seminars for pay, and all FSF projects accept donations.

Revenues fund free-software programs and campaigns, while cash is invested conservatively in socially responsible investing. The financial strategy is designed to maintain the Foundation's long-term future through economic stability.

The FSF is a tax-exempt organization and posts annual IRS Form 990 filings online.[73]

Postal address and headquarters

Through the years the FSF has had its postal address, and until August 31, 2024 when going all remote its physical headquarters,[2] at different locations in Boston, Massachusetts, USA, as indicated in the table below.

As the GNU GPL v2 included the FSF's postal address in one of the first lines of the introduction and the source code license notice template every change of address also caused updates to the license itself.

FSF postal address and headquarters

Start Date	End Date	Address	Notes
September 1, 2024[74][2]	Current	31 Milk St # 960789 Boston, MA 02196 USA	All remote headquarters. USPS postbox in Milk Street Lobby post office[75]
May 1, 2005[76][77][78]	August 31, 2024 (last open to the public on the 16th)	51 Franklin Street, Fifth Floor Boston, MA 02110-1301 USA	Physical headquarters with offices, meeting room, stock and shipping facilities and kitchen[78]
1995[77]	April 30, 2005	59 Temple Place - Suite 330 Boston, MA 02111-1307 USA	Physical headquarters

Criticism

Position on DRM

Linus Torvalds has criticized FSF for using GPLv3 as a weapon in the fight against DRM. Torvalds argues that the issue of DRM and that of a software license should be treated as two separate issues.[79]

Defective by Design campaign

On June 16, 2010, Joe Brockmeier, a journalist at *Linux Magazine*, criticized the Defective by Design campaign by the FSF as "negative" and "juvenile" and not being adequate for providing users with "credible alternatives" to proprietary software.[80] FSF responded to this criticism by saying "that there is a fundamental difference between speaking out against policies or actions and smear campaigns", and "that if one is taking an ethical position, it is justified, and often necessary, to not only speak about the benefits of freedom but against acts of dispossession and disenfranchisement."[81]

GNU LibreDWG license controversy

In 2009, a license update of LibDWG/LibreDWG to version 3 of the GNU GPL made it impossible for the free software projects LibreCAD and FreeCAD to use LibreDWG legally.[82] Many projects voiced their unhappiness about the GPLv3 license selection for LibreDWG, such as FreeCAD, LibreCAD, Assimp, and Blender.[83] Some suggested the selection of a license with a broader license compatibility, for instance the MIT, BSD, or LGPL 2.1.[83] A request went to the FSF to relicense GNU LibreDWG as GPLv2, which was rejected in 2012.[84]

The libDWG has stalled since 2011 for various reasons, including license issues.[85]

Accusations against Richard Stallman

Stallman resigned from the board in 2019 after making controversial comments about one of the victims of Jeffrey Epstein, but rejoined the board 18 months later.[86] Several prominent organizations and individuals who develop free software objected to the decision, citing past writings on Stallman's blog which they considered antithetical to promoting a diverse community.[87][88] As a result of Stallman's reinstatement, prominent members of the Free Software Foundation quit in protest and Red Hat announced that it would stop funding and supporting the Free Software Foundation.[87][89]

Recognition

Key players and industries that have made honorific mention and awards include:

- 2001: GNU Project received the USENIX Lifetime Achievement Award for "the ubiquity, breadth, and quality of its freely available redistributable and modifiable software, which has enabled a generation of research and commercial development".[90]
- 2005: Prix Ars Electronica Award of Distinction in the category of "Digital Communities"[91]

See also

Free and open-source software portal

- Defective by Design – Anti-DRM initiative
- Digital rights – Type of human and legal rights
- Electronic Frontier Foundation – Digital rights group

- Free software movement – Social movement
- League for Programming Freedom
- LibrePlanet – Community project promoting free software
- Foundations promoting Free Software movement:
  - Free Software Foundation Europe
  - Free Software Foundation Latin America
  - Free Software Foundation of India

Notes

1. The FSF annual filings with the Commonwealth of Massachusetts for 1998 and 1999 show that De Icaza was not on the board on 1998-11-01 and was as of 1999-11-01, so he clearly joined sometime between those dates. Those documents further indicate that the 1999 annual meeting occurred in August; usually, new directors are elected at annual meetings.
2. The FSF annual filings with the Commonwealth of Massachusetts for 1999 and 2000 show that Moglen was not on the board on November 1, 1999, and was as of November 1, 2000, so he clearly joined sometime between those dates. Those documents further indicate that the 2000 annual meeting occurred on July 28, 2000; usually, new directors are elected at annual meetings.

References

1. "Corporations Division Entity Summary for ID Number: 042888848" (http://corp.sec.state.ma.us/CorpWeb/CorpSearch/CorpSummary.aspx?sysvalue=AYJWph6_goaQ0uN4EZuhbV50Wm3aPPBRHI8_5qC0vtI-). Secretary of Commonwealth of Massachusetts. Archived (https://web.archive.org/web/20200726223940/http://corp.sec.state.ma.us/CorpWeb/CorpSearch/CorpSummary.aspx?sysvalue=AYJWph6_goaQ0uN4EZuhbV50Wm3aPPBRHI8_5qC0vtI-) from the original on July 26, 2020. Retrieved October 4, 2014.
2. "Join us in saying goodbye to our beloved office on August 16! — Free Software Foundation — Working together for free software" (https://www.fsf.org/blogs/community/fsf-office-closing-party). *www.fsf.org*. Free Software Foundation. August 5, 2024. Retrieved September 1, 2024.
3. "Free Software Foundation announces new executive director, Zoë Kooyman — Free Software Foundation — Working together for free software" (https://www.fsf.org/news/free-software-foundation-announces-new-executive-director-zoe-kooyman). *fsf.org*. Archived (https://web.archive.org/web/20220302010415/https://www.fsf.org/news/free-software-foundation-announces-new-executive-director-zoe-kooyman) from the original on March 2, 2022. Retrieved March 2, 2022.
4. Roberts, Brandon; Schwencke, Ken; Tigas, Mike; Wei, Sisi; Glassford, Alec; Suozzo, Andrea (May 9, 2013). "Free Software Foundation Inc – Nonprofit Explorer" (https://projects.propublica.org/nonprofits/organizations/42888848). *ProPublica*. Archived (https://web.archive.org/web/20220913170959/https://projects.propublica.org/nonprofits/organizations/42888848) from the original on September 13, 2022. Retrieved September 13, 2022.
5. "Staff of the Free Software Foundation" (https://www.fsf.org/about/staff/). Archived (https://web.archive.org/web/20121124045543/http://www.fsf.org/about/staff) from the original on November 24, 2012. Retrieved October 5, 2022.

6. Corrado, Edward M.; Moualison Sandy, Heather; Mitchell, Erik T. (July 3, 2018). "Nullis in Verba: The Free Software Movement as a model for Openness and Transparency" (https://www.tandfonline.com/doi/full/10.1080/07317131.2018.1456849). *Technical Services Quarterly*. **35** (3): 269–279. doi:10.1080/07317131.2018.1456849 (https://doi.org/10.1080%2F07317131.2018.1456849). ISSN 0731-7131 (https://search.worldcat.org/issn/0731-7131). S2CID 196159979 (https://api.semanticscholar.org/CorpusID:196159979). Archived (https://web.archive.org/web/20230710173206/https://www.tandfonline.com/doi/full/10.1080/07317131.2018.1456849) from the original on July 10, 2023. Retrieved May 3, 2022.
7. "Frequently Asked Questions about the GNU Licenses" (https://www.gnu.org/licenses/gpl-faq.html). Free Software Foundation. Archived (https://web.archive.org/web/20200330051354/http://www.gnu.org/licenses/gpl-faq.html) from the original on March 30, 2020. Retrieved July 22, 2012.
8. "What Is Copyleft?" (https://www.gnu.org/copyleft/copyleft.html). Free Software Foundation. Archived (https://web.archive.org/web/20180618030401/http://www.gnu.org/copyleft/copyleft.html) from the original on June 18, 2018. Retrieved July 22, 2012.
9. "Free Software Foundation, Boston, United States" (https://web.archive.org/web/20200227174517/https://bizpages.org/business--United_States--boston--12856). *bizpages.org*. Archived from the original (https://bizpages.org/business--United_States--Boston--12856) on February 27, 2020. Retrieved March 7, 2020.
10. "FREE SOFTWARE FOUNDATION, INC. Summary Screen" (https://web.archive.org/web/20130525155859/http://corp.sec.state.ma.us/corp/corpsearch/CorpSearchSummary.asp?ReadFromDB=True&UpdateAllowed=&FEIN=042888848). The Commonwealth of Massachusetts, Secretary of the Commonwealth, Corporations Division. Archived from the original (http://corp.sec.state.ma.us/corp/corpsearch/CorpSearchSummary.asp?ReadFromDB=True&UpdateAllowed=&FEIN=042888848) on May 25, 2013.
11. Stallman, Richard (May 29, 2001). "Free Software: Freedom and Cooperation" (https://www.gnu.org/philosophy/rms-nyu-2001-transcript.en.html). *www.gnu.org*. Archived (https://web.archive.org/web/20240224234039/https://www.gnu.org/philosophy/rms-nyu-2001-transcript.en.html) from the original on February 24, 2024. Retrieved May 12, 2024.
12. Stallman, Richard M. (2002). "Linux, GNU, and freedom" (https://www.gnu.org/philosophy/linux-gnu-freedom.html). *Philosophy of the GNU Project*. GNU Project. Archived (https://web.archive.org/web/20171223131647/https://www.gnu.org/philosophy/linux-gnu-freedom.html) from the original on December 23, 2017. Retrieved December 10, 2006.
13. "Licenses" (https://www.gnu.org/licenses/licenses.html). Free Software Foundation. Archived (https://web.archive.org/web/20110406092332/http://www.gnu.org/licenses/licenses.html) from the original on April 6, 2011. Retrieved June 24, 2012.
14. "The GNU Project" (https://www.gnu.org/gnu/thegnuproject.html). Free Software Foundation. Archived (https://web.archive.org/web/20110424064815/https://www.gnu.org/gnu/thegnuproject.html) from the original on April 24, 2011. Retrieved June 24, 2012.
15. Meeker, Heather (June 28, 2005). "The Legend of Linksys" (https://web.archive.org/web/20090419230853/http://linuxinsider.com/story/qECCd2x743n32T/The-Legend-of-Linksys.xhtml). Archived from the original (http://linuxinsider.com/story/qECCd2x743n32T/The-Legend-of-Linksys.xhtml) on April 19, 2009. Retrieved August 11, 2007. Hosted on the Wayback Machine.
16. Gillmor, Dan (May 21, 2003). "GPL Legal Battle Coming?" (https://web.archive.org/web/20030524174013/http://weblog.siliconvalley.com/column/dangillmor/archives/001029.shtml). SiliconValley.com (a division of the San Jose Mercury News). Archived from the original (http://weblog.siliconvalley.com/column/dangillmor/archives/001029.shtml) on May 24, 2003. Retrieved August 11, 2007.
17. Turner, David; Kuhn, Bradley M. (September 29, 2003). "Linksys/Cisco GPL Violations" (https://lwn.net/Articles/51570/). LWN.net. Archived (https://web.archive.org/web/20070831033716/http://lwn.net/Articles/51570/) from the original on August 31, 2007. Retrieved August 11, 2007.

18. Kennedy, Dennis (January 11, 2004). "A Great Learning Opportunity for Software Lawyers – Upcoming GPL Seminar" (https://web.archive.org/web/20070928125310/http://www.denniskennedy.com/blog/2004/01/a_great_learning_opportunity_f.html). Archived from the original (https://www.denniskennedy.com/blog/2004/01/a_great_learning_opportunity_f.html) on September 28, 2007.
19. Lord, Timothy (July 18, 2003). "Seminar On Details Of The GPL And Related Licenses" (https://yro.slashdot.org/story/03/07/18/1835252/seminar-on-details-of-the-gpl-and-related-licenses). Slashdot. Archived (https://web.archive.org/web/20110606235944/http://yro.slashdot.org/story/03/07/18/1835252/Seminar-On-Details-Of-The-GPL-And-Related-Licenses) from the original on June 6, 2011. Retrieved August 11, 2007.
20. Heise, Mark (November 5, 2003). "SCO Subpoena of FSF" (https://www.fsf.org/licensing/sco/sco-subpoena.pdf) (PDF). Free Software Foundation. Archived (https://web.archive.org/web/20070925190126/http://www.fsf.org/licensing/sco/sco-subpoena.pdf) (PDF) from the original on September 25, 2007. Retrieved August 11, 2007.
21. Kuhn, Bradley (May 18, 2004). "The SCO Subpoena of FSF" (https://www.fsf.org/licensing/sco/subpoena.html). Free Software Foundation. Archived (https://web.archive.org/web/20070608193032/http://www.fsf.org/licensing/sco/subpoena.html) from the original on June 8, 2007. Retrieved August 11, 2007.
22. "FSF To Host Free Software Licensing Seminars and Discussions on SCO v. IBM in New York" (https://www.gnu.org/press/2004-01-02-nyc-seminars.html) (Press release). Gnu.org. Archived (https://web.archive.org/web/20080517060822/http://www.gnu.org/press/2004-01-02-nyc-seminars.html) from the original on May 17, 2008. Retrieved August 30, 2011.
23. "Seminar On Details Of The GPL And Related Licenses" (http://yro.slashdot.org/article.pl?sid=03/07/18/1835252). July 18, 2003. Archived (https://web.archive.org/web/20230429075633/https://yro.slashdot.org/story/03/07/18/1835252/seminar-on-details-of-the-gpl-and-related-licenses) from the original on April 29, 2023. Retrieved July 4, 2008.
24. Kuhn, Bradley M. (June 2003). "FSF Bulletin Issue 2, June 2003" (https://www.gnu.org/bulletins/bulletin-002.html). Free Software Foundation. Archived (https://web.archive.org/web/20080517080649/http://www.gnu.org/bulletins/bulletin-002.html) from the original on May 17, 2008. Retrieved July 4, 2008. "On Friday 8 August 2003, we will hold a seminar on the GNU GPL. The seminar, titled "Free Software Licensing and the GNU GPL", will be co-led by Daniel Ravicher, Outside Counsel to FSF from Patterson Belknap Webb & Tyler, and Bradley M. Kuhn, Executive Director of FSF."
25. Sullivan, John (August 25, 2005). "FSF Seminar in NYC on September 28" (http://lists.gnu.org/archive/html/info-fsf/2005-08/msg00001.html). *lists.gnu.org*. Free Software Foundation. Archived (https://web.archive.org/web/20191013044127/https://lists.gnu.org/archive/html/info-fsf/2005-08/msg00001.html) from the original on October 13, 2019. Retrieved July 4, 2008.
26. "GNU General Public License" (https://www.gnu.org/licenses/gpl.html). *GNU*. June 29, 2007. Archived (https://web.archive.org/web/20170729122702/http://www.gnu.org/licenses/gpl.html) from the original on July 29, 2017. Retrieved June 15, 2012.
27. "GNU General Public License" (https://web.archive.org/web/20081002034950/http://gplv3.fsf.org/comments/gplv3-draft-4.html). *GPLv3*. Archived from the original (http://gplv3.fsf.org/comments/gplv3-draft-4.html) on October 2, 2008. Retrieved June 25, 2012.
28. Paul, Ryan (December 13, 2007). "Free Software Foundation lawsuit against Cisco a first" (https://arstechnica.com/news.ars/post/20081211-free-software-foundation-lawsuit-against-cisco-a-first.html). Ars Technica. Archived (https://web.archive.org/web/20081212233437/http://arstechnica.com/news.ars/post/20081211-free-software-foundation-lawsuit-against-cisco-a-first.html) from the original on December 12, 2008. Retrieved December 11, 2008.

29. Paul, Ryan (May 21, 2009). "Cisco settles FSF GPL lawsuit, appoints compliance officer" (https://arstechnica.com/open-source/news/2009/05/cisco-settles-fsf-gpl-lawsuit-appoints-compliance-officer.ars). Ars Technica. Archived (https://web.archive.org/web/20090726192021/http://arstechnica.com/open-source/news/2009/05/cisco-settles-fsf-gpl-lawsuit-appoints-compliance-officer.ars) from the original on July 26, 2009. Retrieved October 6, 2009.
30. Bort, Julie (October 10, 2019). "A programmer explains why he's willing to quit rather than work with industry legend Richard Stallman, who resigned from MIT after controversial remarks on Jeffrey Epstein" (https://web.archive.org/web/20200807034442/https://www.businessinsider.de/international/gnu-programmers-call-for-richard-stallman-to-quit-2019-10/?r=US&IR=T). Business Insider. Archived from the original (https://www.businessinsider.de/international/gnu-programmers-call-for-richard-stallman-to-quit-2019-10/?r=US&IR=T) on August 7, 2020. Retrieved May 3, 2020.
31. Larabel, Michael (September 26, 2019). "Richard Stallman To Continue As Head Of The GNU Project" (https://www.phoronix.com/scan.php?page=news_item&px=Richard-Stallman-GNU-Head). *Phoronix*. Archived (https://web.archive.org/web/20220321204310/https://www.phoronix.com/scan.php?page=news_item&px=Richard-Stallman-GNU-Head) from the original on March 21, 2022. Retrieved March 24, 2021.
32. Stallman, Richard. "Richard Stallman's Personal Site" (http://www.stallman.org/). Archived (https://web.archive.org/web/20110224213224/http://stallman.org/) from the original on February 24, 2011. Retrieved March 18, 2021. "I continue to be the Chief GNUisance of the GNU Project. I do not intend to stop any time soon."
33. Brodkin, Jon (March 22, 2021). "Richard Stallman returns to FSF 18 months after controversial rape comments" (https://arstechnica.com/tech-policy/2021/03/richard-stallman-returns-to-fsf-18-months-after-controversial-rape-comments/). *Ars Technica*. Archived (https://web.archive.org/web/20210323002959/https://arstechnica.com/tech-policy/2021/03/richard-stallman-returns-to-fsf-18-months-after-controversial-rape-comments/) from the original on March 23, 2021. Retrieved March 23, 2021.
34. Clark, Mitchell (March 22, 2021). "Richard Stallman returns to the Free Software Foundation after resigning in 2019" (https://www.theverge.com/2021/3/22/22344910/richard-stallman-returns-free-software-foundation-board-comments). *The Verge*. Archived (https://web.archive.org/web/20210322222019/https://www.theverge.com/2021/3/22/22344910/richard-stallman-returns-free-software-foundation-board-comments) from the original on March 22, 2021. Retrieved March 22, 2021.
35. "GNU Press -- Published Documentation" (https://web.archive.org/web/20051125024241/http://www.gnupress.org/). Free Software Foundation, Inc. Archived from the original (https://www.gnupress.org/) on November 25, 2005.
36. List of books published (http://shop.fsf.org/category/books/) Archived (https://web.archive.org/web/20130501022816/http://shop.fsf.org/category/books/) 2013-05-01 at the Wayback Machine in GNU Press home site
37. "FSF and Debian join forces to help free software users find the hardware they need" (https://sdtimes.com/fsf-debian-join-forces-help-free-software-users-find-hardware-need/). September 8, 2014. Archived (https://web.archive.org/web/20191019225752/https://sdtimes.com/fsf-debian-join-forces-help-free-software-users-find-hardware-need/) from the original on October 19, 2019. Retrieved October 19, 2019.
38. "FSFとDebian、GNU/Linuxハードウェア情報サイト「h-node.org」を共同支援へ | OSDN Magazine" (https://mag.osdn.jp/14/09/11/062900). *OSDN*. Archived (https://web.archive.org/web/20191019225751/https://mag.osdn.jp/14/09/11/062900) from the original on October 19, 2019. Retrieved October 19, 2019.
39. "home - h-node.org" (https://h-node.org/). *h-node.org*. Archived (https://web.archive.org/web/20191019115648/https://h-node.org/) from the original on October 19, 2019. Retrieved October 19, 2019.

40. Stross, Randall (January 14, 2007). "Want an iPhone? Beware the iHandcuffs" (https://www.nytimes.com/2007/01/14/business/yourmoney/14digi.html). *The New York Times*. Archived (https://web.archive.org/web/20161101225145/http://www.nytimes.com/2007/01/14/business/yourmoney/14digi.html) from the original on November 1, 2016. Retrieved May 26, 2014.
41. "Digital Restrictions Management and Treacherous Computing" (https://www.fsf.org/campaigns/drm.html). Free Software Foundation. September 18, 2006. Archived (https://web.archive.org/web/20180705233004/https://www.fsf.org/campaigns/drm.html) from the original on July 5, 2018. Retrieved December 17, 2007.
42. Noyes, Katherine (August 21, 2012). "For DRM-Free Content, Look for the New FSF Logo" (https://www.pcworld.com/article/460807/for_drm_free_content_look_for_the_new_fsf_logo.html). *PCWorld*. Archived (https://web.archive.org/web/20230606090737/https://www.pcworld.com/article/460807/for_drm_free_content_look_for_the_new_fsf_logo.html) from the original on June 6, 2023. Retrieved June 6, 2023.
43. "High Priority Free Software Projects" (https://www.fsf.org/campaigns/priority-projects/). Free Software Foundation. Archived (https://web.archive.org/web/20090701092733/http://www.fsf.org/campaigns/priority.html) from the original on July 1, 2009. Retrieved January 7, 2017.
44. "Changelog for the High Priority Projects list" (https://www.fsf.org/campaigns/priority-projects/changelog). Free Software Foundation. Archived (https://web.archive.org/web/20170129093713/https://www.fsf.org/campaigns/priority-projects/changelog) from the original on January 29, 2017. Retrieved January 24, 2017.
45. Larabel, Michael (October 15, 2011). "The Sad State Of FSF's High Priority Projects" (https://www.phoronix.com/scan.php?page=news_item&px=MTAwMTY). Phoronix. Archived (https://web.archive.org/web/20150319093219/http://www.phoronix.com/scan.php?page=news_item&px=MTAwMTY) from the original on March 19, 2015. Retrieved December 29, 2014. "Long story short, being on the Free Software Foundation's high priority list really doesn't mean much with some of these "important" projects not even being actively developed or even discussed."
46. Larabel, Michael (April 22, 2012). "Many FSF Priority Projects Still Not Progressing" (https://www.phoronix.com/scan.php?page=news_item&px=MTA5MDk). Phoronix. Archived (https://web.archive.org/web/20150319092614/http://www.phoronix.com/scan.php?page=news_item&px=MTA5MDk) from the original on March 19, 2015. Retrieved December 29, 2014. "Most of the projects are basically not going anywhere. Many of them at the time were not really advancing in their goals, haven't had releases in a while, or coding hasn't even started. It's been more than a half-year and still there's no significant work towards clearing many of projects from the FSF list."
47. "Free GNU/Linux distributions" (http://www.gnu.org/distros/free-distros.html). GNU. April 30, 2021. Archived (https://web.archive.org/web/20151225192720/http://www.gnu.org/distros/free-distros.html) from the original on December 25, 2015. Retrieved May 27, 2021.
48. Noyes, Katherine (October 11, 2012). "New FSF logo highlights hardware that 'Respects Your Freedom' " (https://www.pcworld.com/article/461621/new-fsf-logo-highlights-hardware-that-respects-your-freedom.html). *PCWorld*. Archived (https://web.archive.org/web/20230606090739/https://www.pcworld.com/article/461621/new-fsf-logo-highlights-hardware-that-respects-your-freedom.html) from the original on June 6, 2023. Retrieved June 6, 2023.
49. Gay, Josh (January 27, 2012). "Respects Your Freedom hardware certification requirements" (https://www.fsf.org/resources/hw/endorsement/criteria). Free Software Foundation. Archived (https://web.archive.org/web/20140207142653/http://www.fsf.org/resources/hw/endorsement/criteria) from the original on February 7, 2014. Retrieved February 4, 2014.
50. "Staff and Board — Free Software Foundation — Working together for free software" (https://www.fsf.org/about/staff-and-board). *www.fsf.org*. Archived (https://web.archive.org/web/20130815225019/https://www.fsf.org/about/staff-and-board) from the original on August 15, 2013. Retrieved December 15, 2024.

51. "Alexandre Oliva joins Free Software Foundation board of directors" (https://www.fsf.org/news/alexandre-oliva-joins-free-software-foundation-board-of-directors). Free Software Foundation. August 28, 2019. Archived (https://web.archive.org/web/20191020202720/https://www.fsf.org/news/alexandre-oliva-joins-free-software-foundation-board-of-directors) from the original on October 20, 2019. Retrieved October 17, 2019.
52. The first GNU's Bulletin ("GNU's Bulletin, Volume 1, No.1" (https://www.gnu.org/bulletins/bull1.txt). Free Software Foundation. February 1986. Archived (https://web.archive.org/web/20150623180723/http://www.gnu.org/bulletins/bull1.txt) from the original on June 23, 2015. Retrieved August 11, 2007.), indicates this list of people as *round[ing] out FSF's board of directors*.
53. The FSF annual filings with the Commonwealth of Massachusetts for 2002 ("2002 Annual Report for Free Software Foundation, Inc" (https://corp.sec.state.ma.us/corp/corpsearch/display_pdf.asp?CORP_DRIVE1/2002/1217/000000000/9152/200228079130_1.pdf) (PDF). The Commonwealth of Massachusetts. December 17, 2002. Archived (https://web.archive.org/web/20070925190125/http://corp.sec.state.ma.us/corp/corpsearch/display_pdf.asp?CORP_DRIVE1/2002/1217/000000000/9152/200228079130_1.pdf) (PDF) from the original on September 25, 2007. Retrieved August 11, 2007.) show that De Icaza has left the board. Changes to board composition are usually made at the annual meeting; which occurred on February 25, 2002.
54. "Matthew Garrett joins Free Software Foundation board of directors" (https://www.fsf.org/news/matthew-garrett-joins-free-software-foundation-board-of-directors). Free Software Foundation. October 16, 2014. Archived (https://web.archive.org/web/20150203232350/http://www.fsf.org/news/matthew-garrett-joins-free-software-foundation-board-of-directors) from the original on February 3, 2015. Retrieved January 22, 2015.
55. "Bradley Kuhn Joins the FSF Board" (https://www.fsf.org/blogs/bradley-kuhn-joins-the-fsf-board). March 25, 2010. Archived (https://web.archive.org/web/20100329200144/http://www.fsf.org/blogs/bradley-kuhn-joins-the-fsf-board) from the original on March 29, 2010. Retrieved March 26, 2010.
56. "On Recent Controversial Events - Bradley M. Kuhn (Brad) (bkuhn)" (http://ebb.org/bkuhn/blog/2019/10/15/fsf-rms.html). *ebb.org*. Archived (https://web.archive.org/web/20191016113625/http://ebb.org/bkuhn/blog/2019/10/15/fsf-rms.html) from the original on October 16, 2019. Retrieved October 17, 2019.
57. Moglen announced his intention to resign in his blog (Moglen, Eben (April 23, 2007). "And Now ... Life After GPLv3" (http://moglen.law.columbia.edu/blog/organizations/SFLC/Transition.html). Archived (https://web.archive.org/web/20070702172544/http://moglen.law.columbia.edu/blog/organizations/SFLC/Transition.html) from the original on July 2, 2007. Retrieved August 11, 2007.). The resignation likely occurred at the 2007 annual meeting of the directors; the exact date of that meeting is unknown.
58. "Kat Walsh joins FSF board of directors" (https://www.fsf.org/news/kat-walsh-joins-fsf-board-of-directors). *fsf.org*. Free Software Foundation, Inc. March 21, 2015. Archived (https://web.archive.org/web/20150321170636/https://www.fsf.org/news/kat-walsh-joins-fsf-board-of-directors) from the original on March 21, 2015. Retrieved March 22, 2015.
59. "Kat (@mindspillage@mastodon.social)" (https://mastodon.social/@mindspillage/105948765536809476). *Mastodon*. March 25, 2021. Archived (https://web.archive.org/web/20220730020134/https://mastodon.social/@mindspillage/105948765536809476) from the original on July 30, 2022. Retrieved April 27, 2022.
60. "Ingénieur de Recherche 2ème classe" (https://www.lri.fr/membre.php?mb=2529). *Laboratoire de Recherche en Informatique*. Archived (https://web.archive.org/web/20220111130452/https://www.lri.fr/membre.php?mb=2529) from the original on January 11, 2022. Retrieved August 8, 2020.

61. "Geoffrey Knauth elected Free Software Foundation president; Odile Bénassy joins the board" (https://www.fsf.org/news/geoffrey-knauth-elected-free-software-foundation-president-odile-benassy-joins-the-board). *Free Software Foundation*. August 5, 2020. Archived (https://web.archive.org/web/20220607085357/https://www.fsf.org/news/geoffrey-knauth-elected-free-software-foundation-president-odile-benassy-joins-the-board) from the original on June 7, 2022. Retrieved August 8, 2020.
62. Kooyman, Zoë. "My appointment as the FSF's new executive director" (https://www.fsf.org/blogs/community/my-appointment-as-the-fsfs-new-executive-director). *Free Software Foundation*. Retrieved May 4, 2024.
63. > Article II, Sec. 1 - Number, Election and Qualification: The present members of the corporation shall constitute the voting members. Thereafter the voting members annually at its annual meeting shall fix the number of voting members and shall elect the number of voting members so fixed. At any special or regular meeting, the voting members then in office may increase the number of voting members and elect new voting members to complete the number so fixed; or they may decrease the number of voting members, but only to eliminate vacancies caused by the death, resignation, removal or disqualification of one or more voting members.
   >
   > —Amended By-laws, Nov. 25, 2002, Free Software Foundation, Inc.
64. > In addition to the right to elect Directors as provided in the bylaws and such other powers and rights as may be vested in them by law, these Articles of Organization or the bylaws, the Voting Members shall have such other powers and rights as the Directors may designate.
   >
   > —Amended By-laws, Nov. 25, 2002, Free Software Foundation, Inc.
65. "Amended Bylaws" (https://www.fsf.org/about/fsf-bylaws). Free Software Foundation. Archived (https://web.archive.org/web/20230124225933/https://www.fsf.org/about/fsf-bylaws) from the original on January 24, 2023. Retrieved January 24, 2023.
66. "Meet the staff of the Free Software Foundation" (https://www.fsf.org/about/staff-and-board). Archived (https://web.archive.org/web/20130815225019/https://www.fsf.org/about/staff-and-board) from the original on August 15, 2013. Retrieved September 4, 2015.
67. "Join us in saying goodbye to our beloved office on August 16!" (https://www.fsf.org/blogs/community/fsf-office-closing-party).
68. "Certificate of Change of Principal Office" (http://corp.sec.state.ma.us/corp/corpsearch/display_pdf.asp?CORP_DRIVE1/2005/0526/000000000/4122/200516698270_1.pdf) (PDF). The Commonwealth of Massachusetts. May 26, 2005. Archived (https://web.archive.org/web/20080826091313/http://corp.sec.state.ma.us/corp/corpsearch/display_pdf.asp?CORP_DRIVE1/2005/0526/000000000/4122/200516698270_1.pdf) (PDF) from the original on August 26, 2008. Retrieved July 4, 2008.
69. The site member.fsf.org first appears in the Internet Archive in December 2002, and that site lists the date of the launch as 25 November 2002. "FSF Membership Page" (https://web.archive.org/web/20021220112452/http://member.fsf.org/). The Internet Archive. Archived from the original (http://member.fsf.org/) on December 20, 2002.

70. Kuhn has an FSF-generated member link that identifies him as the first member on his web page. "Homepage of Bradley M. Kuhn" (http://www.ebb.org/bkuhn). Bradley M. Kuhn. January 5, 2008. Archived (https://web.archive.org/web/20071226122322/http://ebb.org/bkuhn/) from the original on December 26, 2007. Retrieved January 5, 2008.
71. Sullivan, John (October 27, 2016). "FSF announces change in general counsel" (https://www.fsf.org/news/fsf-announces-change-in-general-counsel). Free Software Foundation. Archived (https://web.archive.org/web/20170508115053/http://www.fsf.org/news/fsf-announces-change-in-general-counsel) from the original on May 8, 2017. Retrieved May 18, 2017.
72. Stallman, Richard. "About the GNU Project" (https://www.gnu.org/gnu/thegnuproject.html). *Gnu Project*. FSF. Archived (https://web.archive.org/web/20110424064815/https://www.gnu.org/gnu/thegnuproject.html) from the original on April 24, 2011. Retrieved May 18, 2014.
73. "FSF Financial Information" (https://www.fsf.org/about/financial). Archived (https://web.archive.org/web/20220604214911/https://www.fsf.org/about/financial) from the original on June 4, 2022. Retrieved June 10, 2022.
74. Richard Stallman (August 31, 2024). "FSF office adddress update" (https://lists.gnu.org/archive/html/info-gnu/2024-09/msg00000.html). Retrieved September 1, 2024.
75. "MILK STREET - USPS" (https://tools.usps.com/locations/details/1441502). United States Postal Service. Retrieved September 1, 2024.
76. Lee, Matt (February 12, 2005). "FSF will be moving" (https://www.fsf.org/news/fsf-moving.html). Free Software Foundation. Retrieved September 1, 2024.
77. Lee, Matt (February 15, 2005). "FSF Responds to LinuxWorld in Boston" (https://www.fsf.org/news/fsf-responds-to-linuxworld-in-boston). Free Software Foundation. Retrieved September 1, 2024.
78. Lee, Matt (December 28, 2010). "Virtually tour our previous office at 51 Franklin Street" (https://www.fsf.org/about/contact/tour-2010). Free Software Foundation. Retrieved September 1, 2024.
79. patrick_g. "Linus Torvalds: the anniversary interview of the 20 years of the kernel" (https://linuxfr.org/nodes/85904/comments/1230981). LinuxFr.org. Archived (https://web.archive.org/web/20191023175643/https://linuxfr.org/nodes/85904/comments/1230981) from the original on October 23, 2019. Retrieved October 23, 2019.
80. "The Party of Gno" (https://web.archive.org/web/20100620003719/http://www.linux-mag.com/cache/7806/1.html). Archived from the original on June 20, 2010. Retrieved June 22, 2010.
81. Hill, Benjamin Mako (November 28, 2011). "In defense of negativity" (https://www.fsf.org/bulletin/2011/fall/in-defense-of-negativity). Free Software Foundation. Archived (https://web.archive.org/web/20190917155820/https://www.fsf.org/bulletin/2011/fall/in-defense-of-negativity) from the original on September 17, 2019. Retrieved October 23, 2019.
82. Larabel, Michael (January 24, 2013). "FSF Wastes Away Another "High Priority" Project" (https://web.archive.org/web/20161109200145/https://www.phoronix.com/scan.php?page=news_item&px=MTI4Mjc). Phoronix. Archived from the original (https://www.phoronix.com/scan.php?page=news_item&px=MTI4Mjc) on November 9, 2016. Retrieved August 22, 2013. "*Both LibreCAD and FreeCAD both want to use LibreDWG and have patches available for supporting the DWG file format library, but can't integrate them. The programs have dependencies on the popular GPLv2 license while the Free Software Foundation will only let LibreDWG be licensed for GPLv3 use, not GPLv2.*"

83. Prokoudine, Alexandre (January 26, 2012). "What's up with DWG adoption in free software?" (https://librearts.org/2012/01/whats-up-with-dwg-adoption-in-free-software/). librearts.org. Retrieved March 9, 2025. "*[Assimp's Alexander Gessler:] "Personally, I'm extremely unhappy with their [LibreDWG's — LGW] GPL licensing. It prohibits its use in Assimp and for many other applications as well. I don't like dogmatic ideologies, and freeing software by force (as GPL/GNU does) is something I dislike in particular. It's fine for applications, because it doesn't hurt at this point, but, in my opinion, not for libraries that are designed to be used as freely as possible." [Blender's Toni Roosendaal:] "Blender is also still "GPLv2 or later". For the time being we stick to that, moving to GPL 3 has no evident benefits I know of. My advice for LibreDWG: if you make a library, choosing a widely compatible license (MIT, BSD, or LGPL) is a very positive choice."*"
84. Prokoudine, Alexandre (December 27, 2012). "LibreDWG drama: the end or the new beginning?" (https://librearts.org/2012/12/libredwg-drama-the-end-or-the-new-beginning/). librearts.org. Retrieved March 9, 2025. "*[...]the unfortunate situation with support for DWG files in free CAD software via LibreDWG. We feel, by now it ought to be closed. We have the final answer from FSF. [...] "We are not going to change the license."*"
85. Prokoudine, Alexandre (January 26, 2012). "What's up with DWG adoption in free software?" (https://librearts.org/2012/01/whats-up-with-dwg-adoption-in-free-software/). librearts.org. Retrieved March 9, 2025. "*GPLv3 license. It doesn't work for end-user software, because they tend to use 3rd party components under different licenses that impose restrictions. FSF who are sole copyright holders of LibreDWG objected to relicensing. With regards to FreeCAD project and Yorik van Havre, its contributor, Richard Stallman stated:" You should not change the license of your library. Rather, it is best to make it clear to him what the conditions are." [...] Personally, I'm extremely unhappy with their [LibreDWG's — LGW] GPL licensing. It prohibits its use in Assimp and for many other applications as well. I don't like dogmatic ideologies, and freeing software by force (as GPL/GNU does) is something I dislike in particular. It's fine for applications, because it doesn't hurt at this point, but, in my opinion, not for libraries that are designed to be used as freely as possible.*"
86. Brodkin, Jon (March 22, 2021). "Richard Stallman returns to FSF 18 months after controversial rape comments" (https://arstechnica.com/tech-policy/2021/03/richard-stallman-returns-to-fsf-18-months-after-controversial-rape-comments/). *Ars Technica*. Archived (https://web.archive.org/web/20210323002959/https://arstechnica.com/tech-policy/2021/03/richard-stallman-returns-to-fsf-18-months-after-controversial-rape-comments/) from the original on March 23, 2021. Retrieved March 31, 2021.
87. Vaughan-Nichols, Steven J. "Free Software Foundation leaders and supporters desert sinking ship" (https://www.zdnet.com/article/free-software-foundation-leaders-and-supporters-desert-sinking-ship/). *ZDNet*. Archived (https://web.archive.org/web/20210330142323/https://www.zdnet.com/article/free-software-foundation-leaders-and-supporters-desert-sinking-ship/) from the original on March 30, 2021. Retrieved March 31, 2021.
88. Brodkin, Jon (March 23, 2021). "Free software advocates seek removal of Richard Stallman and entire FSF board" (https://arstechnica.com/tech-policy/2021/03/free-software-advocates-seek-removal-of-richard-stallman-and-entire-fsf-board/). *Ars Technica*. Archived (https://web.archive.org/web/20210402121805/https://arstechnica.com/tech-policy/2021/03/free-software-advocates-seek-removal-of-richard-stallman-and-entire-fsf-board/) from the original on April 2, 2021. Retrieved March 31, 2021.
89. Salter, Jim (March 29, 2021). "Red Hat withdraws from the Free Software Foundation after Stallman's return" (https://arstechnica.com/gadgets/2021/03/red-hat-withdraws-from-the-stallman-led-free-software-foundation/). *Ars Technica*. Archived (https://web.archive.org/web/20210330213008/https://arstechnica.com/gadgets/2021/03/red-hat-withdraws-from-the-stallman-led-free-software-foundation/) from the original on March 30, 2021. Retrieved March 31, 2021.

90. "USENIX Lifetime Achievement Award ("The Flame")" (https://www.usenix.org/directory/flame.html). USENIX. Archived (https://web.archive.org/web/20071228023853/http://www.usenix.org/directory/flame.html) from the original on December 28, 2007. Retrieved December 5, 2007.
91. Free Software Foundation (2005). "FSF honored with Prix Ars Electronica award" (https://www.fsf.org/news/digital-communities.html). *News Releases*. Free Software Foundation. Archived (https://web.archive.org/web/20061011050335/http://www.fsf.org/news/digital-communities.html) from the original on October 11, 2006. Retrieved December 10, 2006.

External links

- Official website (https://fsf.org/)
- LibrePlanet (https://libreplanet.org/wiki/Main_Page)

Academic Free License

The **Academic Free License** (**AFL**) is a permissive free software license written in 2002 by Lawrence E. Rosen, a former general counsel of the Open Source Initiative (OSI).

Academic Free License	
Author	Lawrence E. Rosen
Latest version	1.2, 2.1, 3.0
Publisher	Lawrence E. Rosen
Published	2002
SPDX identifier	AFL-3.0 AFL-2.1 AFL-2.0 AFL-1.2 AFL-1.1
Debian FSG compatible	?
FSF approved	Yes[1]
OSI approved	Yes[2]
GPL compatible	No[1]
Copyleft	No[1]
Linking from code with a different licence	Yes
Website	rosenlaw.com/OSL3.0-explained.htm (http://rosenlaw.com/OSL3.0-explained.htm)

The license grants similar rights to the BSD, MIT, UoI/NCSA and Apache licenses – licenses allowing the software to be made proprietary – but was written to correct perceived problems with those licenses. The AFL:

- makes clear what software is being licensed by including a statement following the software's copyright notice;
- includes a complete copyright grant to the software;
- contains a complete patent grant to the software;
- makes clear that no trademark rights are granted to the licensor's trademarks;
- warrants that the licensor either owns the copyright or is distributing the software under a license;
- is itself copyrighted, with the right granted to copy and distribute without modification.

The Free Software Foundation consider all AFL versions up to and including 3.0 as incompatible with the GNU GPL.[1] though Eric S. Raymond (a co-founder of the OSI) contends that AFL 3.0 is GPL compatible.[3] In late 2002, an OSI working draft considered it a "best practice" license.[4] In mid-2006, however, the OSI's License Proliferation Committee found it "redundant with more popular licenses",[2] specifically version 2 of the Apache Software License.

See also

foss ***Free and open-source software portal***

- License proliferation
- Open Software License – similar, but reciprocal license by the same author
- Software using the Academic Free License (category)

References

1. Stallman, Richard. "Various Licenses and Comments about Them" (https://www.gnu.org/licenses/license-list.html). Free Software Foundation. Retrieved October 20, 2011.
2. "Academic Free License 3.0" (http://opensource.org/licenses/AFL-3.0). Open Source Initiative. Retrieved January 18, 2016.
3. "Licensing HOWTO" (http://www.catb.org/~esr/Licensing-HOWTO.html#id2853340). Archived (https://web.archive.org/web/20100402084554/http://www.catb.org/~esr/Licensing-HOWTO.html) from the original on April 2, 2010. Retrieved May 15, 2010.
4. Raymond, Eric (November 9, 2002). "Licensing HOWTO" (http://www.catb.org/~esr/Licensing-HOWTO.html). Archived (https://web.archive.org/web/20070704015946/http://www.catb.org/~esr/Licensing-HOWTO.html) from the original on July 4, 2007. Retrieved July 7, 2007.

External links

- Text of the Academic Free License v3.0 (http://opensource.org/licenses/AFL-3.0)
- Allocation of the Risk by Lawrence Rosen (http://www.rosenlaw.com/html/GL14.pdf) (PDF) – reasoning behind the Academic Free License

WIKIPEDIA
The Free Encyclopedia

Apache License

The **Apache License** is a permissive free software license written by the Apache Software Foundation (ASF).[4] It allows users to use the software for any purpose, to distribute it, to modify it, and to distribute modified versions of the software under the terms of the license, without concern for royalties. The ASF and its projects release their software products under the Apache License. The license is also used by many non-ASF projects.

Apache License

The Apache Software Foundation logo

Author	The Apache Software Foundation
Latest version	2.0
Publisher	The Apache Software Foundation
Published	January 2004
SPDX identifier	Apache-2.0 Apache-1.1 Apache-1.0
Debian FSG compatible	Yes[1]
FSF approved	Yes[2]
OSI approved	Yes[3]
GPL compatible	Only version 2.0 is compatible with only GPLv3.[2]
Copyleft	No
Linking from code with a different licence	Yes
Website	www.apache.org/licenses (https://www.apache.org/licenses)

History

Beginning in 1995, the Apache Group (later the Apache Software Foundation) released successive versions of the Apache HTTP Server. Its initial license was essentially the same as the original 4-clause BSD license, with only the names of the organizations changed, and with an additional clause forbidding derivative works from bearing the Apache name.

In July 1999, the Berkeley Software Distribution accepted the argument put to it by the Free Software Foundation and retired their *advertising clause* (clause 3) to form the new 3-clause BSD license. In 2000, Apache did likewise and created the Apache License 1.1, in which derived products are no longer required to include attribution in their advertising materials, only in their documentation. Individual packages licensed under the 1.1 version may have used different wording due to varying requirements for attribution or mark identification, but the binding terms were the same.

In January 2004, ASF decided to depart from the BSD model and produced the Apache License 2.0. The stated goals of the license included making it easier for non-ASF projects to use, improving compatibility with GPL-based software, allowing the license to be included by reference instead of listed in every file, clarifying the license on contributions, and requiring a patent license on contributions that necessarily infringe a contributor's own patents.[5] This license requires the preservation of the copyright notice and disclaimer.

Licensing conditions

The Apache License is permissive; unlike copyleft licenses, it does not require a derivative work of the software, or modifications to the original, to be distributed using the same license. It still requires application of the same license to all unmodified parts. In every licensed file, original copyright, patent, trademark, and attribution notices must be preserved (excluding notices that do not pertain to any part of the derivative works). In every licensed file changed, a notification must be added stating that changes have been made to that file.

If a `NOTICE` text file is included as part of the distribution of the original work, then derivative works must include a readable copy of these notices within a `NOTICE` text file distributed as part of the derivative works, within the source form or documentation, or within a display generated by the derivative works (wherever such third-party notices normally appear).

The contents of the `NOTICE` file do not modify the license, as they are for informational purposes only, and adding more attribution notices as addenda to the `NOTICE` text is permissible, provided that these notices cannot be understood as modifying the license. Modifications may have appropriate copyright notices, and may provide different license terms for the modifications.

Unless explicitly stated otherwise, any contributions submitted by a licensee to a licensor will be under the terms of the license without any terms and conditions, but this does not preclude any separate agreements with the licensor regarding these contributions.

Apache License 2.0

The Apache License 2.0 attempts to forestall potential patent litigation in Section 3. The user is granted a patent license from each contributor to "make, have made, use, offer to sell, sell, import, and otherwise transfer the Work." Through an *in terrorem* clause, if the user sues anyone alleging that the software or a contribution within it constitutes patent infringement, any such patent licenses for that work are terminated.

Compatibility

The Apache Software Foundation and the Free Software Foundation agree that the Apache License 2.0 is a free software license, compatible with the GNU General Public License[5] (GPL) version 3,[2] meaning that code under GPLv3 and Apache License 2.0 can be combined, as long as the resulting software is licensed under the GPLv3.[6]

The Free Software Foundation considers all versions of the Apache License to be incompatible with the previous GPL versions 1 and 2.[2] Furthermore, it considers Apache License versions before 2.0 incompatible with GPLv3. Because of version 2.0's patent license requirements, the Free Software Foundation recommends it over other non-copyleft licenses.[7][2] If the Apache License with the LLVM exception is used, then it is compatible with GPLv2.[8]

Reception and adoption

In October 2012, 8,708 projects located at SourceForge.net were available under the terms of the Apache License.[9] In a blog post from May 2008, Google mentioned that over 25% of the nearly 100,000 projects then hosted on Google Code were using the Apache License,[10] including the Android operating system.[11]

As of 2015, according to Black Duck Software[12] and GitHub,[13] the Apache license is the third most popular license in the FOSS domain after MIT License and GPLv2.

The OpenBSD project does not consider the Apache License 2.0 to be an acceptable free license because of its patent provisions. The OpenBSD policy believes that when the license forces one to give up a legal right that one otherwise has, that license is no longer free. Moreover, the project objects to involving contract law with copyright law, stating "...Copyright law is somewhat standardized by international agreements, contract law differs wildly among jurisdictions. So what the license means in different jurisdictions may vary and is hard to predict."[14]

See also

Free and open-source software portal

- Comparison of free and open-source software licenses
- List of Apache Software Foundation projects
- Software using the Apache license (category)

References

1. "The Apache Software License (ASL)" (https://wiki.debian.org/DFSGLicenses#The_Apache_Software_License_.28ASL.29). *The Big DFSG-compatible Licenses*. Debian Project. Retrieved 6 July 2009.
2. "Apache License, Version 2.0" (https://www.gnu.org/licenses/license-list.html#apache2). *Various Licenses and Comments about Them*. Free Software Foundation. Archived (https://web.archive.org/web/20090716201618/https://www.gnu.org/licenses/license-list.html) from the original on 16 July 2009. Retrieved 6 July 2009.
3. "OSI-approved licenses by name David Gutierrez & David Louie Gutierrez" (http://www.opensource.org/licenses/alphabetical). Open Source Initiative. Archived (https://web.archive.org/web/20110428141712/http://opensource.org/licenses/alphabetical) from the original on 28 April 2011. Retrieved 31 March 2011.
4. New Media Rights (12 September 2008). "Open Source Licensing Guide" (http://www.newmediarights.org/open_source/new_media_rights_open_source_licensing_guide). California Western School of Law. Retrieved 28 November 2015. "*The 'BSD-like' licenses such as the BSD, MIT, and Apache licenses are extremely permissive, requiring little more than attributing the original portions of the licensed code to the original developers in your own code and/or documentation.*"

5. "Apache License, Version 2.0" (https://www.apache.org/licenses/LICENSE-2.0). Apache Software Foundation. Retrieved 15 July 2019.
6. Apache Software Foundation. "Apache License v2.0 and GPL Compatibility" (https://www.apache.org/licenses/GPL-compatibility.html). Archived (https://web.archive.org/web/20080115045635/http://www.apache.org/licenses/GPL-compatibility.html) from the original on 15 January 2008. Retrieved 30 January 2008.
7. "How to choose a license for your own work" (https://gnu.org/licenses/license-recommendations.html). *gnu.org*. Free Software Foundation. 15 December 2018. Retrieved 25 August 2019.
8. "LLVM Exception | Software Package Data Exchange (SPDX)" (https://spdx.org/licenses/LLVM-exception.html). *spdx.org*. Retrieved 22 June 2022.
9. "Projects at SourceForge under Apache License" (http://sourceforge.net/search/?&fq%5B%5D=trove%3A401). Retrieved 28 October 2012.
10. Stein, Greg (28 May 2008). "Standing Against License Proliferation" (http://google-opensource.blogspot.com/2008/05/standing-against-license-proliferation.html). *Google Open Source Blog*.
11. "Licenses" (https://source.android.com/setup/start/licenses). *Android Open Source Project*. Retrieved 25 August 2019.
12. "Top 20 licenses" (https://web.archive.org/web/20160719043600/https://www.blackducksoftware.com/top-open-source-licenses). Black Duck Software. 19 November 2015. Archived from the original (http://www.blackducksoftware.com/resources/data/top-20-licenses) on 19 July 2016. Retrieved 19 November 2015.
13. Balter, Ben (9 March 2015). "Open source license usage on GitHub.com" (https://github.com/blog/1964-license-usage-on-github-com). *GitHub*.
14. "OpenBSD copyright policy" (http://www.openbsd.org/policy.html). *openbsd.org*. OpenBSD Project. 28 May 2019.

External links

- Apache Licenses (https://www.apache.org/licenses)
- Quick Summary of the Apache License 2.0 (https://www.tldrlegal.com/license/apache-license-2-0-apache-2-0)

Apple Public Source License

The **Apple Public Source License** (**APSL**) is the open-source and free software license under which Apple's Darwin operating system was released in 2000. A free and open-source software license was voluntarily adopted to further involve the community from which much of Darwin originated.

The first version of the Apple Public Source License was approved by the Open Source Initiative (OSI).[5] Version 2.0, released July 29, 2003, is also approved as a free software license by the Free Software Foundation (FSF) which finds it acceptable for developers to work on projects that are already covered by this license. However, the FSF recommends that developers should not release new projects under this license, because the partial copyleft is not compatible with the GNU General Public License and allows linking with files released entirely as proprietary software.[4] The license does require that if any derivatives of the original source are released externally, their source should be made available; the Free Software Foundation compares this requirement to a similar one in its own GNU Affero General Public License.[4]

Many software releases from Apple have now been relicensed under the more liberal Apache License, such as the Bonjour Zeroconf stack. However, most OS component source code remains under APSL.

Apple Public Source License

Author	Apple Inc.
Latest version	2.0
Published	August 6, 2003
SPDX identifier	APSL-1.0, APSL-1.1, APSL-1.2, APSL-2.0
Debian FSG compatible	No[1]
FSF approved	Yes (Version 2.0, not versions 1.0, 1.1 and 1.2)[2][3]
OSI approved	Yes
GPL compatible	No[2]
Copyleft	Partial[4]
Linking from code with a different licence	Yes[4]
Website	https://opensource.apple.com/apsl/

See also

Free and open-source software portal

- Software using the Apple Public Source License (category)

References

1. "Apple Public Source License (APSL)" (https://wiki.debian.org/DFSGLicenses#Apple_Public_Source_License_.28APSL.29). *The Big DFSG-compatible Licenses*. Debian Project. Retrieved January 27, 2017.
2. "Apple Public Source License (APSL), version 2.x" (https://www.gnu.org/licenses/license-list.html#apsl2). *Various Licenses and Comments about Them*. Free Software Foundation. Archived (https://web.archive.org/web/20090716201618/http://www.gnu.org/licenses/license-list.html) from the original on July 16, 2009. Retrieved July 6, 2009.
3. "Various Licenses and Comments about Them - GNU Project - Free Software Foundation" (https://www.gnu.org/licenses/license-list.html#apsl1). Gnu.org. September 3, 1999. Retrieved May 13, 2022.
4. FSF website (http://www.fsf.org/licensing/essays/apsl.html)
5. Raymond, Eric. "OSI clarifies the status of the APSL" (https://lwn.net/1999/0318/a/raymond.html). *Linux Weekly News*. Retrieved February 14, 2013.

External links

- Text of the Apple Public Source License (https://opensource.apple.com/apsl)
- Free Software Foundation's opinion on the license (https://www.gnu.org/philosophy/apsl.html)

WIKIPEDIA
The Free Encyclopedia

Artistic License

The **Artistic License** is an open-source license used for certain free and open-source software packages, most notably the standard implementation of the Perl programming language and most CPAN modules, which are dual-licensed under the Artistic License and the GNU General Public License (GPL).

Artistic License

Author	The Perl Foundation
Latest version	2.0
Publisher	The Perl Foundation
Published	?
SPDX identifier	Artistic-1.0 Artistic-1.0-cl8 Artistic-1.0-Perl Artistic-2.0 ClArtistic
Debian FSG compatible	Yes[1]
FSF approved	1.0 No (Yes, for Clarified Artistic License), 2.0 Yes
OSI approved	Yes (both)
GPL compatible	1.0 No (Yes, for Clarified Artistic License), 2.0 Yes
Copyleft	No[2]
Linking from code with a different licence	Yes
Website	www.perlfoundation.org/artistic-license-20.html (https://www.perlfoundation.org/artistic-license-20.html)

History

Artistic License 1.0

The original Artistic License was written by Larry Wall. The name of the license is a reference to the concept of artistic license.

Whether or not the original Artistic License is a free software license is largely unsettled. The Free Software Foundation explicitly called the original Artistic License a non-free license,[3] criticizing it as being "too vague; some passages are too clever for their own good, and their meaning is not clear".[4] The FSF recommended that the license not be used on its own, but approved the common AL/GPL dual-licensing approach for Perl projects.

In response to this, Bradley Kuhn, who later worked for the Free Software Foundation, made a minimal redraft to clarify the ambiguous passages. This was released as the **Clarified Artistic License** and was approved by the FSF. It is used by the Paros Proxy, the JavaFBP toolkit and NcFTP.

The terms of the Artistic License 1.0 were at issue in *Jacobsen v. Katzer* in the initial 2009 ruling by the United States District Court for the Northern District of California declared that FOSS-like licenses could only be enforced through contract law rather than through copyright law, in contexts where contract damages would be difficult to establish.[5] On appeal, a federal appellate court "determined that the terms of the Artistic License are enforceable copyright conditions".[6] The case was remanded to the District Court, which did not apply the superior court's criteria on the grounds that, in the interim, the governing Supreme Court precedent applicable to the case had changed.[7] However, this left undisturbed the finding that a free and open-source license nonetheless has economic value.[8][9] Jacobsen ultimately prevailed in 2010, and the Case established a new standard making terms and conditions under Artistic License 1.0 enforceable through copyright statutes and relevant precedents.[10]

Artistic License 2.0

In response to the Request for Comments (RFC) process for improving the licensing position for Raku, Kuhn's draft was extensively rewritten by Roberta Cairney and Allison Randal for readability and legal clarity, with input from the Perl community. This resulted in the **Artistic License 2.0**, which has been approved as both a free software[11] and open source[12] license.

The Artistic license 2.0 is also notable for its excellent license compatibility with other FOSS licenses due to a relicensing clause, a property other licenses like the GPL lack.[13]

> You may Distribute your Modified Version as Source (either gratis or for a Distributor Fee, and with or without a Compiled form of the Modified Version) [...] provided that you do at least ONE of the following:
>
> [...] (c) allow anyone who receives a copy of the Modified Version to make the Source form of the Modified Version available to others under
>
> (i) the Original License or
>
> (ii) **a license** that permits the licensee to freely copy, modify and redistribute the Modified Version using the same licensing terms that apply to the copy that the licensee received, and requires that the Source form of the Modified Version, and of any works derived from it, be made freely available in that license fees are prohibited but Distributor Fees are allowed.

It has been adopted by some of the Raku implementations, the Mojolicious framework and the NPM. It is also used by the SNEeSe emulator, which was formerly licensed under the Clarified Artistic License.

The OSI recommends that all developers and projects licensing their products with the Artistic License adopt Artistic License 2.0.[14]

See also

- Software using the Artistic license (category)

References

1. "DFSG Licenses – The DFSG and Software Licenses" (http://wiki.debian.org/DFSGLicenses#TheArtisticLicense). *Debian Wiki*. Retrieved November 28, 2010.
2. "Re: For Approval: Artistic License 2.0: msg#00055" (https://web.archive.org/web/20111228154514/http://osdir.com/ml/licenses.open-source.general/2007-03/msg00055.html). March 14, 2007. Archived from the original (http://osdir.com/ml/licenses.open-source.general/2007-03/msg00055.html) on December 28, 2011. Retrieved July 11, 2009.
3. "Explaining Why We Don't Endorse Other Systems - GNU Project - Free Software Foundation" (https://www.gnu.org/philosophy/common-distros.html). Free Software Foundation. Retrieved 2013-01-27. "... it permits software released under the original Artistic License to be included, even though that's a nonfree license."

4. "Various Licenses and Comments about Them - GNU Project - Free Software Foundation (FSF)" (https://www.gnu.org/licenses/license-list.html#NonFreeSoftwareLicense). Fsf.org. Archived (https://web.archive.org/web/20100724023833/https://www.gnu.org/licenses/license-list.html) from the original on 24 July 2010. Retrieved 2010-08-07.
5. New Open Source Legal Decision: Jacobsen & Katzer and How Model Train Software Will Have an Important Effect on Open Source Licensing (http://lawandlifesiliconvalley.blogspot.com/2007/08/new-open-source-legal-decision-jacobsen.html), Radcliffe, Mark (Law & Life: Silicon Valley) (2007-08-22)
6. Opinion, Jacobsen v. Katzer (http://www.cafc.uscourts.gov/images/stories/opinions-orders/08-1001.pdf) Archived (https://web.archive.org/web/20110304135243/http://www.cafc.uscourts.gov/images/stories/opinions-orders/08-1001.pdf) 2011-03-04 at the Wayback Machine, United States Court of Appeals for the Federal Circuit (2008-08-13)
7. United States District Court, N.D. California. (2009-01-05). "Robert JACOBSEN, Plaintiff, v. Matthew KATZER and Kamind Associates, Inc., Defendants" (https://www.leagle.com/decision/20091534609cbfsupp2d92511455). Retrieved 2019-07-11.
8. Menon, Yamini (Spring 2011). "Jacobsen Revisted: Conditions and the Future of Open-Source Software Licenses" (http://digital.law.washington.edu/dspace-law/bitstream/handle/1773.1/1054/6WJLTA311.pdf?sequence=3) (PDF). *Washington Journal of Law, Technology and Arts*. **6** (4): 311–357.
9. WSGR ALERT (2010-02-23). "WSGR ALERT: Settlement Reached in Important Open Source Copyright Infringement Case" (https://www.wsgr.com/WSGR/Display.aspx?SectionName=publications/PDFSearch/wsgralert_jacobsen_katzer.htm). Retrieved 2019-07-11.
10. "McDonagh, L. (2013). Copyright, Contract and FOSS. In: N. Shemtov & I. Walden (Eds.), Free and Open Source Software. (pp. 69-108). OUP Oxford. ISBN 9780199680498" (http://openaccess.city.ac.uk/12602/1/McDonagh%20-%20FOSS.pdf) (PDF). Archived (https://ghostarchive.org/archive/20221009/http://openaccess.city.ac.uk/12602/1/McDonagh%20-%20FOSS.pdf) (PDF) from the original on 2022-10-09. Retrieved 2019-07-11.
11. "Various Licenses and Comments about Them - GNU Project - Free Software Foundation (FSF)" (https://www.gnu.org/licenses/license-list.html#ArtisticLicense2). Fsf.org. Archived (https://web.archive.org/web/20100724023833/https://www.gnu.org/licenses/license-list.html) from the original on 24 July 2010. Retrieved 2010-08-07.
12. "Old Nabble - License Committee Report for May 2007" (https://web.archive.org/web/20070930024228/http://www.nabble.com/License-Committee-Report-for-May-2007-tf3872801.html). Nabble.com. Archived from the original (http://www.nabble.com/License-Committee-Report-for-May-2007-tf3872801.html) on 2007-09-30. Retrieved 2010-03-18.
13. Interview with Allison Randal about Artistic License 2.0 (https://www.theperlreview.com/Interviews/allison-randal-artistic-license.html) Archived (https://web.archive.org/web/20150905151027/https://www.theperlreview.com/Interviews/allison-randal-artistic-license.html) September 5, 2015, at the Wayback Machine on www.theperlreview.com
14. "The Artistic License:Licensing" (https://web.archive.org/web/20090324061628/http://www.opensource.org/licenses/artistic-license-1.0.php). Open Source Initiative. October 31, 2006. Archived from the original (http://opensource.org/licenses/artistic-license-1.0.php) on 24 March 2009. Retrieved March 18, 2009.

External links

- Version 1.0
  - The Artistic License (https://web.archive.org/web/20121016115951/http://www.perlfoundation.org/artistic_license_1_0) – The original Artistic License 1.0, the one which is still used by Perl and CPAN; They use a disjunction of the Artistic License 1.0 and the GNU GPL for Perl 5 and above.

    - The Clarified Artistic License (http://www.ncftp.com/ncftp/doc/LICENSE.txt)
- Version 2.0
    - The Artistic License 2.0 (http://www.perlfoundation.org/artistic_license_2_0) – It's e.g. used by Parrot.
    - 2.0 revision RFC process (http://dev.perl.org/rfc/346.html)
- Prominent uses
    - Dusk (http://crepuscule.sourceforge.net/) – The first online Novel and Blog written under Artistic License 2.0.
    - "R.E.M releases videos under Artistic License 2.0 (https://web.archive.org/web/20080610044848/http://perlbuzz.com/2008/02/rem-releases-videos-under-artistic-license-20.html) – Is about R.E.M.'s choice of the Artistic License 2.0 for videos from one of their albums.

WIKIPEDIA
The Free Encyclopedia

Beerware

Beerware is a tongue-in-cheek software license with permissive terms, which grants the right to do anything with the source code, assuming the license notice is preserved.[3]

Beerware	
Author	Poul-Henning Kamp
Latest version	42
Publisher	Yes
Published	1998[a]
SPDX identifier	Beerware
Debian FSG compatible	Yes
FSF approved	Yes (see "informal license" section)[2]
OSI approved	No
GPL compatible	Yes[2]
Copyleft	No[2]
Linking from code with a different licence	Yes

Description

Should the user of the code consider the software useful, they are encouraged to buy the author a beer "in return" if they ever meet. The Humanitarian-FOSS project at Trinity College recognized the *"version 42"* beerware license variant as an extremely permissive "copyright only" and GPL-compatible license.[3] According to the Free Software Foundation the license would be classified as an "informal" free, non-copyleft and GPL-compatible license, however more detailed licenses are recommended.[2]

Poul-Henning Kamp states preference of his Beerware license to other licenses, such as BSD and GPL, the latter of which he has described as a "joke".[4] The full text of Kamp's license is:[5]

> "THE BEER-WARE LICENSE" (Revision 42): <phk@FreeBSD.ORG> wrote this file. As long as you retain this notice you can do whatever you want with this stuff. If we meet some day, and you think this stuff is worth it, you can buy me a beer in return. Poul-Henning Kamp.

See also

foss ***Free and open-source software portal***

- 0BSD a public domain equivalent license used by Toybox and explicitly allowed for Android
- Anti-copyright license
- Careware
- Comparison of free and open-source software licenses
- Donationware
- WTFPL

Notes

a. According to the Wayback Machine, the earliest revision of Poul-Henning Kamp's website containing the license is 1998. It is unknown whether this was indeed the first year of the license's publication.[1]

References

1. Poul-Henning Kamp (http://web.archive.org/web/20010302183711/people.freebsd.org/~phk/) at the Wayback Machine (archived 2001-03-02)
2. "Various Licenses and Comments about Them § Informal license" (https://www.gnu.org/licenses/license-list.html#informal). *www.gnu.org*. Free Software Foundation. 2016-01-05. Retrieved 2016-01-05.
3. "Beerware License" (https://web.archive.org/web/20160322002352/http://www.cs.trincoll.edu/hfoss/wiki/Chris_Fei:_Beerware_License). Humanitarian-FOSS. Archived from the original (http://www.cs.trincoll.edu/hfoss/wiki/Chris_Fei:_Beerware_License) on 2016-03-22. Retrieved 2015-04-20. "The license is compatible with proprietary licenses and the GNU GPL, as code under this license has no restrictions whatsoever."
4. I think the GNU license is a joke, it fights the capitalism it so much is against with their own tools, and no company is ever going to risk any kind of proximity to so many so vague statements assembled in a license. (http://people.freebsd.org/~phk/)
5. Kamp, Poul-Henning (2004-10-24). "Poul-Henning Kamp" (http://people.freebsd.org/~phk/). Archived (https://web.archive.org/web/20060421154730/http://people.freebsd.org/~phk/) from the original on 21 April 2006. Retrieved 2006-04-24.

BSD licenses

BSD licenses are a family of permissive free software licenses, imposing minimal restrictions on the use and distribution of covered software. This is in contrast to copyleft licenses, which have share-alike requirements. The original BSD license was used for its namesake, the Berkeley Software Distribution (BSD), a Unix-like operating system. The original version has since been revised, and its descendants are referred to as modified BSD licenses.

BSD is both a license and a class of license (generally referred to as BSD-like). The modified BSD license (in wide use today) is very similar to the license originally used for the BSD version of Unix. The BSD license is a simple license that merely requires that all code retain the BSD license notice if redistributed in source code format, or reproduce the notice if redistributed in binary format. The BSD license (unlike some other licenses e.g. GPL) does not require that source code be distributed at all.

Terms

In addition to the original (4-clause) license used for BSD, several derivative licenses have emerged that are also commonly referred to as a "BSD license". Today, the typical BSD license is the 3-clause version, which is revised from the original 4-clause version.

In all BSD licenses as following, <year> is the year of the copyright. As published in BSD, <copyright holder> is "Regents of the University of California".

Previous license

Some releases of BSD prior to the adoption of the 4-clause BSD license used a license that is clearly ancestral to the 4-clause BSD license. These releases include some parts of 4.3BSD-Tahoe (1988), about 1000 files,[1] and Net/1 (1989). Although largely replaced by the 4-clause license, this license can be found in 4.3BSD-Reno, Net/2, and 4.4BSD-Alpha.

Prior BSD License

Author	Regents of the University of California
Publisher	Public domain
Published	1988
SPDX identifier	BSD-4.3TAHOE
Debian FSG compatible	Yes
OSI approved	No
GPL compatible	No
Copyleft	No
Linking from code with a different licence	Yes

> Copyright (c) <year> <copyright holder>. All rights reserved.
>
> Redistribution and use in source and binary forms are permitted provided that the above copyright notice and this paragraph are duplicated in all such forms and that any documentation, advertising materials, and other materials related to such distribution and use acknowledge that the software was

developed by the <copyright holder>. The name c
endorse or promote products derived from th
permission. THIS SOFTWARE IS PROVIDED `'A
IMPLIED WARRANTIES, INCLUDING, W
WARRANTIES OF MERCHANTABILITY A
PURPOSE.

4-clause license (original "BSD License")

The original BSD license contained a clause not found in later licenses, known as the "advertising clause". This clause eventually became controversial, as it required authors of all works deriving from a BSD-licensed work to include an acknowledgment of the original source in all advertising material. This was clause number 3 in the original license text:[6]

BSD License

Author	Regents of the University of California
Publisher	Public domain
Published	1990
SPDX identifier	BSD-4-Clause (see list for more[2])
Debian FSG compatible	Yes[3]
FSF approved	Yes[4]
OSI approved	No[5]
GPL compatible	No[4]
Copyleft	No[4]
Linking from code with a different licence	Yes

> Copyright (c) <year>, <copyright holder> 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. All advertising materials mentioning features or use of this software must display the following acknowledgement: This product includes software developed by the <copyright holder>.
> 4. Neither the name of the <copyright holder> 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 <CO
> EXPRESS OR IMPLIED WARRANTIES, INC
> IMPLIED WARRANTIES OF MERCHANTABI
> PURPOSE ARE DISCLAIMED. IN NO EVEN
> LIABLE FOR ANY DIRECT, INDIRECT, INC
> CONSEQUENTIAL DAMAGES (INCLUDING,
> OF SUBSTITUTE GOODS OR SERVICES; L
> BUSINESS INTERRUPTION) HOWEVER C
> LIABILITY, WHETHER IN CONTRACT, STR
> NEGLIGENCE OR OTHERWISE) ARISING IN
>
> SOFTWARE, EVEN IF ADVISED OF THE POS

This clause was objected to on the grounds that as people changed the license to reflect their name or organization it led to escalating advertising requirements when programs were combined in a software distribution: every occurrence of the license with a different name required a separate acknowledgment. In arguing against it, Richard Stallman has stated that he counted 75 such acknowledgments in a 1997 version of NetBSD.[7] In addition, the clause presented a legal problem for those wishing to publish BSD-licensed software which relies upon separate programs using the GNU GPL: the advertising clause is incompatible with the GPL, which does not allow the addition of restrictions beyond those it already imposes; because of this, the GPL's publisher, the Free Software Foundation, recommends developers not use the license, though it states there is no reason not to use software already using it.[4]

3-clause license ("BSD License 2.0", "Revised BSD License", "New BSD License", or "Modified BSD License")

New BSD License

Author	Regents of the University of California
Publisher	Public domain
Published	22 July 1999[6]
SPDX identifier	BSD-3-Clause (see list for more[2])
Debian FSG compatible	Yes[3]
FSF approved	Yes[8]
OSI approved	Yes[5]
GPL compatible	Yes[8]
Copyleft	No[8]
Linking from code with a different licence	Yes

The advertising clause was removed from the license text in the official BSD license on July 22, 1999, by William Hoskins, Director of the Office of Technology Licensing for UC Berkeley.[6][9][10] On January 31, 2012, UC Berkeley Executive Director of the Office of Intellectual Property and Industry Alliances established that licensees and distributors are no longer required to include the acknowledgement within advertising materials. Accordingly, the advertising clause 3 of the original 4-clause BSD license for any and all software officially licensed under a UC Berkeley version of the BSD license, was deleted in its entirety.[11] Other BSD distributions removed the clause, but many similar clauses remain in BSD-derived code from other sources, and unrelated code using a derived license.

While the original license is sometimes referred to as the "BSD-old", the resulting 3-clause version is sometimes referred to by "BSD-new." Other names include **new BSD**, "revised BSD", "BSD-3", or "3-

clause BSD". This version has been vetted as an Open source license by the OSI as "The BSD License".[5] The Free Software Foundation, which refers to the license as the "Modified BSD License", states that it is compatible with the GNU GPL. The FSF encourages users to be specific when referring to the license by name (i.e. not simply referring to it as "a BSD license" or "BSD-style") to avoid confusion with the original BSD license.[8]

This version allows unlimited redistribution for any purpose as long as its copyright notices and the license's disclaimers of warranty are maintained. The license also contains a clause restricting use of the names of contributors for endorsement of a derived work without specific permission.

> Copyright <year> <copyright holder>
>
> 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. Neither the name of the copyright holder 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 COPYRIGHT HOLDERS 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 COPYRIGHT HOLDER 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.[8]

2-clause license ("Simplified BSD License" or "FreeBSD License")

An even more simplified version has come into use, primarily known for its usage in FreeBSD.[13] It was in use there as early as 29 April 1999[14] and likely well before. The primary difference between it and the New BSD (3-clause) License is that it omits the non-endorsement clause. The FreeBSD version of the license also adds a further disclaimer about views and opinions expressed in the software,[15] though this is not commonly included by other projects.

FreeBSD License

Author	The FreeBSD Project

Publisher	The FreeBSD Project
Published	April 1999 or earlier
SPDX identifier	BSD-2-Clause (see list for more[2])
Debian FSG compatible	Yes
FSF approved	Yes[12]
OSI approved	Yes[5]
GPL compatible	Yes[12]
Copyleft	No[12]
Linking from code with a different licence	Yes

The Free Software Foundation, which refers to the license as the FreeBSD License, states that it is compatible with the GNU GPL. In addition, the FSF encourages users to be specific when referring to the license by name (i.e. not simply referring to it as "a BSD license" or "BSD-style"), as it does with the modified/new BSD license, to avoid confusion with the original BSD license.[12]

> Copyright (c) <year>, <copyright holder>
>
> 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.
>
> THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS 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 COPYRIGHT OWNER 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
SOFTWARE, EVEN IF ADVISED OF THE POSS

Other projects, such as NetBSD, use a similar 2-clause license.[16] This version has been vetted as an Open source license by the OSI as the "Simplified BSD License."[5]

The ISC license without the 'and/or' wording is functionally equivalent, and endorsed by the OpenBSD project as a license template for new contributions.[17]

0-clause license ("BSD Zero Clause License")

The BSD 0-clause license goes further than the 2-clause license by dropping the requirements to include the copyright notice, license text, or disclaimer in either source or binary forms. Doing so forms a public-domain-equivalent license,[20] the same way as MIT No Attribution License. It is known as "0BSD", "Zero-Clause BSD", or "Free Public License 1.0.0".[21][22] It was created by Rob Landley and first used in Toybox when he was disappointed after using the GNU General Public License in BusyBox.[23]

BSD Zero Clause License

Author	Rob Landley
Published	2013
SPDX identifier	0BSD
Debian FSG compatible	Yes
FSF approved	Yes[18]
OSI approved	Yes[19]
GPL compatible	Yes[18]
Copyleft	No
Linking from code with a different licence	Yes

> Copyright (C) [year] by [copyright holder] <[email]>
>
> Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted.
>
> THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.[22]

Other variations

The SPDX License List contains extra BSD license variations. Examples include:[2]

- `BSD-1-Clause`, a license with only the source code retaining clause, used by Berkeley Software Design in the 1990s,[24][25] and later used by the Boost Software License. OSI approved since 2020.[26]
- `BSD-2-Clause-Patent`, a variation of BSD-2-Clause with a patent grant. OSI approved since 2017.[27]
- `BSD-3-Clause-No-Nuclear-Warranty`, a variation of BSD-3-Clause that adds that a piece of software is not licensed for use in a nuclear facility.

License compatibility

Commercial license compatibility

The FreeBSD project argues on the advantages of BSD-style licenses for companies and commercial use-cases due to their license compatibility with proprietary licenses and general flexibility, stating that the BSD-style licenses place only "minimal restrictions on future behavior" and are not "legal time-bombs", unlike copyleft licenses.[28] The BSD License allows proprietary use and allows the software released under the license to be incorporated into proprietary products. Works based on the material may be released under a proprietary license as closed source software, allowing usual commercial usages under them.

FOSS compatibility

The 3-clause BSD license, like most permissive licenses, is compatible with almost all FOSS licenses (and as well proprietary licenses).[29][30]

Two variants of the license, the New BSD License/Modified BSD License (3-clause),[8] and the Simplified BSD License/FreeBSD License (2-clause)[12] have been verified as GPL-compatible free software licenses by the Free Software Foundation, and have been vetted as open source licenses by the Open Source Initiative.[5] The original, 4-clause BSD license has not been accepted as an open source license and, although the original is considered to be a free software license by the FSF, the FSF does not consider it to be compatible with the GPL due to the advertising clause.[4]

Reception and usage

The BSD license family is one of the oldest and most broadly used license families in the free and open-source software ecosystem, and has been the inspiration for a number of other licenses. Many FOSS software projects use a BSD license, for instance the BSD OS family (FreeBSD etc.), Google's Bionic

> Over the years I've become convinced that the BSD license is great for code you don't care about. I'll use it myself. If there's a library routine that I just want to say 'hey,

or Toybox. As of 2015 the BSD 3-clause license ranked in popularity number five according to Black Duck Software[32] and sixth according to GitHub data.[33]

> this is useful to anybody and I'm not going to maintain this,' I'll put it under the BSD license.
>
> -- Linus Torvalds at LinuxCon 2016[31]

See also

Free and open-source software portal

- Comparison of free and open-source software licenses
- Software using the BSD license (category)

References

1. Bostic, Keith (15 June 1988). "4.3BSD-tahoe release" (https://groups.google.com/g/comp.sys.tahoe/c/50ManvdM1-s). Newsgroup: comp.sys.tahoe (news:comp.sys.tahoe). Retrieved 5 December 2021.
2. "SPDX License List" (https://spdx.org/licenses/). *spdx.org*. SPDX Working Group.
3. "License information" (http://www.debian.org/legal/licenses/). Debian. Retrieved 18 February 2010.
4. "Original BSD license" (https://www.gnu.org/licenses/license-list.html#OriginalBSD). *Various Licenses and Comments about Them*. Free Software Foundation. Retrieved 2 October 2010.
5. "The BSD License:Licensing" (http://www.opensource.org/licenses/bsd-license.php). Open Source Initiative. 31 October 2006. Archived (https://web.archive.org/web/20091129081849/http://www.opensource.org/licenses/bsd-license.php) from the original on 29 November 2009. Retrieved 6 December 2009.
6. "To All Licensees, Distributors of Any Version of BSD" (ftp://ftp.cs.berkeley.edu/pub/4bsd/README.Impt.License.Change). University of California, Berkeley. 22 July 1999. Retrieved 15 November 2006.
7. Richard Stallman. "The BSD License Problem" (https://www.gnu.org/philosophy/bsd.html). Free Software Foundation. Archived (https://web.archive.org/web/20061112224151/https://www.gnu.org/philosophy/bsd.html) from the original on 12 November 2006. Retrieved 15 November 2006.
8. "Modified BSD license" (https://www.gnu.org/licenses/license-list.html#ModifiedBSD). *Various Licenses and Comments about Them*. Free Software Foundation. Retrieved 2 October 2010.
9. "Berkeley removes Advertising Clause – Slashdot" (https://bsd.slashdot.org/story/99/09/02/189210/berkeley-removes-advertising-clause). *bsd.slashdot.org*. 2 September 1999. Retrieved 2 September 2021.
10. Comparing the BSD and GPL Licenses (http://timreview.ca/article/67) on Technology Innovation Management Review by Bruce Montague (on October 2007)

11. Katz, Michael. "Executive Director, Office of Intellectual Property and Industry Research Alliances, University of California, Berkeley" (https://ipira.berkeley.edu/sites/default/files/amendment_of_4-clause_bsd_software_license.pdf) (PDF). *Office of Intellectual Property and Industry Research Alliances (IPIRA)*. University of California, Berkeley. Retrieved 19 November 2024.
12. "FreeBSD license" (https://www.gnu.org/licenses/license-list.html#FreeBSD). *Various Licenses and Comments about Them*. Free Software Foundation. Retrieved 2 October 2010.
13. "The FreeBSD Copyright" (http://www.freebsd.org/copyright/freebsd-license.html). The FreeBSD Project. Archived (https://web.archive.org/web/20091125231326/http://www.freebsd.org/copyright/freebsd-license.html) from the original on 25 November 2009. Retrieved 6 December 2009.
14. "The FreeBSD Copyright (as available at archive.org)" (https://web.archive.org/web/19990429093111/http://www.freebsd.org/copyright/freebsd-license.html). The FreeBSD Foundation. Archived from the original on 29 April 1999. Retrieved 7 January 2017.
15. "The FreeBSD Copyright" (https://www.freebsd.org/copyright/freebsd-license.html). *freebsd.org*. Retrieved 25 March 2020.
16. "NetBSD Licensing and Redistribution" (http://www.netbsd.org/about/redistribution.html). The NetBSD Foundation. Retrieved 6 December 2009.
17. "OpenBSD Copyright Policy" (https://www.openbsd.org/policy.html). Retrieved 17 July 2016.
18. "Zero-clause license" (https://www.gnu.org/licenses/license-list.html#Zero-BSD). *Various Licenses and Comments about Them*. Free Software Foundation. Retrieved 2 October 2010.
19. "[License-review] Please rename "Free Public License-1.0.0" to 0BSD" (http://lists.opensource.org/pipermail/license-review_lists.opensource.org/2018-November/003830.html). Retrieved 15 February 2019.
20. "BSD 0-Clause License (0BSD) Explained in Plain English" (https://tldrlegal.com/license/bsd-0-clause-license). *tldrlegal.com*. Retrieved 15 February 2019.
21. "BSD Zero Clause License" (https://spdx.org/licenses/0BSD.html). *spdx.org*. Retrieved 19 February 2021.
22. "Zero-Clause BSD / Free Public License 1.0.0 (0BSD)" (https://opensource.org/licenses/0BSD). *opensource.org*. 5 December 2015. Retrieved 19 February 2021.
23. *Toybox vs BusyBox - Rob Landley, hobbyist* (https://www.youtube.com/watch?v=MkJkyMuBm3g&t=1528s), retrieved 28 April 2023
24. "BSD 1-Clause License" (https://spdx.org/licenses/BSD-1-Clause.html). Software Package Data Exchange (SPDX). 2018. Retrieved 30 May 2022.
25. "Log of /head/include/ifaddrs.h" (https://svnweb.freebsd.org/base/head/include/ifaddrs.h?revision=62606&view=markup). svnweb.freebsd.org. Retrieved 30 May 2022.
26. "1-clause BSD License – Open Source Initiative" (https://opensource.org/license/bsd-1-clause). Open Source Initiative. 13 March 2020. Retrieved 26 March 2024.
27. "BSD+Patent – Open Source Initiative" (https://opensource.org/license/bsdpluspatent). Open Source Initiative. 4 April 2017. Retrieved 26 March 2024.

28. Montague, Bruce (13 November 2013). "Why you should use a BSD style license for your Open Source Project – GPL Advantages and Disadvantages" (http://www.freebsd.org/doc/en_US.ISO8859-1/articles/bsdl-gpl/article.html). FreeBSD. Retrieved 28 November 2015. "In contrast to the GPL, which is designed to prevent the proprietary commercialization of Open Source code, the BSD license places minimal restrictions on future behavior. This allows BSD code to remain Open Source or become integrated into commercial solutions, as a project's or company's needs change. In other words, the BSD license does not become a legal time-bomb at any point in the development process. In addition, since the BSD license does not come with the legal complexity of the GPL or LGPL licenses, it allows developers and companies to spend their time creating and promoting good code rather than worrying if that code violates licensing."
29. Hanwell, Marcus D. (28 January 2014). "Should I use a permissive license? Copyleft? Or something in the middle?" (http://opensource.com/business/14/1/what-license-should-i-use-open-source-project). opensource.com. Retrieved 30 May 2015. "Permissive licensing simplifies things One reason the business world, and more and more developers [...], favor permissive licenses is in the simplicity of reuse. The license usually only pertains to the source code that is licensed and makes no attempt to infer any conditions upon any other component, and because of this there is no need to define what constitutes a derived work. I have also never seen a license compatibility chart for permissive licenses; it seems that they are all compatible."
30. "Licence Compatibility and Interoperability" (https://web.archive.org/web/20150617130550/https://joinup.ec.europa.eu/software/page/licence_compatibility_and_interoperability). *Open-Source Software – Develop, share, and reuse open source software for public administrations*. joinup.ec.europa.eu. Archived from the original (https://joinup.ec.europa.eu/software/page/licence_compatibility_and_interoperability) on 17 June 2015. Retrieved 30 May 2015. "The licences for distributing free or open source software (FOSS) are divided in two families: permissive and copyleft. Permissive licences (BSD, MIT, X11, Apache, Zope) are generally compatible and interoperable with most other licences, tolerating to merge, combine or improve the covered code and to re-distribute it under many licences (including non-free or "proprietary")."
31. Torvalds at LinuxCon Part III: Permissive Licenses and Org Charts (https://fossforce.com/2016/09/torvalds-linuxcon-permissive-licenses-org-charts/) FOSS Force, 2016
32. "Top 20 licenses" (https://web.archive.org/web/20160719043600/https://www.blackducksoftware.com/top-open-source-licenses). Black Duck Software. 19 November 2015. Archived from the original (http://www.blackducksoftware.com/resources/data/top-20-licenses) on 19 July 2016. Retrieved 19 November 2015. "1. MIT license 24%, 2. GNU General Public License (GPL) 2.0 23%, 3. Apache License 16%, 4. GNU General Public License (GPL) 3.0 9%, 5. BSD License 2.0 (3-clause, New or Revised) License 6%, 6. GNU Lesser General Public License (LGPL) 2.1 5%, 7. Artistic License (Perl) 4%, 8. GNU Lesser General Public License (LGPL) 3.0 2%, 9. Microsoft Public License 2%, 10. Eclipse Public License (EPL) 2%"
33. Balter, Ben (9 March 2015). "Open source license usage on GitHub.com" (https://github.com/blog/1964-license-usage-on-github-com). github.com. Retrieved 21 November 2015. " "1 MIT 44.69%, 2 Other 15.68%, 3 GPLv2 12.96%, 4 Apache 11.19%, 5 GPLv3 8.88%, 6 BSD 3-clause 4.53%, 7 Unlicense 1.87%, 8 BSD 2-clause 1.70%, 9 LGPLv3 1.30%, 10 AGPLv3 1.05%"

External links

- *Twenty Years of Berkeley Unix: From AT&T-Owned to Freely Redistributable* (http://www.oreilly.com/catalog/opensources/book/kirkmck.html), Marshall Kirk McKusick, in: Open Sources: Voices from the Open Source Revolution, O'Reilly 1999
- The Amazing Disappearing BSD License (https://urchin.earth.li/~twic/The_Amazing_Disappearing_BSD_License.html)
- BSD License Definition (http://www.linfo.org/bsdlicense.html) – by The Linux Information Project (LINFO)

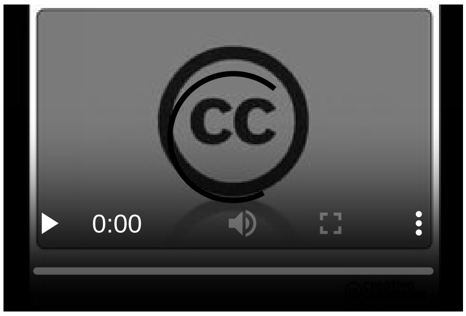
Creative Commons license

(Redirected from Creative Commons licenses)

Creative Commons logo

A **Creative Commons** (**CC**) **license** is one of several public copyright licenses that enable the free distribution of an otherwise copyrighted "work".[a] A CC license is used when an author wants to give other people the right to share, use, and build upon a work that the author has created. CC provides an author flexibility (for example, they might choose to allow only non-commercial uses of a given work) and protects the people who use or redistribute an author's work from concerns of copyright infringement as long as they abide by the conditions that are specified in the license by which the author distributes the work.[1][2][3][4][5]

There are several types of Creative Commons licenses. Each license differs by several combinations that condition the terms of distribution. They were initially released on December 16, 2002, by Creative Commons, a U.S. non-profit corporation founded in 2001. There have also been five versions of the suite of licenses, numbered 1.0 through 4.0.[6] Released in November 2013, the 4.0 license suite is the most current. While the Creative Commons license was originally grounded in the American legal system, there are now several Creative Commons jurisdiction ports which accommodate international laws.[7][8]



A video explaining how Creative Commons licenses can be used in conjunction with commercial licensing arrangements

In October 2014, the Open Knowledge Foundation approved the Creative Commons CC BY, CC BY-SA and CC0 licenses as conformant with the "Open Definition" for content and data.[9][10][11]

History

Lawrence Lessig and Eric Eldred designed the Creative Commons License (CCL) in 2001 because they saw a need for a license between the existing modes of copyright and public domain status. Version 1.0 of the licenses was officially released on 16 December 2002.[12]

Origins

The CCL allows inventors to keep the rights to their innovations while also allowing for some external use of the invention.[13] The CCL emerged as a reaction to the decision in *Eldred v. Ashcroft*, in which the United States Supreme Court ruled constitutional provisions of the Copyright Term Extension Act that extended the copyright term of works to be the last living author's lifespan plus an additional 70 years.[13]

License porting

The original non-localized Creative Commons licenses were written with the U.S. legal system in mind; therefore, the wording may be incompatible with local legislation in other jurisdictions, rendering the licenses unenforceable there. To address this issue, Creative Commons asked its affiliates to translate the various licenses to reflect local laws in a process called "porting".[14] As of July 2011, Creative Commons licenses have been ported to over 50 jurisdictions worldwide.[15]

International use

Chinese use

Working with Creative Commons, the Chinese government adapted the Creative Commons License to the Chinese context, replacing the individual monetary compensation of U.S. copyright law with incentives to Chinese innovators to innovate as a social contribution.[16]

Applicable works

Work licensed under a Creative Commons license is governed by applicable copyright law.[17] This allows Creative Commons licenses to be applied to all work falling under copyright, including: books, plays, movies, music, articles, photographs, blogs, and websites.

Software

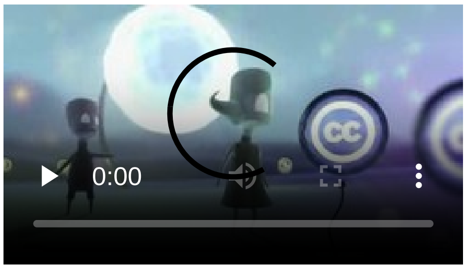
While software is also governed by copyright law and CC licenses are applicable, the CC recommends against using it in software specifically due to backward-compatibility limitations with existing commonly used software licenses.[18][19] Instead, developers may resort to use more software-friendly free and open-source software (FOSS) software licenses. Outside the FOSS licensing use case for software there are several usage examples to utilize CC licenses to specify a "Freeware" license model; examples are The White Chamber, Mari0 or Assault Cube.[20] Despite the status of CC0 as the most free copyright license, the Free Software Foundation does not recommend releasing software into the public domain using the CC0 due to patent concerns.[21]

However, application of a Creative Commons license may not modify the rights allowed by fair use or fair dealing or exert restrictions which violate copyright exceptions.[22] Furthermore, Creative Commons licenses are non-exclusive and non-revocable.[23] Any work or copies of the work obtained under a Creative Commons license may continue to be used under that license.[24]

When works are protected by more than one Creative Commons license, the user may choose any of them.[25]

Aaron Swartz and Lawrence Lessig at the 2002 event for the first release of the licenses

Wanna Work Together? (https://creativecommons.org/videos/wanna-work-together) animation by Creative Commons



The second version (https://creativecommons.org/videos/mayer-and-bettle2) of the Mayer and Bettle (https://creativecommons.org/videos/mayer-and-bettle) promotional animation explaining Creative Commons with Jamendo as an example

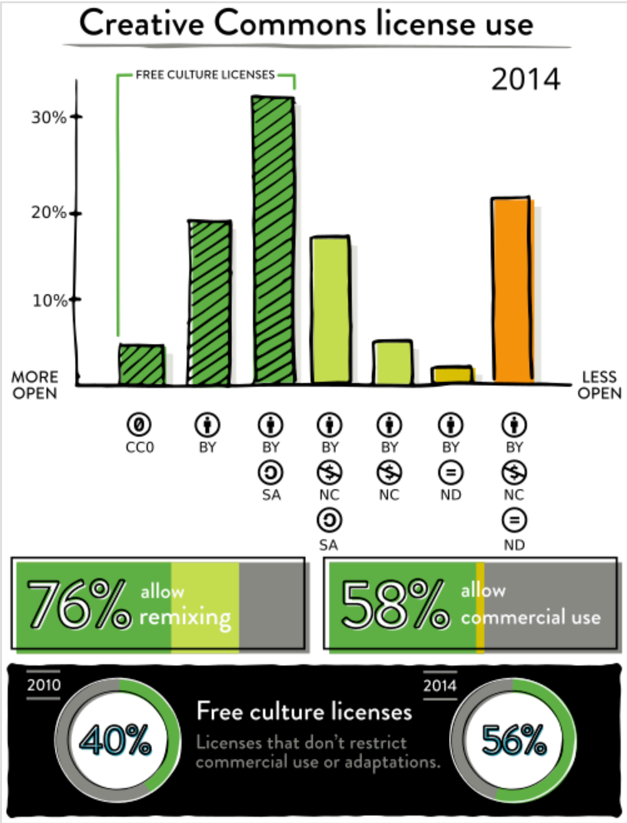
Preconditions

The author, or the licensor in case the author did a contractual transfer of rights, needs to have the exclusive rights on the work. If the work has already been published under a public license, it can be uploaded by any third party, once more on another platform, by using a compatible license, and making reference and attribution to the original license (e.g. by referring to the URL of the original license).[26]

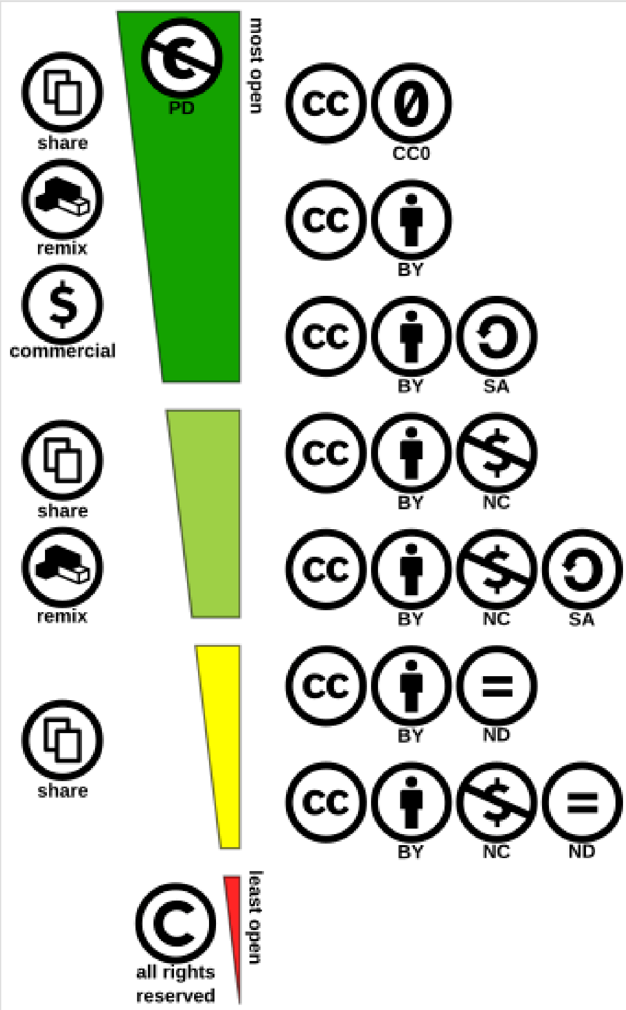
Consequences

The license is non-exclusive, royalty-free, and unrestricted in terms of territory and duration, so it is irrevocable, unless a new license is granted by the author after the work has been significantly modified. Any use of the work that is not covered by other copyright rules triggers the public license. Upon activation of the license, the licensee must adhere to all conditions of the license, otherwise the license agreement is illegitimate, and the licensee would commit a copyright infringement. The author, or the licensor as a proxy, has the legal rights to act upon any copyright infringement. The licensee has a limited period to correct any non-compliance.[26]

Types of licenses



CC license usage in 2014 (top and middle), "Free cultural works" compatible license usage 2010 to 2014 (bottom)



Creative commons license spectrum between public domain (top) and all rights reserved (bottom). Left side indicates the use-cases allowed, right side the license components. The dark green area indicates Free Cultural Works compatible licenses, the two green areas compatibility with the Remix culture.

Four rights

The CC licenses all grant "baseline rights", such as the right to distribute the copyrighted work worldwide for non-commercial purposes and without modification.[27] In addition, different versions of license prescribe different rights, as shown in this table:[28]

Icon	Right	Description
	Attribution (BY)	Licensees may copy, distribute, display, perform and make derivative works and remixes based on it only if they give the author or licensor the credits (attribution) in the manner specified by these. Since version 2.0, all Creative Commons licenses require attribution to the creator and include the BY element. The letters BY are not an abbreviation, unlike the other rights.
	Share-alike (SA)	Licensees may distribute derivative works only under a license identical to ("not more restrictive than") the license that governs the original work. (See also *Copyleft*.) Without share-alike, derivative works might be sublicensed with compatible but more restrictive license clauses, e.g. CC BY to CC BY-NC.
	Non-commercial (NC)	Licensees may copy, distribute, display, perform the work and make derivative works and remixes based on it only for non-commercial purposes.
	No derivative works (ND)	Licensees may copy, distribute, display and perform only verbatim copies of the work, not derivative works and remixes based on it. Since version 4.0, derivative works are allowed but must not be shared.

The last two clauses are not free content licenses, according to definitions such as DFSG or the Free Software Foundation's standards, and cannot be used in contexts that require these freedoms, such as Wikipedia. For software, Creative Commons includes three free licenses created by other institutions: the BSD License, the GNU LGPL, and the GNU GPL.[29]

Mixing and matching these conditions produces sixteen possible combinations, of which eleven are valid Creative Commons licenses and five are not. Of the five invalid combinations, four include both the "ND" and "SA" clauses, which are mutually exclusive; and one includes none of the clauses. Of the eleven valid combinations, the five that lack the "BY" clause have been retired because 98% of licensors requested attribution, though they do remain available for reference on the website.[30][31][32] This leaves six regularly used licenses plus the CC0 public domain declaration.

Six regularly used licenses

The six licenses in most frequent use are shown in the following table. Among them, those accepted by the Wikimedia Foundation – the public domain dedication and two attribution (BY and BY-SA) licenses – allow the sharing and remixing (creating derivative works), including for commercial use, so long as attribution is given.[32][33][34]

License name	Abbreviation	Icon	Attribution required	Allows remix culture	Allows commercial use	Allows Free Cultural Works	Meets the OKF 'Open Definition'
Attribution	CC BY		Yes	Yes	Yes	Yes	Yes
Attribution-ShareAlike	CC BY-SA		Yes	Yes	Yes	Yes	Yes
Attribution-NonCommercial	CC BY-NC		Yes	Yes	No	No	No
Attribution-NonCommercial-ShareAlike	CC BY-NC-SA		Yes	Yes	No	No	No
Attribution-NoDerivatives	CC BY-ND		Yes	No	Yes	No	No
Attribution-NonCommercial-NoDerivatives	CC BY-NC-ND		Yes	No	No	No	No

Zero, public domain

Tool name	Abbreviation	Icon	Attribution required	Allows remix culture	Allows commercial use	Allows Free Cultural Works	Meets the OKF 'Open Definition'
"No Rights Reserved"	CC0		No	Yes	Yes	Yes	Yes

Besides copyright licenses, Creative Commons also offers **CC0**, a tool for relinquishing copyright and releasing material into the public domain.[34] CC0 is a legal tool for waiving as many rights as legally possible.[36] Or, when not legally possible, CC0 acts as fallback as public domain equivalent license.[36] Development of CC0 began in 2007[37] and it was released in 2009.[38][39] A major target of the license was the scientific data community.[40]

CC zero public domain dedication tool logo[35]

In 2010, Creative Commons announced its Public Domain Mark,[41] a tool for labeling works already in the public domain. Together, CC0 and the Public Domain Mark replace the Public Domain Dedication and Certification,[42] which took a U.S.-centric approach and co-mingled distinct operations.

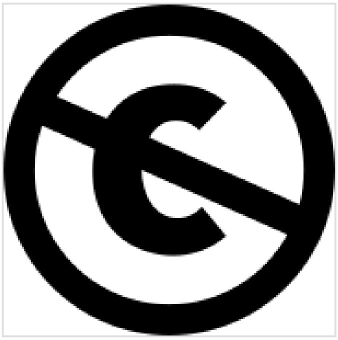
In 2011, the Free Software Foundation added CC0 to its free software licenses. However, the Free Software Foundation currently does not recommend using CC0 to release software into the public domain because it explicitly does not grant a patent license.[21]

In February 2012, CC0 was submitted to Open Source Initiative (OSI) for their approval.[43] However, controversy arose over its clause which excluded from the scope of the license any relevant patents held by the copyright holder. This clause was added for scientific data rather than software, but some members of the OSI believed it could weaken users' defenses against software patents. As a result, Creative Commons withdrew their submission, and the license is not currently approved by the OSI.[40][44]

From 2013 to 2017, the stock photography website Unsplash used the CC0 license,[45][46] distributing several million free photos a month.[47] Lawrence Lessig, the founder of Creative Commons, has contributed to the site.[48] Unsplash moved from using the CC0 license to a custom license in June 2017[49] and to an explicitly nonfree license in January 2018.

In October 2014, the Open Knowledge Foundation approved the Creative Commons CC0 as conformant with the Open Definition and recommend the license to dedicate content to the public domain.[10][11]

In July 2022, Fedora Linux disallowed software licensed under CC0 due to patent rights explicitly not being waived under the license.[50]



Creative Commons Public Domain Mark. Indicates works which have already fallen into (or were given to) the public domain.

Retired licenses

Due to either disuse or criticism, a number of previously offered Creative Commons licenses have since been retired,[30][51] and are no longer recommended for new works. The retired licenses include all licenses lacking the Attribution element other than CC0, as well as the following four licenses:

- **Developing Nations License**: a license which only applies to developing countries deemed to be "non-high-income economies" by the World Bank. Full copyright restrictions apply to people in other countries.[52]
- **Sampling**: parts of the work can be used for any purpose other than advertising, but the whole work cannot be copied or modified[53]
- **Sampling Plus**: parts of the work can be copied and modified for any purpose other than advertising, and the entire work can be copied for noncommercial purposes[54]
- **NonCommercial Sampling Plus**: the whole work or parts of the work can be copied and modified for non-commercial purposes[55]

Version 4.0

The latest version 4.0 of the Creative Commons licenses, released on November 25, 2013, are generic licenses that are applicable to most jurisdictions and do not usually require ports.[56][57][58][28] No new ports have been implemented in version 4.0 of the license.[59] Version 4.0 discourages using ported versions and instead acts as a single global license.[60]

Rights and obligations

Attribution

Since 2004, all current licenses other than the CC0 variant require attribution of the original author, as signified by the BY component (as in the preposition "by").[31] The attribution must be given to "the best of [one's] ability using the information available".[61] Creative Commons suggests the mnemonic "TASL": *title* – *author* – *source* [web link] – [CC] *licence*.
Generally this implies the following:

- **Include any copyright notices (if applicable)**. If the work itself contains any copyright notices placed there by the copyright holder, those notices must be left intact, or reproduced in a way that is reasonable to the medium in which the work is being re-published.
- **Cite the author's name, screen name, or user ID**, etc. If the work is being published on the Internet, it is nice to link that name to the person's profile page, if such a page exists.
- **Cite the work's title or name (if applicable)**, if such a thing exists. If the work is being published on the Internet, it is nice to link the name or title directly to the original work.
- **Cite the specific CC license the work is under**. If the work is being published on the Internet, it is nice if the license citation links to the license on the CC website.

- **Mention if the work is a derivative work or adaptation**. In addition to the above, one needs to identify that their work is a derivative work, e.g., "This is a Finnish translation of [original work] by [author]." or "Screenplay based on [original work] by [author]."

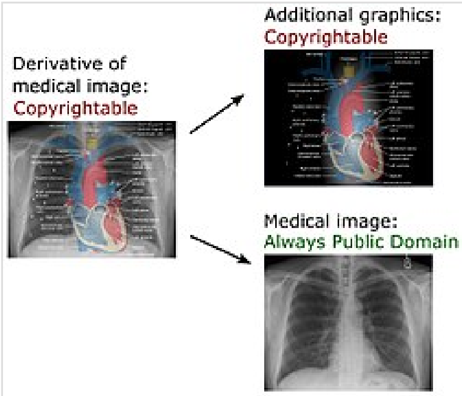
Non-commercial licenses

The NonCommercial license allows image creators to restrict selling and profiting from their works by other parties and thus maintaining free of charge access to images.

The "non-commercial" option included in some Creative Commons licenses is controversial in definition,[62] as it is sometimes unclear what can be considered a non-commercial setting, and application, since its restrictions differ from the principles of open content promoted by other permissive licenses.[63] In 2014 Wikimedia Deutschland published a guide to using Creative Commons licenses as wiki pages for translations and as PDF.[26]

Adaptability

Rights in an adaptation can be expressed by a CC license that is compatible with the status or licensing of the original work or works on which the adaptation is based.[64]



An example of a permitted combination of two works, one being CC BY-SA and the other being public domain

License compatibility chart for combining or mixing two CC licensed works[65][66]

	PUBLIC DOMAIN / PUBLIC DOMAIN	BY	BY SA	BY NC / BY NC SA	BY ND / BY NC ND
PUBLIC DOMAIN / PUBLIC DOMAIN	✓	✓	✓	✓	✗
BY	✓	✓	✓	✓	✗
BY SA	✓	✓	✓	✗	✗
BY NC / BY NC SA	✓	✓	✗	✓	✗
BY ND / BY NC ND	✗	✗	✗	✗	✗

Legal aspects

The legal implications of large numbers of works having Creative Commons licensing are difficult to predict, and there is speculation that media creators often lack insight to be able to choose the license which best meets their intent in applying it.[67]

Some works licensed using Creative Commons licenses have been involved in several court cases.[68] Creative Commons itself was not a party to any of these cases; they only involved licensors or licensees of Creative Commons licenses. When the cases went as far as decisions by judges (that is, they were not dismissed for lack of jurisdiction or were not settled privately out of court), they have all validated the legal robustness of Creative Commons public licenses.

Dutch tabloid

In early 2006, podcaster Adam Curry sued a Dutch tabloid who published photos from Curry's Flickr page without Curry's permission. The photos were licensed under the Creative Commons Non-Commercial license. While the verdict was in favor of Curry, the tabloid avoided having to pay restitution to him as long as they did not repeat the offense. Professor Bernt Hugenholtz, main creator of the Dutch CC license and director of the Institute for Information Law of the University of Amsterdam, commented, "The Dutch Court's decision is especially noteworthy because it confirms that the conditions of a Creative Commons license automatically apply to the content licensed under it, and binds users of such content even without expressly agreeing to, or having knowledge of, the conditions of the license."[69][70][71][72]

Virgin Mobile

In 2007, Virgin Mobile Australia launched an advertising campaign promoting their cellphone text messaging service using the work of amateur photographers who uploaded their work to Flickr using a Creative Commons-BY (Attribution) license. Users licensing their images this way freed their work for use by any other entity, as long as the original creator was attributed credit, without any other compensation required. Virgin upheld this single restriction by printing a URL leading to the photographer's Flickr page on each of their ads. However, one picture, depicting 15-year-old Alison Chang at a fund-raising carwash for her church,[73] caused some controversy when she sued Virgin Mobile. The photo was taken by Alison's church youth counselor, Justin Ho-Wee Wong, who uploaded the image to Flickr under the Creative Commons license.[73] In 2008, the case (concerning personality rights rather than copyright as such) was thrown out of a Texas court for lack of jurisdiction.[74][75]

SGAE vs Fernández

In the fall of 2006, the collecting society Sociedad General de Autores y Editores (SGAE) in Spain sued Ricardo Andrés Utrera Fernández, owner of a disco bar located in Badajoz who played CC-licensed music. SGAE argued that Fernández should pay royalties for public performance of the music between November 2002 and August 2005. The Lower Court rejected the collecting society's claims because the owner of the bar proved that the music he was using was not managed by the society.[76]

In February 2006, the Cultural Association Ladinamo (based in Madrid, and represented by Javier de la Cueva) was granted the use of copyleft music in their public activities. The sentence said:

> Admitting the existence of music equipment, a joint evaluation of the evidence practiced, this court is convinced that the defendant prevents communication of works whose management is entrusted to the plaintiff [SGAE], using a repertoire of authors who have not assigned the exploitation of their rights to the SGAE, having at its disposal a database for that purpose and so it is manifested both by the legal representative of the Association and by Manuela Villa Acosta, in charge of the cultural programming of the association, which is compatible with the alternative character of the Association and its integration in the movement called 'copy left'.[77]

GateHouse Media, Inc. v. That's Great News, LLC

On June 30, 2010, GateHouse Media filed a lawsuit against *That is Great News, LLC*. GateHouse Media owns a number of local newspapers, including *Rockford Register Star*, which is based in Rockford, Illinois. *That is Great News* makes plaques out of newspaper articles and sells them to the people featured in the articles.[78] GateHouse sued *That is Great News* for copyright infringement and breach of contract. GateHouse claimed that *That is Great News* violated the non-commercial and no-derivative works restrictions on GateHouse Creative Commons licensed work when they published the material on their website. The case was settled on August 17, 2010, though the terms of the settlement were not made public.[78][79]

Drauglis v. Kappa Map Group, LLC

In 2007, photographer Art Drauglis uploaded several pictures to the photo-sharing website Flickr, giving them the Creative Commons Attribution-ShareAlike 2.0 Generic License (CC BY-SA). One photo, titled "Swain's Lock, Montgomery Co., MD.", was downloaded by Kappa Map Group, a map-making company, and published in 2012 on the front cover of *Montgomery Co. Maryland Street Atlas*. The text "Photo: Swain's Lock, Montgomery Co., MD Photographer: Carly Lesser & Art Drauglis, Creative Commoms [*sic*], CC-BY-SA-2.0" was placed on the back cover, but nothing on the front indicated authorship.

The validity of CC BY-SA 2.0 as a license was not in dispute. CC BY-SA 2.0 requires that the licensee use nothing less restrictive than the CC BY-SA 2.0 terms. The atlas was sold commercially and not for free reuse by others. The dispute was whether Drauglis' license terms that would apply to "derivative works" applied to the entire atlas. Drauglis sued the defendants in June 2014 for copyright infringement and license breach, seeking declaratory and injunctive relief, damages, fees, and costs. Drauglis asserted, among other things, that Kappa Map Group "exceeded the scope of the License because defendant did not publish the Atlas under a license with the same or similar terms as those under which the Photograph was originally licensed."[80] The judge dismissed the case on that count, ruling that the atlas was not a derivative work of the photograph in the sense of the license, but rather a collective work. Since the atlas was not a derivative work of the photograph, Kappa Map Group did not need to license the entire atlas under the CC BY-SA 2.0 license. The judge also determined that the work had been properly attributed.[81]

In particular, the judge determined that it was sufficient to credit the author of the photo as prominently as authors of similar authorship (such as the authors of individual maps contained in the book) and that the name "CC-BY-SA-2.0" is sufficiently precise to locate the correct license on the internet and can be considered a valid identifier for the license.[3]

Verband zum Schutz geistigen Eigentums im Internet (VGSE)

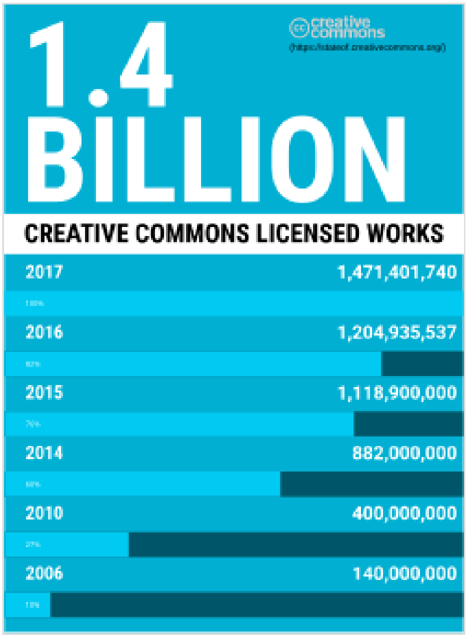
In July 2016, German computer magazine *LinuxUser* reported that a German blogger, Christoph Langner, used two CC BY-licensed photographs from Berlin photographer Dennis Skley on his private blog Linuxundich. Langner duly mentioned the author and the license and added a link to the original. Langner was later contacted by the *Verband zum Schutz geistigen Eigentums im Internet* (VGSE) (Association for the Protection of Intellectual Property in the Internet) with a demand for €2300 for failing to provide the full name of the work, the full name of the author, the license text, and a source link, as is required by the fine print in the license. Of this sum, €40 was to go to the photographer, with the remainder retained by VGSE.[82][83] The Higher Regional Court of Cologne dismissed the claim in May 2019.[84]

Works with a Creative Commons license

Creative Commons maintains a content directory wiki of organizations and projects using Creative Commons licenses.[85] On its website CC also provides case studies of projects using CC licenses across the world.[86] CC licensed content can also be accessed through a number of content directories and search engines.

Unicode symbols

After being proposed by Creative Commons in 2017,[87] Creative Commons license symbols were added to Unicode with version 13.0 in 2020.[88] The circle with an equal sign (meaning *no derivatives*) is present in older versions of Unicode, unlike all the other symbols.



Number of Creative Commons licensed works as of 2017, per *State of the Commons (https://stateof.creativecommons.org/)* report

Name	Unicode	Decimal	UTF-8	Image	Displayed	Unicode block
Circled equals meaning *no derivatives*	U+229C	⊜	E2 8A 9C	⊜	⊜	Mathematical Operators
Circled zero with slash meaning *no rights reserved*	U+1F10D	🄍	F0 9F 84 8D	🄍	🄍	Enclosed Alphanumeric Supplement
Circled anticlockwise arrow meaning *share alike*	U+1F10E	🄎	F0 9F 84 8E	🄎	🄎	Enclosed Alphanumeric Supplement
Circled dollar sign with overlaid backslash meaning *non-commercial*	U+1F10F	🄏	F0 9F 84 8F	🄏	🄏	Enclosed Alphanumeric Supplement
Circled CC meaning *Creative Commons license*	U+1F16D	🅭	F0 9F 85 AD	🅭	🅭	Enclosed Alphanumeric Supplement
Circled C with overlaid backslash meaning *public domain*	U+1F16E	🅮	F0 9F 85 AE	🅮	🅮	Enclosed Alphanumeric Supplement
Circled human figure meaning *attribution, credit*	U+1F16F	🅯	F0 9F 85 AF	🅯	🅯	Enclosed Alphanumeric Supplement

These symbols can be used in succession to indicate a particular Creative Commons license, for example, *CC-BY-SA* (CC-Attribution-ShareAlike) can be expressed with Unicode symbols `CIRCLED CC`, `CIRCLED HUMAN FIGURE` and `CIRCLED ANTICLOCKWISE ARROW` placed next to each other: 🅭🅯🄎

Case law database

In December 2020, the Creative Commons organization launched an online database covering licensing case law and legal scholarship.[89][90]

See also

Free and open-source software portal

Law portal

- Closed captioning – uses a similar CC logo
- Free-culture movement
- Free music
- Free software
- Non-commercial educational station

Notes

a. A "work" is any creative material made by a person. A painting, a graphic, a book, a song and its lyrics, or a photograph of almost anything are all examples of "works".

References

1. Shergill, Sanjeet (May 6, 2017). "The teacher's guide to Creative Commons licenses" (https://web.archive.org/web/20180626111219/https://www.openeducationeuropa.eu/en/blogs/teachers-guide-creative-commons-licenses). *Open Education Europa*. Archived from the original (https://www.openeducationeuropa.eu/en/blogs/teachers-guide-creative-commons-licenses) on June 26, 2018. Retrieved March 15, 2018.
2. "What are Creative Commons licenses?" (https://www.wur.nl/en/article/What-are-Creative-Commons-licenses.htm). *Wageningen University & Research*. June 16, 2015. Archived (https://web.archive.org/web/20180315133811/https://www.wur.nl/en/article/What-are-Creative-Commons-licenses.htm) from the original on March 15, 2018. Retrieved March 15, 2018.
3. "Creative Commons licenses" (http://guides.lib.umich.edu/creativecommons/licenses). *University of Michigan Library*. Archived (https://web.archive.org/web/20181121140426/http://guides.lib.umich.edu/creativecommons/licenses) from the original on November 21, 2018. Retrieved March 15, 2018.
4. "Creative Commons licenses" (https://www.gla.ac.uk/media/media_464806_en.pdf) (PDF). *University of Glasgow*. Archived (https://web.archive.org/web/20180315133739/https://www.gla.ac.uk/media/media_464806_en.pdf) (PDF) from the original on March 15, 2018. Retrieved March 15, 2018.
5. "The Creative Commons licenses" (https://en.unesco.org/open-access/creative-commons-licenses). *UNESCO*. Archived (https://web.archive.org/web/20180315134320/https://en.unesco.org/open-access/creative-commons-licenses) from the original on March 15, 2018. Retrieved March 15, 2018.
6. "License Versions" (https://wiki.creativecommons.org/wiki/License_Versions). *Creative Commons Wiki*. Archived (https://web.archive.org/web/20170630063933/https://wiki.creativecommons.org/wiki/License_Versions) from the original on June 30, 2017. Retrieved July 4, 2017.
7. "Creative Commons | University of Minnesota Libraries" (https://www.lib.umn.edu/services/copyright/creative-commons). *www.lib.umn.edu*. Retrieved October 9, 2024.

8. "What Is a Creative Commons License?" (https://copyrightalliance.org/faqs/what-is-creative-commons-license/). *Copyright Alliance*. September 7, 2016. Archived (https://web.archive.org/web/20240930152512/https://copyrightalliance.org/faqs/what-is-creative-commons-license/) from the original on September 30, 2024. Retrieved October 9, 2024.
9. "Open Definition 2.1" (https://web.archive.org/web/20170127074917/http://opendefinition.org/od/2.1/en/). *Open Definition*. Archived from the original (http://opendefinition.org/od/2.1/en/) on January 27, 2017. Retrieved January 25, 2023.
10. "Conformant Licenses" (http://opendefinition.org/licenses/). *Open Definition*. Archived (https://web.archive.org/web/20160301222106/http://opendefinition.org/licenses/) from the original on March 1, 2016. Retrieved January 25, 2023.
11. Vollmer, Timothy (December 27, 2013). "Creative Commons 4.0 BY and BY-SA licenses approved conformant with the Open Definition" (https://web.archive.org/web/20160304102501/https://blog.creativecommons.org/2013/12/27/creative-commons-4-0-by-and-by-sa-licenses-approved-conformant-with-the-open-definition/). *Creative Commons*. Archived from the original (https://creativecommons.org/2013/12/27/creative-commons-4-0-by-and-by-sa-licenses-approved-conformant-with-the-open-definition/) on March 4, 2016. Retrieved January 25, 2023.
12. "Creative Commons Unveils Machine-Readable Copyright Licenses" (https://web.archive.org/web/20021222175127/http://www.creativecommons.org/press-releases/entry/3476). *Creative Commons*. December 16, 2002. Archived from the original (http://www.creativecommons.org/press-releases/entry/3476) on December 22, 2002.
13. "1.1 The Story of Creative Commons" (https://certificates.creativecommons.org/cccertedu/chapter/1-1-the-story-of-creative-commons/). *Creative Commons Certificate for Educators, Academic Librarians and GLAM*. Archived (https://web.archive.org/web/20230408214258/https://certificates.creativecommons.org/cccertedu/chapter/1-1-the-story-of-creative-commons/) from the original on April 8, 2023. Retrieved April 28, 2021.
14. Murray, Laura J. (2014). *Putting intellectual property in its place : rights discourses, creative labor, and the everyday* (https://www.worldcat.org/oclc/844373100). S. Tina Piper, Kirsty Robertson. Oxford: Oxford University Press. ISBN 978-0-19-933626-5. OCLC 844373100 (https://search.worldcat.org/oclc/844373100). Archived (https://web.archive.org/web/20241005164356/https://search.worldcat.org/title/844373100) from the original on October 5, 2024. Retrieved April 29, 2021.
15. "Worldwide". Creative Commons. Archived from the original on October 15, 2008.
16. Meng, Bingchun (January 26, 2009). "Articulating a Chinese Commons: An Explorative Study of Creative Commons in China" (https://ijoc.org/index.php/ijoc/article/view/441). *International Journal of Communication*. **3**: 16. ISSN 1932-8036 (https://search.worldcat.org/issn/1932-8036).
17. "Creative Commons Legal Code" (https://creativecommons.org/licenses/by/2.5/au/legalcode). Creative Commons. January 9, 2008. Archived (https://web.archive.org/web/20100211175916/http://creativecommons.org/licenses/by/2.5/au/legalcode) from the original on February 11, 2010. Retrieved February 22, 2010.
18. "Creative Commons FAQ: Can I use a Creative Commons license for software?" (http://wiki.creativecommons.org/FAQ#Can_I_use_a_Creative_Commons_license_for_software.3F). Wiki.creativecommons.org. July 29, 2013. Archived (https://web.archive.org/web/20101127205528/http://wiki.creativecommons.org/FAQ#Can_I_use_a_Creative_Commons_license_for_software.3F) from the original on November 27, 2010. Retrieved September 20, 2013.
19. "Non-Software Licenses" (https://choosealicense.com/non-software/). *Choose a License*. Archived (https://web.archive.org/web/20220102152501/https://choosealicense.com/non-software/) from the original on January 2, 2022. Retrieved November 13, 2020.
20. "AssaultCube – License" (http://assault.cubers.net/docs/license.html#entirepackage). *assault.cubers.net*. Archived (https://web.archive.org/web/20101225031357/http://assault.cubers.net/docs/license.html) from the original on December 25, 2010. Retrieved January 30, 2011. "*AssaultCube is FREEWARE. [...] The content, code and images of the AssaultCube website and all documentation are licensed under "Attribution-NonCommercial-ShareAlike 3.0 Unported*"
21. "Various Licenses and Comments about Them" (https://www.gnu.org/licenses/license-list.html#CC0). GNU Project. Archived (https://web.archive.org/web/20100724023833/https://www.gnu.org/licenses/license-list.html) from the original on July 24, 2010. Retrieved April 4, 2015.

22. "Do Creative Commons licenses affect exceptions and limitations to copyright, such as fair dealing and fair use?" (https://wiki.creativecommons.org/wiki/Frequently_Asked_Questions#Do_Creative_Commons_licenses_affect_exceptions_and_limitations_to_copyright.2C_such_as_fair_dealing_and_fair_use.3F). *Frequently Asked Questions – Creative Commons*. Archived (https://web.archive.org/web/20150808182345/https://wiki.creativecommons.org/wiki/Frequently_Asked_Questions#Do_Creative_Commons_licenses_affect_exceptions_and_limitations_to_copyright.2C_such_as_fair_dealing_and_fair_use.3F) from the original on August 8, 2015. Retrieved July 26, 2015.
23. "What if I change my mind about using a CC license?" (https://wiki.creativecommons.org/wiki/Frequently_Asked_Questions#What_if_I_change_my_mind_about_using_a_CC_license.3F). *Frequently Asked Questions – Creative Commons*. Archived (https://web.archive.org/web/20150808182345/https://wiki.creativecommons.org/wiki/Frequently_Asked_Questions#What_if_I_change_my_mind_about_using_a_CC_license.3F) from the original on August 8, 2015. Retrieved July 26, 2015.
24. "What happens if the author decides to revoke the CC license to material I am using?" (https://wiki.creativecommons.org/wiki/Frequently_Asked_Questions#What_happens_if_the_author_decides_to_revoke_the_CC_license_to_material_I_am_using.3F). *Frequently Asked Questions – Creative Commons*. Archived (https://web.archive.org/web/20150808182345/https://wiki.creativecommons.org/wiki/Frequently_Asked_Questions#What_happens_if_the_author_decides_to_revoke_the_CC_license_to_material_I_am_using.3F) from the original on August 8, 2015. Retrieved July 26, 2015.
25. "How do CC licenses operate?" (https://wiki.creativecommons.org/wiki/Frequently_Asked_Questions#How_do_CC_licenses_operate.3F). *Frequently Asked Questions – Creative Commons*. Archived (https://web.archive.org/web/20150808182345/https://wiki.creativecommons.org/wiki/Frequently_Asked_Questions#How_do_CC_licenses_operate.3F) from the original on August 8, 2015. Retrieved July 26, 2015.
26. Till Kreutzer (2014). *Open Content – A Practical Guide to Using Creative Commons Licenses* (https://commons.wikimedia.org/wiki/File:Open_Content_A_Practical_Guide_to_Using_Creative_Commons_Licences_web.pdf) (PDF). Wikimedia Deutschland e.a. ISBN 978-3-940785-57-2. Archived (https://web.archive.org/web/20150404014103/https://commons.wikimedia.org/wiki/File:Open_Content_A_Practical_Guide_to_Using_Creative_Commons_Licences_web.pdf) (PDF) from the original on April 4, 2015. Retrieved March 23, 2015.
27. "Baseline Rights" (http://wiki.creativecommons.org/Baseline_Rights). Creative Commons. June 12, 2008. Archived (https://web.archive.org/web/20100208081209/http://wiki.creativecommons.org/Baseline_Rights) from the original on February 8, 2010. Retrieved February 22, 2010.
28. "Frequently Asked Questions" (https://creativecommons.org/faq/#what-are-creative-commons-licenses). *Creative Commons*. Creative Commons Corporation. August 28, 2020. Archived (https://web.archive.org/web/20101127205528/http://wiki.creativecommons.org/FAQ#what-are-creative-commons-licenses) from the original on November 27, 2010. Retrieved November 26, 2020.
29. "Creative Commons GNU LGPL" (https://web.archive.org/web/20090622222019/http://creativecommons.org/license/cc-lgpl). Archived from the original (https://creativecommons.org/license/cc-lgpl) on June 22, 2009. Retrieved July 20, 2009.
30. "Retired Legal Tools" (https://creativecommons.org/retiredlicenses/). Creative Commons. Archived (https://web.archive.org/web/20160503224254/https://creativecommons.org/retiredlicenses/) from the original on May 3, 2016. Retrieved May 31, 2012.
31. "Announcing (and explaining) our new 2.0 licenses" (https://creativecommons.org/weblog/entry/4216). Creativecommons.org. May 25, 2004. Archived (https://web.archive.org/web/20130921060401/http://creativecommons.org/weblog/entry/4216) from the original on September 21, 2013. Retrieved September 20, 2013.
32. "About The Licenses – Creative Commons" (https://creativecommons.org/licenses/). Creative Commons. Archived (https://web.archive.org/web/20150726003931/http://creativecommons.org/licenses/) from the original on July 26, 2015. Retrieved July 26, 2015.
33. "Creative Commons — Attribution 3.0 United States" (https://creativecommons.org/licenses/by/3.0/us/). Creative Commons. November 16, 2009. Archived (https://web.archive.org/web/20100224022605/http://creativecommons.org/licenses/by/3.0/us/) from the original on February 24, 2010. Retrieved February 22, 2010.
34. "CC0" (https://creativecommons.org/choose/zero). Creative Commons. Archived (https://web.archive.org/web/20100226071411/http://creativecommons.org/choose/zero) from the original on February 26, 2010. Retrieved February 22, 2010.

35. "Downloads" (https://creativecommons.org/about/downloads). Creative Commons. December 16, 2015. Archived (https://web.archive.org/web/20151225052559/https://creativecommons.org/about/downloads) from the original on December 25, 2015. Retrieved December 24, 2015.
36. Kreutzer, Till. "Validity of the Creative Commons Zero 1.0 Universal Public Domain Dedication and its usability for bibliographic metadata from the perspective of German Copyright Law" (https://rd-alliance.org/sites/default/files/cc0-analysis-kreuzer.pdf) (PDF). Archived (https://web.archive.org/web/20170525023540/https://www.rd-alliance.org/sites/default/files/cc0-analysis-kreuzer.pdf) (PDF) from the original on May 25, 2017. Retrieved July 4, 2017.
37. "Creative Commons Launches CC0 and CC+ Programs" (https://web.archive.org/web/20100223053554/http://creativecommons.org/press-releases/entry/7919) (Press release). Creative Commons. December 17, 2007. Archived from the original (https://creativecommons.org/press-releases/entry/7919) on February 23, 2010. Retrieved February 22, 2010.
38. Baker, Gavin (January 16, 2009). "Report from CC board meeting" (http://www.earlham.edu/~peters/fos/2009/01/report-from-cc-board-meeting.html). *Open Access News*. Archived (https://web.archive.org/web/20100919010433/http://www.earlham.edu/~peters/fos/2009/01/report-from-cc-board-meeting.html) from the original on September 19, 2010. Retrieved February 22, 2010.
39. "Expanding the Public Domain: Part Zero" (https://creativecommons.org/weblog/entry/13304). Creativecommons.org. March 11, 2009. Archived (https://web.archive.org/web/20130921060406/http://creativecommons.org/weblog/entry/13304) from the original on September 21, 2013. Retrieved September 20, 2013.
40. Christopher Allan Webber. "CC withdrawl [sic] of CC0 from OSI process" (https://web.archive.org/web/20150906005825/https://lists.opensource.org/pipermail/license-review/2012-February/000231.html). *In the Open Source Initiative Licence review mailing list*. Archived from the original (http://projects.opensource.org/pipermail/license-review/2012-February/000231.html) on September 6, 2015. Retrieved February 24, 2012.
41. "Marking and Tagging the Public Domain: An Invitation to Comment" (https://creativecommons.org/weblog/entry/22940). Creativecommons.org. August 10, 2010. Archived (https://web.archive.org/web/20130921060643/http://creativecommons.org/weblog/entry/22940) from the original on September 21, 2013. Retrieved September 20, 2013.
42. "Copyright-Only Dedication (based on United States law) or Public Domain Certification" (https://creativecommons.org/licenses/publicdomain/). Creative Commons. August 20, 2009. Archived (https://web.archive.org/web/20100223035835/http://creativecommons.org/licenses/publicdomain/) from the original on February 23, 2010. Retrieved February 22, 2010.
43. Carl Boettiger. "OSI recognition for Creative Commons Zero License?" (http://lists.opensource.org/pipermail/license-review_lists.opensource.org/2012-February/). *In the Open Source Initiative Licence review mailing list*. Open Source Initiative. Archived (https://web.archive.org/web/20130926193810/http://projects.opensource.org/pipermail/license-review/2012-February/) from the original on September 26, 2013. Retrieved February 1, 2012.
44. The Open Source Initiative FAQ (October 21, 2007). "What about the Creative Commons "CC0" ("CC Zero") public domain dedication? Is that Open Source?" (http://opensource.org/faq#cc-zero). Open Source Initiative. Archived (https://web.archive.org/web/20130519153636/http://opensource.org/faq#cc-zero) from the original on May 19, 2013. Retrieved May 25, 2013.
45. "Unsplash is a site full of free images for your next splash page" (https://thenextweb.com/entrepreneur/2013/08/14/unsplash-is-a-site-full-of-images-you-can-use-freely-for-your-next-startups-splash-page/). *The Next Web*. August 14, 2013. Archived (https://web.archive.org/web/20151117024732/http://thenextweb.com/entrepreneur/2013/08/14/unsplash-is-a-site-full-of-images-you-can-use-freely-for-your-next-startups-splash-page/) from the original on November 17, 2015. Retrieved November 13, 2015.
46. "License | Unsplash" (https://unsplash.com/license). *unsplash.com*. Archived (https://web.archive.org/web/20151117015514/https://unsplash.com/license) from the original on November 17, 2015. Retrieved November 13, 2015.
47. "Why Building Something Useful For Others Is The Best Marketing There Is" (http://www.fastcompany.com/3042417/how-i-get-it-done/why-building-something-useful-for-others-is-the-best-marketing-there-is). *Fast Company*. February 18, 2015. Archived (https://web.archive.org/web/20151114225726/http://www.fastcompany.com/3042417/how-i-get-it-done/why-building-something-useful-for-others-is-the-best-marketing-there-is) from the original on November 14, 2015. Retrieved November 13, 2015.

48. "Lawrence Lessig | Unsplash Book" (https://web.archive.org/web/20151117022116/https://book.unsplash.com/lawrence-lessig/). *book.unsplash.com*. Archived from the original (https://book.unsplash.com/lawrence-lessig/) on November 17, 2015. Retrieved November 13, 2015.
49. "Community update: Unsplash branded license and ToS changes" (https://creativecommons.org/2017/06/22/unsplash/). June 22, 2017. Archived (https://web.archive.org/web/20180107174853/https://creativecommons.org/2017/06/22/unsplash/) from the original on January 7, 2018. Retrieved January 7, 2018.
50. Claburn, Thomas (July 25, 2022). "Fedora sours on CC 'No Rights Reserved' license" (https://www.theregister.com/2022/07/25/fedora_sours_on_creative_commons/). *The Register*. Archived (https://web.archive.org/web/20221012114806/https://www.theregister.com/2022/07/25/fedora_sours_on_creative_commons/) from the original on October 12, 2022. Retrieved September 14, 2022.
51. Lessig, Lawrence (June 4, 2007). "Retiring standalone DevNations and one Sampling license" (https://creativecommons.org/weblog/entry/7520). Creative Commons. Archived (https://web.archive.org/web/20070707023305/http://creativecommons.org/weblog/entry/7520) from the original on July 7, 2007. Retrieved July 5, 2007.
52. "Developing Nations License" (https://creativecommons.org/licenses/devnations/2.0/). Creative Commons. Archived (https://web.archive.org/web/20120412101407/http://creativecommons.org/licenses/devnations/2.0/) from the original on April 12, 2012. Retrieved April 9, 2012.
53. "Sampling 1.0" (https://creativecommons.org/licenses/sampling/1.0/). Creative Commons. Archived (https://web.archive.org/web/20120316011404/http://creativecommons.org/licenses/sampling/1.0/) from the original on March 16, 2012. Retrieved April 9, 2012.
54. "Sampling Plus 1.0" (https://creativecommons.org/licenses/sampling+/1.0/). Creative Commons. November 13, 2009. Archived (https://web.archive.org/web/20120411100732/http://creativecommons.org/licenses/sampling+/1.0/) from the original on April 11, 2012. Retrieved April 9, 2012.
55. "NonCommercial Sampling Plus 1.0" (https://creativecommons.org/licenses/nc-sampling+/1.0/). Creative Commons. November 13, 2009. Archived (https://web.archive.org/web/20120325110847/http://creativecommons.org/licenses/nc-sampling+/1.0/) from the original on March 25, 2012. Retrieved April 9, 2012.
56. Peters, Diane (November 25, 2013). "CC's Next Generation Licenses — Welcome Version 4.0!" (https://creativecommons.org/weblog/entry/40768). *Creative Commons*. Archived (https://web.archive.org/web/20131126124736/http://creativecommons.org/weblog/entry/40768) from the original on November 26, 2013. Retrieved November 26, 2013.
57. "What's new in 4.0?" (https://creativecommons.org/Version4). *Creative Commons*. 2013. Archived (https://web.archive.org/web/20131129150056/http://creativecommons.org/Version4) from the original on November 29, 2013. Retrieved November 26, 2013.
58. "CC 4.0, an end to porting Creative Commons licences?" (http://www.technollama.co.uk/cc-4-0-an-end-to-porting-creative-commons-licenses). TechnoLlama. September 25, 2011. Archived (https://web.archive.org/web/20130902134000/http://www.technollama.co.uk/cc-4-0-an-end-to-porting-creative-commons-licenses) from the original on September 2, 2013. Retrieved August 11, 2013.
59. "CC Affiliate Network" (http://wiki.creativecommons.org/CC_Affiliate_Network#The_Licensing_Suite). Creative Commons. Archived (https://web.archive.org/web/20110709065022/http://wiki.creativecommons.org/CC_Affiliate_Network#The_Licensing_Suite) from the original on July 9, 2011. Retrieved July 8, 2011.
60. "Frequently Asked Questions: What if CC licenses have not been ported to my jurisdiction?" (http://wiki.creativecommons.org/Frequently_Asked_Questions#What_if_CC_licenses_have_not_been_ported_to_my_jurisdiction.3F). *Creative Commons*. Archived (https://web.archive.org/web/20131127113420/http://wiki.creativecommons.org/Frequently_Asked_Questions#What_if_CC_licenses_have_not_been_ported_to_my_jurisdiction.3F) from the original on November 27, 2013. Retrieved November 26, 2013.
61. "Frequently Frequently Asked Questions" (http://wiki.creativecommons.org/FFAQ#How_do_I_properly_attribute_a_Creative_Commons_licensed_work.3F). Creative Commons. February 2, 2010. Archived (https://web.archive.org/web/20100226070130/http://wiki.creativecommons.org/FFAQ#How_do_I_properly_attribute_a_Creative_Commons_licensed_work.3F) from the original on February 26, 2010. Retrieved February 22, 2010.

62. "Defining Noncommercial report published" (https://creativecommons.org/weblog/entry/17127). Creativecommons.org. September 14, 2009. Archived (https://web.archive.org/web/20130921060638/http://creativecommons.org/weblog/entry/17127) from the original on September 21, 2013. Retrieved September 20, 2013.
63. "The Case for Free Use: Reasons Not to Use a Creative Commons -NC License" (http://freedomdefined.org/Licenses/NC). Freedomdefined.org. August 26, 2013. Archived (https://archive.today/20120709051540/http://freedomdefined.org/Licenses/NC) from the original on July 9, 2012. Retrieved September 20, 2013.
64. "Frequently Asked Questions" (http://wiki.creativecommons.org/Frequently_Asked_Questions#Can_I_combine_material_that_uses_different_Creative_Commons_licenses_into_my_work.3F). *CC Wiki*. Archived (https://web.archive.org/web/20140325054613/http://wiki.creativecommons.org/Frequently_Asked_Questions#Can_I_combine_material_that_uses_different_Creative_Commons_licenses_into_my_work.3F) from the original on March 25, 2014. Retrieved March 25, 2014.
65. "Frequently Asked Questions" (https://creativecommons.org/faq/#can-i-combine-material-under-different-creative-commons-licenses-in-my-work). *Creative Commons*. July 14, 2016. Archived (https://web.archive.org/web/20101127205528/http://wiki.creativecommons.org/FAQ#can-i-combine-material-under-different-creative-commons-licenses-in-my-work) from the original on November 27, 2010. Retrieved August 1, 2016.
66. Creative Commons licenses without a non-commercial or no-derivatives requirement, including public domain/CC0, are all cross-compatible. Non-commercial licenses are compatible with each other and with less restrictive licenses, except for Attribution-ShareAlike. No-derivatives licenses are not compatible with any license, including themselves.
67. Katz, Zachary (2005). "Pitfalls of Open Licensing: An Analysis of Creative Commons Licensing". *IDEA: The Intellectual Property Law Review*. **46** (3): 391.
68. "Creative Commons Case Law" (http://wiki.creativecommons.org/Case_Law). Archived (https://web.archive.org/web/20110901220753/http://wiki.creativecommons.org/Case_Law) from the original on September 1, 2011. Retrieved August 31, 2011.
69. "Creative Commons license upheld by court" (http://news.cnet.com/2100-1030_3-6052292.html). News.cnet.com. Archived (https://web.archive.org/web/20121025165735/http://news.cnet.com/2100-1030_3-6052292.html) from the original on October 25, 2012. Retrieved December 24, 2012.
70. Rimmer, Matthew (January 2007). *Digital Copyright and the Consumer Revolution: Hands Off My Ipod – Matthew Rimmer – Google Böcker* (https://books.google.com/books?id=1ONyncVruj8C&q=The+Dutch+Court%27s+decision+is+especially+noteworthy+because+it+confirms+that+the+conditions+of+a+Creative+Commons+license+automatically&pg=PA271). Edward Elgar. ISBN 9781847207142. Archived (https://web.archive.org/web/20160414003634/https://books.google.com/books?id=1ONyncVruj8C&pg=PA271&dq=The+Dutch+Court%27s+decision+is+especially+noteworthy+because+it+confirms+that+the+conditions+of+a+Creative+Commons+license+automatically&hl=sv&sa=X&ei=SzKvUMOkJseF4gSp4oEY&redir_esc=y#v=onepage&q=The%20Dutch%20Court%27s%20decision%20is%20especially%20noteworthy%20because%20it%20confirms%20that%20the%20conditions%20of%20a%20Creative%20Commons%20license%20automatically&f=false) from the original on April 14, 2016. Retrieved December 24, 2012.
71. "Creative Commons License Upheld by Dutch Court" (http://www.groklaw.net/article.php?story=20060316052623594). *Groklaw*. March 16, 2006. Archived (https://web.archive.org/web/20100505060734/http://www.groklaw.net/article.php?story=20060316052623594) from the original on May 5, 2010. Retrieved September 2, 2006.
72. "Creative Commons Licenses Enforced in Dutch Court" (https://creativecommons.org/weblog/entry/5823). March 16, 2006. Archived (https://web.archive.org/web/20110906074741/http://creativecommons.org/weblog/entry/5823) from the original on September 6, 2011. Retrieved August 31, 2011.
73. Cohen, Noam. "Use My Photo? Not Without Permission" (https://www.nytimes.com/2007/10/01/technology/01link.html). *The New York Times*. Archived (https://web.archive.org/web/20110615133400/http://www.nytimes.com/2007/10/01/technology/01link.html) from the original on June 15, 2011. Retrieved September 25, 2007. "One moment, Alison Chang, a 15-year-old student from Dallas, is cheerfully goofing around at a local church-sponsored car wash, posing with a friend for a photo. Weeks later, that photo is posted online and catches the eye of an ad agency in Australia, and the altered image of Alison appears on a billboard in Adelaide as part of a Virgin Mobile advertising campaign."

74. Evan Brown (January 22, 2009). "No personal jurisdiction over Australian defendant in Flickr right of publicity case" (https://web.archive.org/web/20110713050011/http://blog.internetcases.com/2009/01/22/no-personal-jurisdiction-over-australian-defendant-in-flickr-right-of-publicity-case/). Internet Cases, a blog about law and technology. Archived from the original (http://blog.internetcases.com/2009/01/22/no-personal-jurisdiction-over-australian-defendant-in-flickr-right-of-publicity-case/) on July 13, 2011. Retrieved September 25, 2010.
75. "Lawsuit Against Virgin Mobile and Creative Commons – FAQ" (https://creativecommons.org/weblog/entry/7680). September 27, 2007. Archived (https://web.archive.org/web/20110907054202/http://creativecommons.org/weblog/entry/7680) from the original on September 7, 2011. Retrieved August 31, 2011.
76. Mia Garlick (March 23, 2006). "Spanish Court Recognizes CC-Music" (https://creativecommons.org/2006/03/23/spanishcourtrecognizesccmusic/). Creative Commons. Archived (https://web.archive.org/web/20230926051259/https://creativecommons.org/2006/03/23/spanishcourtrecognizesccmusic/) from the original on September 26, 2023. Retrieved November 22, 2023.
77. "Sentencia nº 12/2006 Juzgado de lo Mercantil nº 5 de Madrid" (https://web.archive.org/web/20151126165923/http://www.derecho-internet.org/node/359). *Derecho de Internet* (in Spanish). Archived from the original (http://www.derecho-internet.org/node/359) on November 26, 2015. Retrieved December 24, 2015.
78. Evan Brown (July 2, 2010). "New Copyright Lawsuit Involves Creative Commons" (http://blog.internetcases.com/2010/07/02/new-copyright-lawsuit-involves-creative-commons/). Internet Cases. Archived (https://web.archive.org/web/20120621215029/http://blog.internetcases.com/2010/07/02/new-copyright-lawsuit-involves-creative-commons/) from the original on June 21, 2012. Retrieved April 20, 2012.
79. "GateHouse Media v. That's Great News" (http://www.citmedialaw.org/threats/gatehouse-media-v-thats-great-news#description). Citizen Media Law Project. August 5, 2010. Archived (https://web.archive.org/web/20120502165158/http://www.citmedialaw.org/threats/gatehouse-media-v-thats-great-news#description) from the original on May 2, 2012. Retrieved April 20, 2012.
80. "Memorandum Opinion" (https://www.gpo.gov/fdsys/pkg/USCOURTS-dcd-1_14-cv-01043/pdf/USCOURTS-dcd-1_14-cv-01043-0.pdf) (PDF). United States District Court for the District of Columbia. August 18, 2015. Archived (https://web.archive.org/web/20160921202242/https://www.gpo.gov/fdsys/pkg/USCOURTS-dcd-1_14-cv-01043/pdf/USCOURTS-dcd-1_14-cv-01043-0.pdf) (PDF) from the original on September 21, 2016. Retrieved August 29, 2016.
81. Guadamuz, Andres (October 24, 2015). "US Court interprets copyleft clause in Creative Commons licenses" (http://www.technollama.co.uk/us-court-interprets-copyleft-clause-in-creative-commons-licenses). *TechnoLlama*. Archived (https://web.archive.org/web/20151222123232/http://www.technollama.co.uk/us-court-interprets-copyleft-clause-in-creative-commons-licenses) from the original on December 22, 2015. Retrieved December 10, 2015.
82. Luther, Jörg (July 2016). "Kleingedrucktes – Editorial" (http://www.linux-community.de/Internal/Artikel/Print-Artikel/LinuxUser/2016/07/Kleingedrucktes) [Fine print – Editorial]. *LinuxUser* (in German) (7/2016). ISSN 1615-4444 (https://search.worldcat.org/issn/1615-4444). Archived (https://web.archive.org/web/20160915150620/http://www.linux-community.de/Internal/Artikel/Print-Artikel/LinuxUser/2016/07/Kleingedrucktes) from the original on September 15, 2016. Retrieved September 9, 2016.
83. "Abmahnung des Verbandes zum Schutz geistigen Eigentums im Internet (VSGE)" (https://web.archive.org/web/20160914075031/https://www.recht-freundlich.de/abmahnung-urheberrechtlich-filesharing/abmahnung-des-verbandes-zum-schutz-geistigen-eigentums-im-internet-vsge) [Notice to cease and desist from the Association for the Protection of Intellectual Property in the Internet (VSGE)] (in German). Hannover, Germany: Feil Rechtsanwaltsgesellschaft. January 8, 2014. Archived from the original (https://www.recht-freundlich.de/abmahnung-urheberrechtlich-filesharing/abmahnung-des-verbandes-zum-schutz-geistigen-eigentums-im-internet-vsge) on September 14, 2016. Retrieved September 9, 2016.
84. "Creative Commons-Foto-Abmahnung: Rasch Rechtsanwälte setzen erfolgreich Gegenansprüche durch" (https://www.anwalt.de/rechtstipps/creative-commons-foto-abmahnung-rasch-rechtsanwaelte-setzen-erfolgreich-gegenansprueche-durch_155491.html) [Creative Commons photo notice: Rasch attorneys successfully enforce counterclaims]. *anwalt.de* (in German). May 22, 2019. Archived (https://web.archive.org/web/20191219023851/https://www.anwalt.de/rechtstipps/creative-commons-foto-abmahnung-rasch-rechtsanwaelte-setzen-erfolgreich-gegenansprueche-durch_155491.html) from the original on December 19, 2019. Retrieved December 18, 2019.

85. "Content Directories" (http://wiki.creativecommons.org/Content_Directories). creativecommons.org. Archived (https://web.archive.org/web/20090430010815/http://wiki.creativecommons.org/Content_Directories) from the original on April 30, 2009. Retrieved April 24, 2009.
86. "Case Studies" (http://wiki.creativecommons.org/Casestudies). Creative Commons. Archived (https://web.archive.org/web/20111224111652/http://wiki.creativecommons.org/Casestudies) from the original on December 24, 2011. Retrieved December 20, 2011.
87. "Proposal to add CC license symbols to UCS" (http://www.unicode.org/L2/L2017/17242r-creative-commons.pdf) (PDF). *Unicode*. July 24, 2017. Archived (https://web.archive.org/web/20210326234933/http://www.unicode.org/L2/L2017/17242r-creative-commons.pdf) (PDF) from the original on March 26, 2021. Retrieved August 21, 2020.
88. Steuer, Eric (March 18, 2020). "The Unicode Standard Now Includes CC License Symbols" (https://creativecommons.org/2020/03/18/the-unicode-standard-now-includes-cc-license-symbols/). *Creative Commons*. Archived (https://web.archive.org/web/20200727183045/https://creativecommons.org/2020/03/18/the-unicode-standard-now-includes-cc-license-symbols/) from the original on July 27, 2020. Retrieved July 6, 2020.
89. Salazar, Krystle (December 3, 2020). "Explore the new CC legal database site!" (https://creativecommons.org/2020/12/03/explore-the-new-cc-legal-database-site/). *Creative Commons*. Mountain View, California, US. Archived (https://web.archive.org/web/20210103130857/https://creativecommons.org/2020/12/03/explore-the-new-cc-legal-database-site/) from the original on January 3, 2021. Retrieved January 3, 2021.
90. Creative Commons. "Creative Commons Legal Database" (https://legaldb.creativecommons.org/). *Creative Commons*. Mountain View, California, US. Archived (https://web.archive.org/web/20210117035744/https://legaldb.creativecommons.org/) from the original on January 17, 2021. Retrieved January 3, 2021.

External links

- Official website (https://creativecommons.org)
- Full selection of licenses (https://creativecommons.org/licenses/)
- CC License options (https://creativecommons.org/share-your-work/cclicenses/)
- *Licenses*. Overview of free licenses. freedomdefined.org
- Web-friendly formatted summary of CC BY-SA 4.0 (https://creativecommons.org/licenses/by-sa/4.0/)

Common Development and Distribution License

Common Development and Distribution License	
Author	Sun Microsystems, Oracle Corporation
Latest version	1.1
Publisher	Oracle Corporation
SPDX identifier	CDDL-1.1 CDDL-1.0
Debian FSG compatible	Yes
FSF approved	Yes (only 1.0)[1]
OSI approved	Yes (only 1.0)[2]
GPL compatible	No[1]
Copyleft	Yes, file-level[1]
Linking from code with a different licence	Yes[1]

The **Common Development and Distribution License** (**CDDL**) is a free and open-source software license,[3] produced by Sun Microsystems, based on the Mozilla Public License (MPL). Files licensed under the CDDL can be combined with files licensed under other licenses, whether open source or proprietary.[2] In 2005 the Open Source Initiative approved the license. The Free Software Foundation (FSF) considers it a free software license, but one which is incompatible with the GNU General Public License (GPL).[1]

Terms

Derived from the Mozilla Public License 1.1,[4] the CDDL tries to address some of the problems of the MPL.[5] Like the MPL, the CDDL is a weak copyleft license in-between GPL license and BSD/MIT permissive licenses, requiring only source code files under CDDL to remain under CDDL.

Unlike strong copyleft licenses like the GPL, mixing of CDDL licensed source code files with source code files under other licenses is permitted without relicensing. The resulting compiled software product ("binary") can be licensed and sold under a different license, as long as the source code is still available under CDDL, which should enable more commercial business cases, according to Sun.[5][6][7]

Like the MPL the CDDL includes a patent grant to the licensee from all contributors ("patent peace"). However, in section 2.1(d), the patent grant is lost if the code implementing a patented feature is modified.[8]

History

The previous software license used by Sun for its open source projects was the Sun Public License (SPL), also derived from the Mozilla Public License. The CDDL license is considered by Sun (now Oracle) to be SPL version 2.[9]

The CDDL was developed by a Sun Microsystems team (among them Solaris kernel engineer Andrew Tucker[10][11] and Claire Giordano[12]), based on the MPL version 1.1. On December 1, 2004 the CDDL was submitted for approval to the Open Source Initiative[12] and was approved as an open source license in mid January 2005. The second CDDL proposal, submitted in early January 2005, includes some corrections that prevent the CDDL from being in conflict with European Copyright law and to allow single developers to use the CDDL for their work.

In 2006, in the first draft of the OSI's license proliferation committee report, the CDDL is one of nine preferred licenses listed as popular, widely used, or with strong communities.[13]

While the Free Software Foundation (FSF) also considered the CDDL a free software license, they saw some incompatibilities with their GNU General Public License (GPL).[1]

GPL compatibility

The question of whether and when both licenses are incompatible sparked debates in the free software domain in 2004 to 2006.[14][15] For instance, the FSF considered the CDDL incompatible to their GPL license, without going into detail until 2016.[16]

CDDL is one of several Open Source Licenses which are incompatible with GPL. This characteristic was inherited from the MPL 1.1 (fixed with the MPL 2.0 according to the FSF[1]) and results from a complex interaction of several clauses;[14][17] the root of the problem being GPL virality, similar to other cases of GPL incompatibility.[18] Some people argue that Sun (or the Sun engineer) as creator of the license made the CDDL intentionally GPL incompatible.[14] According to Danese Cooper one of the reasons for basing the CDDL on the Mozilla license was that the Mozilla license is GPL-incompatible. Cooper stated, at the 6th annual Debian conference, that the engineers who had written the Solaris kernel requested that the license of OpenSolaris be GPL-incompatible.[19]

> Mozilla was selected partially because it is GPL incompatible. That was part of the design when they released OpenSolaris. ... the engineers who wrote Solaris ... had some biases about how it should be released, and you have to respect that.

Simon Phipps (Sun's Chief Open Source Officer at the time), who had introduced Cooper as "the one who actually wrote the CDDL",[20] did not immediately comment, but later in the same video, he says, referring back to the license issue, "I actually disagree with Danese to some degree",[21] while describing the strong preference among the engineers who wrote the code for a BSD-like license, which was in conflict with Sun's preference for something copyleft, and that waiting for legal clearance to release some parts of the code under the then unreleased GNU GPL v3 would have taken several years, and would probably also have involved mass resignations from engineers (unhappy with either the delay, the GPL, or both—this is not clear from the video).

Later, in September 2006, Phipps rejected Cooper's assertion in even stronger terms.[22] Similarly, Bryan Cantrill, who was at Sun at that time and involved in the release of CDDL licensed software stated in 2015 that he and his colleagues expected in 2006 the fast emergence of CDDL licensed software into the Linux ecosystem and the CDDL being not an obstacle.[23]

cdrtools controversy

The GPL compatibility question was also the source of a controversy behind a partial relicensing of cdrtools to the CDDL which had been previously all GPL. In 2006, the Debian project declared the cdrtools legally undistributable because the build system was licensed under the CDDL.[24]

The author, Jörg Schilling, claimed that smake is an independent project and does not violate the GPLv3.[25] Schilling also argued that even though the GPL requires all scripts required to build the work to be licensed freely, they do not necessarily have to be under the GPL.[26][27] Thus not causing an incompatibility that violates the license.

He also argued that in "combined works" (in contrast to "derived works") GPL and CDDL licensed code is compatible.[28][29]

Red Hat's attorneys have prevented cdrtools from being in Fedora or Red Hat Enterprise Linux, arguing that Schilling has an "unorthodox" view of copyright law that isn't shared by their legal counsel or the Free Software Foundation.[30]

ZFS in the Linux kernel

In 2015, the CDDL to GPL compatibility question reemerged when Ubuntu announced inclusion of OpenZFS by default.[31]

In 2016 Ubuntu announced that a legal review resulted in the conclusion that it is legally acceptable to use ZFS as binary kernel module in Linux. (As opposed to building it into the kernel image itself.)[32]

Others followed Ubuntu's conclusion, for instance James E. J. Bottomley argued there cannot be "a convincing theory of harm" developed, making it impossible to bring the case to court.[33]

Eben Moglen, co-author of the GPLv3 and founder of the SFLC, argued that while the letter of the GPL might be violated, the spirit of both licenses is unharmed, which would be the relevant aspect in court.[34]

The SFLC mentioned also that a precedent exists with the Andrew File System's kernel module, which is not considered a derivative work of the kernel by the kernel developers.[35][36]

On the other hand, Bradley M. Kuhn and attorney Karen M. Sandler from the Software Freedom Conservancy[37] argued that Ubuntu would violate both licenses, as a binary ZFS module would be a derivative work of the kernel.[38] In April 2016, the Ubuntu 16.04 LTS release included the CDDL-licensed ZFS on Linux.[39]

Adoption

Example projects released under CDDL:

- OpenSolaris (including DTrace, initially released alone, and ZFS)
- illumos (as OpenSolaris OS/Net, continuation project) and illumos distributions[40]
- OpenZFS multi platform open source volume manager and file system
- NetBeans IDE and RCP

- GlassFish
- JWSDP
- Project DReaM
- cdrtools
- OpenDJ

See also

Free and open-source software portal

- Dual-licensing
- GNAT Modified General Public License
- List of software licenses

References

1. "Various Licenses and Comments About Them - Common Development and Distribution License" (https://www.gnu.org/licenses/license-list.html#CDDL). Free Software Foundation. Retrieved 2020-10-14.
2. "Can code licensed under the CDDL be combined with code licensed under other open source licenses?" (https://web.archive.org/web/20091006181308/http://opensolaris.org/os/about/faq/licensing_faq/#CDDL-combo). *OpenSolaris FAQ: Common Development and Distribution License (CDDL)*. OpenSolaris. Archived from the original (http://www.opensolaris.org/os/about/faq/licensing_faq/#CDDL-combo) on 2009-10-06.
3. *Common Development and Distribution License (CDDL) Information* (https://web.archive.org/web/20090304142159/http://www.sun.com/cddl/), archived from the original (http://www.sun.com/cddl/) on 2009-03-04, "We have drafted a new open source license ..."
4. CDDL_MPL_redline.pdf (https://web.archive.org/web/20060816050912/http://www.sun.com/cddl/CDDL_MPL_redline.pdf) on sun.com (archived)
5. CDDL Why Summary (https://web.archive.org/web/20050214114513/http://www.sun.com/cddl/CDDL_why_summary.html) on sun.com (archived, 2005)
6. McNealy: CDDL is 'best of both worlds' (https://www.zdnet.com/article/mcnealy-cddl-is-best-of-both-worlds/) on zdnet.com by Aaron Tan (September 14, 2005)
7. CDDL (https://tldrlegal.com/license/common-development-and-distribution-license-%28cddl-1.0%29-explained) on tldrlegal.com
8. "Common Development and Distribution License 1.0 | Open Source Initiative" (https://opensource.org/licenses/CDDL-1.0). *opensource.org*. 31 October 2006. Retrieved 2020-07-18.
9. "SPL to CDDL as of NetBeans 5.0 - Why change licenses?" (https://web.archive.org/web/20070224082827/http://www.netbeans.org/about/legal/license-change.html). NetBeans. Archived from the original (http://www.netbeans.org/about/legal/license-change.html#Why_change_licenses?) on 2007-02-24. Retrieved 2006-12-31. "The SPL was based on the Mozilla license - as CDDL is as well. [..] One way to think of the CDDL is as a cleaned-up version of the Mozilla license - anyone can reuse it as-is. It's the SPL version 2.0."
10. "Andy Tucker on the CDDL" (https://alanhargreaves.wordpress.com/2005/04/12/andy-tucker-on-the-cddl/). *Alan Hargreaves' Blog*. 12 April 2005.

11. Open source licenses, IP, and CDDL (https://web.archive.org/web/20061111055302/http://blogs.sun.com/tucker/) on Andrew Tuckers blog, "as one of the drafters of the CDDL I can at least comment on what the license says, and on our intentions in creating it." (Tuesday April 12, 2005)
12. "For Approval: Common Development and Distribution License (CDDL)" (https://lwn.net/Articles/114840/). 1 December 2004.
13. First draft of OSI's license proliferation report (https://web.archive.org/web/20120205011112/http://www.crynwr.com/cgi-bin/ezmlm-cgi?3%3Amss%3A11636%3A200607%3Anknhhdligldemhkfbhpd). Archived from the original (http://crynwr.com/cgi-bin/ezmlm-cgi?3:mss:11636:200607:nknhhdligldemhkfbhpd) Archived (https://web.archive.org/web/20140104020911/http://crynwr.com/cgi-bin/ezmlm-cgi?3:mss:11636:200607:nknhhdligldemhkfbhpd) 2014-01-04 at the Wayback Machine on 2012-02-05. Retrieved 2013-01-03.
14. "Sun Proposes New Open-Source License" (https://archive.today/20150817133221/http://www.eweek.com/c/a/Linux-and-Open-Source/Sun-Proposes-New-OpenSource-License). Archived from the original (http://www.eweek.com/c/a/Linux-and-Open-Source/Sun-Proposes-New-OpenSource-License) on August 17, 2015.
15. "The Blog of Ben Rockwood" (http://cuddletech.com/blog/pivot/entry.php?id=31).
16. "Interpreting, enforcing and changing the GNU GPL, as applied to combining Linux and ZFS" (https://www.fsf.org/licensing/zfs-and-linux). Free Software Foundation. April 11, 2016. Retrieved 2017-07-27.
17. "MPL / GPL Incompatibility" (http://www.tomhull.com/ocston/docs/mozgpl.html). Retrieved 2007-12-03.
18. chandan (2006-09-18). "Copyrights, Licenses and CDDL Illustrated" (https://web.archive.org/web/20150529114715/https://blogs.oracle.com/chandan/entry/copyrights_licenses_and_cddl_illustrated). blogs.oracle.com. Archived from the original (https://blogs.oracle.com/chandan/entry/copyrights_licenses_and_cddl_illustrated) on 2015-05-29. Retrieved 2015-05-29. "A common misconception is about CDDL and GPL incompatibility. (Incompatibility in the sense: to combine two source files, one under GPL and another under CDDL, to create a common executable.) GPL is incompatible with most licenses like Mozilla Public License, Apache, and CDDL. GPL wants you erase those licenses and use GPL in that place, where as these licenses do not permit erasing them. Hence the incompatibility deadlock."
19. Danese Cooper (2006). *OpenSolaris and CDDL discussion at Debconf 2006* (http://meetings-archive.debian.net/pub/debian-meetings/2006/debconf6/theora-small/2006-05-14/tower/OpenSolaris_Java_and_Debian-Simon_Phipps__Alvaro_Lopez_Ortega.ogg) (Ogg Theora). Event occurs at 27:26. "Mozilla was selected partially because it is GPL incompatible. That was part of the design when they released OpenSolaris. ... the engineers who wrote Solaris ... had some biases about how it should be released, and you have to respect that" (alternate URL (https://web.archive.org/web/20110722120048/http://caesar.acc.umu.se/pub/debian-meetings/2006/debconf6/theora-small/2006-05-14/tower/OpenSolaris_Java_and_Debian-Simon_Phipps__Alvaro_Lopez_Ortega.ogg), see 27:27 through 28:24)
20. Simon Phipps (2006). *OpenSolaris and CDDL discussion at Debconf 2006* (http://meetings-archive.debian.net/pub/debian-meetings/2006/debconf6/theora-small/2006-05-14/tower/OpenSolaris_Java_and_Debian-Simon_Phipps__Alvaro_Lopez_Ortega.ogg) (Ogg Theora). *Debconf 2006*. Event occurs at 13:00. "...we have got Danese Cooper in the room, and she is the one who actually wrote the CDDL..."
21. Simon Phipps (2006). *OpenSolaris and CDDL discussion at Debconf 2006* (http://meetings-archive.debian.net/pub/debian-meetings/2006/debconf6/theora-small/2006-05-14/tower/OpenSolaris_Java_and_Debian-Simon_Phipps__Alvaro_Lopez_Ortega.ogg) (Ogg Theora). Event occurs at 36:00. "I actually disagree with Danese to some degree..."

22. Phipps, Simon (2006-09-04). "Re: Danese Cooper claims CDDL made incompatible with GPL on purpose" (https://marc.info/?l=opensolaris-discuss&m=115740406507420). *OpenSolaris-Discuss List*. Retrieved 2019-03-07. "Nonetheless she is wrong to characterise the opinion of the Solaris engineering team in the way she does. She is speaking this way because she lost an argument inside Sun, not because her view is representative of the views of Sun or its staff in the way she claims. She, along with many actual engineers, was an advocate of using GPL for OpenSolaris but the need to release rather than wait for one of {GPL v3, Mozilla license revision, encumbrance removal} meant that this was not possible. I am still furious with her for the statement she made at DebConf, which was spiteful and an obstacle to a united FOSS movement."
23. Bryan Cantrill (2015-04-06). "I am the CTO of Joyent, the father of DTrace and an OS kernel developer for 20 years. AMA!" (https://www.reddit.com/r/IAmA/comments/31ny87/i_am_the_cto_of_joyent_the_father_of_dtrace_and/cq3bs9z?context=3). reddit.com. Retrieved 2016-03-11. "Question: Was the CDDL designed to prevent Sun technologies from entering Linux? - BC: Great question, and the answer was that we didn't know -- but the expectation was that it would be ported to Linux relatively quickly. I remember vividly standing over a terminal with a bunch of people as we actually launched OpenSolaris (like, clicked carriage return on making the DTrace code live -- which was the first in the chute), and the Sun Legal guy and I were chatting. We were both wondering if DTrace was going to show up in Linux in a month or if it would take two years. But that was the range of guesses: neither of us believed that the Linux community themselves would hold up CDDL as an obstacle, and certainly if you told me that a decade later, DTrace wouldn't be in Linux because of licensing FUD, I wouldn't have believed you. Of course, in hindsight, it all seems so clear: NIH is enormously powerful, and we were fools for discounting it."
24. "cdrtools - a tale of two licenses [LWN.net]" (https://lwn.net/Articles/195167/). *lwn.net*. Retrieved 2020-07-18.
25. "Cdrtools (Cdrecord) release information" (http://cdrtools.sourceforge.net/private/cdrecord.html). *cdrtools.sourceforge.net*. Retrieved 2020-07-18.
26. "The GNU General Public License" (https://www.gnu.org/licenses/gpl.html). Retrieved 2009-10-24.
27. "Die GPL kommentiert und erklärt Online-Version" (https://web.archive.org/web/20150908033333/http://www.oreilly.de/german/freebooks/gplger/) (in German). O'Reilly. Archived from the original (http://www.oreilly.de/german/freebooks/gplger) on 2015-09-08. Retrieved 2010-11-17.
28. "Neuer Streit um cdrtools" (http://www.pro-linux.de/news/1/10155/neuer-streit-um-cdrtools.html). *Pro-Linux* (in German). "Laut Aussagen von Jörg Schilling sind die Lizenzen durchaus miteinander kompatibel. Die Regeln werden oftmals falsch ausgelegt. Die Aussagen der FSF-Verantwortlichen seien oft widersprüchlich und in sich nicht schlüssig."
29. "OSSCC GPL" (http://www.osscc.net/en/gpl.html).
30. "Forbidden items - Fedora Project Wiki" (https://fedoraproject.org/wiki/Forbidden_items#cdrtools). *fedoraproject.org*. Retrieved 2020-07-18.
31. Michael Larabel (6 October 2015). "Ubuntu Is Planning To Make The ZFS File-System A "Standard" Offering" (https://www.phoronix.com/scan.php?page=news_item&px=Ubuntu-ZFS-Standard-Plans). Phoronix.
32. Dustin Kirkland (18 February 2016). "ZFS Licensing and Linux" (https://insights.ubuntu.com/2016/02/18/zfs-licensing-and-linux/). *Ubuntu Insights*. Canonical.

33. Are GPLv2 and CDDL incompatible? (http://blog.hansenpartnership.com/are-gplv2-and-cddl-incompatible/) on hansenpartnership.com by James E. J. Bottomley, "What the above analysis shows is that even though we presumed combination of GPLv2 and CDDL works to be a technical violation, there's no way actually to prosecute such a violation because we can't develop a convincing theory of harm resulting. Because this makes it impossible to take the case to court, effectively it must be concluded that the combination of GPLv2 and CDDL, provided you're following a GPLv2 compliance regime for all the code, is allowable." (23 February 2016)
34. Moglen, Eben; Choudhary, Mishi (26 February 2016). "The Linux Kernel, CDDL and Related Issues" (https://softwarefreedom.org/resources/2016/linux-kernel-cddl.html).
35. Moglen, Eben; Choudhary, Mishi (26 February 2016). "The Linux Kernel, CDDL and Related Issues" (https://softwarefreedom.org/resources/2016/linux-kernel-cddl.html). "Historically, there's been things like the original Andrew filesystem module: a standard filesystem that really wasn't written for Linux in the first place, and just implements a UNIX filesystem. Is that derived just because it got ported to Linux that had a reasonably similar VFS interface to what other UNIXes did? ... Personally, I think that case wasn't a derived work, and I was willing to tell the AFS guys so."
36. Copying (https://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/tree/COPYING) on git.kernel.org "NOTE! This copyright does *not* cover user programs that use kernel services by normal system calls – this is merely considered normal use of the kernel, and does *not* fall under the heading of "derived work"."
37. Software Freedom Law Center Appoints Two New Attorneys to Defend and Support Free and Open Source Software (http://www.softwarefreedom.org/news/2005/oct/31/new-attorneys/) (October 31, 2005)
38. GPL Violations Related to Combining ZFS and Linux (https://sfconservancy.org/blog/2016/feb/25/zfs-and-linux/) on sfconservancy.org by Bradley M. Kuhn and Karen M. Sandler, "Conservancy (as a Linux copyright holder ourselves), along with the members of our coalition in the GPL Compliance Project for Linux Developers, all agree that Canonical and others infringe Linux copyrights when they distribute zfs.ko."
39. "openzfs/zfs" (https://github.com/openzfs/zfs). *GitHub*. Retrieved 2020-07-18.
40. "illumos Distributions" (https://wiki.illumos.org/display/illumos/Distributions). *The illumos Family*. illumos. 20 March 2017.

External links

- *Common Development and Distribution License (CDDL) Information* (https://web.archive.org/web/20090304142159/http://www.sun.com/cddl/), archived from the original (http://www.sun.com/cddl/) on 2009-03-04
  - "CDDL 1.0 copy at opensource.org" (http://opensource.org/licenses/CDDL-1.0). 31 October 2006. Retrieved 9 April 2013.
  - *Redline diffs between MPL1.1 and CDDL* (https://web.archive.org/web/20090304142159/http://www.sun.com/cddl/CDDL_MPL_redline.pdf) (PDF), p. 9, archived from the original (http://www.sun.com/cddl/CDDL_MPL_redline.pdf) (PDF) on 2009-03-04
  - *Summary description of changes* (https://web.archive.org/web/20090304142159/http://www.sun.com/cddl/CDDL_why_summary.html), archived from the original (http://www.sun.com/cddl/CDDL_why_summary.html) on 2009-03-04
  - *Detailed description of changes* (https://web.archive.org/web/20090304142159/http://www.sun.com/cddl/CDDL_why_details.html), archived from the original (http://www.sun.com/cddl/CDDL_why_details.html) on 2009-03-04

- FAQ on CDDL on Open Solaris Site (https://web.archive.org/web/20071027082141/http://www.opensolaris.org/os/about/faq/licensing_faq/)
- Copyrights, Licenses and CDDL Illustrated (https://web.archive.org/web/20150529114715/https://blogs.oracle.com/chandan/entry/copyrights_licenses_and_cddl_illustrated) on oracle.com (2006)
- The Common Development and Distribution License (https://lwn.net/Articles/114839/), Linux Weekly News Editorial (December 8, 2004)
- CDDL Analysis from a DFSG perspective, and Opinion Piece (http://soundadvice.id.au/blog/2005/02/04/#cddl) (2005)

Wikipedia
The Free Encyclopedia

Eclipse Public License

The **Eclipse Public License** (**EPL**) is a free and open source software license most notably used for the Eclipse IDE and other projects by the Eclipse Foundation. It replaces the Common Public License (CPL) and removes certain terms relating to litigations related to patents.[6]

The Eclipse Public License is designed to be a business-friendly free software license, and features weaker copyleft provisions than licenses such as the GNU General Public License (GPL).[7] The receiver of EPL-licensed programs can use, modify, copy and distribute the work and modified versions, in some cases being obligated to release their own changes.[8]

The EPL is listed as a free software license by the Free Software Foundation (FSF) and approved by the Open Source Initiative (OSI).[3][2]

Discussion of a new version of the EPL began in May 2013.[9] Version 2.0 was announced on 24 August 2017.[4]

On 20 January 2021, the license steward for the license was changed from Eclipse.org Foundation, Inc. (Delaware, USA) to Eclipse Foundation AISBL (Brussels, Belgium).[10]

Eclipse Public License

Author	Eclipse Foundation
Latest version	2.0
Published	24 August 2017
SPDX identifier	EPL-2.0 EPL-1.0
Debian FSG compatible	Yes[1]
FSF approved	Yes[2]
OSI approved	Yes[3]
GPL compatible	Optionally but not by default[4]
Copyleft	Limited[2]
Linking from code with a different licence	Yes[5]
Website	eclipse.org/legal/eplfaq.php (https://eclipse.org/legal/eplfaq.php)

Compatibility

The EPL 1.0 is not compatible with the GPL, and a work created by combining a work licensed under the GPL with a work licensed under the EPL cannot be lawfully distributed.[7] The GPL requires that "[any distributed work] that ... contains or is derived from the [GPL-licensed] Program ... be licensed as a whole ... under the terms of [the GPL]", and that the distributor not "impose any further restrictions on the recipients' exercise of the rights granted". The EPL, however, requires that anyone distributing the work grant every recipient a license to any patents that they might hold that cover the modifications they have made.[7] Because this is a "further restriction" on the recipients, distribution of such a combined work does not satisfy the GPL.[2]

The EPL, in addition, does not contain a patent retaliation clause.[2]

Derivative works

According to article 1(b) of the EPL, additions to the original work may be licensed independently, including under a proprietary license, provided such additions are "separate modules of software" and do not constitute a derivative work.[11][8] Changes and additions which do constitute a derivative work must be licensed under the same terms and conditions of the EPL, which includes the requirement to make source code available.[8]

Linking to code (for example to a library) licensed under EPL automatically does not mean that your program is a derivative work. Eclipse Foundation interprets the term "derivative work" in a way that is consistent with the definition in the U.S. Copyright Act, as applicable to computer software.[12]

Later versions

If a new version of the EPL is published the user/contributor can choose to distribute the software under the version with which he or she received it or upgrade to the new version.[8]

Comparison with the CPL

The EPL was based on the Common Public License (CPL),[13] but there are some differences between the two licenses:

- The Eclipse Foundation replaces IBM as the Agreement Steward in the EPL
- The EPL patent clause is revised by deleting the sentence from section 7 of the CPL[6]

The Eclipse Foundation sought permission from contributors to re-licence their CPL code under the EPL.[14]

Version 2.0

Version 2.0 of the Eclipse Public License (SPDX code `EPL-2.0`) was announced on 24 August 2017.[4] The Eclipse Foundation maintains an FAQ.[15] The FSF has analyzed the license in relation to GPL license compatibility and added it to their official list.[16] The bare license notice is available in several formats, including plain text.[17]

In terms of GPL compatibility, the new license allows the initial contributor to a new project to opt in to a secondary license that provides explicit compatibility with the GNU General Public License version 2.0, or any later version. If this optional designation is absent, then the Eclipse license remains source incompatible with the GPL (any version).[4][16]

Other changes include:[15]

- the license now applies to "files" not "modules"
- the choice of law provision has been removed

- the new license is suitable for scripting languages, including JavaScript

The Eclipse Foundation advises that version 1.0 is deprecated and that projects should migrate to version 2.0. Relicensing is a straightforward matter and does not require the consent of all contributors, past and present. Rather, the version 1.0 license allows a project (preferably after forming a consensus) to adopt any new version by simply updating the relevant file headers and license notices.[15]: §3

Notable projects

In addition to the Eclipse Foundation, the EPL is used in some other projects, especially those running on the Java virtual machine.

Licensed solely under the EPL

- AT&T KornShell[18]
- Clojure (and ClojureScript)
- Graphviz
- Jikes RVM
- JUnit[19]
- Mondrian
- OpenDaylight Project[20]
- UWIN

Multi-licensed under the EPL and one or more other licenses

- Eclipse OMR (https://eclipse.org/omr)
- Eclipse OpenJ9
- Jetty
- JRuby[21]

See also

Free and open-source software portal

- Software using the Eclipse Public License (category)

References

1. "Copyright information for Debian package eclipse-emf" (https://web.archive.org/web/20110607225635/http://packages.debian.org/changelogs/pool/main/e/eclipse-emf/eclipse-emf_2.5.0-1/eclipse-emf.copyright). Archived from the original (http://packages.debian.org/changelogs/pool/main/e/eclipse-emf/eclipse-emf_2.5.0-1/eclipse-emf.copyright) on 7 June 2011. Retrieved 18 February 2011.
2. "Various Licenses and Comments about Them" (https://www.gnu.org/philosophy/license-list.html#EPL). Retrieved 3 June 2016.

3. "OSI approval" (http://www.opensource.org/licenses/alphabetical). Retrieved 21 June 2007.
4. Milinkovich, Mike (24 August 2017). "Eclipse Public License version 2.0 approved by OSI and Eclipse Foundation Board of Directors" (https://www.eclipse.org/org/press-release/20170829eplv2.php). *Eclipse Foundation*. Ottawa, Canada. Retrieved 19 October 2017. "The new EPL v2 will now optionally allow EPL licensed projects to be compatible with the GPL. ... The new EPL v2 will allow initial contributors to new projects to specify if they want their EPL v2 licensed project to be GPL compatible."
5. "Eclipse Public License (EPL) Frequently Asked Questions : linking section" (http://www.eclipse.org/legal/eplfaq.php#LINK). Retrieved 2 October 2013.
6. In section 7, this sentence is in CPL 1.0, but not EPL 1.0: "If Recipient institutes patent litigation against a Contributor with respect to a patent applicable to software (including a cross-claim or counterclaim in a lawsuit), then any patent licenses granted by that Contributor to such Recipient under this Agreement shall terminate as of the date such litigation is filed."
7. "Open Source Software: a legal guide | LawGives" (https://web.archive.org/web/20200730221407/https://www.lawgives.com/guide/55ce85817777775593000144/Open-Source-Software-a-legal-guide). *LawGives*. Archived from the original (https://www.lawgives.com/guide/55ce85817777775593000144/Open-Source-Software-a-legal-guide) on 30 July 2020. Retrieved 30 July 2020.
8. "Eclipse Public License - v 1.0" (http://www.eclipse.org/legal/epl-v10.html). Retrieved 12 September 2006.
9. "Community Review of the Eclipse Public License" (https://mmilinkov.wordpress.com/2013/05/31/community-review-of-the-eclipse-public-license/). 31 May 2013.
10. "[License-review] Change in Steward for the Eclipse Public License 2.0" (http://lists.opensource.org/pipermail/license-review_lists.opensource.org/2021-April/005153.html). *lists.opensource.org*. Retrieved 2 May 2021.
11. "Eclipse Public License (EPL) Frequently Asked Questions" (http://www.eclipse.org/legal/eplfaq.php). Retrieved 18 December 2009.
12. Beaton, Wayne. "EPL-2.0 FAQ | The Eclipse Foundation" (https://www.eclipse.org/legal/epl-2.0/faq.php#h.dywt3xhx2jg). *www.eclipse.org*. Retrieved 25 June 2019.
13. "CPL to EPL Transition Plan" (http://www.eclipse.org/legal/CPL2EPLTransitionPlan.pdf) (PDF). 12 September 2006.
14. "CPL To EPL Transition Plan Frequently Asked Questions" (http://www.eclipse.org/legal/cpl2epl/cpl2eplfaq.php). Retrieved 21 December 2015.
15. Beaton, Wayne. "Eclipse Public License 2.0 FAQ" (https://www.eclipse.org/org/documents/epl-2.0/faq.php). *Eclipse Foundation*. Ottawa, Canada. Retrieved 20 October 2017.
16. Robertson, Donald (17 October 2017). "Eclipse Public License version 2.0 added to license list" (http://www.fsf.org/blogs/licensing/eclipse-public-license-version-2-0-added-to-license-list-1). *Free Software Foundation*. Retrieved 20 October 2017.
17. *Eclipse Public License — v 2.0* (https://www.eclipse.org/org/documents/epl-2.0/EPL-2.0.txt) (TXT). Ottawa, Canada: Eclipse Foundation. 2017. Retrieved 20 October 2017.
18. "ast-open package" (https://web.archive.org/web/20150402135842/http://www2.research.att.com/~astopen/download/gen/ast-open.html). Archived from the original (http://www2.research.att.com/~astopen/download/gen/ast-open.html) on 2 April 2015.
19. "JUnit 5" (https://github.com/junit-team/junit5). *GitHub*. 22 October 2021.
20. "FAQ | OpenDaylight" (https://web.archive.org/web/20140429061958/http://www.opendaylight.org/resources/faq). *www.opendaylight.org*. Archived from the original (http://www.opendaylight.org/resources/faq#5) on 29 April 2014.
21. "jruby/COPYING at master - jruby/jruby" (https://github.com/jruby/jruby/blob/master/COPYING). GitHub. 1 January 2014. Retrieved 11 February 2014.

External links

- The Eclipse Public License, version 1.0 (http://www.eclipse.org/legal/epl-v10.html)
- The Eclipse Public License, version 2.0 (http://www.eclipse.org/legal/epl-v20.html)
- Eclipse Public License FAQ (http://www.eclipse.org/legal/eplfaq.php)
- EPL v1.0 on OSI (http://opensource.org/licenses/EPL-1.0)
- EPL v2.0 on OSI (http://opensource.org/licenses/EPL-2.0)

WIKIPEDIA
The Free Encyclopedia

GNU General Public License

The **GNU General Public Licenses** (**GNU GPL**, or simply **GPL**) are a series of widely used free software licenses, or copyleft licenses, that guarantee end users the freedoms to run, study, share, or modify the software.[7] The GPL was the first copyleft license available for general use. It was originally written by Richard Stallman, the founder of the Free Software Foundation (FSF), for the GNU Project. The license grants the recipients of a computer program the rights of the Free Software Definition.[8] The licenses in the GPL series are all copyleft licenses, which means that any derivative work must be distributed under the same or equivalent license terms. It is more restrictive than the Lesser General Public License, and even further distinct from the more widely used permissive software licenses such as BSD, MIT, and Apache.

Historically, the GPL license family has been one of the most popular software licenses in the free and open-source software (FOSS) domain.[7][9][10][11][12] Prominent free software programs licensed under the GPL include the Linux kernel and the GNU Compiler Collection (GCC). David A. Wheeler argues that the copyleft provided by the GPL was crucial to the success of Linux-based systems, giving the programmers who contributed to the kernel assurance that their work would benefit the whole world and remain free, rather than being exploited by software companies that would not have to give anything back to the community.[13]

In 2007, the third version of the license (GPLv3) was released to address some perceived problems with the second version (GPLv2) which were discovered during the latter's long-time usage.

To keep the license current, the GPL license includes an optional "any later version" clause, allowing users to choose between the original terms or the terms in new versions as updated by the FSF. Software projects licensed with the optional "or later" clause include the GNU Project, while projects like the

GNU General Public License

GPLv3 Free Software
Free as in Freedom

Author	Richard Stallman
Latest version	3
Publisher	Free Software Foundation
Published	25 February 1989
SPDX identifier	GPL-3.0-or-later GPL-3.0-only GPL-2.0-or-later GPL-2.0-only GPL-1.0-or-later GPL-1.0-only
Debian FSG compatible	Yes[1]
FSF approved	Yes[2]
OSI approved	Yes (applies to GPLv3-only and GPLv2-only)[3]
Copyleft	Yes[2][4][5]
Linking from code with a different licence	Software licensed under GPL compatible licenses only, with the exception of the LGPL which allows all programs.[6]
Website	www.gnu.org/licenses/gpl.html (https://www.gnu.org/licenses/gpl.html)

Linux kernel are licensed under GPLv2 only. The "or any later version" clause is sometimes known as a "lifeboat clause" since it allows combinations between different versions of GPL-licensed software to maintain compatibility.

Usage of the license has steadily declined since the 2010s, particularly due to these complexities, but also a perception it holds back the modern open source landscape from growth and commercialization.[14][15]

History

The original GPL was written by Richard Stallman in 1989, for use with programs released as part of the GNU Project. It was based on a unification of similar licenses used for early versions of GNU Emacs,[16] the GNU Debugger, and the GNU C Compiler.[17] These licenses contained similar provisions to the modern GPL, but were specific to each program, rendering them incompatible, despite being the same license.[18] Stallman's goal was to produce one license that could be used for any project, thus making it possible for many projects to share code.

The second version of the license, GPLv2, was released in 1991. Over the following 15 years, members of the free software community became concerned over problems in the GPLv2 license that could let someone exploit GPL-licensed software in ways contrary to the license's intent.[19] These problems included tivoization (the inclusion of GPL-licensed software in hardware that refuses to run modified versions of its software), compatibility issues similar to those of the AGPL (v1), and patent deals between Microsoft and distributors of free and open-source software, which some viewed as an attempt to use patents as a weapon against the free software community.

Version 3 was developed as an attempt to address these concerns and was officially released on 29 June 2007.[20]

Version 1

Version 1 of the GNU GPL,[21] released on 25 February 1989,[22] was written to protect against the two main methods by which software distributors restricted the freedoms that define free software. The first problem is that distributors might publish only binary files that are executable, but not readable or modifiable by humans. To prevent this, the GPLv1 states that copying and distributing copies of any portion of the program must also make the human-readable source code available under the same licensing terms.[a]

GNU General Public License, version 1

Published	25 February 1989
Website	www.gnu.org/licenses/old-licenses/gpl-1.0.html (https://www.gnu.org/licenses/old-licenses/gpl-1.0.html)
Deprecated	yes

The second problem is that distributors might add restrictions, either to the license or by combining the software with other software that had other restrictions on distribution. The union of two sets of restrictions would apply to the combined work, thus adding unacceptable constrictions. To prevent this, the GPLv1 states that modified versions, as a whole, had to be distributed under the terms of GPLv1.[b] Therefore, software distributed under the terms of the GPLv1 could be combined with software under more permissive terms, as this would not change the terms under which the whole could be distributed.

However, software distributed under GPLv1 could not be combined with software distributed under a more restrictive license, as this would conflict with the requirement that the whole be distributable under the terms of GPLv1.

Version 2

According to Richard Stallman, the major change in version 2 of the GPL was the "Liberty or Death" clause, as he calls it[18] – Section 7. The section says that licensees may distribute a GPL-covered work *only* if they can satisfy all of the license's obligations, despite any other legal obligations they might have. In other words, the obligations of the license may not be severed due to conflicting obligations. This provision is intended to discourage any party from using a patent infringement claim or other litigation to impair users' freedom under the license.[18]

GNU General Public License, version 2

Published	June 1991
Website	www.gnu.org/licenses/old-licenses/gpl-2.0.html (https://www.gnu.org/licenses/old-licenses/gpl-2.0.html)

By 1990, it was becoming apparent that a less restrictive license would be strategically useful for the C library and for software libraries that did the same job of existing proprietary ones.[23] When the GPLv2 was released in June 1991, a second license – the GNU Library General Public License – was introduced at the same time and numbered with version 2 to show that both were complementary.[24] The version numbers diverged in 1999 when version 2.1 of the LGPL was released, which renamed it the GNU Lesser General Public License to reflect its place in the philosophy. The GPLv2 was also modified to refer to the new name of the LGPL, but its version number remained the same, resulting in the original GPLv2 not being recognized by the Software Package Data Exchange (SPDX).[25]

The license includes instructions to specify "version 2 of the License, or (at your option) any later version" to allow the flexible optional use of either version 2 or 3, but some developers change this to specify "version 2" only.

Version 3

In late 2005, the Free Software Foundation (FSF) announced work on version 3 of the GPL. On 16 January 2006, the first "discussion draft" of GPLv3 was published, and the public consultation began. The official GPLv3 was released by the FSF on 29 June 2007. GPLv3 was written by Richard Stallman, with legal counsel from Eben Moglen and Richard Fontana from the Software Freedom Law Center.[26][27]

GNU General Public License, version 3

Published	29 June 2007
Website	www.gnu.org/licenses/gpl-3.0.html (https://www.gnu.org/licenses/gpl-3.0.html)

According to Stallman, the most important changes were in relation to software patents, free software license compatibility, the definition of "source code", and hardware restrictions on software modifications, such as tivoization.[26][28] Other changes related to internationalization, how license violations are handled, and how additional permissions could be granted by the copyright holder. The concept of "software propagation," was explicitly defined as a term for the copying and duplication of software.

The public consultation process was coordinated by the Free Software Foundation with assistance from Software Freedom Law Center, Free Software Foundation Europe,[29] and other free software groups. Comments were collected from the public via the gplv3.fsf.org web portal,[30] using purpose-written software called stet. By the end of the comment period, a total of 2,636 comments had been submitted.[31]

The third draft was released on 28 March 2007.[32] This draft included language intended to prevent patent-related agreements such as the controversial Microsoft-Novell patent agreement, and restricted the anti-tivoization clauses to a legal definition of a "user" and a "consumer product". It also explicitly removed the section on "Geographical Limitations", the probable removal of this section having been announced at the launch of the public consultation.

Richard Stallman at the launch of the first draft of the GNU GPLv3 at MIT, Cambridge, Massachusetts, United States. To his right is Columbia Law Professor Eben Moglen, chairman of the Software Freedom Law Center.

The fourth and final discussion draft[33] was released on 31 May 2007. It introduced Apache License version 2.0 compatibility (prior versions are incompatible), clarified the role of outside contractors, and made an exception to avoid the perceived problems of a Microsoft–Novell style agreement, saying in Section 11 paragraph 6 that:

> You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license ...

This aimed to make such future deals ineffective. The license was also meant to cause Microsoft to extend the patent licenses it granted to Novell customers for the use of GPLv3 software to *all* users of that GPLv3 software; this was possible only if Microsoft was legally a "conveyor" of the GPLv3 software.[34]

Early drafts of GPLv3 also let licensors add an AGPL-like requirement that would have plugged a loophole in the GPL regarding application service providers.[35][36] The freedom to run, study, and share the source code and guarantee copyleft protections is somewhat ambiguous in the context of web services. However, there were concerns expressed about the administrative costs of checking code for the additional requirements in the GPLv3 drafts, and it was decided to keep the GPL and the AGPL license separated.[37]

Others, notably high-profile Linux kernel developers such as Linus Torvalds, Greg Kroah-Hartman, and Andrew Morton, commented to the mass media and made public statements about their objections to parts of the GPLv3 drafts.[38] The kernel developers disapproved of GPLv3 draft clauses regarding DRM/tivoization, patents, and "additional restrictions", and warned of a Balkanisation of the "Open Source Universe".[38][39] Linus Torvalds, who decided not to adopt the GPLv3 for the Linux kernel,[40] reiterated his criticism several years later.[41][42]

GPLv3 improved compatibility with several free software licenses such as the Apache License, version 2.0, and the GNU Affero General Public License, which GPLv2 could not be combined with.[43] However, GPLv3 software could only be combined and share code with GPLv2 software if the GPLv2 license used had the optional "or later" clause and the software was upgraded to GPLv3. While the "GPLv2 or any later version" clause is considered by FSF as the most common form of licensing GPLv2 software,[44] Toybox developer Rob Landley described it as a *lifeboat clause*.[c] Software projects licensed with the optional "or later" clause include Joomla[47] and the GNU Project,[48] while a prominent example without the clause is the Linux kernel.[40][49]

The final version of the license text was published on 29 June 2007.[50]

Terms and conditions

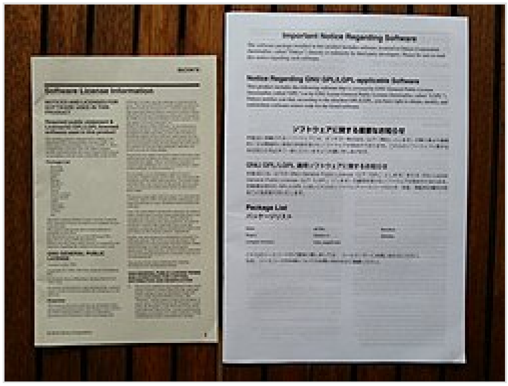
The terms and conditions of the GPL must be made available to anybody receiving a copy of a work that has a GPL license applied to it ("the licensee"). Any licensee who adheres to the terms and conditions is given permission to modify the work, as well as to copy and redistribute the work or any derivative version. The licensee is allowed to charge a fee for this service or do this free of charge. This latter point distinguishes the GPL from software licenses that prohibit commercial redistribution. The FSF argues that free software should not place restrictions on commercial use,[51] and the GPL explicitly states that GPL works may be sold at any price.

The GPL additionally states that a distributor may not impose "further restrictions on the rights granted by the GPL". This forbids activities such as distributing the software under a non-disclosure agreement or contract.

The fourth section of the GPLv2 and the seventh section the GPLv3 require that programs distributed as pre-compiled binaries be accompanied by a copy of the source code, a written offer to distribute the source code via the same mechanism as the pre-compiled binary, or the written offer to obtain the source code that the user got when they received the pre-compiled binary under the GPL. The second section of the GPLv2 and the fifth section of the GPLv3 also require distributing the license along with the program. The GPLv3 allows making the source code available in additional ways in fulfillment of the seventh section. These include downloading source code from an adjacent network server or by peer-to-peer transmission, provided that is how the compiled code was available and there are "clear directions" on where to find the source code.

The FSF does not hold the copyright for a work released under the GPL unless an author explicitly assigns copyrights to the FSF (which seldom happens except for programs that are part of the GNU Project). Only the individual copyright holders have the authority to sue when a license violation is suspected.

Use of licensed software



Printed GPL statements for consumer entertainment devices which incorporate GPL components

Software under the GPL may be run for all purposes, including commercial purposes and even as a tool for creating proprietary software, such as when using GPL-licensed compilers.[52] Users or companies who distribute GPL-licensed works (e.g. software), may charge a fee for copies or give them free of charge. This distinguishes the GPL from shareware software licenses that allow copying for personal use but prohibit commercial distribution or proprietary licenses where copying is prohibited by copyright law. The FSF argues that freedom-respecting free software should also not restrict commercial use and distribution (including redistribution):[51][53]

> In purely private (or internal) use—with no sales and no distribution—the software code may be modified and parts reused without requiring the source code to be released. For sales or distribution, the entire source code needs to be made available to end users, including any code changes and additions—in that case, copyleft is applied to ensure that end users retain the freedoms defined above.

However, software running as an application program under a GPL-licensed operating system such as Linux is not required to be licensed under GPL or to be distributed with source-code availability—the licensing depends only on the used libraries and software components and not on the underlying platform.[54] For example, if a program consists only of original source code, or is combined with source code from other software components,[d] then the custom software components need not be licensed under GPL and need not make their source code available; even if the underlying operating system used is licensed under the GPL, applications running on it are not considered derivative works.[54] Only if GPL-licensed parts are used in a program (and the program is distributed), then all other source code of the program needs to be made available under the same license terms. The GNU *Lesser* General Public License (LGPL) was created to have a weaker copyleft than the GPL, in that it does not require custom-developed source code (distinct from the LGPL-licensed parts) to be made available under the same license terms.

The fifth section of the GPLv3 states that no GPL-licensed code shall be considered an effective "technical protection measure" as defined by Article 11 of the WIPO Copyright Treaty, and that those who convey the work waive all legal power to prohibit circumvention of the technical protection measure "to the extent such circumvention is effected by exercising rights under this License with respect to the covered work". This means that users cannot be held liable for circumventing DRM implemented using GPLv3-licensed code under laws such as the US Digital Millennium Copyright Act (DMCA).[55]

Copyleft

The distribution rights granted by the GPL for modified versions of the work are not unconditional. When someone distributes a GPL-licensed work plus their own modifications, the requirements for distributing the whole work cannot be any greater than the requirements that are in the GPL. This requirement is known as copyleft. It earns its legal power from the use of copyright on software programs.

Because a GPL work is copyrighted, a licensee has no right to redistribute it, not even in modified form (barring fair use), except under the terms of the license. One is only required to adhere to the terms of the GPL if one wishes to exercise rights normally restricted by copyright law, such as redistribution. Conversely, if one distributes copies of the work without abiding by the terms of the GPL (for instance, by keeping the source code secret), they can be sued by the original author under copyright law.

Copyright law has historically been used to prevent distribution of work by parties not authorized by the creator. Copyleft uses the same copyright laws to accomplish a very different goal. It grants rights to distribution to all parties insofar as they provide the same rights to subsequent ones, and they to the next, etc. In this way, the GPL and other copyleft licenses attempt to enforce libre access to the work and all derivatives.[56]

Many distributors of GPL-licensed programs bundle the source code with the executables. An alternative method of satisfying the copyleft is to provide a written offer to provide the source code on a physical medium (such as a CD) upon request. In practice, many GPL-licensed programs are distributed over the Internet, and the source code is made available over FTP or HTTP. For Internet distribution, this complies with the license.

Copyleft applies only when a person seeks to redistribute the program. Developers may make private modified versions with no obligation to divulge the modifications, as long as they do not distribute the modified software to anyone else. Copyleft applies only to the software, and not to its output (unless that output is itself a derivative work of the program).[e] For example, a public web portal running a modified derivative of a GPL-licensed content management system is not required to distribute its changes to the underlying software, because the modified web portal is not being redistributed but rather hosted, and also because the web portal output is also not a derivative work of the GPL-licensed content management system.

There has been debate on whether it is a violation of the GPLv1 to release the source code in obfuscated form, such as in cases in which the author is less willing to make the source code available. The consensus was that while unethical, it was not considered a violation. The issue was clarified when the license was altered with v2 to require that the "preferred" version of the source code be made available.[58]

License versus contract

The GPL was designed as a license, rather than a contract.[59] In some common law jurisdictions, the legal distinction between a license and a contract is an important one: contracts are enforceable by contract law, whereas licenses are enforced under copyright law. However, this distinction is not useful in the many jurisdictions where there are no differences between contracts and licenses, such as civil law systems.[60]

Those who do not accept the GPL's terms and conditions do not have permission, under copyright law, to copy or distribute GPL-licensed software or derivative works. However, if they do not redistribute the GPL-licensed program, they may still use the software within their organization however they like, and works (including programs) constructed by the use of the program are not required to be covered by this license.

Software developer Allison Randal argued that the GPLv3 as a license is unnecessarily confusing for lay readers, and could be simplified while retaining the same conditions and legal force.[61]

In April 2017, a US federal court ruled that an open-source license is an enforceable contract.[62]

In October 2021 Software Freedom Conservancy sued Vizio over breach of contract as an end user to request source code for Vizio's TVs. A federal judge has ruled in the interim that the GPL is an enforceable contract by end users as well as a license for copyright holders.[63]

Derivations

The text of the GPL is itself copyrighted, and the copyright is held by the Free Software Foundation.

The FSF permits people to create new licenses based on the GPL, as long as the derived licenses do not use the GPL preamble without permission. This is discouraged, however, since such a license might be incompatible with the GPL[64] and causes a perceived license proliferation.

Other licenses created by the GNU Project include the GNU Lesser General Public License, GNU Free Documentation License, and GNU Affero General Public License.

The text of the GPL is not itself under the GPL. The license's copyright disallows modification of the license. Copying and distributing the license is allowed since the GPL requires recipients to get "a copy of this License along with the Program."[65] According to the GPL FAQ, anyone can create a new license using a modified version of the GPL as long as they use a different name for the license, do not mention "GNU," and remove the preamble. However, the preamble can be used in a modified license if permission to use it is obtained from the Free Software Foundation (FSF).[66]

Linking and derived works

Libraries

According to the FSF, "The GPL does not require you to release your modified version or any part of it. You are free to make modifications and use them privately, without ever releasing them."[67] However, if one releases a GPL-licensed entity to the public, there is an issue regarding linking: namely, whether a proprietary program that uses a GPL library is in violation of the GPL.

This key dispute is whether non-GPL software can legally statically link or dynamically link to GPL libraries. Different opinions exist on this issue. The GPL is clear in requiring that all derivative works of code under the GPL must themselves be under the GPL. Ambiguity arises with regard to using GPL libraries and bundling GPL software into a larger package (perhaps mixed into a binary via static linking). This is ultimately a question not of the GPL *per se*, but of how copyright law defines derivative works. The following points of view exist:

Point of view: dynamic and static linking violate GPL

The Free Software Foundation (which holds the copyright of several notable GPL-licensed software products and of the license text itself) asserts that an executable that uses a dynamically linked library is indeed a derivative work. This does not, however, apply to separate programs communicating with one another.[68]

The Free Software Foundation also created the LGPL, which is nearly identical to the GPL, but with additional permissions to allow linking for the purposes of "using the library".

Richard Stallman and the FSF specifically encourage library writers to license under the GPL so that proprietary programs cannot use the libraries, in an effort to protect the free software world by giving it more tools than the proprietary world.[69]

Point of view: static linking violates GPL but unclear as of dynamic linking

Some people believe that while static linking produces derivative works, it is not clear whether an executable that dynamically links to a GPL code should be considered a derivative work (see weak copyleft). Linux author Linus Torvalds agrees that dynamic linking can create derived works but disagrees over the circumstances.[70]

A Novell lawyer has written that dynamic linking not being derivative "makes sense" but is not "clear-cut", and that evidence for good-intentioned dynamic linking can be seen by the existence of proprietary Linux kernel drivers.[71]

In *Galoob v. Nintendo*, the United States Ninth Circuit Court of Appeals defined a derivative work as having "'form' or permanence" and noted that "the infringing work must incorporate a portion of the copyrighted work in some form",[72] but there have been no clear court decisions to resolve this particular conflict.

Point of view: linking is irrelevant

According to an article in the *Linux Journal*, Lawrence Rosen (a one-time Open Source Initiative general counsel) argues that the method of linking is mostly irrelevant to the question about whether a piece of software is a derivative work; more important is the question about whether the software was intended to interface with client software or libraries.[73] He states, "The primary indication of whether a new program is a derivative work is whether the source code of the original program was used [in a copy-paste sense], modified, translated or otherwise changed in any way to create the new program. If not, then I would argue that it is not a derivative work,"[73] and lists numerous other points regarding intent, bundling, and linkage mechanism. He further argues on his firm's website[74] that such "market-based" factors are more important than the linking technique.

There is also the specific issue of whether a plugin or module (such as the NVidia or ATI graphics card kernel modules) must also be GPL if it could reasonably be considered its own work. This point of view suggests that reasonably separate plugins, or plugins for software designed to use plugins, could be licensed under an arbitrary license if the work is GPLv2. Of particular interest is the GPLv2 paragraph:

> You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions: ...
>
> b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License. ... These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

The GPLv3 has a different clause:

> You may convey a work based on the Program or the modifications to produce it from the Program, in the form of source code under the terms of Section 4, provided that you also meet all of these conditions: ...
>
> c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable Section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it. ... A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

As a case study, some supposedly proprietary plugins and themes/skins for GPLv2 CMS software such as Drupal and WordPress have come under fire, with both sides of the argument taken.[75]

The FSF differentiates on how the plugin is being invoked. If the plugin is invoked through dynamic linkage and it performs function calls to the GPL program then it is most likely a derivative work.[76]

Communicating and bundling with non-GPL programs

The mere act of communicating with other programs does not, by itself, require all software to be GPL; nor does distributing GPL software with non-GPL software. However, minor conditions must be followed that ensure the rights of GPL software are not restricted. The following is a quote from the gnu.org GPL FAQ, which describes to what extent software is allowed to communicate with and be bundled with GPL programs:[77]

> *What is the difference between an "aggregate" and other kinds of "modified versions"?*
>
> An "aggregate" consists of a number of separate programs, distributed together on the same CD-ROM or other media. The GPL permits you to create and distribute an aggregate, even when the licenses of the other software are non-free or GPL-incompatible. The only condition is that you cannot release the aggregate under a license that prohibits users from exercising rights that each program's individual license would grant them.
>
> Where's the line between two separate programs, and one program with two parts? This is a legal question, which ultimately judges will decide. We believe that a proper criterion depends both on the mechanism of communication (exec, pipes, rpc, function calls within a shared address space, etc.) and the semantics of the communication (what kinds of information are interchanged).
>
> If the modules are included in the same executable file, they are definitely combined in one program. If modules are designed to run linked together in a shared address space, that almost surely means combining them into one program.
>
> By contrast, pipes, sockets, and command-line arguments are communication mechanisms normally used between two separate programs. So when they are used for communication, the modules normally are separate programs. But if the semantics of the communication are intimate enough, exchanging complex internal data structures, that too could be a basis to consider the two parts as combined into a larger program.

The FSF thus draws the line between "library" and "other program" via 1) "complexity" and "intimacy" of information exchange and 2) mechanism (rather than semantics), but resigns that the question is not clear-cut and that in complex situations, case law will decide.

Legal status

The first known violation of the GPL was in 1989, when NeXT extended the GCC compiler to support Objective-C, but did not publicly release the changes.[78] After an inquiry they created a public patch. There was no lawsuit filed for this violation.[79]

In 2002, MySQL AB sued Progress NuSphere for copyright and trademark infringement in US federal court. NuSphere had allegedly violated MySQL's copyright by linking MySQL's GPL-licensed code with NuSphere Gemini table without complying with the license. After a preliminary hearing on 27 February 2002, the parties entered settlement talks and eventually settled.[f] After the hearing, FSF commented that the judge "made clear that she sees the GNU GPL to be an enforceable and binding license."[80]

In August 2003, the SCO Group stated that they believed the GPL to have no legal validity and that they intended to pursue lawsuits over sections of code supposedly copied from SCO Unix into the Linux kernel. This was a problematic stand for them, as they had distributed Linux and other GPL-licensed code in their Caldera OpenLinux distribution, and there is little evidence that they had any legal right to do so except under the terms of the GPL. In February 2018, after a federal circuit court judgment, appeal, and the case being (partially) remanded to the circuit court, the parties restated their remaining claims and provided a plan to move toward final judgement.[81] The remaining claims revolved around Project Monterey and were finally settled in November 2021 by IBM paying $14.25 million to the TSG (previously SCO) bankruptcy trustee.[82]

In April 2004, the netfilter/iptables project was granted a preliminary injunction against Sitecom Germany by Munich District Court after Sitecom refused to desist from distributing Netfilter's GPL-licensed software in violation of the terms of the GPL. Harald Welte of Netfilter was represented by ifrOSS co-founder Till Jaeger. In July 2004, the German court confirmed this injunction as a final ruling against Sitecom.[83] The court's justification was that:

> Defendant has infringed on the copyright of the plaintiff by offering the software 'netfilter/iptables' for download and by advertising its distribution, without adhering to the license conditions of the GPL. Said actions would only be permissible if the defendant had a license grant. ... This is independent of the questions whether the licensing conditions of the GPL have been effectively agreed upon between plaintiff and defendant or not. If the GPL were not agreed upon by the parties, defendant would notwithstanding lack the necessary rights to copy, distribute, and make the software 'netfilter/iptables' publicly available.

This exactly mirrored the predictions given previously by the FSF's Eben Moglen. This ruling was important because it was the first time that a court had confirmed that violating terms of the GPL could be a copyright violation and established jurisprudence as to the enforceability of the GPLv2 under German law.[84]

In May 2005, Daniel Wallace filed suit against the Free Software Foundation in the Southern District of Indiana, contending that the GPL is an illegal attempt to fix prices (at zero). The suit was dismissed in March 2006, on the grounds that Wallace had failed to state a valid antitrust claim; the court noted that "the GPL encourages, rather than discourages, free competition and the distribution of computer operating systems, the benefits of which directly pass to consumers".[85] Wallace was denied the possibility of further amending his complaint, and was ordered to pay the FSF's legal expenses.

On 8 September 2005, the Seoul Central District Court ruled that the GPL was not material to a case dealing with trade secrets derived from GPL-licensed work.[86] Defendants argued that since it is impossible to maintain trade secrets while being compliant with GPL and distributing the work, they are not in breach of trade secrets. This argument was considered without ground.

On 6 September 2006, the gpl-violations.org project prevailed in court litigation against D-Link Germany GmbH regarding D-Link's copyright-infringing use of parts of the Linux kernel in storage devices they distributed.[87] The judgment stated that the GPL is valid, legally binding, and stands in a German court.[88]

In late 2007, BusyBox developers and the Software Freedom Law Center embarked upon a program to gain GPL compliance from distributors of BusyBox in embedded systems, suing those who would not comply. These were claimed to be the first US uses of courts for enforcement of GPL obligations. (See BusyBox GPL lawsuits.)

On 11 December 2008, the Free Software Foundation sued Cisco Systems, Inc. for copyright violations by its Linksys division, of the FSF's GPL-licensed coreutils, readline, Parted, Wget, GNU Compiler Collection, binutils, and GNU Debugger software packages, which Linksys distributes in the Linux firmware[89] of its WRT54G wireless routers, as well as numerous other devices including DSL and Cable modems, Network Attached Storage devices, Voice-Over-IP gateways, virtual private network devices, and a home theater/media player device.[90]

After six years of repeated complaints to Cisco by the FSF, claims by Cisco that they would correct, or were correcting, their compliance problems (not providing complete copies of all source code and their modifications), of repeated new violations being discovered and reported with more products, and lack of action by Linksys (a process described on the FSF blog as a "five-years-running game of Whack-a-Mole"[90]) the FSF took them to court.

Cisco settled the case six months later by agreeing "to appoint a Free Software Director for Linksys" to ensure compliance, "to notify previous recipients of Linksys products containing FSF programs of their rights under the GPL," to make source code of FSF programs freely available on its website, and to make a monetary contribution to the FSF.[91]

In 2011, it was noticed that GNU Emacs had been accidentally releasing some binaries without corresponding source code for two years, contrary to the intended spirit of the GPL, resulting in a copyright violation.[92] Richard Stallman described this incident as a "very bad mistake",[93] which was promptly fixed. The FSF did not sue any downstream redistributors who also unknowingly violated the GPL by distributing these binaries.

In 2017 Artifex, the maker of Ghostscript, sued Hancom, the maker of an office suite that included Ghostscript. Artifex offers two licenses for Ghostscript; one is the AGPL License and the other is a commercial license. Hancom did not acquire a commercial license from Artifex nor did it release its office suite as free software. Artifex sued Hancom in US District Court and made two claims. First, Hancom's use of Ghostscript was a violation of copyright; and second, Hancom's use of Ghostscript was a license violation. The court found the GPL license was an enforceable contract and Hancom was in breach of contract.[94][95]

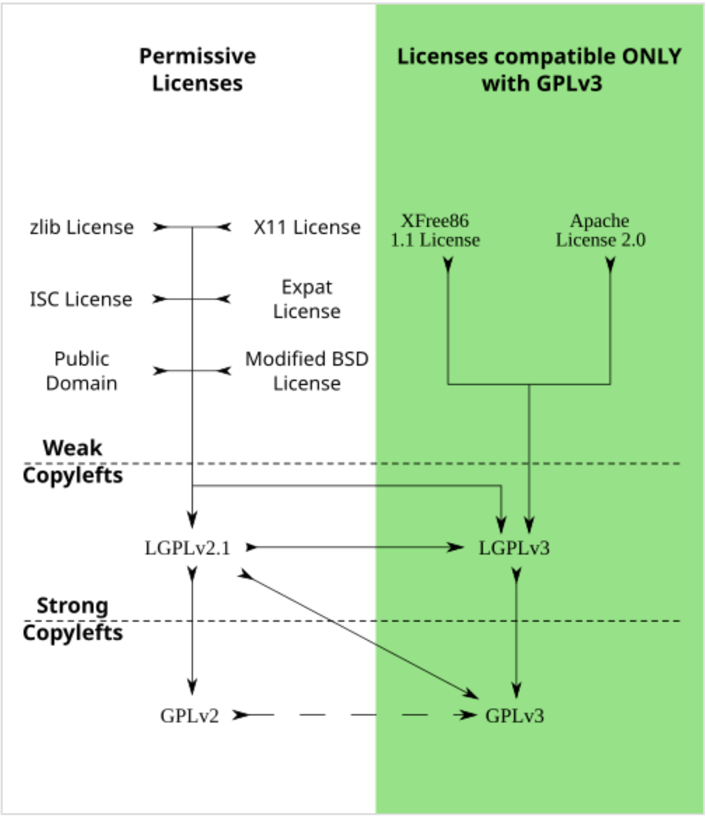
On 20 July 2021, the developers of the open-source Stockfish chess engine sued ChessBase, a creator of chess software, for violating the GPLv3 license.[96] It was claimed that Chessbase had made only slight modifications to the Stockfish code and sold the new engines (Fat Fritz 2 and Houdini 6) to their customers.[97] Additionally, Fat Fritz 2 was marketed as if it was an innovative engine. ChessBase had infringed on the license by not distributing these products as Free Software in accordance with the GPL.

A year later on 7 November 2022, the parties reached an agreement and ended the dispute. In the near future ChessBase will no longer sell products containing Stockfish code, while informing their customers of this fact with an appropriate notice on their web pages. However, one year later, Chessbase's license would be reinstated. Stockfish did not seek damages or financial compensation.[98][99][100]

Compatibility and multi-licensing

Code licensed under several other licenses can be combined with a program under the GPL without conflict, as long as the combination of restrictions on the work as a whole does not put any additional restrictions beyond what GPL allows.[101] In addition to the regular terms of the GPL, there are additional restrictions and permissions one can apply:

1. If a user wants to combine code licensed under different versions of GPL, then this is only allowed if the code with the earlier GPL version includes an "or any later version" statement.[102] For instance, the GPLv3-licensed GNU LibreDWG library cannot be used by LibreCAD and FreeCAD who have GPLv2-only dependencies.[103]
2. Code licensed under LGPL is permitted to be linked with any other code no matter what license that code has,[104] though the LGPL does add additional requirements for the combined work. LGPLv3 and GPLv2-only can thus commonly not be linked, as the combined Code work would add additional LGPLv3 requirements on top of the GPLv2-only licensed software. Code licensed under LGPLv2.x without the "any later version" statement can be relicensed if the whole combined work is licensed to GPLv2 or GPLv3.[105]



Quick guide of license compatibility with GPLv3 according to the FSF. Dashed line indicates that the GPLv2 is only compatible with the GPLv3 with the clause "or any later version".

FSF maintains a list[106] of GPL-compatible free software licenses[107] containing many of the most common free software licenses, such as the original MIT/X license, the BSD license (in its current 3-clause form), and the Artistic License 2.0.[108]

Starting from GPLv3, it is unilaterally compatible for materials (like text and other media) under Creative Commons Attribution-ShareAlike 4.0 International License to be remixed into the GPL-licensed materials (prominently software), not vice versa, for niche use cases like game engine (GPL) with game scripts (CC BY-SA).[109][110]

David A. Wheeler has advocated that free/open source software developers use only GPL-compatible licenses, because doing otherwise makes it difficult for others to participate and contribute code.[111] As a specific example of license incompatibility, Sun Microsystems' ZFS cannot be included in the GPL-licensed Linux kernel, because it is licensed under the GPL-incompatible Common Development and Distribution License. Furthermore, ZFS is protected by patents, so distributing an independently developed GPL-ed implementation would still require Oracle's permission.[112]

A number of businesses use multi-licensing to distribute a GPL version and sell a proprietary license to companies wishing to combine the package with proprietary code, using dynamic linking or not. Examples of such companies include MySQL AB, Digia PLC (Qt framework, before 2011 from Nokia), Red Hat (Cygwin), and Riverbank Computing (PyQt). Other companies, like the Mozilla Foundation (products include Mozilla Application Suite, Mozilla Thunderbird, and Mozilla Firefox), used multi-licensing to distribute versions under the GPL and some other open-source licenses.

Text and other media

It is possible to use the GPL for text documents (or more generally for all kinds of media) if it is clear what constitutes the source code (defined as "the preferred form of the work for making changes in it").[113] For manuals and textbooks, though, the FSF recommends the GNU Free Documentation License (GFDL) instead, which it created for this purpose.[114] Nevertheless, the Debian developers recommended (in a resolution adopted in 2006) to license documentation for their project under the GPL, because of the incompatibility of the GFDL with the GPL (text licensed under the GFDL cannot be incorporated into GPL software).[115][116] Also, the FLOSS Manuals foundation, an organization devoted to creating manuals for free software, decided to eschew the GFDL in favor of the GPL for its texts in 2007.[117]

If the GPL is used for computer fonts, any documents or images made with such fonts might also have to be distributed under the terms of the GPL. This is not the case in countries that recognize typefaces (the appearance of fonts) as being a useful article and thus not eligible for copyright, but font files as copyrighted computer software (which can complicate font embedding, since the document could be considered 'linked' to the font; in other words, embedding a vector font in a document could force it to be released under the GPL, but a rasterized rendering of the font would not be subject to the GPL). The FSF provides an exception for cases where this is not desired.[118]

Adoption

Historically, the GPL license family has been one of the most popular software licenses in the FOSS domain.[7][119][9][10][11][120]

A 1997 survey of MetaLab, then the largest free software archive, showed that the GPL accounted for about half of the software licensed therein.[119] Similarly, a 2000 survey of Red Hat Linux 7.1 found that 53% of the source code was licensed under the GPL.[9] As of 2003, about 68% of all projects and 82.1% of the open source industry certified licensed projects listed on SourceForge.net were from the GPL license family.[121] As of August 2008, the GPL family accounted for 70.9% of the 44,927 free software projects listed on Freecode.[10]

After the release of the GPLv3 in June 2007, adoption of this new GPL version was much discussed[122] and some projects decided against upgrading. For instance the Linux kernel,[40][42] MySQL,[123] BusyBox,[124] AdvFS,[125] Blender,[126][127] VLC media player,[128] and MediaWiki[129] decided against adopting GPLv3. On the other hand, in 2009, two years after the release of GPLv3, Google open-source programs office manager Chris DiBona reported that the number of open-source project licensed software that had moved from GPLv2 to GPLv3 was 50%, counting the projects hosted at Google Code.[11]

In 2011, four years after the release of the GPLv3, 6.5% of all open-source license projects are GPLv3 while 42.5% are GPLv2 according to Black Duck Software data.[130][131] Following in 2011 *451 Group* analyst Matthew Aslett argued in a blog post that copyleft licenses went into decline and permissive licenses increased, based on statistics from Black Duck Software.[132] Similarly, in February 2012 Jon Buys reported that among the top 50 projects on GitHub five projects were under a GPL license, including dual licensed and AGPL projects.[133]

GPL usage statistics from 2009 to 2013 was extracted from Freecode data by Walter van Holst while analyzing license proliferation.[12]

Usage of GPL family licenses in % on Freecode[12]

2009	2010	2011	2012	2013	2014-06-18[134][135]
72%	63%	61%	59%	58%	approx. 54%

In August 2013, according to Black Duck Software, the website's data shows that the GPL license family is used by 54% of open-source projects, with a breakdown of the individual licenses shown in the following table.[120] However, a later study in 2013 showed that software licensed under the GPL license family has increased, and that even the data from Black Duck Software has shown a total increase of software projects licensed under GPL. The study used public information gathered from repositories of the Debian Project, and the study criticized Black Duck Software for not publishing their methodology used in collecting statistics.[136] Daniel German, Professor in the Department of Computer Science at the University of Victoria in Canada, presented a talk in 2013 about the methodological challenges in determining which are the most widely used free software licenses, and showed how he could not replicate the result from Black Duck Software.[137]

In 2015, according to Black Duck, GPLv2 lost its first position to the MIT license and is now second, the GPLv3 dropped to fourth place while the Apache license kept its third position.[7]

Usage of GPL family licenses in the FOSS domain in % according to Black Duck Software

License	2008-05-08[138]	2009-03-11[139]	2011-11-22[130]	2013-08-12[120]	2015-11-19[7]	2016-06-06[140]	2017-01-02[141]	2018-06-04[142]
GPLv2	58.69%	52.2%	42.5%	33%	23%	21%	19%	14%
GPLv3	1.64%	4.15%	6.5%	12%	9%	9%	8%	6%
LGPLv2.1	11.39%	9.84%	?	6%	5%	4%	4%	3%
LGPLv3	? (<0.64%)	0.37%	?	3%	2%	2%	2%	1%
GPL family together	71.72% (+ <0.64%)	66.56%	?	54%	39%	36%	33%	24%

A March 2015 analysis of the GitHub repositories revealed, for the GPL license family, a usage percentage of approximately 25% among licensed projects.[143] In June 2016, an analysis of Fedora Project's packages revealed the GNU GPLv2 or later as the most popular license, and the GNU GPL family as the most popular license family (followed by the MIT, BSD, and GNU LGPL families).[144]

An analysis of whitesourcesoftware.com in April 2018 of the FOSS ecosystem saw the GPLv3 on third place (18%) and the GPLv2 on fourth place (11%), after MIT license (26%) and Apache 2.0 license (21%).[145]

Reception

Legal barrier to application stores

The GPL is incompatible with many application digital distribution systems, like the Mac App Store, and certain other software distribution platforms (on smartphones as well as PCs). The problem lies in the right "to make a copy for your neighbour", as this right is violated by digital rights management systems embedded within the platform to prevent copying of paid software. Even if the application is free in the application store in question, it might result in a violation of that application store's terms.[146]

There is a distinction between an app *store*, which sells DRM-restricted software under proprietary licenses, and the more general concept of digital distribution via some form of online software repository. Virtually all modern Unix systems and Linux distributions have application repositories, including NetBSD, FreeBSD, Ubuntu, Fedora, and Debian. These specific application repositories all contain GPL-licensed software apps, in some cases even when the core project does not permit GPL-licensed code in the base system (for instance OpenBSD[147]). In other cases, such as the Ubuntu App Store, proprietary commercial software applications *and* GPL-licensed applications are both available via the same system; the reason that the Mac App Store (and similar projects) is incompatible with GPL-licensed apps is not inherent in the concept of an app store, but is rather specifically due to Apple's terms-of-use requirement[146] that all apps in the store utilize Apple DRM restrictions. Ubuntu's app store does not demand any such requirement: "These terms do not limit or restrict your rights under any applicable open source software licenses."[148]

Microsoft

In 2001, Microsoft CEO Steve Ballmer referred to Linux as "a cancer that attaches itself in an intellectual property sense to everything it touches".[149][150] In response to Microsoft's attacks on the GPL, several prominent Free Software developers and advocates released a joint statement supporting the license.[151] Microsoft has released Microsoft Windows Services for UNIX, which contains GPL-licensed code. In July 2009, Microsoft itself released a body of around 20,000 lines of Linux driver code under the GPL.[152] The Hyper-V code that is part of the submitted code used open-source components licensed under the GPL and was originally statically linked to proprietary binary parts, the latter being inadmissible in GPL-licensed software.[153]

"Viral" nature

The description of the GPL as "viral", when called 'General Public Virus' or 'GNU Public Virus' (GPV), dates back to a year after the GPLv1 was released.[154]

In 2001, the term received broader public attention when Craig Mundie, Microsoft Senior Vice President, described the GPL as being "viral".[155] Mundie argues that the GPL has a "viral" effect in that it only allows the conveyance of whole programs, which means programs that link to GPL libraries must themselves be under a GPL-compatible license, else they cannot be combined and distributed.

In 2006, Richard Stallman responded in an interview that Mundie's metaphor of a "virus" is wrong as software under the GPL does not "attack" or "infect" other software. Accordingly, Stallman believes that comparing the GPL to a virus is inappropriate, and that a better metaphor for software under the GPL would be a spider plant: if one takes a piece of it and puts it somewhere else, it grows there too.[156]

On the other hand, the concept of a viral nature of the GPL was taken up by others later too.[157][158] For instance, a 2008 article stated: "The GPL license is 'viral,' meaning any derivative work you create containing even the smallest portion of the previously GPL-licensed software must also be licensed under the GPL license."[159]

Barrier to commercialization

The FreeBSD project has stated that "a less publicized and unintended use of the GPL is that it is very favorable to large companies that want to undercut software companies. In other words, the GPL is well suited for use as a marketing weapon, potentially reducing overall economic benefit and contributing to monopolistic behavior" and that the GPL can "present a real problem for those wishing to commercialize and profit from software."[160]

Richard Stallman wrote about the practice of selling license exceptions to free software licenses as an example of ethically acceptable commercialization practice. Selling exceptions here means that the copyright holder of a given software releases it (along with the corresponding source code) to the public under a free software license, "then lets customers pay for permission to use the same code under different terms, for instance allowing its inclusion in proprietary applications". Stallman considered selling exceptions "acceptable since the 1990s, and on occasion I've suggested it to companies. Sometimes this approach has made it possible for important programs to become free software". Although the FSF does not practice selling exceptions, a comparison with the X11 license (which is a non-copyleft free software license) is proposed for suggesting that this commercialization technique should be regarded as ethically acceptable. Releasing a given program under a non-copyleft free software license would permit embedding the code in proprietary software. Stallman comments that "either we have to conclude that it's wrong to release anything under the X11 license—a conclusion I find unacceptably extreme—or reject this implication. Using a non-copyleft license is weak, and usually an inferior choice, but it's not wrong. In other words, selling exceptions permits some embedding in proprietary software, and the X11 license permits even more embedding. If this doesn't make the X11 license unacceptable, it doesn't make selling exceptions unacceptable".[161]

Open-source criticism

In 2000, developer and author Nikolai Bezroukov published an analysis and comprehensive critique of GPL's foundations and Stallman's software development model, called "Labyrinth of Software Freedom".[162][163]

Version 2 of the WTFPL (Do What The Fuck You Want To Public License) was created by Debian project leader Sam Hocevar in 2004 as a parody of the GPL.[164]

In 2005, open source software advocate Eric S. Raymond questioned the relevance of GPL then for the FOSS ecosystem, stating: "We don't need the GPL anymore. It's based on the belief that open source software is weak and needs to be protected. Open source would be succeeding faster if the GPL didn't make lots of people nervous about adopting it."[165] Richard Stallman replied: "GPL is designed to ...

ensure that every user of a program gets the essential freedoms—to run it, to study and change the source code, to redistribute copies, and to publish modified versions ... [Raymond] addresses the issue in terms of different goals and values—those of 'open source,' which do not include defending software users' freedom to share and change software."[166]

In 2007, Allison Randal, who took part in the GPL draft committee, criticized GPLv3 for being incompatible with the GPLv2[167] and for missing clarity in the formulation.[168] Similarly, Whurley prophesied in 2007 the downfall of the GPL due to the lack of focus on developers with GPLv3 which would drive them towards permissive licenses.[169]

In 2009, David Chisnall described in an InformIT article, "The Failure of the GPL", the problems with the GPL such as its incompatibility and complexity of the license text.[170]

In 2014, dtrace developer and Joyent CTO Bryan Cantrill called the copyleft GPL a "Corporate Open Source Anti-pattern" by being "anti-collaborative" and recommended permissive software licenses instead.[171]

GPLv3 criticism

In September 2006, during the draft process of the GPLv3, several high-profile developers of the Linux kernel like Linus Torvalds, Greg Kroah-Hartman, and Andrew Morton, warned of a split in the FOSS community: "the release of GPLv3 portends the Balkanisation of the entire Open Source Universe upon which we rely."[38] Similarly, Benjamin Mako Hill also argued in 2006 during the GPLv3 draft that a united, collaborating community is more important than a single license.[172]

Following the GPLv3 release in 2007, some journalists[42][130][173] and Toybox developer Rob Landley[45][46] criticized that with the introduction of the GPLv3 the split between the open source and free software community became wider than ever because the significantly extended GPLv3 is essentially incompatible with the GPLv2.[102] Compatibility is only given under the optional "or later" clause of the GPL, which was not taken by the Linux kernel, among others.[40] Bruce Byfield noted that before the release of GPLv3, GPLv2 was a unifying element between the open-source and the free software community.[130]

For the LGPLv3, GNU TLS maintainer Nikos Mavrogiannopoulos similarly argued, "If we assume that its [the LGPLv3] primary goal is to be used by free software, then it blatantly fails that",[174] after he re-licensed *GNU TLS* from LGPLv3 back to LGPLv2.1 due to license compatibility issues.[175]

Lawrence Rosen, attorney and computer specialist, praised in 2007 how the community using the Apache license was now able to work together with the GPL community in a compatible manner, as the problems of GPLv2 compatibility with Apache licensed software were resolved with the GPLv3. He said, "I predict that one of the biggest success stories of GPLv3 will be the realization that the entire universe of free and open-source software can thus be combined into comprehensive open source solutions for customers worldwide."[176]

In July 2013, Flask developer Armin Ronacher drew a less optimistic conclusion on the GPL compatibility in the FOSS ecosystem: "When the GPL is involved the complexities of licensing becomes a non fun version of a riddle", also noting that the conflict between Apache License 2.0 and GPLv2 still has impact on the ecosystem.[177]

See also

Free and open-source software portal

- Criticism of copyright
- Multi-licensing
- European Union Public Licence (EUPL)
- GPL font exception
- GPL linking exception
- Comparison of free and open-source software licenses
- Free-software license
- Category:Software using the GNU General Public License
- Public information licence

Notes

a. Sections 3a and 3b of the license
b. Sections 2b and 4 of the license
c. "GPLv3 broke "the" GPL into incompatible forks that can't share code....FSF expected universal compliance, but hijacked lifeboat clause when boat wasn't sinking...."[45][46]
d. example: if *only* GNU *Lesser* General Public License- (LGPL-) libraries, LGPL-software-components and components with permissive free software licenses are used (thus not GPL itself), then *only* the source code of LGPL parts has to be made available—for the developer's own self-developed software components this is not required (even when the underlying operating system used is licensed under GPL, as is the case with Linux).
e. A counterexample is the GPL'ed GNU Bison: the parsers it outputs *do* contain parts of itself and are therefore derivatives, which would fall under the GPL if not for a special exception granted by GNU Bison.[57]
f. See *Progress Software Corporation v. MySQL AB*, 195 F. Supp. 2d 328 (D. Mass. 2002), on defendant's motion for preliminary injunction.

References

1. "License information" (https://www.debian.org/legal/licenses/). *The Debian Project*. Software in the Public Interest (published 12 July 2017). 1997–2017. Archived (https://web.archive.org/web/20170720142826/https://www.debian.org/legal/licenses/) from the original on 20 July 2017. Retrieved 20 July 2017. "... This page presents the opinion of some debian-legal contributors on how certain licenses follow the Debian Free Software Guidelines (DFSG). ... Licenses currently found in Debian main include:

    - ...
    - Expat/MIT-style licenses
    - ...

   "

2. "Various Licenses and Comments about Them" (https://www.gnu.org/licenses/license-list.en.html). *The GNU Project*. Free Software Foundation (published 4 April 2017). 2014–2017. GNU General Public License (GPL) version 3. Archived (https://web.archive.org/web/20170720140022/https://www.gnu.org/licenses/license-list.en.html) from the original on 20 July 2017. Retrieved 20 July 2017. "... This is the latest version of the GNU GPL: a free software license, and a copyleft license. ... GPLv3 is not compatible with GPLv2 by itself. However, most software released under GPLv2 allows you to use the terms of later versions of the GPL as well. When this is the case, you can use the code under GPLv3 to make the desired combination. ..."
3. "Licenses by Name" (https://opensource.org/licenses/alphabetical). *Open Source Initiative*. n.d. Archived (https://web.archive.org/web/20170720141629/https://opensource.org/licenses/alphabetical) from the original on 20 July 2017. Retrieved 20 July 2017. "... The following licenses have been approved by the OSI. ...

    - GNU General Public License version 2 (GPL-2.0)
    - GNU General Public License version 3 (GPL-3.0)
    - ...

    "
4. "Various Licenses and Comments about Them" (https://www.gnu.org/licenses/license-list.en.html). *The GNU Project*. Free Software Foundation (published 4 April 2017). 2014–2017. GNU General Public License (GPL) version 2. Archived (https://web.archive.org/web/20170720140022/https://www.gnu.org/licenses/license-list.en.html) from the original on 20 July 2017. Retrieved 20 July 2017. "... This is the previous version of the GNU GPL: a free software license, and a copyleft license. ... GPLv2 is, by itself, not compatible with GPLv3. However, most software released under GPLv2 allows you to use the terms of later versions of the GPL as well. When this is the case, you can use the code under GPLv3 to make the desired combination. ..."
5. "Copyleft: Pragmatic Idealism – Free Software Foundation" (http://www.fsf.org/licensing/essays/pragmatic.html). Free Software Foundation. Retrieved 10 December 2009.
6. "GPL FAQ: If a library is released under the GPL (not the LGPL)" (https://www.gnu.org/licenses/gpl-faq.html#IfLibraryIsGPL). *GNU Project*. Free Software Foundation.
7. "Top 20 licenses" (https://web.archive.org/web/20160719043600/https://www.blackducksoftware.com/top-open-source-licenses). Black Duck Software. 19 November 2015. Archived from the original (http://www.blackducksoftware.com/resources/data/top-20-licenses) on 19 July 2016. Retrieved 19 November 2015.
8. "GPL FAQ: Does using the GPL for a program make it GNU Software?" (https://www.gnu.org/licenses/gpl-faq.html#DoesUsingTheGPLForAProgramMakeItGNUSoftware). *GNU Project*. Free Software Foundation.
9. David A. Wheeler. "Estimating Linux's Size" (http://www.dwheeler.com/sloc/redhat62-v1/redhat62sloc.html).
10. "Freecode's statistics page" (https://web.archive.org/web/20080828100318/http://www.freecode.com/stats/). Archived from the original (http://www.freecode.com/stats/) on 28 August 2008. "GPL 60.5%, lGPLv2 6.9%, GPLv2 1.9% GPLv3 1.6%"
11. Asay, Matt (23 July 2009). "GPLv3 hits 50 percent adoption | The Open Road – CNET News" (https://web.archive.org/web/20131029190819/http://news.cnet.com/8301-13505_3-10294452-16.html). News.cnet.com. Archived from the original (http://news.cnet.com/8301-13505_3-10294452-16.html) on 29 October 2013. Retrieved 2 September 2013.

12. License proliferation: a naive quantitative analysis (https://lwn.net/Articles/547400/) on lwn.net "Walter van Holst is a legal consultant at the Dutch IT consulting company mitopics... Walter instead chose to use data from a software index, namely Freecode... Walter's 2009 data set consisted of 38,674 projects... The final column in the table shows the number of projects licensed under "any version of the GPL". In addition, Walter presented pie charts that showed the proportion of projects under various common-licenses. Notable in those data sets was that, whereas in 2009 the proportion of projects licensed GPLv2-only and GPLv3 was respectively 3% and 2%, by 2013, those numbers had risen to 7% and 5%."
13. "Why the GPL rocketed Linux to success" (https://web.archive.org/web/20130516005712/http://www.dwheeler.com/blog/2006/09/01/#gpl-bsd). Archived from the original (http://www.dwheeler.com/blog/2006/09/01/#gpl-bsd) on 16 May 2013. Retrieved 22 September 2006. "So while the BSDs have lost energy every time a company gets involved, the GPL'ed programs gain every time a company gets involved."
14. Bacon, Jono. "The decline of GPL? | Opensource.com" (https://opensource.com/article/17/2/decline-gpl#comments). *opensource.com*. Retrieved 21 May 2025.
15. "GNU General Public License (GPL) 2.0—commentary | Legal Guidance | LexisNexis" (https://www.lexisnexis.co.uk/legal/guidance/gnu-general-public-license-gpl-2-0-commentary). *www.lexisnexis.co.uk*. Retrieved 21 May 2025.
16. "GNU Emacs Copying Permission Notice (1985)" (https://github.com/larsbrinkhoff/emacs-16.56/blob/master/etc/COPYING). *GitHub*. Retrieved 8 November 2015.
17. "The History of the GPL" (http://www.free-soft.org/gpl_history/). Retrieved 24 November 2011.
18. Stallman, Richard (21 April 2006). "Presentation at the second international GPLv3 conference, held in Porto Alegre" (https://fsfe.org/activities/gplv3/fisl-rms-transcript.en.html).
19. "Why Upgrade to GPL Version 3 --GPLv3" (http://gplv3.fsf.org/rms-why.html). Fsf.org. Retrieved 17 March 2011.
20. "FSF releases the GNU General Public License, version 3 – Free Software Foundation – working together for free software" (http://www.fsf.org/news/gplv3_launched). Fsf.org. Retrieved 15 January 2011.
21. "GNU General Public License, version 1" (https://www.gnu.org/licenses/old-licenses/gpl-1.0.html).
22. "New General Public License" (https://groups.google.com/group/gnu.announce/msg/bf254a45c6f512f3).
23. For the reasoning see The GNU project (https://www.gnu.org/gnu/thegnuproject.html).
24. "GNU Library General Public License, version 2.0" (https://www.gnu.org/copyleft/library.html). Retrieved 21 November 2018.
25. "SPDX License List – Software Package Data Exchange (SPDX)" (https://spdx.org/licenses/). Retrieved 14 September 2021.
26. Stallman, Richard (25 February 2006). "Presentation in Brussels, Belgium—the first day of that year's FOSDEM conference" (https://archive.today/20120907170559/http://www.ifso.ie/documents/rms-gplv3-2006-02-25.html). Archived from the original (http://www.ifso.ie/documents/rms-gplv3-2006-02-25.html) on 7 September 2012.
27. "GPLv3 authors comment on final draft" (http://www.linux-watch.com/news/NS8853808815.html). Retrieved 4 March 2008.
"The GPLv3 process: Public consultation and private drafting" (https://web.archive.org/web/20080624185521/http://www.linux.com/articles/56554). Archived from the original (http://www.linux.com/articles/56554) on 24 June 2008. Retrieved 4 March 2008.
28. Interview with Richard Stallman (http://www.freesoftwaremagazine.com/articles/interview_with_richard_stallman) Archived (https://web.archive.org/web/20171120024741/http://freesoftwaremagazine.com/articles/interview_with_richard_stallman/) 20 November 2017 at the Wayback Machine, Free Software Magazine, 23 January 2008.

29. "GPLv3: Drafting version 3 of the GNU General Public License" (http://fsfeurope.org/projects/gplv3/). Free Software Foundation Europe.
30. "gplv3.fsf.org comments for discussion draft 4" (https://web.archive.org/web/20081002034950/http://gplv3.fsf.org/comments/gplv3-draft-4.html). Archived from the original (http://gplv3.fsf.org/comments/gplv3-draft-4.html) on 2 October 2008. Retrieved 31 March 2008.
31. "gplv3.fsf.org comments for draft 2" (https://web.archive.org/web/20080724014550/http://gplv3.fsf.org/comments/gplv3-draft-2.html). Archived from the original (http://gplv3.fsf.org/comments/gplv3-draft-2.html) on 24 July 2008. Retrieved 31 March 2008. "Showing comments in file 'gplv3-draft-1' ... found 727"
"gplv3.fsf.org comments for draft 3" (https://web.archive.org/web/20080703072053/http://gplv3.fsf.org/comments/gplv3-draft-3). Archived from the original (http://gplv3.fsf.org/comments/gplv3-draft-3) on 3 July 2008. Retrieved 31 March 2008. "Showing comments in file 'gplv3-draft-3' ... found 649"
"gplv3.fsf.org comments for draft 4" (https://web.archive.org/web/20081002034950/http://gplv3.fsf.org/comments/gplv3-draft-4.html). Archived from the original (http://gplv3.fsf.org/comments/gplv3-draft-4.html) on 2 October 2008. Retrieved 31 March 2008. "Showing comments in file 'gplv3-draft-4' ... found 298"
32. "Guide to the third draft of GPLv3" (http://gplv3.fsf.org/gpl3-dd3-guide).
33. "Final Discussion Draft" (http://gplv3.fsf.org/gpl-draft-2007-05-31.html). Retrieved 4 June 2007.
34. "GPL version 3 FAQ" (http://gplv3.fsf.org/dd3-faq). Retrieved 4 June 2007.
"Fourth Discussion Draft Rationale" (http://gplv3.fsf.org/gpl3-dd4-rationale.pdf) (PDF). Retrieved 4 June 2007.
35. Tiemann, Michael (7 June 2007). "GNU Affero GPL version 3 and the "ASP loophole" " (https://web.archive.org/web/20200814210433/https://opensource.org/node/152). OSI. Archived from the original (http://opensource.org/node/152) on 14 August 2020. Retrieved 19 August 2013.
36. List of free-software licences on the FSF website (http://www.fsf.org/licensing/licenses/index.html): "We recommend that developers consider using the GNU AGPL for any software which will commonly be run over a network."
37. "GPL FAQ: Why did you decide to write the GNU Affero GPLv3 as a separate license?" (https://www.gnu.org/licenses/gpl-faq.html#SeparateAffero). *GNU Project*. Free Software Foundation.
38. James E.J. Bottomley; Mauro Carvalho Chehab; Thomas Gleixner; Christoph Hellwig; Dave Jones; Greg Kroah-Hartman; Tony Luck; Andrew Morton; Trond Myklebust; David Woodhouse (15 September 2006). "Kernel developers' position on GPLv3 – The Dangers and Problems with GPLv3" (https://lwn.net/Articles/200422/). LWN.net. Retrieved 11 March 2015. "The current version (Discussion Draft 2) of GPLv3 on first reading fails the necessity test of section 1 on the grounds that there's no substantial and identified problem with GPLv2 that it is trying to solve. However, a deeper reading reveals several other problems with the current FSF draft: 5.1 DRM Clauses ... 5.2 Additional Restrictions Clause ... 5.3 Patents Provisions ... since the FSF is proposing to shift all of its projects to GPLv3 and apply pressure to every other GPL-licensed project to move, we foresee the release of GPLv3 portends the Balkanisation of the entire Open Source Universe upon which we rely."
39. Petreley, Nicholas (27 September 2006). "A fight against evil or a fight for attention?" (http://www.linuxjournal.com/node/1000100). Linux Journal. Retrieved 11 March 2015. "Second, the war between Linus Torvalds and other Kernel developers and the Free Software Foundation over GPLv3 is continuing, with Torvalds saying he's fed up with the FSF."
40. Torvalds, Linus. "COPYING" (https://git.kernel.org/cgit/linux/kernel/git/stable/linux-stable.git/tree/COPYING). kernel.org. Retrieved 13 August 2013. "[T]he only valid version of the GPL as far as the kernel is concerned is _this_ particular version of the license (ie v2, not v2.2 or v3.x or whatever), unless explicitly otherwise stated."

41. Linus Torvalds says GPL v3 violates everything that GPLv2 stood for (https://www.youtube.com/watch?v=PaKIZ7gJlRU) Debconf 2014, Portland, Oregon (accessed 11 March 2015)
42. Kerner, Sean Michael (8 January 2008). "Torvalds Still Keen On GPLv2" (http://www.internetnews.com/dev-news/article.php/3720371/Torvalds+Still+Keen+On+GPLv2.htm). *Internet News*. internetnews.com. Retrieved 12 February 2015. "In some ways, Linux was the project that really made the split clear between what the FSF is pushing which is very different from what open source and Linux has always been about, which is more of a technical superiority instead of a – this religious belief in freedom," Torvalds told Zemlin. So, the GPL Version 3 reflects the FSF's goals and the GPL Version 2 pretty closely matches what I think a license should do and so right now, Version 2 is where the kernel is."
43. "GPL 3 Overview" (http://www.techlawforum.net/post.cfm/gpl-3-overview). Tech LawForum. 29 June 2007. Retrieved 2 September 2013.
44. "A Quick Guide to GPLv3" (https://www.gnu.org/licenses/quick-guide-gplv3.html#new-compatible-licenses). *GNU Software Project*. Free Software Foundation.
45. Landley, Rob (13 March 2013). "Embedded Linux Conference 2013 – Toybox: Writing a New Command Line" (https://ghostarchive.org/varchive/youtube/20211027/SGmtP5Lg_t0). The Linux Foundation. Archived from the original (https://www.youtube.com/watch?v=SGmtP5Lg_t0&t=15m10s) (video) on 27 October 2021. Retrieved 24 June 2016.
46. Landley, Rob. "Transcript of CELF 2013 Toybox talk" (http://landley.net/talks/celf-2013.txt). landley.net. Retrieved 21 August 2013.
47. "Licenses | ER Software Design" (https://www.ersd.net/licenses). *www.ersd.net*. Retrieved 21 May 2025.
48. "Various Licenses and Comments about Them - GNU Project - Free Software Foundation" (https://www.gnu.org/licenses/license-list.en.html). *www.gnu.org*. Retrieved 21 May 2025.
49. Linus Torvalds (8 September 2000). "Linux-2.4.0-test8" (http://lkml.iu.edu/hypermail/linux/kernel/0009.1/0096.html). lkml.iu.edu. Retrieved 21 November 2015. "The only one of any note that I'd like to point out directly is the clarification in the COPYING file, making it clear that it's only _that_particular version of the GPL that is valid for the kernel. This should not come as any surprise, as that's the same license that has been there since 0.12 or so, but I thought I'd make that explicit"
50. "GNU General Public License" (https://www.gnu.org/licenses/gpl.html). 29 June 2007. Retrieved 15 June 2012.
51. "Selling Free Software" (https://www.gnu.org/philosophy/selling.html). *GNU Project*. Free Software Foundation. Archived (https://web.archive.org/web/20231107142639/https://www.gnu.org/philosophy/selling.html) from the original on 7 November 2023.
52. "Frequently Asked Questions about the GNU Licenses" (https://www.gnu.org/licenses/gpl-faq.html#CanIUseGPLToolsForNF). *GNU Project*. Free Software Foundation. Can I use GPL-covered editors such as GNU Emacs to develop nonfree programs? Can I use GPL-covered tools such as GCC to compile them?. Archived (https://web.archive.org/web/20231117151759/http://www.gnu.org/licenses/gpl-faq.html#CanIUseGPLToolsForNF) from the original on 17 November 2023.
53. "Frequently Asked Questions about the GNU Licenses" (https://www.gnu.org/licenses/gpl-faq.html#GPLRequireSourcePostedPublic). *GNU Project*. Free Software Foundation. Does the GPL require that source code of modified versions be posted to the public?. Archived (https://web.archive.org/web/20231117151759/http://www.gnu.org/licenses/gpl-faq.html#GPLRequireSourcePostedPublic) from the original on 17 November 2023.
"GPL FAQ: Unreleased modifications" (https://www.gnu.org/licenses/gpl-faq.html#UnreleasedMods). *GNU Project*. Free Software Foundation.
"GPL FAQ: Internal Distribution" (https://www.gnu.org/licenses/gpl-faq.html#InternalDistribution). *GNU Project*. Free Software Foundation.
54. "GPL FAQ: Port program to GNU/Linux" (https://www.gnu.org/licenses/gpl-faq.html#PortProgramToGPL). *GNU Project*. Free Software Foundation.

55. "A Quick Guide to GPLv3 – GNU Project – Free Software Foundation (FSF)" (https://www.gnu.org/licenses/quick-guide-gplv3.html#neutralizing-laws-that-prohibit-free-software-but-not-forbidding-drm). Free Software Foundation.
56. De Laat, Paul B. (December 2005). "Copyright or copyleft?". *Research Policy*. **34** (10): 1511–1532. doi:10.1016/j.respol.2005.07.003 (https://doi.org/10.1016%2Fj.respol.2005.07.003).
57. "Conditions for Using Bison" (https://www.gnu.org/software/bison/manual/html_node/Conditions.html). *GNU Project*. Free Software Foundation. Retrieved 11 December 2008.
58. "Reasoning behind the "preferred form" language in the GPL" (https://lwn.net/Articles/431651/). LWN.net. 7 March 2011.
59. Stallman, Richard M. (9 June 2006). "Don't Let 'Intellectual Property' Twist Your Ethos" (https://www.gnu.org/philosophy/no-ip-ethos.html).
Moglen, Eben (22 June 2006). "Will the "Not a contract" phrase go?". *Transcript* (http://fsfeurope.org/projects/gplv3/barcelona-moglen-transcript.en.html#q7-a-contract). 3rd international GPLv3 conference (https://fsfe.org/campaigns/gplv3/europe-gplv3-conference.en.html). Barcelona.
60. Guadamuz-Gonzalez, Andres (2004). "Viral contracts or unenforceable documents? Contractual validity of copyleft licenses". *European Intellectual Property Review*. **26** (8): 331–339. SSRN 569101 (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=569101).
61. Allison Randal (14 May 2007). "GPLv3, Clarity and Simplicity" (https://web.archive.org/web/20081015194558/http://radar.oreilly.com/archives/2007/05/gplv3_clarity_a.html). Archived from the original (http://radar.oreilly.com/archives/2007/05/gplv3_clarity_a.html) on 15 October 2008.
62. Keith Collins (11 May 2017). "A federal court has ruled that an open-source license is an enforceable contract" (https://qz.com/981029/a-federal-court-has-ruled-that-an-open-source-license-is-an-enforceable-contract/).
63. "SFC v. Vizio remanded back to California state courts [LWN.net]" (https://lwn.net/Articles/895405/). *lwn.net*. Retrieved 10 June 2022.
64. "GPL FAQ: Can I modify the GPL and make a modified license?" (http://www.fsf.org/licensing/licenses/gpl-faq.html#ModifyGPL). *GNU Project*. Free Software Foundation.
65. "The GNU General Public License Version 3" (https://www.gnu.org/copyleft/gpl.html). Free Software Foundation. 29 June 2007. Retrieved 21 July 2009.
66. "GPL FAQ: Can I modify the GPL and make a modified license?" (https://www.gnu.org/licenses/gpl-faq.en.html#ModifyGPL). *GNU Project*. Free Software Foundation. 9 June 2017. Retrieved 11 July 2017.
67. "GPL FAQ: Does the GPL require that source code of modified versions be posted to the public?" (https://www.gnu.org/licenses/gpl-faq.html#GPLRequireSourcePostedPublic). *GNU Project*. Free Software Foundation.
68. "Frequently Asked Questions about the GNU Licenses (GPL FAQ)" (https://www.gnu.org/licenses/gpl-faq.html). *GNU Project*. Free Software Foundation. Retrieved 15 March 2011.
69. "Why you shouldn't use the Lesser GPL for your next library" (https://www.gnu.org/philosophy/why-not-lgpl.html). *GNU Project*. Free Software Foundation. Retrieved 15 January 2011.
70. Linus Torvalds, GPL only modules (https://lkml.org/lkml/2006/12/17/79), *linux-kernel* mailing list (17 December 2006).
71. Matt Asay, The GPL: Understanding the License that Governs Linux (http://www.novell.com/coolsolutions/feature/1532.html) Archived (https://web.archive.org/web/20081006073434/http://www.novell.com/coolsolutions/feature/1532.html) 6 October 2008 at the Wayback Machine, *Novell Cool Solutions Feature* (16 Jan 2004).
72. *Lewis Galoob Toys, Inc. v. Nintendo of America, Inc.*, 964 F.2d 965 (http://law.justia.com/cases/federal/appellate-courts/F2/964/965/341457/), ¶10 (9th Cir. 21 May 1992).

73. Lawrence Rosen, "Derivative Works (http://www.linuxjournal.com/article/6366)", *Linux Journal* (1 January 2003).
74. "Derivative Works" (http://www.rosenlaw.com/lj19.htm). *Rosenlaw & Einschlag*. 25 May 2004.
75. Matt (29 January 2009). "Why They're Wrong: WordPress Plugins Shouldn't Have to be GPL" (http://www.webmaster-source.com/2009/01/29/why-theyre-wrong-wordpress-plugins-shouldnt-have-to-be-gpl/). Webmaster-Source. Retrieved 15 January 2011. "Licensing" (https://drupal.org/licensing/faq#q7). Drupal.org. Retrieved 15 January 2011.
76. "GPL FAQ: Can I apply the GPL when writing a plug-in for a nonfree program?" (https://www.gnu.org/licenses/gpl-faq.html#GPLPluginsInNF). *GNU Project*. Free Software Foundation. Retrieved 15 January 2011.
77. MereAggregation (https://www.gnu.org/licenses/old-licenses/gpl-2.0-faq.html#MereAggregation) "What constitutes combining two parts into one program? This is a legal question, which ultimately judges will decide. We believe that a proper criterion depends both on the mechanism of communication (exec, pipes, rpc, function calls within a shared address space, etc.) and the semantics of the communication (what kinds of information are interchanged)." on gnu.org
78. 12 Years of GPL Compliance: A Historical Perspective (http://ebb.org/bkuhn/talks/LinuxTag-2011/compliance.html), by Bradley Kuhn Slide 10
79. Common Lisp, Readline and GPL (http://clisp.cvs.sourceforge.net/viewvc/clisp/clisp/doc/Why-CLISP-is-under-GPL), Mail on 6 November, 21:31
80. "Judge Saris defers GNU GPL Questions for Trial in MySQL vs. Progress Software" (https://www.gnu.org/press/2002-03-01-pi-MySQL.html). gnu.org. Retrieved 24 March 2011.
81. "JOINT STATUS REPORT, Case 2:03-cv-00294-DN, Document 1179" (https://web.archive.org/web/20190126061200/https://sco-vs-ibm.org/review/acrobat/180216.pdf) (PDF). IN THE UNITED STATES DISTRICT COURT FOR THE DISTRICT OF UTAH, CENTRAL DIVISION. 16 February 2018. Archived from the original on 26 January 2019. Retrieved 25 January 2019.
82. Vaughan-Nichols, Steven J. "Last of original SCO v IBM Linux lawsuit settled" (https://www.zdnet.com/article/last-of-original-sco-v-ibm-linux-lawsuit-settled/). *ZDNet*. Retrieved 4 December 2021.
83. "Groklaw – The German GPL Order – Translated" (http://www.groklaw.net/article.php?story=20040725150736471). *groklaw.net*.
84. Bird & Bird, A Review of German Case Law on the GNU General Public License (http://www.twobirds.com/English/News/Articles/Pages/2007/Review_German_GNU_General_Public_License.aspx) Archived (https://web.archive.org/web/20130517081201/http://www.twobirds.com/English/News/Articles/Pages/2007/Review_German_GNU_General_Public_License.aspx) 17 May 2013 at the Wayback Machine, 17 December 2007, retrieved 1 March 2012
85. Dismissal (http://www.groklaw.net/pdf/WallaceFSFGrantingDismiss.pdf) Archived (https://web.archive.org/web/20200116194140/http://www.groklaw.net/pdf/WallaceFSFGrantingDismiss.pdf) 16 January 2020 at the Wayback Machine of Wallace v. FSF. From this article (http://www.groklaw.net/article.php?story=20060320201540127) Archived (https://web.archive.org/web/20100505060626/http://www.groklaw.net/article.php?story=20060320201540127) 5 May 2010 at the Wayback Machine on Groklaw.
86. 판결문 – GNU 프로젝트 – 자유 소프트웨어 재단(FSF) (https://web.archive.org/web/20071018211119/http://korea.gnu.org/gv/sentence.html). *gnu.org*. Archived from the original (http://korea.gnu.org/gv/sentence.html) on 18 October 2007.
87. Welte, Harald (22 September 2006). "gpl-violations.org project prevails in court case on GPL violation by" (http://gpl-violations.org/news/20060922-dlink-judgement_frankfurt/). *gpl-violations.org*.

88. "D-Link Judgement" (https://web.archive.org/web/20130426055334/http://www.jbb.de/judgment_dc_frankfurt_gpl.pdf) (PDF). Archived from the original (http://www.jbb.de/fileadmin/download/urteil_lg_muenchen_gpl.pdf) (PDF) on 26 April 2013.
89. Ewing, James (1 August 2004). "Linux on Linksys Wi-Fi Routers" (http://www.linuxjournal.com/article/7322). Linux Journal. Retrieved 23 January 2012.
90. "Free Software Foundation Files Suit Against Cisco For GPL Violations" (https://www.fsf.org/news/2008-12-cisco-suit) (Press release). Free Software Foundation. 11 December 2008. Retrieved 22 August 2011.
91. "FSF Settles Suit Against Cisco" (https://www.fsf.org/news/2009-05-cisco-settlement.html) (Press release). Free Software Foundation. 20 May 2009. Retrieved 22 August 2011.
92. Brockmeier, Joe (29 July 2011). "Say what? GNU Emacs violates the GPL" (https://www.networkworld.com/article/737764/opensource-subnet-say-what-gnu-emacs-violates-the-gpl.html). Network World. Retrieved 19 January 2016.
Naughton, Edward J. (8 August 2011). "License revoked: Applying Section 4 of the GPL and the lessons of Best Buy to Google's Android" (https://web.archive.org/web/20160127150256/http://brownrudnick.com/blog/emerging-technologies/license-revoked-applying-section-4-of-the-gpl-and-the-lessons-of-best-buy-to-googles-android/). *brownrudnick.com*. Archived from the original (http://brownrudnick.com/blog/emerging-technologies/license-revoked-applying-section-4-of-the-gpl-and-the-lessons-of-best-buy-to-googles-android/) on 27 January 2016.
Soulskill (29 July 2011). "Emacs Has Been Violating the GPL Since 2009" (http://developers.slashdot.org/story/11/07/29/1445252/Emacs-Has-Been-Violating-the-GPL-Since-2009). *slashdot.org*.
93. Stallman, Richard (28 July 2011). "Re: Compiled files without sources????" (http://lists.gnu.org/archive/html/emacs-devel/2011-07/msg01155.html). *lists.gnu.org*.
94. "A federal court has ruled that the GPL is an enforceable contract" (https://lwn.net/Articles/722791/). LWN.net. 15 May 2017. Retrieved 17 May 2021.
95. Robertson, Donald (11 October 2017). "Update on Artifex v. Hancom GNU GPL compliance case" (https://www.fsf.org/blogs/licensing/update-on-artifex-v-hancom-gnu-gpl-compliance-case-1). Free Software Foundation. Retrieved 17 May 2021.
96. "Our lawsuit against ChessBase – Stockfish – Open Source Chess Engine" (https://stockfishchess.org/blog/2021/our-lawsuit-against-chessbase/). *stockfishchess.org*. Retrieved 22 November 2022.
97. "Statement on Fat Fritz 2 – Stockfish – Open Source Chess Engine" (https://stockfishchess.org/blog/2021/statement-on-fat-fritz-2/). *stockfishchess.org*. Retrieved 22 November 2022.
98. "ChessBase GmbH and the Stockfish team reach an agreement and end their legal dispute – Stockfish – Open Source Chess Engine" (https://stockfishchess.org/blog/2022/chessbase-stockfish-agreement/). *stockfishchess.org*. Retrieved 22 November 2022.
99. "Settlement (German)" (https://stockfishchess.org/files/settlement_german.pdf) (PDF). 7 November 2022. Retrieved 22 November 2022.
100. "Settlement (English)" (https://stockfishchess.org/files/settlement_english.pdf) (PDF). 7 November 2022. Retrieved 22 November 2022.
101. "The GNU General Public License v3.0" (https://www.gnu.org/licenses/gpl.html). *GNU Project*. Free Software Foundation. Retrieved 24 March 2010.
102. "GPL FAQ: Is GPLv3 compatible with GPLv2?" (https://www.gnu.org/licenses/gpl-faq.html#v2v3Compatibility). *GNU Project*. Free Software Foundation. Retrieved 3 June 2014. "No. Some of the requirements in GPLv3, such as the requirement to provide Installation Information, do not exist in GPLv2. As a result, the licenses are not compatible: if you tried to combine code released under both these licenses, you would violate section 6 of GPLv2. However, if code is released under GPL "version 2 or later," that is compatible with GPLv3 because GPLv3 is one of the options it permits."

103. Larabel, Michael (24 January 2013). "FSF Wastes Away Another "High Priority" Project" (https://web.archive.org/web/20161109200145/https://www.phoronix.com/scan.php?page=news_item&px=MTI4Mjc). Phoronix. Archived from the original (https://www.phoronix.com/scan.php?page=news_item&px=MTI4Mjc) on 9 November 2016. Retrieved 22 August 2013. "Both LibreCAD and FreeCAD both want to use LibreDWG and have patches available for supporting the DWG file format library, but can't integrate them. The programs have dependencies on the popular GPLv2 license while the Free Software Foundation will only let LibreDWG be licensed for GPLv3 use, not GPLv2."
Prokoudine, Alexandre (27 December 2012). "LibreDWG drama: the end or the new beginning?" (https://librearts.org/2012/12/libredwg-drama-the-end-or-the-new-beginning/). librearts.org. Retrieved 4 February 2025. "... the unfortunate situation with support for DWG files in free CAD software via LibreDWG. We feel, by now it ought to be closed. We have the final answer from FSF. ... "We are not going to change the license." "
104. "GNU Lesser General Public License v2.1 – GNU Project – Free Software Foundation (FSF)" (https://www.gnu.org/licenses/old-licenses/lgpl-2.1.html#SEC2). fsf.org. Retrieved 26 April 2011.
105. "GPL FAQ: How are the various GNU licenses compatible with each other?" (https://www.gnu.org/licenses/gpl-faq.html#AllCompatibility). *GNU Project*. Free Software Foundation. Retrieved 13 April 2011.
106. "Various licenses with comments – GPL-Compatible Free Software Licenses" (https://www.gnu.org/licenses/license-list.html#GPLCompatibleLicenses). FSF. Retrieved 20 April 2012.
107. "GPL FAQ: What does it mean to say that two licenses are "compatible"?" (https://www.gnu.org/licenses/gpl-faq.html#WhatIsCompatible). *GNU Project*. Free Software Foundation. Retrieved 14 April 2011.
"GPL FAQ: What does it mean to say a license is "compatible with the GPL?" " (https://www.gnu.org/licenses/gpl-faq.html#WhatDoesCompatMean). *GNU Project*. Free Software Foundation. Retrieved 14 April 2011.
108. "Black Duck Open Source Resource Center" (https://web.archive.org/web/20121014125844/http://www.blackducksoftware.com/oss/licenses/). blackducksoftware.com. Archived from the original (http://www.blackducksoftware.com/oss/licenses/) on 14 October 2012. Retrieved 26 April 2011.
109. "Compatible Licenses" (https://creativecommons.org/share-your-work/licensing-considerations/compatible-licenses/).
110. "ShareAlike compatibility: GPLv3 – Creative Commons" (https://wiki.creativecommons.org/wiki/ShareAlike_compatibility:_GPLv3#Considerations_for_adapters_applying_the_GPLv3). *wiki.creativecommons.org*.
111. "Make Your Open Source Software GPL-Compatible. Or Else" (http://www.dwheeler.com/essays/gpl-compatible.html). David A. Wheeler. Retrieved 26 April 2011.
112. "Linux: ZFS, Licenses and Patents" (https://web.archive.org/web/20110612175303/http://kerneltrap.org/node/8066). *Tag1 Consulting*. Archived from the original (https://www.tag1consulting.com/blog/interview-linus-torvalds-linux-and-git) on 12 June 2011.
113. "GPL FAQ: Can I use the GPL for something other than software?" (https://www.gnu.org/licenses/gpl-faq.html#GPLOtherThanSoftware). *GNU Project*. Free Software Foundation. Retrieved 20 June 2009.
114. "GPL FAQ: Why don't you use the GPL for manuals?" (https://www.gnu.org/licenses/gpl-faq.html#WhyNotGPLForManuals). *GNU Project*. Free Software Foundation. Retrieved 20 June 2009.

115. Srivastava, Manoj (2006). "Draft Debian Position Statement about the GNU Free Documentation License (GFDL)" (http://people.debian.org/~srivasta/Position_Statement.xhtml). Retrieved 25 September 2007. "It is not possible to borrow text from a GFDL'd manual and incorporate it in any free software program whatsoever. This is not a mere license incompatibility. It's not just that the GFDL is incompatible with this or that free software license: it's that it is fundamentally incompatible with any free software license whatsoever. So if you write a new program, and you have no commitments at all about what license you want to use, saving only that it be a free license, you cannot include GFDL'd text. The GNU FDL, as it stands today, does not meet the Debian Free Software Guidelines. There are significant problems with the license, as detailed above; and, as such, we cannot accept works licensed under the GNU FDL into our distribution."
116. Debian Project: Resolution: Why the GNU Free Documentation License is not suitable for Debian (http://www.debian.org/vote/2006/vote_001.en.html#amendmenttexta). Voted February–March 2006. Retrieved 20 June 2009.
117. "License Change" (https://archive.today/20080228211105/http://en.flossmanuals.net/bin/view/Blog/LicenseChange). *FLOSS Manuals Foundation*. 6 June 2007. Archived from the original (http://en.flossmanuals.net/bin/view/Blog/LicenseChange) on 28 February 2008. Retrieved 20 June 2009.
118. "Font Licensing" (http://www.fsf.org/blogs/licensing/20050425novalis). *Free Software Foundation*. 25 April 2005.
"GPL FAQ: How does the GPL apply to fonts?" (https://www.gnu.org/licenses/gpl-faq.html#FontException). *GNU Project*. Free Software Foundation.
119. David A. Wheeler. "Make Your Open Source Software GPL-Compatible. Or Else" (http://www.dwheeler.com/essays/gpl-compatible.html).
120. "Top 20 licenses" (https://web.archive.org/web/20160719043600/https://www.blackducksoftware.com/top-open-source-licenses). Black Duck Software. 23 August 2013. Archived from the original (http://www.blackducksoftware.com/resources/data/top-20-licenses) on 19 July 2016. Retrieved 23 August 2013.
121. "SourceForge.net: Software Map" (https://web.archive.org/web/20170213162500/https://www.dwheeler.com/frozen/sourceforge-stats-20031110.html). Dwheeler.com. Archived from the original (http://www.dwheeler.com/frozen/sourceforge-stats-20031110.html) on 13 February 2017. Retrieved 17 November 2008. " "License -> OSI: ... GNU General Public License (GPL) (32641 projects), GNU Library or Lesser General Public License (LGPL) (4889 projects" (of 45727, 82.1%)"
122. Mark (8 May 2008). "The Curse of Open Source License Proliferation" (https://web.archive.org/web/20151208112000/http://socializedsoftware.com/2008/05/08/the-curse-of-open-source-license-proliferation/). socializedsoftware.com. Archived from the original (http://socializedsoftware.com/2008/05/08/the-curse-of-open-source-license-proliferation/) on 8 December 2015. Retrieved 30 November 2015. "Currently the decision to move from GPL v2 to GPL v3 is being hotly debated by many open source projects. According to Palamida, a provider of IP compliance software, there have been roughly 2489 open source projects that have moved from GPL v2 to later versions."
123. "MySQL changes license to avoid GPLv3" (https://web.archive.org/web/20070206060251/http://www.businessreviewonline.com/os/archives/2007/01/mysql_changes_l.html). *Computer business review online*. 4 January 2007. Archived from the original (http://www.businessreviewonline.com/os/archives/2007/01/mysql_changes_l.html) on 6 February 2007.

124. corbet (1 October 2006). "Busy busy busybox" (https://lwn.net/Articles/202106/). lwn.net. Retrieved 21 November 2015. "Since BusyBox can be found in so many embedded systems, it finds itself at the core of the GPLv3 anti-DRM debate. ... The real outcomes, however, are this: BusyBox will be GPLv2 only starting with the next release. It is generally accepted that stripping out the "or any later version" is legally defensible, and that the merging of other GPLv2-only code will force that issue in any case"
Landley, Rob (9 September 2006). "Re: Move GPLv2 vs v3 fun..." (https://lwn.net/Articles/202110/) lwn.net. Retrieved 21 November 2015. "Don't invent a straw man argument please. I consider licensing BusyBox under GPLv3 to be useless, unnecessary, overcomplicated, and confusing, and in addition to that it has actual downsides. 1) Useless: We're never dropping GPLv2."
125. "HP Press Release: HP Contributes Source Code to Open Source Community to Advance Adoption of Linux" (http://www.hp.com/hpinfo/newsroom/press/2008/080623a.html). *www.hp.com*.
126. Prokoudine, Alexandre (26 January 2012). "What's up with DWG adoption in free software?" (https://librearts.org/2012/01/whats-up-with-dwg-adoption-in-free-software/). librearts.org. Retrieved 4 February 2025. "[Blender's Toni Roosendaal:] "Blender is also still 'GPLv2 or later'. For the time being we stick to that, moving to GPL 3 has no evident benefits I know of." "
127. "License – blender.org" (https://www.blender.org/about/license/). Retrieved 17 December 2016. "The source code we develop at blender.org is default being licensed as GNU GPL Version 2 or later. "
128. Denis-Courmont, Rémi. "VLC media player to remain under GNU GPL version 2" (http://www.videolan.org/press/2007-1.html). videolan.org. Retrieved 21 November 2015. "In 2001, VLC was released under the OSI-approved GNU General Public version 2, with the commonly-offered option to use 'any later version' thereof (though there was not any such later version at the time). Following the release by the Free Software Foundation (FSF) of the new version 3 of its GNU General Public License (GPL) on the 29th of June 2007, contributors to the VLC media player, and other software projects hosted at videolan.org, debated the possibility of updating the licensing terms for future version of the VLC media player and other hosted projects, to version 3 of the GPL. ... There is strong concern that these new additional requirements might not match the industrial and economic reality of our time, especially in the market of consumer electronics. It is our belief that changing our licensing terms to GPL version 3 would currently not be in the best interest of our community as a whole. Consequently, we plan to keep distributing future versions of VLC media player under the terms of the GPL version 2."
129. "Copyright" (https://www.mediawiki.org/wiki/Copyright). *MediaWiki*.
130. Byfield, Bruce (22 November 2011). "7 Reasons Why Free Software Is Losing Influence: Page 2" (http://www.datamation.com/open-source/7-reasons-why-free-software-is-losing-influence_2.html). Datamation.com. Retrieved 23 August 2013. "At the time, the decision seemed sensible in the face of a deadlock. But now, GPLv2 is used for 42.5% of free software, and GPLv3 for less than 6.5%, according to Black Duck Software."
131. GPL, copyleft use declining faster than ever (http://www.itworld.com/article/2832379/it-management/gpl--copyleft-use-declining-faster-than-ever.html) Archived (https://web.archive.org/web/20170904082743/https://www.itworld.com/article/2832379/it-management/gpl--copyleft-use-declining-faster-than-ever.html) 4 September 2017 at the Wayback Machine on ITworld on 16 December 2011 by Brian Proffitt

132. Proffitt, Brian (16 December 2011). "GPL, copyleft use declining faster than ever – Data suggests a sharper rate of decline, which raises the question: why?" (https://web.archive.org/web/20131203213915/http://www.itworld.com/it-managementstrategy/233753/gpl-copyleft-use-declining-faster-ever). IT world. Archived from the original (http://www.itworld.com/it-managementstrategy/233753/gpl-copyleft-use-declining-faster-ever) on 3 December 2013. Retrieved 23 August 2013.
Aslett, Matthew (15 December 2011). "On the continuing decline of the GPL" (https://web.archive.org/web/20161209212934/https://blogs.the451group.com/opensource/2011/12/15/on-the-continuing-decline-of-the-gpl/). the451group.com. Archived from the original (https://blogs.the451group.com/opensource/2011/12/15/on-the-continuing-decline-of-the-gpl/) on 9 December 2016. Retrieved 26 November 2015.
133. The Top Licenses on Github (http://ostatic.com/blog/the-top-licenses-on-github) Archived (https://web.archive.org/web/20160304221127/http://ostatic.com/blog/the-top-licenses-on-github) 4 March 2016 at the Wayback Machine on ostatic.com by Jon Buys (7 February 2012)
134. tags letter g (http://freecode.com/tags/letter/G?page=4) tagged as GPL family (including misnamed variants) 21000+100+3000+2000+400 of 47985 projects on freecode (18 June 2014 frozen)
135. About Freecode (http://freecode.com/about) Archived (https://web.archive.org/web/20111031160805/http://freecode.com/about) 31 October 2011 at the Wayback Machine on freecode.com "The Freecode site has been moved to a static state effective 18 June 2014 due to low traffic levels and so that folks will focus on more useful endeavors than site upkeep."
136. "GPL use in Debian on the rise: study" (http://www.itwire.com/business-it-news/open-source/52838-gpl-use-in-debian-on-the-rise-study). Itwire.com. Retrieved 2 September 2013.
137. "Surveying open-source licenses" (https://lwn.net/Articles/547400/). Lwn.net. Retrieved 2 September 2013.
138. Mark (8 May 2008). "The Curse of Open Source License Proliferation" (https://web.archive.org/web/20151208112000/http://socializedsoftware.com/2008/05/08/the-curse-of-open-source-license-proliferation/). socializedsoftware.com. Archived from the original (http://socializedsoftware.com/2008/05/08/the-curse-of-open-source-license-proliferation/) on 8 December 2015. Retrieved 30 November 2015.
139. Top 20 Most Commonly Used Open Source Licenses (http://connollyshaun.blogspot.com/2009/03/top-20-most-commonly-used-open-source.html) Shaun Connolly, 11 March 2009
140. "Top 20 licenses" (https://web.archive.org/web/20160719043600/https://www.blackducksoftware.com/top-open-source-licenses). Black Duck Software. 6 June 2016. Archived from the original (http://www.blackducksoftware.com/resources/data/top-20-licenses) on 19 July 2016. Retrieved 6 June 2016.
141. "Top 20 licenses" (https://web.archive.org/web/20160719043600/https://www.blackducksoftware.com/top-open-source-licenses). Black Duck Software. 2 January 2017. Archived from the original (http://www.blackducksoftware.com/resources/data/top-20-licenses) on 19 July 2016. Retrieved 2 January 2017.
142. "Top 20 licenses" (https://web.archive.org/web/20160719043600/https://www.blackducksoftware.com/top-open-source-licenses). Black Duck Software. 4 June 2018. Archived from the original (http://www.blackducksoftware.com/resources/data/top-20-licenses) on 19 July 2016. Retrieved 4 June 2018.
143. Balter, Ben (9 March 2015). "Open source license usage on GitHub.com" (https://github.com/blog/1964-license-usage-on-github-com). github.com. Retrieved 21 November 2015.
144. Anwesha Das (22 June 2016). "Software Licenses in Fedora Ecosystem" (https://anweshadas.in/software-licenses-in-fedora-ecosystem/). anweshadas.in. Retrieved 1 November 2016. "From the above chart it is clear that the GPL family is the highest used (I had miscalculated it as MIT before). The other major licenses are MIT, BSD, the LGPL family, Artistic (for Perl packages), LPPL (fo[r] texlive packages), ASL."

145. Open Source Licensing Trends: 2017 vs. 2016 (https://resources.whitesourcesoftware.com/blog-whitesource/open-source-licensing-trends-2017-vs-2016) on whitesourcesoftware.com by Sivan Michaeli (12 April 2018)
146. "The GPL, the App Store and You (https://www.engadget.com/2011-01-09-the-gpl-the-app-store-and-you.html)" on engadget.com (2011)
147. "Copyright Policy" (http://www.openbsd.org/policy.html), OpenBSD
148. "Ubuntu One : Terms and Conditions" (https://archive.today/20130925165451/https://one.ubuntu.com/terms). One.ubuntu.com. 29 August 2013. Archived from the original (https://one.ubuntu.com/terms) on 25 September 2013. Retrieved 25 September 2013.
149. 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)
150. "GNU GENERAL PUBLIC LICENSE Version 2, June 1991" (https://web.archive.org/web/20130515070507/http://www.dwheeler.com/frozen/microsoft-interix-gpl.txt). *dwheeler.com*. Wheeler, David A. Archived from the original (http://www.dwheeler.com/frozen/microsoft-interix-gpl.txt) on 15 May 2013. Retrieved 6 December 2008.
151. *Free Software Leaders Stand Together* (https://en.wikisource.org/wiki/Free_Software_Leaders_Stand_Together) – via Wikisource.
152. Clarke, Gavin (20 July 2009). "Microsoft embraces Linux cancer to sell Windows servers" (https://www.theregister.co.uk/2009/07/20/microsoft_windows_drivers_linux/). The Register.
153. Clarke, Gavin (23 July 2009). "Microsoft opened Linux-driver code after 'violating' GPL" (https://www.theregister.co.uk/2009/07/23/microsoft_hyperv_gpl_violation/). The Register.
154. 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.
"General Public Virus" (http://catb.org/esr/jargon/oldversions/jarg221.txt). *Jargon File 2.2.1*. 15 December 1990. Retrieved 29 April 2007.
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.
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.
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.
Griffis, L. Adrian (15 July 2000). "The GNU Public Virus" (https://web.archive.org/web/20070930210455/http://themes.freshmeat.net/articles/view/172#comment-5548). Archived from the original (http://themes.freshmeat.net/articles/view/172#comment-5548) on 30 September 2007. Retrieved 29 April 2007.
155. "Speech Transcript – Craig Mundie, The New York University Stern School of Business (http://news.microsoft.com/speeches/speech-transcript-craig-mundie-the-new-york-university-stern-school-of-business/)", *Prepared Text of Remarks by Craig Mundie*, Microsoft Senior Vice President, The Commercial Software Model The New York University Stern School of Business 3 May 2001

156. 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.
Chopra, Samir; Dexter, Scott (14 August 2007). *Decoding liberation: the promise of free and open source software* (https://books.google.com/books?id=c7ppFih2mSwC&pg=PT74). Routledge. p. 56. ISBN 978-0-415-97893-4.
Williams, Sam (March 2002). *Free as in Freedom: Richard Stallman's Crusade for Free Software* (https://archive.org/details/freeasinfreedomr00will). O'Reilly Media. ISBN 0-596-00287-4.
157. Nikolai Bezroukov (2001). "Comparative merits of GPL, BSD and Artistic licences (Critique of Viral Nature of GPL v.2 – or In Defense of Dual Licensing Idea)" (https://web.archive.org/web/20011222205401/http://icfcst.kiev.ua/panorama/OSS/bsd_vs_gpl.shtml). Archived from the original (http://icfcst.kiev.ua/panorama/OSS/bsd_vs_gpl.shtml) on 22 December 2001. "Viral property stimulates proliferation of licenses and contributes to the "GPL-enforced nightmare" – a situation when many other licenses are logically incompatible with the GPL and make life unnecessary difficult for developers working in the Linux environment (KDE is a good example here, Python is a less known example)."
158. Geere, Duncan (16 December 2011). "Some rights reserved: the alternatives to copyright" (https://www.wired.co.uk/news/archive/2011-12/16/alternatives-to-copyright). *Wired UK*. Retrieved 30 May 2015.

"Inoculating Your Purchase – Contractual Protection from Viral Licenses in M&A Transactions" (http://www.friedfrank.com/siteFiles/Publications/ACC-GNY_Newsletter_OpenSourceInMATransactions.pdf) (PDF). Friedfrank.com. Retrieved 30 May 2015.

"Legal update on Information and Communication Technology – July 2013" (https://web.archive.org/web/20150316223946/http://www.buddlefindlay.com/article/2013/07/01/legal-update-on-information-and-communication-technology-%E2%80%93-july-2013). *buddlefindlay.com*. 1 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 16 March 2015.

159. New Media Rights (12 September 2008). "Open Source Licensing Guide" (http://www.newmediarights.org/open_source/new_media_rights_open_source_licensing_guide). California Western School of Law. Retrieved 28 November 2015.
160. Montague, Bruce (13 November 2013). "GPL Advantages and Disadvantages" (http://www.freebsd.org/doc/en_US.ISO8859-1/articles/bsdl-gpl/article.html#GPL-ADVANTAGES). FreeBSD. Retrieved 28 November 2015.
161. Richard Stallman (2010). On Selling Exceptions to the GNU GPL (http://www.fsf.org/blogs/rms/selling-exceptions). Free Software Foundation.
162. Bezroukov, Nikolai, Labyrinth of Software Freedom (http://www.softpanorama.org/Copyright/License_classification/index.shtml) – "BSD vs GPL and social aspects of free licensing debate" on softpanorama.org by Nikolai Bezroukov Accessed 23 September 2010.
163. The Scope of Open Source Licensing (http://www.eecs.harvard.edu/itm/seminars03-04/OSLicense.ver2.doc) Archived (https://web.archive.org/web/20160109200847/http://www.eecs.harvard.edu/itm/seminars03-04/OSLicense.ver2.doc) 9 January 2016 at the Wayback Machine – Harvard University by Josh Lerner and Jean Tirole (2002)
164. Sam Hocevar (21 September 2015). "Should I change the name of the WTFPL?" (http://programmers.stackexchange.com/questions/149050/should-i-change-the-name-of-the-wtfpl#comment617907_161949). *Programmers Stack Exchange* (User comment). Retrieved 19 July 2016. "The WTFPL is a parody of the GPL, which has a similar copyright header and list of permissions to modify (i.e. none), see for instance gnu.org/licenses/gpl-3.0.en.html. The purpose of the WTFPL wording is to give more freedom than the GPL does."

165. Biancuzzi, Federico (30 June 2005). "ESR: "We Don't Need the GPL Anymore" " (https://web.archive.org/web/20180417163519/http://www.onlamp.com/pub/a/onlamp/2005/06/30/esr_interview.html). onlamp.com. Archived from the original (http://www.onlamp.com/pub/a/onlamp/2005/06/30/esr_interview.html) on 17 April 2018. Retrieved 10 February 2015. "We don't need the GPL anymore. It's based on the belief that open source software is weak and needs to be protected. Open source would be succeeding faster if the GPL didn't make lots of people nervous about adopting it."
166. "RMS: The GNU GPL Is Here to Stay" (https://web.archive.org/web/20150117151743/http://www.onlamp.com/pub/a/onlamp/2005/09/22/gpl3.html). onlamp.com. 22 September 2005. Archived from the original (http://www.onlamp.com/pub/a/onlamp/2005/09/22/gpl3.html) on 17 January 2015. Retrieved 12 February 2015. "ESR addresses the issue in terms of different goals and values—those of "open source," which do not include defending software users' freedom to share and change software. Perhaps he thinks the GNU GPL is not needed to achieve those goals."
167. Randal, Allison (13 April 2007). "GPLv3, Linux and GPLv2 Compatibility" (http://radar.oreilly.com/2007/04/gplv3-linux-and-gplv2-compatib.html). *radar.oreilly.com*. O'Reilly Media. Retrieved 19 January 2016. "*You might think the FSF would have to be insane to unleash this licensing hell. ... If the license were purely a cleaned up version of the GPLv2, there would be no incompatibility, the FSF would have no agenda involved in getting projects to update to the new license, and at the same time there would be no reason for projects to object to updating. Smooth sailing.*"
168. Randal, Allison (14 May 2007). "GPLv3, Clarity and Simplicity" (http://radar.oreilly.com/2007/05/gplv3-clarity-and-simplicity.html). *radar.oreilly.com*. O'Reilly Media. Retrieved 19 January 2016. "Looking at the near-finished draft, I have to say it's unlikely that they ever considered simplicity a priority, if they considered it at all. ... The language choices of an open source license can support that freedom, can empower the users and the developers. The GPLv3 doesn't."
169. Whurley (6 June 2007). "The Death Of A Software License" (https://web.archive.org/web/20081011022019/http://talk.bmc.com/blogs/blog-whurley/whurley/the-death-of-a-software-license). Archived from the original (http://talk.bmc.com/blogs/blog-whurley/whurley/the-death-of-a-software-license) on 11 October 2008. Retrieved 24 June 2016. "Version 3 is going to distance Richard Stallman and the Free Software Foundation from the developers that make the organization so influential to begin."
170. Chisnall, David (31 August 2009). "The Failure of the GPL" (http://www.informit.com/articles/article.aspx?p=1390172&seqNum=3). informit.com. Retrieved 24 January 2016.
171. Cantrill, Bryan (17 September 2014). "Corporate Open Source Anti-patterns" (https://ghostarchive.org/varchive/youtube/20211027/Pm8P4oCIY3g). *YouTube*. Archived from the original (https://www.youtube.com/watch?v=Pm8P4oCIY3g&t=37m05s) on 27 October 2021. Retrieved 26 December 2015. "Anti-pattern: Anti-collaborative licensing"
172. Hill, Benjamin Mako (28 January 2006). "Notes on the GPLv3" (https://web.archive.org/web/20150922064257/http://archive09.linux.com/feature/51666). linux.com. Archived from the original (http://archive09.linux.com/feature/51666) on 22 September 2015. Retrieved 25 January 2016. "The GPL is one thing that almost everyone in the free and open-source software communities have in common. For that reason, the revision has the potential to highlight disagreements, differences in opinion, differences in business models, and differences in tactics. ... We would be wise to remember that the potential for the GPL to hinder our ability to work together is far more dangerous than the even the most radical change textual change the FSF might suggest. ... Above all, we must remember that our community and its goals are more important than any single license – no matter how widespread."

173. McDougall, Paul (10 July 2007). "Linux Creator Calls GPLv3 Authors 'Hypocrites' As Open Source Debate Turns Nasty" (https://web.archive.org/web/20080413091038/http://www.informationweek.com/blog/main/archives/2007/07/linux_creator_c.html). informationweek.com. Archived from the original (http://www.informationweek.com/blog/main/archives/2007/07/linux_creator_c.html) on 13 April 2008. Retrieved 12 February 2015. "... the latest sign of a growing schism in the open source community between business-minded developers like Torvalds and free software purists."
174. Mavrogiannopoulos, Nikos (26 March 2013). "The perils of LGPLv3" (http://nmav.gnutls.org/2013/03/the-perils-of-lgplv3.html). gnutls.org. Retrieved 18 November 2015. "LGPLv3 is the latest version of the GNU Lesser General Public License. It follows the successful LGPLv2.1 license, and was released by Free Software Foundation as a counterpart to its GNU General Public License version 3. The goal of the GNU Lesser General Public Licenses is to provide software that can be used by both proprietary and free software. This goal has been successfully handled so far by LGPLv2.1, and there is a multitude of libraries using that license. Now we have LGPLv3 as the latest, and the question is how successful is LGPLv3 on this goal? In my opinion, very little. If we assume that its primary goal is to be used by free software, then it blatantly fails that."
175. "GnuTLS 3.1.10: changelog" (http://www.gnutls.org/abi-tracker/changelog/gnutls/3.1.10/log.html). *www.gnutls.org*.
Nikos Mavrogiannopoulos (18 December 2012). "gnutls is moving" (https://lwn.net/Articles/529558/). Retrieved 11 December 2012.
176. Rosen, Lawrence (2007). "Comments on GPLv3" (http://www.rosenlaw.com/GPLv3-Comments.htm). Rosenlaw.com. Retrieved 22 August 2014.
177. Ronacher, Armin (23 July 2013). "Licensing in a Post Copyright World" (http://lucumr.pocoo.org/2013/7/23/licensing/). *lucumr.pocoo.org*. Retrieved 18 November 2015. "The License Compatibility Clusterfuck – When the GPL is involved the complexities of licensing becomes a non fun version of a riddle. So many things to consider and so many interactions to consider. And that GPL incompatibilities are still an issue that actively effects people is something many appear to forget. For instance one would think that the incompatibility of the GPLv2 with the Apache Software License 2.0 should be a thing of the past now that everything upgrades to GPLv3, but it turns out that enough people are either stuck with GPLv2 only or do not agree with the GPLv3 that some Apache Software licensed projects are required to migrate. For instance Twitter's Bootstrap is currently migrating from ASL2.0 to MIT precisely because some people still need GPLv2 compatibility. Among those projects that were affected were Drupal, WordPress, Joomla, the MoinMoin Wiki and others. And even that case shows that people don't care that much about licenses any more as Joomla 3 just bundled bootstrap even though they were not licenses in a compatible way (GPLv2 vs ASL 2.0). The other traditional case of things not being GPL compatible is the OpenSSL project which has a license that does not go well with the GPL. That license is also still incompatible with the GPLv3. The whole ordeal is particularly interesting as some not so nice parties have started doing license trolling through GPL licenses."
Ronacher, Armin (2009). "Are you sure you want to use the GPL?" (http://lucumr.pocoo.org/2009/2/12/are-you-sure-you-want-to-use-gpl/). *lucumr.pocoo.org*.

External links

- Official website (https://www.gnu.org/licenses/gpl.html)
- GNU General Public License v3.0 (https://www.gnu.org/licenses/gpl-3.0.html)
- GNU General Public License v2.0 (https://www.gnu.org/licenses/old-licenses/gpl-2.0.html)— This version is deprecated by the FSF, but is still used by many software projects, including Linux kernel and GNU packages.

- GNU General Public License v1.0 (https://www.gnu.org/licenses/old-licenses/gpl-1.0.html)—This version is deprecated by the FSF.
- The Emacs General Public License (http://www.free-soft.org/gpl_history/emacs_gpl.html), a February 1988 version, a direct predecessor of the GNU GPL
- History of the GPL (http://www.free-soft.org/gpl_history/) by Li-Cheng Tai, 4 July 2001
- A Practical Guide to GPL Compliance (https://softwarefreedom.org/resources/2008/compliance-guide.html) (Covers GPLv2 and v3)—from the Software Freedom Law Center
- A paper on enforcing the GPL (https://web.archive.org/web/20130502102348/http://sapnakumar.org/Publications_files/EnfGPL.pdf)
- Frequently Asked Questions about the GPL (https://gnu.org/licenses/gpl-faq.html)
- GNU General Public License and Commentaries (https://web.archive.org/web/20060109051638/http://www.rattlesnake.com/software-law/GNU-GPL-and-Commentaries.html), edited by Robert Chassell
- List of presentation transcripts about the GPL and free software licenses (https://wiki.fsfe.org/Transcripts#Licences_and_the_GNU_GPL) Archived (https://web.archive.org/web/20151112175145/https://wiki.fsfe.org/Transcripts#Licences_and_the_GNU_GPL) 12 November 2015 at the Wayback Machine by the FSFE
- The Labyrinth of Software Freedom (http://www.softpanorama.org/Copyright/License_classification/index.shtml) BSD vs GPL and social aspects of free licensing debate, by Nikolai Bezroukov

---

GNU Affero General Public License

The **GNU Affero General Public License** (**GNU AGPL**) is a free, copyleft license published by the Free Software Foundation in November 2007, and based on the GNU GPL version 3 and the *Affero General Public License* (non-GNU).

It is intended for software designed to be run over a network, adding a provision requiring that the corresponding source code of modified versions of the software be prominently offered to all users who interact with the software over a network.[6]

The Open Source Initiative approved the GNU AGPLv3[3] as an open source license in March 2008 after the company Funambol submitted it for consideration through its CEO Fabrizio Capobianco.[7]

GNU Affero General Public License

Author	Free Software Foundation
Latest version	3
Publisher	Free Software Foundation, Inc.
Published	November 19, 2007
SPDX identifier	AGPL-3.0-or-later AGPL-3.0-only
Debian FSG compatible	Yes[1]
FSF approved	Yes[2]
OSI approved	Yes[3][4]
GPL compatible	Yes (permits linking with GPLv3)[5]
Copyleft	Yes,[2] incl. use over network
Linking from code with a different licence	Only with GPLv3; AGPL terms will apply for the AGPL part in a combined work.[2][5]
Website	www.gnu.org/licenses/agpl.html (https://www.gnu.org/licenses/agpl.html)

History

In 2000, while developing an e-learning and e-service business model at Mandriva, Henry Poole met with Richard Stallman in Amsterdam and discussed the issue of the GPLv2 license not requiring Web application providers to share source code with the users interacting with their software over a network. Over the following months, Stallman and Poole discussed approaches to solve the problem. In 2001, Poole founded Affero Inc. (a web services business), and he needed a license that would require distribution by other organizations who used Affero code to create derivative web services. At that time, Poole contacted Bradley M. Kuhn and Eben Moglen of the Free Software Foundation to get advice on a new license that would resolve this matter in GPLv2.

Around late February 2002, Kuhn suggested, based on the idea of a quine (a program that prints its own source code), that GPLv2 be supplemented with a section 2(d) that would require derivative works to maintain a "download source" feature that would provide complete and corresponding source code. Kuhn argued that there was precedent for such a requirement in GPLv2 section 2(c), which required preserving certain features by downstream distributors and modifiers.[8]

Moglen and Kuhn wrote the text of the proposed new section 2(d), and provided it to Poole, who then requested and received permission from the FSF to publish a derivative of GPLv2 for this purpose. In March 2002, Affero, Inc. published the original Affero General Public License (AGPLv1) for use with the Affero project and made the new license available for use by other software-as-a-service developers.[9][10][11]

The FSF contemplated including the special provision of AGPLv1 into GPLv3 but ultimately decided to publish a separate license, nearly identical to GPLv3 but containing a provision similar in purpose and effect to section 2(d) of AGPLv1. The new license was named the GNU Affero General Public License. Retaining the Affero name indicated its close historic relationship with AGPLv1. The GNU AGPL was given version number 3 for parity with the GPL, and the current GNU Affero General Public License is often abbreviated *AGPLv3*.

The finalized version of GNU AGPLv3[12] was published by the FSF on November 19, 2007.

Compatibility with the GPL

Both versions of the AGPL, like the corresponding versions of the GNU GPL on which they are based, are strong copyleft licenses. In the Free Software Foundation's judgment, the added requirement in section 2(d) of Affero GPL v1 made it incompatible with the otherwise nearly identical GPLv2. That is to say, one cannot distribute a single work formed by combining components covered by each license.

By contrast, the GPLv3 and GNU AGPLv3 licenses include clauses (in section 13 of each license) that together achieve a form of mutual compatibility for the two licenses. These clauses explicitly allow the "conveying" of a work formed by linking code licensed under the one license against code licensed under the other license,[13] despite the licenses otherwise not allowing relicensing under the terms of each other.[2] In this way, the copyleft of each license is relaxed to allow distributing such combinations.[2]

To establish an upgrade path from Affero's original AGPLv1 to the GNU AGPLv3, Affero, Inc. published the Affero General Public License version 2 in November 2007,[14] which is merely a transitional license that allows recipients of software licensed under "AGPLv1 or any later version as published by Affero, Inc." to distribute the software, or derivative works, under the GNU AGPLv3 or any later version.

Examples of applications under GNU AGPL

Stet was the first software system known to be released under the GNU AGPL, on November 21, 2007,[8] and is the only known program to be used mainly for the production of its own license.

Flask developer Armin Ronacher noted in 2013 that the GNU AGPL is a "terrible success, especially among the startup community" as a "vehicle for dual commercial licensing", and gave HumHub, MongoDB, Odoo, RethinkDB, Shinken, Slic3r, SugarCRM, and WURFL as examples.[15]

MongoDB dropped the AGPL in late-2018 in favor of the "Server Side Public License" (SSPL), a modified version which requires those who offer the licensed software as a service accessible to third-parties, to make the entire source code of all software used to facilitate the service (including without limitation all "management software, user interfaces, application program interfaces, automation software, monitoring software, backup software, storage software and hosting software, all such that a user could run an instance of the service using the Service Source Code you make available") available under the same license.[16] As approval for the SSPL by the Open Source Initiative was not forthcoming, the application for certification was withdrawn. It was banned by both Debian and the Fedora Project, who state that the license's intent is to discriminate against cloud computing providers offering services based on the software without purchasing its commercial license.[17][18]

Software continues to be released under AGPLv3, various examples include many servers and clients for the fediverse such as Mastodon, Pixelfed and PeerTube, office suite software OnlyOffice, the RStudio IDE for the R programming language, system monitoring platform Grafana, the document/bibliography management system Zotero and more.

Decentralized chat and collaboration software Element was relicensed from Apache 2.0 to both AGPLv3 and GPLv3, with a separate commercial license for Element Commercial.[19][20][21]

See also

Free and open-source software portal

- List of software under the GNU AGPL
- Free-software license
- GNU General Public License
- GNU Lesser General Public License
- GNAT Modified General Public License
- GPL linking exception
- GNU Free Documentation License
- Comparison of free and open-source software licenses

References

1. Jaspert, Joerg (November 28, 2008). "ftp.debian.org: Is AGPLv3 DFSG-free?" (https://bugs.debian.org/cgi-bin/bugreport.cgi?msg=17;bug=495721). The Debian Project. Retrieved December 1, 2008.
2. "Various Licenses and Comments about Them" (https://www.gnu.org/licenses/license-list.html#AGPL). Free Software Foundation. 2020-05-07. Retrieved 2021-01-03. "We recommend that developers consider using the GNU AGPL for any software which will commonly be run over a network."
3. "OSI approved licenses" (https://ghostarchive.org/archive/20211023/https://opensource.org/licenses/alphabetical). Open Source initiative. Archived from the original (https://opensource.org/licenses/alphabetical) on 2021-10-23.
4. "OSI approved" (https://web.archive.org/web/20211128165931/https://tldrlegal.com/licenses/tags/OSI-Approved), *Licenses*, TL;DR legal, archived from the original (https://tldrlegal.com/licenses/tags/OSI-Approved) on 2021-11-28, retrieved 2016-02-17.
5. "Licenses section 13" (https://www.gnu.org/licenses/agpl-3.0.html#section13), *GNU AGPLv3*, GNU Project.

6. "Why the Affero GPL" (https://ghostarchive.org/archive/20211023/https://www.gnu.org/licenses/why-affero-gpl.html). *The GNU Project*. Archived from the original (https://www.gnu.org/licenses/why-affero-gpl.html) on 2021-10-23.
7. "Funambol Helps New AGPLv3 Open Source License Gain Formal OSI Approval" (https://web.archive.org/web/20130607064424/http://www.funambol.com/news/pressrelease_2008.3.13.php) (Press release). Funambol. Mar 13, 2008. Archived from the original (https://www.funambol.com/news/pressrelease_2008.3.13.php) on 2013-06-07.
8. Kuhn, Bradley M. (November 21, 2007). "stet and AGPLv3" (https://web.archive.org/web/20080315231323/http://www.softwarefreedom.org/technology/blog/2007/nov/21/stet-and-agplv3/). Software Freedom Law Center. Archived from the original (http://www.softwarefreedom.org/technology/blog/2007/nov/21/stet-and-agplv3/) on March 15, 2008. Retrieved June 14, 2008.
9. "Affero GPLv3: Why It Exists & Who It's For?" (https://ebb.org/bkuhn/talks/SCALE-2013/agplv3.html). *ebb.org*. Retrieved 2024-04-25.
10. "SCALE: The life and times of the AGPL [LWN.net]" (https://lwn.net/Articles/541981/). *lwn.net*. Retrieved 2024-04-25.
11. "Free Software Foundation Announces Support of the Affero General Public License, the First Copyleft License for Web Services" (https://www.gnu.org/press/2002-03-19-Affero.html). Free Software Foundation. 2002-03-19. Retrieved 2021-01-03.
12. "License text of GNU AGPLv3" (https://www.gnu.org/licenses/agpl-3.0.html). *Free Software Foundation (US)*. November 19, 2007. Archived (https://web.archive.org/web/20071204202305/https://www.gnu.org/licenses/agpl-3.0.html) from the original on 2007-12-04. Retrieved November 19, 2007.
13. "GNU General Public License" (https://www.gnu.org/licenses/gpl-3.0.html). Free Software Foundation. 2007-06-29. Retrieved 2021-01-03.
14. "Affero General Public License" (https://web.archive.org/web/20191123102313/http://www.affero.org/agpl2.html). November 2007. Archived from the original on 2019-11-23. Retrieved 2021-01-03.
15. Ronacher, Armin (2013-07-23). "Licensing in a Post Copyright World" (http://lucumr.pocoo.org/2013/7/23/licensing/). lucumr.pocoo.org. Archived (https://web.archive.org/web/20130727124604/http://lucumr.pocoo.org/2013/7/23/licensing) from the original on 2013-07-27. Retrieved 2015-11-18. "*The AGPLv3 was a terrible success, especially among the startup community that found the perfect base license to make dual licensing with a commercial license feasible. MongoDB, RethinkDB, OpenERP, SugarCRM as well as WURFL all now utilize the AGPLv3 as a vehicle for dual commercial licensing. The AGPLv3 makes that generally easy to accomplish as the original copyright author has the rights to make a commercial license possible but nobody who receives the sourcecode itself through the APLv3 inherits that right. I am not sure if that was the intended use of the license, but that's at least what it's definitely being used for now.*"
16. "Server Side Public License (SSPL)" (https://ghostarchive.org/archive/20211023/https://www.mongodb.com/licensing/server-side-public-license). *MongoDB*. Archived from the original (https://www.mongodb.com/licensing/server-side-public-license) on 2021-10-23. Retrieved 2021-01-25.
17. Vaughan-Nichols, Steven J. "MongoDB "open-source" Server Side Public License rejected" (https://ghostarchive.org/archive/20211023/https://www.zdnet.com/article/mongodb-open-source-server-side-public-license-rejected/). *ZDNet*. Archived from the original (https://www.zdnet.com/article/mongodb-open-source-server-side-public-license-rejected/) on 2021-10-23. Retrieved 2019-01-17.

18. "MongoDB's licensing changes led Red Hat to drop the database from the latest version of its server OS" (https://ghostarchive.org/archive/20211023/https://www.geekwire.com/2019/mongodbs-licensing-changes-led-red-hat-drop-database-latest-version-server-os/). *GeekWire*. 2019-01-16. Archived from the original (https://www.geekwire.com/2019/mongodbs-licensing-changes-led-red-hat-drop-database-latest-version-server-os/) on 2021-10-23. Retrieved 2019-01-17.
19. "Element Copyright License - AGPL 3.0" (https://github.com/element-hq/element-web/blob/develop/LICENSE-AGPL-3.0). *Github*. Retrieved 20 January 2025.
20. "Element Copyright License - GPL 3.0" (https://github.com/element-hq/element-web/blob/develop/LICENSE-GPL-3.0). *Github*. Retrieved 20 January 2025.
21. "Element Copyright License - Commercial License" (https://github.com/element-hq/element-web/blob/develop/LICENSE-COMMERCIAL). *Github*. Retrieved 20 January 2025.

External links

- Official website (https://www.gnu.org/licenses/agpl.html)
- GNU Affero General Public License v3.0 (https://www.gnu.org/licenses/agpl-3.0.html)
- Smith, Brett (November 19, 2007). "Free Software Foundation Releases GNU Affero General Public License Version 3" (https://www.fsf.org/agplv3-pr) (Press release).
- Smith, Brett (March 29, 2007), *GPLv3 and Software as a Service* (https://www.fsf.org/blogs/licensing/2007-03-29-gplv3-saas) – also includes info on version 2 of the Affero GPL.
- Kuhn, Bradley M. (March 19, 2002). "Free Software Foundation Announces Support of the Affero General Public License, the First Copyleft License for Web Services" (https://www.gnu.org/press/2002-03-19-Affero.html) (Press release).
- Internet Archive 2018 snapshot of AGPL Frequently Asked Questions (https://web.archive.org/web/20180322021749/http://www.affero.org/oagf.html) Affero
- Internet Archive 2018 snapshot of AGPL text (https://web.archive.org/web/20180307004233/http://www.affero.org/oagpl.html) Affero

WIKIPEDIA
The Free Encyclopedia

GNU Lesser General Public License

The **GNU Lesser General Public License** (**LGPL**) is a free-software license published by the Free Software Foundation (FSF). The license allows developers and companies to use and integrate a software component released under the LGPL into their own (even proprietary) software without being required by the terms of a strong copyleft license to release the source code of their own components. However, any developer who modifies an LGPL-covered component is required to make their modified version available under the same LGPL license. For proprietary software, code under the LGPL is usually used in the form of a shared library, so that there is a clear separation between the proprietary and LGPL components. The LGPL is primarily used for software libraries, although it is also used by some stand-alone applications.

The LGPL was developed as a compromise between the strong copyleft of the GNU General Public License (GPL) and more permissive licenses such as the BSD licenses and the MIT License. The word "Lesser" in the title shows that the LGPL does not guarantee the end user's complete freedom in the use of software; it only guarantees the freedom of modification for components licensed under the LGPL, but not for any proprietary components.

GNU Lesser General Public License

LGPL 3 Free Software
Free as in Freedom
Logo

Published	1991
SPDX identifier	LGPL-3.0-or-later LGPL-3.0-only LGPL-2.1-or-later LGPL-2.1-only LGPL-2.0-or-later LGPL-2.0-only
Debian FSG compatible	Yes[1]
FSF approved	Yes[2]
OSI approved	Yes[3]
GPL compatible	Yes[2]
Copyleft	Yes[2] (library or dynamic linking-based)
Linking from code with a different licence	Yes
Website	www.gnu.org/licenses/lgpl.html (https://www.gnu.org/licenses/lgpl.html)

History

The license was originally called the **GNU Library General Public License** and was first published in 1991, and adopted the version number 2 for parity with GPL version 2. The LGPL was revised in minor ways in the 2.1 point release, published in 1999, when it was renamed the GNU Lesser General Public License to reflect the FSF's position that not all libraries should use it. Version 3 of the LGPL was published in 2007 as a list of additional permissions applied to GPL version 3.

In addition to the term "work based on the Program" of GPL, LGPL version 2 introduced two additional clarification terms "work based on the library" and "work that uses the library".[4] LGPL version 3 partially dropped these terms.

Differences from the GPL

The main difference between the GPL and the LGPL is that the latter allows the work to be linked with (in the case of a library, "used by") a non-(L)GPLed program, regardless of whether it is licensed under a license of GPL family or other licenses.[5] In LGPL 2.1, the non-(L)GPLed program can then be distributed under any terms if it is not a derivative work. If it is a derivative work, then the program's terms must allow for "modification of the work for the customer's own use and reverse engineering for debugging such modifications". Whether a work that uses an LGPL program is a derivative work or not is a legal issue. A standalone executable that dynamically links to a library through a .so, .dll, or similar medium is generally accepted as not being a derivative work as defined by the LGPL. It would fall under the definition of a "work that uses the Library". Paragraph 5 of the LGPL version 2.1 states:

> A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

Essentially, if it is a "work that uses the library", then it must be possible for the software to be linked with a newer version of the LGPL-covered program. The most commonly used method for doing so is to use "a suitable shared library mechanism for linking". Alternatively, a statically linked library is allowed if either source code or linkable object files are provided.[6]

Compatibility

One feature of the LGPL is the permission to sublicense[7] under the GPL any piece of software which is received under the LGPL (see section 3 of the LGPL version 2.1, and section 2 option b of the LGPL version 3). This feature allows for direct reuse of LGPLed code in GPLed libraries and applications.

Version 3 of the LGPL is not inherently compatible with version 2 of the GPL. However, works using the latter that have given permission to use a later version of the GPL are compatible:[8] a work released under the GPLv2 "or any later version" may be combined with code from a LGPL version 3 library, with the combined work as a whole falling under the terms of the GPLv3.[9]

FSF recommendations on library licensing

The former name *GNU Library General Public License* gave some the impression that the FSF recommended that all software libraries should use the LGPL and programs should use the GPL. In 1999 essay *Why you shouldn't use the Lesser GPL for your next library* Richard Stallman explained that while the LGPL had not been deprecated, one should not necessarily use the LGPL for all libraries, as using GPL can give advantage to free-software developers.[5]

Programming language specifications

The license uses terminology which is mainly intended for applications written in the C programming language or its family. Franz Inc., the developers of Allegro Common Lisp, published their own preamble to the license to clarify terminology in the Lisp context. The LGPL with this preamble is sometimes referred to as the LLGPL.[10]

In addition, Ada has a special feature, generics, which may prompt the use of the GNAT Modified General Public License (GMGPL): it allows code to link against or instantiate GMGPL-covered units without the code itself becoming covered by the GPL.

C++ templates and header-only libraries have the same problem as Ada generics. Version 3 of the LGPL addresses such cases in section 3.[11]

Class inheritance

Some concern has risen about the suitability of object-oriented classes in LGPL-licensed code being inherited by non-(L)GPL code. Clarification is given on the official GNU website:

> The LGPL does not contain special provisions for inheritance, because none are needed. Inheritance creates derivative works in the same way as traditional linking, and the LGPL permits this type of derivative work in the same way as it permits ordinary function calls.[12]

See also

Free and open-source software portal

- GNU Affero General Public License
- GNU Free Documentation License
- GNAT Modified General Public License
- GPL linking exception
- Software using the GNU Lesser General Public License (category)

References

1. License information (https://www.debian.org/legal/licenses/) Debian
2. "Various Licenses and Comments about Them" (https://www.gnu.org/licenses/license-list.en.html#LGPL). *Free Software Foundation*. 26 June 2018. Retrieved 2 August 2018.
3. Licenses & Standards (https://opensource.org/licenses) Open Source Initiative
4. "Chapter 10. The Lesser GPL" (https://copyleft.org/guide/comprehensive-gpl-guidech11.html). *Copyleft and the GNU General Public License: A Comprehensive Tutorial and Guide*. 2018.
5. Stallman, Richard. "Why you shouldn't use the Lesser GPL for your next library (http://www.fsf.org/licensing/licenses/why-not-lgpl.html)". Free Software Foundation official website.

6. Gnu Lesser General Public License, Section 4 (https://www.gnu.org/copyleft/lesser.html#section4), pt. d) 0)
7. "sublicense" (https://en.wiktionary.org/wiki/sublicense). *Wiktionary*. 7 December 2020.
8. GNU Lesser General Public License (LGPL) version 3 (https://www.gnu.org/licenses/license-list.html#LGPL). Free Software Foundation official website.
9. Frequently Asked Questions about the GNU Licenses - GNU Project - Free Software Foundation (https://www.gnu.org/licenses/gpl-faq.html#gpl-compat-matrix). Free Software Foundation official website.
10. Preamble to the Gnu Lesser General Public License (http://opensource.franz.com/preamble.html)
11. LGPLv3 for a C++ Pure Template Library ? (https://listengine.tuxfamily.org/lists.tuxfamily.org/eigen/2008/02/msg00003.html)
12. Turner, David. The LGPL and Java (https://www.gnu.org/licenses/lgpl-java.html). GNU official website.

External links

- Official website (https://www.gnu.org/licenses/lgpl.html)
- GNU Lesser General Public License v3.0 (https://www.gnu.org/licenses/lgpl-3.0.html)
- GNU Lesser General Public License v2.1 (https://www.gnu.org/licenses/old-licenses/lgpl-2.1.html)—This version is deprecated by the FSF.
- GNU Lesser General Public License v2.0 (https://www.gnu.org/licenses/old-licenses/lgpl-2.0.html)—This version is deprecated by the FSF.
- Derivative Works (http://www.linuxjournal.com/article/6366)
- Lisping Copyleft: A Close Reading of the Lisp LGPL, 5 International Free and Open Source Software Law Review 15 (2013) (http://www.ifosslr.org/ifosslr/article/view/75)

Wikipedia
The Free Encyclopedia

ISC license

The **ISC license** is a permissive free software license published by the Internet Software Consortium, now called Internet Systems Consortium (ISC). It is functionally equivalent to the simplified BSD and MIT licenses, but without language deemed unnecessary following the Berne Convention.[nb 1][nb 2]

Originally used for ISC software such as BIND and dig, it has become the preferred license for contributions to OpenBSD[6] and the default license for npm packages.[7] The ISC license is also used for Linux wireless drivers contributed by Qualcomm Atheros,[8] as well as by the LV2 plugin system.[9]

ISC license

Publisher	Internet Systems Consortium
SPDX identifier	ISC
Debian FSG compatible	Yes[1]
FSF approved	Yes[2]
OSI approved	Yes[3]
GPL compatible	Yes[2]
Copyleft	No[2]
Linking from code with a different licence	Yes
Website	www.isc.org/licenses/ (https://www.isc.org/licenses/)

License terms

```
ISC License

Copyright <YEAR> <OWNER>

Permission to use, copy, modify, and/or
distribute this software for any purpose
with or without fee is hereby granted,
provided that the above copyright notice
and this permission notice appear in all
copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE
AUTHOR DISCLAIMS ALL WARRANTIES WITH
REGARD TO THIS SOFTWARE INCLUDING ALL
IMPLIED WARRANTIES OF MERCHANTABILITY AND
FITNESS. IN NO EVENT SHALL THE AUTHOR BE
LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT,
OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE,
DATA OR PROFITS, WHETHER IN AN ACTION OF
CONTRACT, NEGLIGENCE OR OTHER TORTIOUS
ACTION, ARISING OUT OF OR IN CONNECTION
WITH THE USE OR PERFORMANCE OF THIS
SOFTWARE.
```

— [10][11]

When initially released, the license did not include the term "and/or", which was changed from "and" by ISC in 2007.[12] Paul Vixie stated on the BIND mailing list that the ISC license started using the term "and/or" to avoid controversy similar to the events surrounding the University of Washington's refusal to allow distribution of the Pine email software.[12] Both versions of the license are compatible with the GNU GPL.[2]

OpenBSD license

The OpenBSD project began using the ISC license in 2003, before the 2007 addition of the term "and/or" to the license.[13]

```
Copyright (c) YYYY YOUR-NAME-HERE <user@your.dom.ain>

Permission to use, copy, modify, and distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
```

—[6]

Theo de Raadt of OpenBSD chose to retain the original wording, stating that he "disagrees with what ISC did" and is "not confident that their change is good" because "some country's legal systems might not understand 'and/or' in the way the old 'or' was used in the sentence".[5]

Reception

In 2015, ISC announced they would release their Kea DHCP Software under the Mozilla Public License 2.0, stating, "There is no longer a good reason for ISC to have its own license, separate from everything else".[14] They also preferred a copyleft license, stating, "If a company uses our software but improves it, we really want those improvements to go back into the master source". Throughout the following years, they re-licensed all ISC-hosted (https://www.isc.org/download) software, including BIND in 2016[15] and ISC DHCP Server in 2017.[16]

The Publications Office of the European Union advises using the MIT license instead of the ISC License in order to reduce license proliferation.[17]

The GNU project states the inclusion of "and/or" still allows the license to be interpreted as prohibiting distribution of modified versions. Although they state there is no reason to avoid software released under this license, they advise against using the license to keep the problematic language from causing trouble in the future.[18]

See also

Free and open-source software portal

- Comparison of free and open-source software licenses
- Software using the ISC license

Footnotes

1. "The ISC copyright is functionally equivalent to a two-term BSD copyright with language removed that is made unnecessary by the Berne convention."[4]
2. "In OpenBSD we use an ISC-style copyright text [...] that is enough to satisfy every legal system on the planet which follows the Berne Convention."[5]

References

1. "License information" (https://www.debian.org/legal/licenses/). *The Debian Project*. Software in the Public Interest (published 12 July 2017). 1997–2017. Archived (https://web.archive.org/web/20170720142826/https://www.debian.org/legal/licenses/) from the original on 20 July 2017. Retrieved 20 July 2017. "... This page presents the opinion of some debian-legal contributors on how certain licenses follow the Debian Free Software Guidelines (DFSG). ... Licenses currently found in Debian main include:

    - ...
    - Expat/MIT-style licenses
    - ...

    "
2. "Various Licenses and Comments about Them" (https://www.gnu.org/licenses/license-list.en.html). *The GNU Project*. Free Software Foundation (published 4 April 2017). 2014–2017. ISC License. Archived (https://web.archive.org/web/20170720140022/https://www.gnu.org/licenses/license-list.en.html) from the original on 20 July 2017. Retrieved 20 July 2017. "... This is a lax, permissive free software license, and compatible with the GNU GPL. ... This license is sometimes also known as the OpenBSD License, although there is one minor difference between the two licenses. The OpenBSD license was updated to remove the ambiguous term: "and/or". ... At the time the ISC license was released, the use of "and/or" construct was a concern because it is similar language used in the license of Pine that the University of Washington later claimed prohibited people from distributing modified versions of the software. ..."
3. "Licenses by Name" (https://opensource.org/licenses/alphabetical). *Open Source Initiative*. n.d. Archived (https://web.archive.org/web/20170720141629/https://opensource.org/licenses/alphabetical) from the original on 20 July 2017. Retrieved 20 July 2017. "... The following licenses have been approved by the OSI. ...

    - ...
    - ISC License (ISC)
    - ...

    "

4. "Copyright Policy" (http://www.openbsd.org/policy.html). *OpenBSD*. Retrieved 6 June 2016.
5. de Raadt, Theo (21 March 2008). "Re: BSD Documentation License?" (https://marc.info/?l=openbsd-misc&m=120618313520730&w=2). *openbsd-misc* (Mailing list). Retrieved 17 March 2017.
6. "OpenBSD license template" (http://cvsweb.openbsd.org/cgi-bin/cvsweb/src/share/misc/license.template?rev=HEAD). Retrieved 17 March 2017.
7. Oxley, Tim (8 January 2014). "Update default license to ISC" (https://github.com/npm/init-package-json/pull/9). *npm/init-package-json*. Pull Request #9. Retrieved 17 March 2017.
8. "en/users/Drivers/Atheros - Linux Wireless" (https://web.archive.org/web/20170621011348/http://linuxwireless.org/en/users/Drivers/Atheros/). *linuxwireless.org*. Archived from the original (http://linuxwireless.org/en/users/Drivers/Atheros/) on 21 June 2017. Retrieved 15 July 2017.
9. "COPYING · master · LV2 / lv2 · GitLab" (https://gitlab.com/lv2/lv2/-/blob/master/COPYING). *gitlab.com*. Retrieved 10 August 2023.
10. "ISC License (ISC)" (https://opensource.org/licenses/ISC). *Open Source Initiative*. Retrieved 15 July 2017.
11. "ISC License" (https://spdx.org/licenses/ISC.html). *Software Package Data Exchange*. Retrieved 9 July 2021.
12. Vixie, Paul (20 July 2007). "BIND covered under which license and does it conatin any cryptographic content ?" (https://groups.google.com/forum/#!msg/comp.protocols.dns.bind/D8VLXloVfxc/4yweyAjRCecJ). Newsgroup: comp.protocols.dns.bind (news:comp.protocols.dns.bind). Retrieved 15 July 2017 – via Google Groups.
13. "src/share/misc/license.template - view - 1.1" (https://cvsweb.openbsd.org/cgi-bin/cvsweb/src/share/misc/license.template?rev=1.1&content-type=text/x-cvsweb-markup). 3 June 2003.
14. "Kea to be released under Mozilla Public License 2.0 - Internet Systems Consortium" (https://www.isc.org/blogs/kea-license-2-0/). *www.isc.org*. 8 December 2015. Retrieved 18 July 2020.
15. "BIND 9 Adopts the MPL 2.0 License with BIND 9.11.0 - Internet Systems Consortium" (https://www.isc.org/blogs/bind9-adopts-the-mpl-2-0-license-with-bind-9-11-0/). *www.isc.org*. 27 June 2016. Retrieved 18 July 2020.
16. "ISC DHCP Moves to MPL 2.0 License - Internet Systems Consortium" (https://www.isc.org/blogs/isc-dhcp-moves-to-mpl-2-0-license/). *www.isc.org*. 19 June 2017. Retrieved 18 July 2020.
17. "ISC License (ISC) - EU Vocabularies - Publications Office of the EU" (https://op.europa.eu/en/web/eu-vocabularies/concept/-/resource?uri=http://publications.europa.eu/resource/authority/licence/ISC). *op.europa.eu*. Retrieved 18 July 2020.
18. "Various Licenses and Comments about Them" (https://www.gnu.org/licenses/license-list.en.html#ISC). *www.gnu.org*. Retrieved 18 July 2020.

External links

- Internet Systems Consortium's License Text (http://www.isc.org/downloads/software-support-policy/isc-license/)
- License template at the Open Source Initiative (http://opensource.org/licenses/ISC)

WIKIPEDIA
The Free Encyclopedia

MIT License

The **MIT License** is a permissive software license originating at the Massachusetts Institute of Technology (MIT)[6] in the late 1980s.[7] As a permissive license, it puts very few restrictions on reuse and therefore has high license compatibility.[8][9]

MIT License	
Publisher	Massachusetts Institute of Technology
SPDX identifier	MIT (see list for more)[1]
Debian FSG compatible	Yes[2]
FSF approved	Yes[3][4]
OSI approved	Yes[5]
GPL compatible	Yes[3][4]
Copyleft	No[3][4]
Linking from code with a different licence	Yes

Unlike copyleft software licenses, the MIT License also permits reuse within proprietary software, provided that all copies of the software or its substantial portions include a copy of the terms of the MIT License and also a copyright notice.[9][10] In 2015, the MIT License was the most popular software license on GitHub,[11] and was still the most popular in 2025.[12]

Notable projects that use the MIT License include the X Window System, Ruby on Rails, Node.js, Lua, jQuery, .NET, Angular, and React.

License terms

The MIT License has the identifier `MIT` in the SPDX License List.[13][14] It is also known as the "Expat License".[3] It has the following terms:[15]

```
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.
```

Variations

X11 License

The **X11 License**, also known as the **MIT/X Consortium License**, is a variation on the MIT license, most known for its usage by the X Consortium.[16] It has the identifier `X11` in the SPDX License List.[17][4]

It differs from the MIT License mainly by an additional clause restricting use of the copyright holders' name for advertisement.

It has the following terms:[18]

```
Copyright (C) <date> <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 X CONSORTIUM 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.

Except as contained in this notice, the name of <copyright holders> shall not be used in advertising
or otherwise to promote the sale, use or other dealings in this Software without prior written
authorization from <copyright holders>.
```

MIT No Attribution License

The MIT No Attribution License, a variation of the MIT License, has the identifier `MIT-0` in the SPDX License List.[20] A request for legacy approval to the Open Source Initiative was filed on May 15, 2020,[21] which led to a formal approval on August 5, 2020.[19] By doing so, it forms a public-domain-equivalent license, the same way as BSD Zero Clause. It has the following terms:

```
MIT No Attribution

Copyright <YEAR> <COPYRIGHT HOLDER>

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
```

MIT No Attribution License

Author	Roman Mamedov, Amazon Web Services
Published	28 March 2018
SPDX identifier	MIT-0
Debian FSG compatible	n/a
FSF approved	n/a
OSI approved	Yes[19]
GPL compatible	Yes
Copyleft	No
Linking from code with a	Yes

```
whom the Software is furnished to do so.

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.
```

different licence	
Website	https://github.com/aws/mit-0

Other variations

The SPDX License List contains extra MIT license variations. Examples include:[1]

- `MIT-advertising`, a variation with an additional advertising clause.

There is also the Anti-Capitalist Software License (ACSL),[22] built off of the MIT license. The ACSL is not OSI-approved, nor does it qualify as a free software license as defined by the FSF, since it limits the permissions granted to individuals and organizations that do not operate under capitalist structures, like non-profits and cooperatives.

Ambiguity and variants

The name "MIT License" is potentially ambiguous. The Massachusetts Institute of Technology has used many licenses for software since its creation; for example, MIT offers four licensing options for the FFTW[23] C source code library, one of which is the GPL v2.0 and the other three of which are not open-source. The term "MIT License" has also been used to refer to the **Expat License** (used for the XML parsing library Expat) and to the **X11 License** (also called "**MIT/X Consortium License**"; used for X Window System by the MIT X Consortium).[3] Furthermore, the "MIT License" as published by the Open Source Initiative is the same as the Expat License.[14] Due to this differing use of terms, some prefer to avoid the name "MIT License".[7] The Free Software Foundation argues that the term is misleading and ambiguous, and recommends against its use.[3]

The X Consortium was dissolved late in 1996, and its assets transferred to The Open Group,[24] which released X11R6 initially under the same license. The X11 License[4] and the X11R6 "MIT License" chosen for ncurses by the Free Software Foundation[25] both include the following clause, absent in the Expat License:[3]

> Except as contained in this notice, the name(s) of the above copyright holders shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization.

As of 2020, the successor to the X Window System is the X.Org Server, which is licensed under what is effectively the common MIT license, according to the X.org licensing page:[26]

> The X.Org Foundation has chosen the following format of the MIT License as the preferred format for code included in the X Window System distribution. This is a slight variant of the common MIT license form published by the Open Source Initiative

The "slight variant" is the addition of the phrase "(including the next paragraph)" to the second paragraph of the license text, resulting in: "The above copyright notice and this permission notice (including the next paragraph) shall be included in all copies or substantial portions of the Software." This inclusion clarifies that the liability paragraph must also be included for the conditions of the license to be met.[26]

The license-management features at popular source code repository GitHub, as well as its "Choose a License" service, do not differentiate between MIT/Expat license variants. The text of the Expat variant is presented as simply the "MIT License" (represented by the metadata tag `mit`).[27][28]

Comparison to other licenses

BSD

The original BSD license also includes a clause requiring all advertising of the software to display a notice crediting its authors. This "advertising clause" (since disavowed by UC Berkeley[29]) is present in the modified MIT License used by XFree86.

The University of Illinois/NCSA Open Source License combines text from both the MIT and BSD licenses; the license grant and disclaimer are taken from the MIT License.

The ISC license contains similarities to both the MIT and simplified BSD licenses, the biggest difference being that language deemed unnecessary by the Berne Convention is omitted.[30][31]

GNU General Public License

The GPL is explicit about the patent rights an owner grants when the code or derivative work is distributed,[32] while the MIT license does not discuss patents. Moreover, the GPL license impacts derivative works, but the MIT license does not.

Relation to patents

Like the BSD license, the MIT license does not include an express patent license although some commentators[33][34] state that the grant of rights covers all potential restrictions including patents. Both the BSD and the MIT licenses were drafted before the patentability of software was generally recognized under US law.[35] The Apache License version 2.0[3] is a similarly permissive license that includes an explicit contributor's patent license. Of specific relevance to US jurisdictions, the MIT license uses the terms "sell" and "use" that are also used in defining the rights of a patent holder in Title 35 of the United States Code section 154. This has been construed by some commentators[36][37] as an unconventional but implicit license in the US to use any underlying patents.

Origins

One of the originators of the MIT license, computer scientist Jerry Saltzer, has published his recollections of its early development, along with documentary evidence.[38][7]

Reception

As of 2020, according to WhiteSource Software[39] the MIT license was used in 27% of four million open source packages. As of 2015, according to Black Duck Software[40] and a 2015 blog[11] from GitHub, the MIT license was the most popular open-source license, with the GNU GPLv2 coming second in their sample of repositories.

See also

Free and open-source software portal

- Comparison of free and open-source software licenses
- ISC license—similar to the MIT license, but with language deemed unnecessary removed
- Category:Software using the MIT license

References

1. "SPDX License List" (https://spdx.org/licenses/). *spdx.org*. SPDX Working Group.
2. "License information" (https://www.debian.org/legal/licenses/). *The Debian Project*. Software in the Public Interest (published July 12, 2017). 1997–2017. Archived (https://web.archive.org/web/20170720142826/https://www.debian.org/legal/licenses/) from the original on July 20, 2017. Retrieved July 20, 2017. "This page presents the opinion of some debian-legal contributors on how certain licenses follow the Debian Free Software Guidelines (DFSG). ... Licenses currently found in Debian main include: ... Expat/MIT-style licenses ..."
3. "Various Licenses and Comments about Them" (https://www.gnu.org/licenses/license-list.en.html#Expat). *The GNU Project*. Free Software Foundation (published April 4, 2017). 2014–2017. Expat License. Archived (https://web.archive.org/web/20170720140022/https://www.gnu.org/licenses/license-list.en.html) from the original on July 20, 2017. Retrieved July 20, 2017. "This is a lax, permissive non-copyleft free software license, compatible with the GNU GPL. It is sometimes ambiguously referred to as the MIT License."
4. "Various Licenses and Comments about Them" (https://www.gnu.org/licenses/license-list.en.html#X11License). *The GNU Project*. Free Software Foundation (published April 4, 2017). 2014–2017. X11 License. Archived (https://web.archive.org/web/20170720140022/https://www.gnu.org/licenses/license-list.en.html) from the original on July 20, 2017. Retrieved July 20, 2017. "This is a lax permissive non-copyleft free software license, compatible with the GNU GPL. ... This license is sometimes called the MIT license, but that term is misleading, since MIT has used many licenses for software."

5. "Licenses by Name" (https://opensource.org/licenses/alphabetical). *Open Source Initiative*. n.d. Archived (https://web.archive.org/web/20170720141629/https://opensource.org/licenses/alphabetical) from the original on July 20, 2017. Retrieved July 20, 2017. "The following licenses have been approved by the OSI. ... MIT License (MIT) ..."
6. Rosen, Lawrence E. (2005). *Open Source Licensing: Software Freedom and Intellectual Property Law* (https://www.worldcat.org/oclc/56012651). Upper Saddle River, NJ: Prentice Hall PTR. ISBN 0-13-148787-6. OCLC 56012651 (https://search.worldcat.org/oclc/56012651).
7. Haff, Gordon. "The mysterious history of the MIT License" (https://opensource.com/article/19/4/history-mit-license). *opensource.com*. Retrieved July 30, 2019. "The date? The best single answer is probably 1987. But the complete story is more complicated and even a little mysterious. [...] Precursors from 1985. The X Consortium or X11 License variant from 1987. Or the Expat License from 1998 or 1999."
8. Hanwell, Marcus D. (January 28, 2014). "Should I use a permissive license? Copyleft? Or something in the middle?" (http://opensource.com/business/14/1/what-license-should-i-use-open-source-project). opensource.com. Retrieved May 30, 2015. "Permissive licensing simplifies things One reason the business world, and more and more developers [...], favor permissive licenses is in the simplicity of reuse. The license usually only pertains to the source code that is licensed and makes no attempt to infer any conditions upon any other component, and because of this there is no need to define what constitutes a derived work. I have also never seen a license compatibility chart for permissive licenses; it seems that they are all compatible."
9. "Licence Compatibility and Interoperability" (https://web.archive.org/web/20150617130550/https://joinup.ec.europa.eu/software/page/licence_compatibility_and_interoperability). *Open-Source Software*. Joinup. Archived from the original (https://joinup.ec.europa.eu/software/page/licence_compatibility_and_interoperability) on June 17, 2015. Retrieved May 30, 2015. "The licences for distributing free or open source software (FOSS) are divided in two families: permissive and copyleft. Permissive licences (BSD, MIT, X11, Apache, Zope) are generally compatible and interoperable with most other licences, tolerating to merge, combine or improve the covered code and to re-distribute it under many licences (including non-free or "proprietary")."
10. "Paid software includes MIT licensed library, does that put my app under MIT too?" (https://softwareengineering.stackexchange.com/questions/264700/paid-software-includes-mit-licensed-library-does-that-put-my-app-under-mit-too). *Software Engineering Stack Exchange*. Retrieved July 21, 2021.
11. Balter, Ben (March 9, 2015). "Open source license usage on GitHub.com" (https://github.com/blog/1964-license-usage-on-github-com). *The GitHub Blog*. Retrieved November 21, 2015. "1 MIT 44.69%, 2 Other 15.68%"
12. "GitHub Innovation Graph" (https://innovationgraph.github.com/global-metrics/licenses). *innovationgraph.github.com*. Retrieved January 8, 2025.
13. "MIT License" (https://spdx.org/licenses/MIT.html). SPDX. Archived (https://web.archive.org/web/20240508230029/https://spdx.org/licenses/MIT.html) from the original on May 8, 2024.
14. "The MIT License" (http://opensource.org/licenses/MIT). Open Source Initiative. Retrieved December 5, 2010.
15. "MIT License Explained in Plain English - TLDRLegal" (https://tldrlegal.com/license/mit-license). *tldrlegal.com*. Retrieved July 7, 2021.
16. "3.3. X Consortium", *3. X/MIT Licenses* (http://www.xfree86.org/4.0/LICENSE3.html#6), The XFree86 Project, March 2004
17. "X11 License" (https://spdx.org/licenses/X11.html). *spdx.org*. SPDX Working Group.
18. "X11 License Explained in Plain English - TLDRLegal" (https://web.archive.org/web/20210303122101/https://tldrlegal.com/license/x11-license). *tldrlegal.com*. Archived from the original (https://tldrlegal.com/license/x11-license) on March 3, 2021. Retrieved March 26, 2021.

19. Chestek, Pamela (August 5, 2020). "[License-review] Request for Legacy Approval of MIT No Attribution License" (https://lists.opensource.org/pipermail/license-review_lists.opensource.org/2020-August/004915.html).
20. "MIT No Attribution" (https://spdx.org/licenses/MIT-0.html). *spdx.org*. SPDX Working Group.
21. Langel, Tobie (May 15, 2020). "[License-review] Request for Legacy Approval of MIT No Attribution License" (https://lists.opensource.org/pipermail/license-review_lists.opensource.org/2020-May/004856.html).
22. "The Anti-Capitalist Software License" (https://anticapitalist.software/). *anticapitalist.software*. Retrieved May 8, 2024.
23. "FFTW - Fastest Fourier Transform in the West" (https://tlo.mit.edu/technologies/fftw-fastest-fourier-transform-west). Massachusetts Institute of Technology. Retrieved July 17, 2020.
24. Dickey, Thomas E. "Copyrights/comments" (https://invisible-island.net/personal/copyrights.html#bad_dating). Retrieved October 6, 2020.
25. Dickey, Thomas E. "NCURSES — Frequently Asked Questions (FAQ)" (http://invisible-island.net/ncurses/ncurses.faq.html#is_it_gpl).
26. "Licenses" (https://www.x.org/archive/current/doc/xorg-docs/License.html). *X.Org Foundation*.
27. "Licensing a repository" (https://docs.github.com/en/free-pro-team@latest/github/creating-cloning-and-archiving-repositories/licensing-a-repository). *GitHub Docs*.
28. "MIT License" (https://choosealicense.com/licenses/mit/). *ChooseALicense.com*. GitHub. September 5, 2022.
29. "To All Licensees, Distributors of Any Version of BSD" (ftp://ftp.cs.berkeley.edu/pub/4bsd/README.Impt.License.Change). University of California, Berkeley. July 22, 1999. Retrieved November 15, 2006.
30. "Copyright Policy" (http://www.openbsd.org/policy.html). *OpenBSD*. Retrieved June 6, 2016. "The ISC copyright is functionally equivalent to a two-term BSD copyright with language removed that is made unnecessary by the Berne convention."
31. de Raadt, Theo (March 21, 2008). "Re: BSD Documentation License?" (https://marc.info/?l=openbsd-misc&m=120618313520730&w=2). *openbsd-misc* (Mailing list).
32. "Patents and GPLv3 - FSFE" (https://fsfe.org/activities/gplv3/patents-and-gplv3.html). *FSFE - Free Software Foundation Europe*. Retrieved December 1, 2020.
33. "Why so little love for the patent grant in the MIT License?" (https://web.archive.org/web/20210123091649/https://opensource.com/article/18/3/patent-grant-mit-license). January 23, 2021. Archived from the original (https://opensource.com/article/18/3/patent-grant-mit-license) on January 23, 2021. Retrieved January 23, 2021.
34. "Free and open source software and your patents" (https://web.archive.org/web/20200503103939/http://oss-watch.ac.uk/resources/fossandpatents). May 3, 2020. Archived from the original (http://oss-watch.ac.uk/resources/fossandpatents) on May 3, 2020. Retrieved May 3, 2020.
35. Stern and Allen, Open Source Licensing, p. 495 in Understanding the Intellectual Property License 2013 (Practicing Law Institute 2013)
36. "The MIT License, Line by Line" (https://web.archive.org/web/20200503110018/https://writing.kemitchell.com/2016/09/21/MIT-License-Line-by-Line.html). May 3, 2020. Archived from the original (https://writing.kemitchell.com/2016/09/21/MIT-License-Line-by-Line.html) on May 3, 2020. Retrieved May 3, 2020.
37. Christian H. Nadan (2009), "Closing the Loophole: Open Source Licensing & the Implied Patent License" (https://download.pli.edu/WebContent/chbs/185480/185480_Chapter22_Adv_Licensing_Agreements_2017_Vol_02_CC121601854800207140.htm), *The Computer & Internet Lawyer*, **26** (8), Aspen Law & Business, "By using patent terms like "deal in", "use", and "sell", the MIT license grant is more likely to be deemed to include express patent rights than the BSD license."

38. Saltzer, Jerome H (November 18, 2020). "The origin of the "MIT license" " (https://doi.org/10.1109%2FMAHC.2020.3020234). *IEEE Annals of the History of Computing*. **42** (4): 94–98. doi:10.1109/MAHC.2020.3020234 (https://doi.org/10.1109%2FMAHC.2020.3020234). ISSN 1934-1547 (https://search.worldcat.org/issn/1934-1547).
39. "Open Source Licenses in 2020: Trends and Predictions" (https://web.archive.org/web/20200503111426/https://resources.whitesourcesoftware.com/blog-whitesource/top-open-source-licenses-trends-and-predictions). May 3, 2020. Archived from the original (https://resources.whitesourcesoftware.com/blog-whitesource/top-open-source-licenses-trends-and-predictions) on May 3, 2020. Retrieved May 3, 2020.
40. "Top 20 licenses" (https://web.archive.org/web/20130904051147/http://www.blackducksoftware.com/resources/data/top-20-licenses). Black Duck Software. November 19, 2015. Archived from the original (http://www.blackducksoftware.com/resources/data/top-20-licenses) on September 4, 2013. Retrieved November 19, 2015. "1. MIT license 24%, 2. GNU General Public License (GPL) 2.0 23%"

Further reading

- Mitchell, Kyle E. (September 21, 2016). "The MIT License, Line by Line" (https://writing.kemitchell.com/2016/09/21/MIT-License-Line-by-Line.html). */dev/lawyer*. Archived (https://web.archive.org/web/20160928235700/https://writing.kemitchell.com/2016/09/21/MIT-License-Line-by-Line.html) from the original on September 28, 2016. Retrieved September 28, 2016.

External links

- MIT License variants (https://fedoraproject.org/wiki/Licensing:MIT?rd=Licensing/MIT)
- The MIT License template (http://www.opensource.org/licenses/MIT) (Open Source Initiative official site)
- Expat License (http://www.jclark.com/xml/copying.txt) Archived (https://web.archive.org/web/20220721145318/http://www.jclark.com/xml/copying.txt) July 21, 2022, at the Wayback Machine
- X11 License (https://xfree86.org/3.3.6/COPYRIGHT2.html#3)

Mozilla Public License

The **Mozilla Public License** (**MPL**) is a free and open-source weak copyleft license for most Mozilla Foundation software such as Firefox and Thunderbird.[8] The MPL is developed and maintained by Mozilla,[9] which seeks to balance the concerns of both open-source and proprietary developers. It is distinguished from others as a middle ground between the permissive software BSD-style licenses and the GNU General Public License.[10] As such, it allows the integration of MPL-licensed code into proprietary codebases, as long as the MPL-licensed components remain accessible under the terms of the MPL.

MPL has been used by others, such as Adobe to license their Flex product line,[11] and The Document Foundation to license LibreOffice 4.0 (also on LGPL 3+).[12][13] Version 1.1 was adapted by several projects to form derivative licenses like Sun Microsystems' Common Development and Distribution License.[14] It has undergone two revisions:[15] the minor update 1.1, and a major update version 2.0[16] nearing the goals of greater simplicity and better compatibility with other licenses.[17]

Mozilla Public License

Author	Mozilla Foundation[1]
Latest version	2.0[1]
Publisher	Mozilla Foundation[1]
Published	January 3, 2012[1]
SPDX identifier	MPL-2.0 MPL-1.1 MPL-1.0 (see list for more[2])
Debian FSG compatible	Yes[3]
FSF approved	Yes[4]
OSI approved	Yes[5]
GPL compatible	2.0: Yes[4] (by default, unless marked as "Incompatible With Secondary Licenses") 1.1: No[6]
Copyleft	Yes, file-based[7]
Linking from code with a different licence	Yes
Website	www.mozilla.org/MPL (https://www.mozilla.org/MPL)

Terms

The MPL defines rights as passing from "contributors", who create or modify source code, through an optional auxiliary distributor (itself a licensee), to the licensee. It grants liberal copyright and patent licenses allowing for free use, modification, distribution, and "exploit[ation]" of the work, but does not grant the licensee any rights to a contributor's trademarks.[7] These rights will terminate if the licensee fails to comply with the license's terms and conditions, but a violating licensee who returns to compliance regains its rights, and even receiving written notice from a contributor will result in losing rights to that contributor's code only. A patent retaliation clause, similar to that of the Apache License, is included to protect an auxiliary distributor's further recipients against patent trolling. The contributors disclaim warranty and liability, but allow auxiliary distributors to offer such things on their own behalf.

In exchange for the rights granted by license, the licensee must meet certain responsibilities concerning the distribution of licensed source code. Covered source code files must remain under the MPL, and distributors "may not attempt to alter or restrict recipients' rights" to it. The MPL treats the source code file as the boundary between MPL-licensed and proprietary parts, meaning that all or none of the code in a given source file falls under the MPL. An executable consisting solely of MPL-covered files may be sublicensed, but the licensee must ensure access to or provide all the source code within it. Recipients can combine licensed source code with other files under a different, even proprietary license, thereby forming a "larger work" which can be distributed under any terms, but again the MPL-covered source files must be made freely available.[7] This makes the MPL a compromise between the MIT or BSD licenses, which permit all derived works to be relicensed as proprietary, and the GPL, which requires the derived work as a whole to be licensed under the GPL. By allowing proprietary modules in derived projects while requiring core files to remain open source, the MPL is designed to motivate both businesses and the open-source community to help develop core software.[18]

The one exception to covered source files remaining under the MPL occurs when code under version 2.0 or later is combined with separate code files under the GNU GPL, GNU Lesser GPL (LGPL), or Affero GPL (AGPL). In this case, the program as a whole will be under the chosen GNU license, but the MPL-covered files will be dual-licensed, so that recipients can choose to distribute them under that GNU License or the MPL.[4] The initial author of MPL code may choose to opt out of this GPL compatibility by adding a notice to its source files.[7]

It is explicitly granted that MPL-covered code may be distributed under the terms of the license version under which it was received or any later version.[1]:10.2 If code under version 1.0 or 1.1 is upgraded to version 2.0 by this mechanism, the 1.x-covered code must be marked with the aforementioned GPL-incompatible notice. The MPL can be modified to form a new license, provided that said license does not refer to Mozilla or Netscape.

History

Version 1.0 of the MPL was written by Mitchell Baker in 1998 while working as a lawyer at Netscape Communications Corporation.[19] Netscape was hoping that an open-source strategy for developing its own Netscape web browser would allow it to compete better with Microsoft's browser, Internet Explorer.[20] To cover the browser's code, the company drafted a license known as the Netscape Public License (NPL), which included a clause allowing even openly developed code to be theoretically relicensed as proprietary.[21]

However, at the same time, Baker developed a second license similar to the NPL. It was called the Mozilla Public License after Netscape's project name for the new open-source codebase, and, although it was originally only intended for software that supplemented core modules covered by the NPL, it would become much more popular than the NPL and eventually earn approval from the Open Source Initiative.[22]

Less than a year later, Baker and the Mozilla Organization would make some changes to the MPL, resulting in version 1.1, a minor update.[23] This revision was done through an open process that considered comments from both institutional and individual contributors. The primary goals were to clarify terms regarding patents and allow for multiple licensing. This last feature was meant to encourage cooperation with developers that preferred stricter licenses like the GPL.[24] Not only would many projects derive their own licenses from this version, but its structure, legal precision, and explicit terms for patent rights would strongly influence later revisions of popular licenses like the GPL (version 3).[14]

Both versions 1.0 and 1.1 are incompatible with the GPL, which led the Free Software Foundation to discourage using version 1.1.[6] For these reasons, earlier versions of Firefox were released under multiple licenses: the MPL 1.1, GPL 2.0, and LGPL 2.1.[25] Some old software, such as the Mozilla Application Suite, is still under the three licenses. Therefore, in early 2010, after more than a decade without modification, an open process for creating version 2.0 of the MPL began. Over the next 21 months, the MPL was not only changed to make the license clearer and easier to apply, but also to achieve compatibility with the GPL and Apache licenses.[17][26] The revision team was overseen by Baker and led by Luis Villa with key support from Gervase Markham and Harvey Anderson. They would publish three alpha drafts, two beta drafts, and two release candidates for comment before releasing the final draft of version 2.0 on January 3, 2012.[17]

Notable users

- Apache Flex (Formerly known as Adobe Flex)[11]
- Armadillo
- Boulder,[27] the software that runs the Let's Encrypt certificate authority
- Cairo[28]
- Celtx[29]
- Cemu
- Eigen[30]
- H2 (DBMS)[31]
- Internet Systems Consortium[32]
- LibreOffice[13]
- Mozilla Firefox
- OpenMRS[33]
- OpenTofu
- Syncthing[34]
- Servo[35]
- Brave Browser[36]
- MonetDB (marked as "Incompatible With Secondary Licenses")
- RabbitMQ[37]
- VLC (iOS Version)
- ZeroMQ (since version 4.3.5)[38]

Licenses based on pre-MPL 2.0

- AROS Public License 1.1 (based on MPL 1.1)
- Common Development and Distribution License
- Common Public Attribution License

- Erlang Public License 1.1[39] (modified MPL 1.0,[40] where "disagreements are settled under Swedish law in English"[41])
- Firebird's Initial Developer's Public License (based on MPL v1.1)[42]
- Sun Public License
- Yahoo! Public License
- Openbravo's Openbravo Public License (based on MPL 1.1)[43]
- Netscape Public License 1.1 (based on MPL 1.1)[44]

See also

Free and open-source software portal

- Software using the Mozilla Public License (category)
- Comparison of free and open-source software licenses

References

1. "Mozilla Public License, version 2.0" (https://www.mozilla.org/MPL/2.0/). Mozilla Foundation. Retrieved February 28, 2012.
2. "SPDX License List" (https://spdx.org/licenses/). *spdx.org*.
3. "Mozilla Public License (MPL)" (http://wiki.debian.org/DFSGLicenses#MozillaPublicLicense.28MPL.29). *The Big DFSG-compatible Licenses*. Debian Project. Retrieved June 6, 2009.
4. "Mozilla Public License (MPL) version 2.0" (https://www.gnu.org/licenses/license-list.html#MPL-2.0). *Various Licenses and Comments about Them*. Free Software Foundation. Retrieved January 3, 2012.
5. "Open Source Licenses" (http://www.opensource.org/licenses/MPL-2.0). Open Source Initiative. December 19, 2011. Retrieved January 7, 2012. "*Mozilla Public License 2.0 (MPL-2.0)*"
6. "Mozilla Public License (MPL) version 1.1" (https://www.gnu.org/licenses/license-list.html#MPL). *Various Licenses and Comments about Them*. Free Software Foundation. Retrieved August 12, 2016.
7. "MPL 2.0 FAQ" (https://www.mozilla.org/MPL/2.0/FAQ.html). Mozilla Foundation. Retrieved October 14, 2020.
8. "Mozilla Foundation License Policy" (https://www.mozilla.org/MPL/license-policy.html). Mozilla Foundation. Retrieved February 29, 2012.
9. "Open Source Software: a legal guide" (https://web.archive.org/web/20200730221407/https://www.lawgives.com/guide/55ce85817777775593000144/Open-Source-Software-a-legal-guide). *LawGives*. Archived from the original (https://www.lawgives.com/guide/55ce85817777775593000144/Open-Source-Software-a-legal-guide) on July 30, 2020. Retrieved September 8, 2015.
10. Andrew Laurent (2004). *Understanding Open Source and Free Software Licensing* (https://books.google.com/books?id=XhXQZgn_QxAC&q=9780596005818&pg=PT73). O'Reilly Media, Inc. p. 62. ISBN 978-0-596-00581-8.
11. "Adobe Flex FAQ: Licensing" (http://www.adobe.com/products/flex/faq.html#licensing). Adobe Systems. Retrieved February 29, 2012.
12. "The meaning of the 4.0" (http://standardsandfreedom.net/index.php/2013/01/24/the-meaning-of-the-4-0/). January 24, 2013.

13. "Licenses" (https://www.libreoffice.org/download/license/). LibreOffice.
14. Fontana, Richard (January 9, 2012). "The new MPL" (http://opensource.com/law/12/1/the-new-mpl). Retrieved March 1, 2012.
15. "Historical Licensing Documents" (https://www.mozilla.org/MPL/historical.html). Mozilla Foundation. Retrieved February 29, 2012.
16. "Mozilla Public License 2.0 (MPL-2.0) | Open Source Initiative" (https://opensource.org/licenses/MPL-2.0). *opensource.org*. December 19, 2011. Retrieved March 9, 2022.
17. "About MPL 2.0: Revision Process and Changes FAQ" (https://www.mozilla.org/MPL/2.0/Revision-FAQ.html). Mozilla Foundation. Retrieved February 29, 2012.
18. O'Hara, Keith J.; Kay, Jennifer S. (February 2003). "Open source software and computer science education" (http://elvis.rowan.edu/~kay/papers/OSSEdu.pdf) (PDF). *Journal of Computing Sciences in Colleges*. **18** (3). Consortium for Computing Sciences in Colleges: 3–4. ISSN 1937-4771 (https://search.worldcat.org/issn/1937-4771). Retrieved February 29, 2012.
19. Rosen, Lawrence (2004). "7. The Mozilla Public License (MPL)". *Open Source Licensing – Software Freedom and Intellectual Property Law* (https://archive.org/details/opensourcelicens00rose_0). Prentice Hall PTR. ISBN 0-13-148787-6.
20. Andreessen, Marc (April 18, 2005). "The Time 100: Mitchell Baker: The "Lizard Wrangler" ". *Time Magazine*. **165** (16). Time. ISSN 0040-781X (https://search.worldcat.org/issn/0040-781X). OCLC 1311479 (https://search.worldcat.org/oclc/1311479).
21. "Netscape Public License" (https://website-archive.mozilla.org/www.mozilla.org/mpl/MPL/NPL/1.1/). Netscape Communications. Archived from the original (https://www.mozilla.org/mpl/MPL/NPL/1.1) on August 27, 2015. Retrieved August 16, 2016.
22. Wilson, Rowan (November 15, 2011). "The Mozilla Public License - An Overview" (http://www.oss-watch.ac.uk/resources/mpl.xml). *OSS-Watch*. University of Oxford. Retrieved February 29, 2012.
23. Hecker, Frank (April 2, 1999). "Mozilla at One: A Look Back and Ahead" (https://web.archive.org/web/20080628174020/http://www.mozilla.org/mozilla-at-one.html). Archived from the original (https://www.mozilla.org/mozilla-at-one.html) on June 28, 2008. Retrieved March 1, 2012.
24. "NPL Version 1.0M FAQ" (https://web.archive.org/web/20110105015147/http://www.mozilla.org/MPL/NPL-1.0M-FAQ.html). September 24, 1999. Archived from the original (https://www.mozilla.org/MPL/NPL-1.0M-FAQ.html) on January 5, 2011. Retrieved March 1, 2012.
25. "Mozilla Relicensing FAQ" (https://web.archive.org/web/20090505215642/http://www.mozilla.org/MPL/relicensing-faq.html). Mozilla Foundation. August 14, 2007. Archived from the original on May 5, 2009. Retrieved February 28, 2012.
26. Villa, Luis (August 29, 2011). "The Mozilla Public License - almost 2.0 (part 1)" (https://opensource.com/law/11/8/mozilla-public-license-almost-20-part-1). Opensource.com. Retrieved August 16, 2016.
27. "Boulder - an ACME CA" (https://github.com/letsencrypt/boulder). *GitHub*. October 19, 2021.
28. "cairographics.org" (https://www.cairographics.org/).
29. "Celtx - Policies" (https://www.celtx.com/license.html).
30. "Eigen" (http://eigen.tuxfamily.org/index.php?title=Main_Page#License). *eigen.tuxfamily.org*. Retrieved February 23, 2017.
31. "License" (http://www.h2database.com/html/license.html).
32. "Kea to be released under Mozilla Public License 2.0 - Internet Systems Consortium" (https://www.isc.org/blogs/kea-license-2-0/). December 8, 2015.
33. "OpenMRS Licensing Moves to MPLv2 - OpenMRS" (http://openmrs.org/2013/04/openmrs-licensing-moves-to-mplv2/).

34. "syncthing/syncthing" (https://github.com/syncthing/syncthing/blob/master/LICENSE). *GitHub*. October 20, 2021.
35. "servo/servo" (https://github.com/servo/servo/blob/master/LICENSE). *GitHub*. Retrieved November 8, 2020.
36. "brave/brave-browser" (https://github.com/brave/brave-browser/blob/master/LICENSE). *GitHub*. Retrieved November 8, 2020.
37. "Mozilla Public License — RabbitMQ" (https://www.rabbitmq.com/mpl.html). *www.rabbitmq.com*. Retrieved August 18, 2021.
38. "ØMQ Licensing - zeromq" (http://wiki.zeromq.org/area:licensing). June 6, 2023. Retrieved December 5, 2024.
39. "Erlang Programming Language" (http://www.erlang.org/about.html).
40. "ERLANG PUBLIC LICENSE: Version 1.1" (http://www.erlang.org/EPLICENSE). "This Erlang License is a derivative work of the Mozilla Public License, Version 1.0."
41. "English translation of the Erlang Public License legal text" (https://web.archive.org/web/20120607165043/http://www.erlang.org/static/download/EPL1x0-explained.html). Archived from the original (http://www.erlang.org/static/download/EPL1x0-explained.html) on June 7, 2012.
42. "Initial Developer's Public License" (http://www.firebirdsql.org/en/initial-developer-s-public-license-version-1-0/).
43. "Openbravo Public License" (https://web.archive.org/web/20130820143736/http://openbravo.com/content/openbravo-public-license). Archived from the original (http://www.openbravo.com/content/openbravo-public-license) on August 20, 2013. Retrieved January 5, 2019.
44. "Netscape Public License (NPL), versions 1.0 and 1.1" (https://www.gnu.org/licenses/license-list.html#NPL). *Various Licenses and Comments about Them*. Free Software Foundation. Retrieved April 13, 2025.

External links

- Official website (https://www.mozilla.org/MPL/)
  - Mozilla Public License Version 2.0 (https://www.mozilla.org/MPL/2.0/)
    - Comparison between versions 2.0 and 1.1 (https://www.mozilla.org/en-US/MPL/2.0/differences/)
  - Mozilla Public License Version 1.1 (https://www.mozilla.org/MPL/1.1/)
  - Mozilla Public License Version 1.0 (https://www.mozilla.org/MPL/1.0/)

Python License

The **Python License** is a deprecated permissive computer software license created by the Corporation for National Research Initiatives (CNRI). It was used for versions 1.6 and 2.0 of the Python programming language, both released in the year 2000.

The Python License is similar to the BSD License and, while it is a free software license, its wording in some versions meant that it was incompatible with the GNU General Public License (GPL) used by a great deal of free software including the Linux kernel. For this reason CNRI retired the license in 2001, and the license of current releases is the Python Software Foundation License.[3]

Python License

Author	Corporation for National Research Initiatives
Latest version	2.11 and newer
SPDX identifier	Python-2.0, Python-2.0.1
FSF approved	Yes[1]
OSI approved	Yes
GPL compatible	Some versions[1][2]
Copyleft	No

Origin

Python was created by Guido van Rossum and the initial copyright was held by his employer, the Centrum Wiskunde & Informatica (CWI). During this time Python was distributed under a GPL-compatible variant of the Historical Permission Notice and Disclaimer license.[4] CNRI obtained ownership of Python when Van Rossum became employed there, and after some years they drafted a new license for the language.

Retirement

The Python License includes a clause stating that the license is governed by the State of Virginia, United States. The Python Software Foundation License; Python 1.6.1 differs from Python 1.6 only in some minor bug fixes and new GPL-compatible licensing terms.

References

1. "License of Python 1.6a2 and earlier versions" (https://www.gnu.org/licenses/license-list.html#Python1.6a2). Free Software Foundation. List of licenses.
"License of Python 1.6b1 through 2.0 and 2.1" (https://www.gnu.org/licenses/license-list.html#PythonOld). Free Software Foundation. List of licenses.
"License of Python 2.0.1, 2.1.1, and newer versions" (https://www.gnu.org/licenses/license-list.html#Python). Free Software Foundation. List of licenses.
2. History and License (https://docs.python.org/3/license.html) Python

3. "Python 3.3.4 license" (https://www.python.org/download/releases/3.3.4/license). Retrieved April 15, 2014.
4. "Python 2.5 license (with historical notes)" (https://www.python.org/download/releases/2.5/license). Python.org. Retrieved March 31, 2008.

Python Software Foundation License

The **Python Software Foundation License** (**PSFL**) is a BSD-style, permissive software license which is compatible with the GNU General Public License (GPL).[1] Its primary use is for distribution of the Python project software and its documentation.[3] Since the license is permissive, it allows proprietization of the derivations. The PSFL is listed as approved on both FSF's approved licenses list,[1] and OSI's approved licenses list.

Python Software Foundation License

SPDX identifier	PSF-2.0
Debian FSG compatible	Yes
FSF approved	Yes[1]
OSI approved	Yes[2]
GPL compatible	Yes[1]
Copyleft	No

This license is also known as "Python License 2.0.1".[4]

In 2000, Python *(specifically version 2.1)* was briefly available under the Python License, which is incompatible with the GPL. The reason given for this incompatibility by Free Software Foundation was that "*this Python license is governed by the laws of the 'State of Virginia', in the USA*", which the GPL does not permit.[5]

Guido van Rossum

Guido van Rossum, Python's creator, was awarded the 2001 Free Software Foundation Award for the Advancement of Free Software[6] for changing the license to fix this incompatibility.

See also

Free and open-source software portal

- Python Software Foundation
- Software using the Python Software Foundation license (category)

References

External links

- The Python Software Foundation License (https://www.python.org/psf/license/)

1. Stallman, Richard. "Various Licenses and Comments about Them, new Python license" (https://web.archive.org/web/20211227002639/https://www.gnu.org/licenses/license-list.html#Python). Free Software Foundation. Archived from the original (https://www.gnu.org/licenses/license-list.html#Python) on December 27, 2021. Retrieved July 7, 2007.
2. "License Search" (https://opensource.org/license/python-2-0). Open Source Initiative. OSI Approved Licenses.
3. "Terms and conditions for accessing or otherwise using Python" (https://web.archive.org/web/20211225221920/https://docs.python.org/3/license.html#terms-and-conditions-for-accessing-or-otherwise-using-python). Python Software Foundation. Archived from the original (https://docs.python.org/3/license.html#terms-and-conditions-for-accessing-or-otherwise-using-python) on December 25, 2021. Retrieved December 29, 2021.
4. "Various Licenses and Comments about Them" (https://www.gnu.org/licenses/license-list.html#Python). Free Software Foundation. Retrieved April 13, 2023.
"Python License 2.0.1" (https://spdx.org/licenses/). Software Package Data Exchange. SPDX License List.
5. Stallman, Richard. "Various Licenses and Comments about Them, old Python license" (https://www.gnu.org/licenses/license-list.html#PythonOld). Free Software Foundation. Retrieved July 7, 2007.
6. "2001 Free Software Awards" (https://web.archive.org/web/20210814060943/http://www.gnu.org/award/2001/2001.html). GNU Project - Free Software Foundation. Archived from the original (https://www.gnu.org/award/2001/2001.html) on August 14, 2021. Retrieved July 7, 2007.

Shared Source Initiative

The **Shared Source Initiative** (**SSI**) is a source-available software licensing scheme launched by Microsoft in May 2001.[1] The program includes a spectrum of technologies and licenses, and most of its source code offerings are available for download after eligibility criteria are met.[2]

Overview

Microsoft's Shared Source Initiative allows individuals and organizations to access Microsoft's source code for reference (e.g. when developing complementary systems), for review and auditing from a security perspective (mostly wanted by some large corporations and governments), and for development (academic institutions, OEMs, ODMs, IHVs, IBVs, ISVs, individual developers). For example, higher-level OEMs, such as Dell, HP and Nokia, can get more source code of Microsoft Windows.

As part of the framework, Microsoft released 5 licenses for general use. Two of them, Microsoft Public License and Microsoft Reciprocal License, have been approved by the Open Source Initiative as open source licenses[3][4] and are regarded by the Free Software Foundation as free software licenses.[5] Other shared source licenses are proprietary, and thus allow the copyright holder to retain tighter control over the use of their product.

Microsoft's Shared Source Initiative has been imitated by other companies such as RISC OS Open Ltd.[6]

Microsoft also uses specific licenses for some of their products, such as the Shared Source CLI License[7] and the Microsoft Windows Embedded CE 6.0 Shared Source License.[8]

Free and open-source licenses

The following licenses are considered open-source by the Open Source Initiative and free by the Free Software Foundation.

Microsoft Public License (Ms-PL)

This is the least restrictive of the Microsoft licenses and allows for distribution of *compiled* code for either commercial or non-commercial purposes under any license that complies with the Ms-PL. Redistribution of the source code itself is permitted only under the Ms-PL.[9] Initially titled *Microsoft Permissive License*, it was renamed to *Microsoft Public License* while being reviewed for approval by the Open Source Initiative (OSI).[10] The license was approved on October 12, 2007, along with the Ms-RL.[11] According to the Free Software Foundation, it is a free software license but not compatible with the GNU GPL.[5] Ms-PL provides a free and flexible licensing for developers using source codes under this license. However, the Ms-PL is a copyleft license because it requires the source code of software it governs to be distributed only under the same license (the Ms-PL).[12]

Microsoft Reciprocal License (Ms-RL)

This Microsoft license allows for distribution of derived code so long as the modified source files are included and retain the Ms-RL.[13] The Ms-RL allows those files in the distribution that do not contain code originally licensed under Ms-RL to be licensed according to the copyright holder's choosing. This is similar, but not the same as the CDDL, EPL or LGPL (GPL with a typical "linking exception"). Initially known as the *Microsoft Community License*, it was renamed during the OSI approval process.

On December 9, 2005, the Ms-RL license was submitted to the Open Source Initiative for approval by John Cowan.[14] OSI then contacted Microsoft and asked if they wanted OSI to proceed. Microsoft replied that they did not wish to be reactive and that they needed time to review such a decision.[15]

At the O'Reilly Open Source Convention in July 2007, Bill Hilf, director of Microsoft's work with open source projects, announced that Microsoft had formally submitted Ms-PL and Ms-RL to OSI for approval.[16] It was approved on October 12, 2007, along with the Ms-PL.[11] According to the Free Software Foundation, it is a free software license but not compatible with the GNU GPL.[5]

Restricted licenses

The following source-available software licenses have limitations that prevent them from being open-source according to the Open Source Initiative and free to the Free Software Foundation.

Microsoft Limited Public License (Ms-LPL)

This is a version of the Microsoft Public License in which rights are only granted to developers of Microsoft Windows-based software.[17] This license is not open source, as defined by the OSI, because the restriction limiting use of the software to Windows violates the stipulation that open-source licenses must be technology-neutral.[18] It is also considered to be non-free by the Free Software Foundation due to this restriction.[5]

Microsoft Limited Reciprocal License (Ms-LRL)

This is a version of the Microsoft Reciprocal License in which rights are only granted when developing software for a Microsoft Windows platform.[19] Like the Ms-LPL, this license is not open source because it is not technology-neutral[18] due to its restriction that licensed software must be used on Windows, and is also not considered free by the Free Software Foundation due to this restriction.[5]

Microsoft Reference Source License (Ms-RSL)

This is the most restrictive of the Microsoft Shared Source licenses. The source code is made available to view for reference purposes only, mainly to be able to view Microsoft classes source code while debugging.[20] Developers may not distribute or modify the code for commercial or non-commercial purposes.[21] The license has previously been abbreviated *Ms-RL*, but *Ms-RL* now refers to the *Microsoft Reciprocal License*.[13]

Criticism

Two specific shared source licenses are interpreted as free software and open source licenses by FSF and OSI. However, former OSI president Michael Tiemann considers the phrase "Shared Source" itself to be a marketing term created by Microsoft. He argues that it is "an insurgent term that distracts and dilutes the Open Source message by using similar-sounding terms and offering similar-sounding promises".[22]

The Shared Source Initiative has also been noted to increase the problem of license proliferation.[23]

See also

Free and open-source software portal

- Open Source Initiative
- Source-available software
- Software using the Microsoft Public License (category)

References

1. "Microsoft announces expansion of Shared Source Initiative" (https://www.geekzone.co.nz/content.asp?contentid=4253).
2. "Microsoft Shared Source Initiative" (https://cyber.harvard.edu/ilaw/brazil03/ms.html). *Berkman Klein Center*. Retrieved 24 Aug 2018.
3. "Microsoft Public License (MS-PL) – Open Source Initiative" (http://opensource.org/licenses/ms-pl.html). *opensource.org*.
4. "Microsoft Reciprocal License (MS-RL) – Open Source Initiative" (http://opensource.org/licenses/ms-rl.html). *opensource.org*.
5. "Various Licenses and Comments about Them" (https://www.gnu.org/licenses/license-list.html). *GNU Operating System*. Free Software Foundation. Retrieved 23 Jul 2018.
6. "RISC OS Open: Welcome" (http://www.riscosopen.org/). *www.riscosopen.org*.
7. "Shared Source Common Language Infrastructure License" (http://msdn.microsoft.com/MSDN-FILES/027/002/097/ShSourceCLILicense.htm).
8. Archiveddocs. "Windows Embedded" (http://msdn2.microsoft.com/en-us/embedded/bb190212.aspx). *msdn2.microsoft.com*.
9. "Microsoft Public License (Ms-PL)" (http://www.microsoft.com/opensource/licenses.mspx#Ms-PL). *Microsoft*.
10. Foley, Mary. "Microsoft gets the open-source licensing nod from the OSI" (https://www.zdnet.com/article/microsoft-gets-the-open-source-licensing-nod-from-the-osi/). *ZDNet*. Retrieved 22 May 2015.
11. "OSI Approves Microsoft License Submissions" (http://opensource.org/node/207). 2007-10-17. Retrieved 2013-08-08. "*Acting on the advice of the License Approval Chair, the OSI Board today approved the Microsoft Public License (Ms-PL) and the Microsoft Reciprocal License (Ms-RL). The decision to approve was informed by the overwhelming (though not unanimous) consensus from the open source community that these licenses satisfied the 10 criteria of the Open Source definition, and should therefore be approved.*"

12. "Microsoft Public License (MS-PL) | Open Source Initiative" (https://opensource.org/licenses/MS-PL). *opensource.org*. Retrieved 2020-08-28.
13. "Microsoft Reciprocal License (Ms-RL)" (http://www.microsoft.com/opensource/licenses.mspx#Ms-RL). *Microsoft*.
14. John Cowan (2005-12-09). "For Approval: Microsoft Community License" (http://www.crynwr.com/cgi-bin/ezmlm-cgi?3:msp:11322:cokmgmoknbgepfbongjn). license-discuss mailing list.
15. Peter Galli (2006-08-22). "Blogger Can't Tempt Microsoft To Drink OSI Kool-Aid" (http://www.eweek.com/article2/0,1895,2007088,00.asp). eWeek.
16. Tim O'Reilly (2007-07-26). "Microsoft to Submit Shared Source Licenses to OSI" (http://radar.oreilly.com/2007/07/microsoft-to-submit-shared-sou.html). O'Reilly Radar.
17. "Microsoft Limited Public License (Ms-LPL)" (http://www.microsoft.com/resources/sharedsource/licensingbasics/limitedpubliclicense.mspx). *Microsoft*.
18. "The Open Source Definition" (http://www.opensource.org/docs/osd). Open Source Initiative.
19. "Microsoft Limited Reciprocal License (Ms-LRL)" (http://www.microsoft.com/resources/sharedsource/licensingbasics/limitedreciprocallicense.mspx). *Microsoft*.
20. "Microsoft Reference Source License" (http://referencesource.microsoft.com/license.html). Microsoft. 2016-07-06. Retrieved 2016-07-06. "*"Reference use" means use of the software within your company as a reference, in read only form, for the sole purposes of debugging your products, maintaining your products, or enhancing the interoperability of your products with the software, and specifically excludes the right to distribute the software outside of your company.*"
21. "Microsoft Reference Source License" (http://referencesource.microsoft.com/license.html).
22. Michael Tiemann (2007-11-11). "Who Is Behind "Shared Source" Misinformation Campaign?" (http://opensource.org/node/225). Retrieved 2011-03-12. "*Shared source is a marketing term created and controlled by Microsoft. Shared source is not open source by another name. Shared source is an insurgent term that distracts and dilutes the Open Source message by using similar-sounding terms and offering similar-sounding promises. And to date, 'shared source' has been a marketing dud as far as Open Source is concerned.*"
23. Evaluating Microsoft's Shared Source licenses (https://arstechnica.com/information-technology/2007/08/evaluating-microsofts-shared-source-licenses/) Ars Technica, 2007

External links

- Official website (https://www.microsoft.com/en-us/sharedsource/)

Sleepycat Software

(Redirected from Sleepycat License)

Sleepycat Software, Inc. was the software company primarily responsible for maintaining the Berkeley DB packages from 1996 to 2006.[1]

Sleepycat Software, Inc.

Company type	Private company
Industry	Computer software
Genre	Database software
Founded	1997
Founder	Margo Seltzer and Keith Bostic
Defunct	2006
Fate	Acquired
Successor	Oracle Corporation
Headquarters	Lincoln, Massachusetts, U.S.
Key people	Michael Olson (CEO)
Products	Berkeley DB

Company

Berkeley DB is freely-licensed database software originally developed at the University of California, Berkeley for 4.4BSD Unix. Developers from that project founded Sleepycat in 1996 to provide commercial support after a request by Netscape to provide new features in the software.[2] In February 2006, Sleepycat was acquired by Oracle Corporation, which continued developing Berkeley DB.[3]

The founders of the company were spouses Margo Seltzer and Keith Bostic, who are also original authors of Berkeley DB. Another original author, Michael Olson, was the President and CEO of Sleepycat. They attended University of California, Berkeley, where they developed the software that grew to become Berkeley DB. Sleepycat was originally based in Carlisle, Massachusetts[4] and in 2000 moved to Lincoln, Massachusetts.[5]

Sleepycat distributed Berkeley DB under a proprietary software license that included standard commercial features, and simultaneously under the newly created Sleepycat License, which allows open source use and distribution of Berkeley DB with a copyleft redistribution condition similar to the GNU General Public License.[2]

Sleepycat had offices in California, Massachusetts and the United Kingdom, and was profitable during its entire existence.[6]

See also

- Berkeley Software Design
- Computer Systems Research Group

References

1. Fowler, Adam (2015). *NoSQL For Dummies* (https://books.google.com/books?id=O_UwBgAAQBAJ&dq=sleepycat+software+maintained+Berkeley+DB+packages&pg=PA125). John Wiley & Sons. p. 125. ISBN 9781118905623.
2. Brunelli, Mark (March 28, 2005). "A Berkeley DB primer" (https://web.archive.org/web/20080906125402/http://searchenterpriselinux.techtarget.com/originalContent/0,289142,sid39_gci1071880,00.html). *Enterprise Linux News*. Archived from the original (http://searchenterpriselinux.techtarget.com/originalContent/0,289142,sid39_gci1071880,00.html) on 2008-09-06. Retrieved 2009-04-28.
3. Babcock, Charles (February 14, 2006). "Oracle Buys Sleepycat, Is JBoss Next?" (https://web.archive.org/web/20110515023408/http://www.informationweek.com/software/showArticle.jhtml?articleID=180200853). *InformationWeek*. Archived from the original (http://www.informationweek.com/software/showArticle.jhtml?articleID=180200853) on 2011-05-15. Retrieved 2007-01-17.
4. "The Sleepycat Software Contact Page" (https://web.archive.org/web/19971210224023/http://www.sleepycat.com/). Carlisle, Massachusetts: Sleepycat Software, Inc. 1997-05-12. Archived from the original (http://www.sleepycat.com/) on 1997-12-10. Retrieved 2010-04-14.
5. "The Sleepycat Software Contact Page" (https://web.archive.org/web/20001205052100/http://sleepycat.com/contact.html). Lincoln, Massachusetts: Sleepycat Software, Inc. 2000-06-08. Archived from the original (http://www.sleepycat.com/contact.html) on 2000-12-05. Retrieved 2010-04-14.
6. "About Sleepycat" (https://web.archive.org/web/20060315211506/http://www.sleepycat.com/company/pdfs/sc_about_1205.pdf) (PDF). Lincoln, Massachusetts: Sleepycat Software, Inc. 2006-01-04. Archived from the original (http://www.sleepycat.com/company/pdfs/sc_about_1205.pdf) (PDF) on 2006-03-15.

External links

- Oracle Berkeley DB (http://www.oracle.com/technology/products/berkeley-db/) — successor to Sleepycat's web site

Wikipedia
The Free Encyclopedia

Unlicense

The **Unlicense** is a public domain equivalent license for software which provides a public domain waiver with a fall-back public-domain-like license, similar to the CC Zero for cultural works.[3] It includes language used in earlier software projects and has a focus on an anti-copyright message.[4][5]

Unlicense

Unlicense logo

Author	Arto Bendiken
Published	2010
SPDX identifier	Unlicense
FSF approved	Yes[1]
OSI approved	Yes[2]
GPL compatible	Yes[1]
Copyleft	No[1]
Linking from code with a different licence	Yes
Website	unlicense.org (https://unlicense.org)

License terms

The text of the Unlicense is as follows:[5]

```
This is free and unencumbered software released
into the public domain.

Anyone is free to copy, modify, publish, use,
compile, sell, or
distribute this software, either in source code
form or as a compiled
binary, for any purpose, commercial or non-
commercial, and by any
means.

In jurisdictions that recognize copyright laws, the
author or authors
of this software dedicate any and all copyright
interest in the
software to the public domain. We make this
dedication for the benefit
of the public at large and to the detriment of our
heirs and
successors. We intend this dedication to be an
overt act of
relinquishment in perpetuity of all present and
future rights to this
software under copyright law.

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 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.

For more information, please refer to
<https://unlicense.org>
```

Reception

The Free Software Foundation states that "Both public domain works and the lax license provided by the Unlicense are compatible with the GNU GPL."[1]

Google does not allow its employees to contribute to projects under public domain equivalent licenses like the Unlicense (and CC0), while allowing contributions to 0BSD licensed and US government PD projects.[6]

Notable projects that use the Unlicense include youtube-dl,[7] Second Reality,[8] and the source code of the 1995 video game *Gloom*.[9]

History

In a post published on January 1 (Public Domain Day), 2010, Arto Bendiken, the author of the Unlicense, outlined his reasons for preferring public domain software, namely: the nuisance of dealing with licensing terms (for instance license incompatibility), the threat inherent in copyright law, and the impracticability of copyright law.[10]

On January 23, 2010, Bendiken followed-up on his initial post. In this post, he explained that the Unlicense is based on the copyright waiver of SQLite with the no-warranty statement from the MIT License. He then walked through the license, commenting on each part.[11]

In a post published in December 2010, Bendiken further clarified what it means to "license" and "unlicense" software.[12]

In December 2010, Mike Linksvayer, the vice president of Creative Commons at the time, wrote in an identi.ca conversation "I like the movement" in speaking of the Unlicense effort, considering it compatible with the goals of the CC Zero (CC0) license, released in 2009.[13][14] On January 1, 2011, Bendiken reviewed the progress and adoption of the Unlicense, saying it was "difficult to give estimates of current Unlicense adoption" but there were "many hundreds of projects using the Unlicense".[15]

In January 2012, when discussed on OSI's license-review mailing list, the Unlicense was brushed off as a crayon license. In particular, it was criticized for being possibly inconsistent and non-standard, and for making it difficult for some projects to accept Unlicensed code as third-party contributions; leaving too much room for interpretation; and possibly being incoherent in some legal systems.[16][17][18] A request for legacy approval was filed in March 2020,[19] which led to a formal approval in June 2020, with an acknowledgement of a "general agreement that the document is poorly drafted".[2]

In 2015, GitHub reported that approximately 102,000 of their 5.1 million licensed projects (2% of licensed projects on GitHub.com) used the Unlicense.[20]

Until 2022, the Fedora Project recommended CC0 over the Unlicense because the former is "a more comprehensive legal text".[3] However, in July 2022, the CC0 license became unsupported and software to be released in the Fedora distribution must not be under CC0, due to CC0 not waiving patent rights.[21]

See also

- 0BSD a public domain equivalent license used by Toybox and explicitly allowed for Android
- WTFPL
- Comparison of free and open-source software licenses

References

1. "Various Licenses and Comments about Them - GNU Project § The Unlicense" (https://www.gnu.org/licenses/license-list.en.html#Unlicense). Free Software Foundation. Retrieved February 10, 2017.
2. Chestek, Pamela (June 16, 2020). "[License-review] Request for legacy approval: The Unlicense" (http://lists.opensource.org/pipermail/license-review_lists.opensource.org/2020-June/004890.html). Archived (https://web.archive.org/web/20200908144433/http://lists.opensource.org/pipermail/license-review_lists.opensource.org/2020-June/004890.html) from the original on September 8, 2020. "There is general agreement that the document is poorly drafted. It is an attempt to dedicate a work to the public domain (which, taken alone, would not be approved as an open source license) but it also has wording commonly used for license grants. There was some discussion about the legal effectiveness of the document, in particular how it would operate in a jurisdiction where one cannot dedicate a work to the public domain. The lawyers who opined on the issue, both US and non-US, agreed that the document would most likely be interpreted as a license and that the license met the OSD. It is therefore recommended for approval."
3. "Licensing/Unlicense" (https://fedoraproject.org/w/index.php?title=Licensing/Unlicense&direction=next&oldid=384273). Fedora Project. August 14, 2014. Retrieved February 28, 2017. "Fedora recommends use of CC-0 over this license, because it is a more comprehensive legal text around this tricky issue. It is also noteworthy that some MIT variant licenses which contain the right to "sublicense" are closer to a true Public Domain declaration than the one in the "Unlicense" text."
4. Joe Brockmeier (January 11, 2010). "The Unlicense: A License for No License" (https://web.archive.org/web/20170122063132/http://ostatic.com/blog/the-unlicense-a-license-for-no-license). OStatic. Archived from the original (http://ostatic.com/blog/the-unlicense-a-license-for-no-license) on January 22, 2017.
5. "Unlicense Yourself: Set Your Code Free" (https://unlicense.org). Retrieved February 28, 2017.
6. "Open Source Patching" (https://opensource.google/docs/patching/). Retrieved 2020-09-29.
7. "youtube-dl GitHub page" (https://github.com/ytdl-org/youtube-dl/blob/master/LICENSE). *GitHub*. Retrieved 2 October 2016.
8. Mika Tuomi (August 1, 2013). "SecondReality/UNLICENSE at master · mtuomi/SecondReality" (https://github.com/mtuomi/SecondReality/blob/master/UNLICENSE). GitHub. Retrieved February 28, 2017.
9. GloomAmiga (https://github.com/earok/GloomAmiga) on GitHub - Source code of Gloom released in May 2017
10. Arto Bendiken (January 1, 2010). "Set Your Code Free" (http://ar.to/2010/01/set-your-code-free). Retrieved February 10, 2017. "anybody affixing a licensing statement to open-source software is guilty of either magical thinking or of having an intention to follow up on the implied threat"

11. Arto Bendiken (January 23, 2010). "Dissecting the Unlicense: Software Freedom in Four Clauses and a Link" (http://ar.to/2010/01/dissecting-the-unlicense). Retrieved February 10, 2017.
12. Arto Bendiken (December 19, 2010). "Licensed, License-Free, and Unlicensed Code" (http://ar.to/2010/12/licensing-and-unlicensing). Retrieved February 10, 2017.
13. Mike Linksvayer (December 17, 2010). "Conversation" (https://web.archive.org/web/20110816083109/https://identi.ca/conversation/59986314). Identi.ca. Archived from the original (http://identi.ca/conversation/59986314) on August 16, 2011. Retrieved February 28, 2017. "@bendiken surely there's a better name than copyfree, but I like the movement and look fwd to your roundup."
14. Arto Bendiken (December 18, 2010). "CC0 and the Unlicense" (https://groups.google.com/forum/#!topic/unlicense/an9PHJ0NGxA). Google Groups. Retrieved February 28, 2017. "In case it's of interest, I'm engaged in an ongoing Identi.ca conversation with Mike Linksvayer, the vice president of Creative Commons [...] In short, the folks at Creative Commons are aware of the Unlicense initiative, and apparently supportive of it."
15. Arto Bendiken (January 1, 2011). "The Unlicense: The First Year in Review" (http://ar.to/2011/01/unlicense-1st-year). Retrieved February 10, 2017.
16. Val Markovic (Valloric) (July 6, 2014). "Use a working license instead of UNLICENSE" (https://github.com/docopt/docopt.rs/issues/1). GitHub. Retrieved February 9, 2017.
17. cgt (May 3, 2012). "What is wrong with the Unlicense?" (https://softwareengineering.stackexchange.com/questions/147111/what-is-wrong-with-the-unlicense). Software Engineering Stack Exchange. Retrieved February 10, 2017.
18. Moen, Rick (January 3, 2012). "[License-review] OSI, legal conditions outside the "four corners" of the license, and PD/CC 0 [was Re: Can OSI specify that public domain is open source?]" (https://web.archive.org/web/20170301020915/https://lists.opensource.org/pipermail/license-review/2012-January/001386.html). Open Source Initiative. Archived from the original (https://lists.opensource.org/pipermail/license-review/2012-January/001386.html) on March 1, 2017. Retrieved February 10, 2017.
19. Jaeckel, Steffen (March 28, 2020). "[License-review] Request for legacy approval: The Unlicense" (https://web.archive.org/web/20200908143823/http://lists.opensource.org/pipermail/license-review_lists.opensource.org/2020-March/004795.html). Archived from the original (http://lists.opensource.org/pipermail/license-review_lists.opensource.org/2020-March/004795.html) on September 8, 2020.
20. Balter, Ben (2015-03-09). "Open source license usage on GitHub.com" (https://github.com/blog/1964-license-usage-on-github-com). github.com. Retrieved 2015-11-21. "1 MIT 44.69%, 2 Other 15.68%, 3 GPLv2 12.96%, 4 Apache 11.19%, 5 GPLv3 8.88%, 6 BSD 3-clause 4.53%, 7 Unlicense 1.87%, 8 BSD 2-clause 1.70%, 9 LGPLv3 1.30%, 10 AGPLv3 1.05% (30 million × 2% × 17% = 102k)"
21. Claburn, Thomas (2022-07-25). "Fedora sours on CC 'No Rights Reserved' license" (https://www.theregister.com/2022/07/25/fedora_sours_on_creative_commons/). *The Register*. Retrieved 2022-09-14.

External links

- Official website (https://unlicense.org/)
- Official mailing list (https://groups.google.com/forum/#!forum/unlicense)

Wikipedia
The Free Encyclopedia

WTFPL

The **WTFPL** is a permissive free software license.[1] As a public domain like license, the WTFPL is essentially the same as dedication to the public domain.[2] It allows redistribution and modification of the work under any terms. The name is an abbreviation of **Do What The Fuck You Want To Public License**.

The first version of the WTFPL, released in March 2000, was written by Banlu Kemiyatorn for his own software project.[3] Sam Hocevar, Debian's former project leader, wrote version 2.[4]

Do What the Fuck You Want To Public License

The WTFPL logo

Author	Banlu Kemiyatorn, Sam Hocevar
Latest version	2
Publisher	Sam Hocevar
Published	2004
SPDX identifier	WTFPL
FSF approved	Yes[1]
OSI approved	No
GPL compatible	Yes[1]
Copyleft	No[1]
Linking from code with a different licence	Yes
Website	www.wtfpl.net (http://www.wtfpl.net)

Characteristics

The WTFPL intends to be a permissive, public-domain-like license. The license is not a copyleft license.[1] The license differs from public domain in that an author can use it even if they do not necessarily have the ability to place their work in the public domain according to their local laws.[5]

The WTFPL does not include a no-warranty disclaimer, unlike other permissive licenses, such as the MIT License.[6] Though the WTFPL is untested in court, the official website offers a disclaimer to be used in software source code.[3]

Terms

Version 2

The text of Version 2, the most current version of the license, written by Sam Hocevar:[4]

```
           DO WHAT THE FUCK YOU WANT TO PUBLIC LICENSE
                   Version 2, December 2004

Copyright (C) 2004 Sam Hocevar <sam@hocevar.net>

Everyone is permitted to copy and distribute verbatim or modified
copies of this license document, and changing it is allowed as long
as the name is changed.

           DO WHAT THE FUCK YOU WANT TO PUBLIC LICENSE
```

```
  TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

 0. You just DO WHAT THE FUCK YOU WANT TO.
```

Version 1

```
do What The Fuck you want to Public License

Version 1.0, March 2000
Copyright (C) 2000 Banlu Kemiyatorn (]d).
136 Nives 7 Jangwattana 14 Laksi Bangkok
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Ok, the purpose of this license is simple
and you just

DO WHAT THE FUCK YOU WANT TO.
```

Reception

Usage

The WTFPL is not in wide use among open-source software projects; according to Black Duck Software, the WTFPL is used by less than one percent of open-source projects.[7] Examples include the OpenStreetMap Potlatch online editor,[8] the video game Liero (version 1.36),[9] yalu102[10] and MediaWiki extensions.[11] More than 7,000 Wikimedia Commons files[12] and more than 34,000 Projects on GitHub were published under the terms of the WTFPL.[13]

Discussion

The license was confirmed as a GPL-compatible free software license by the Free Software Foundation, but its use is "not recommended".[1] In 2009, the Open Source Initiative chose not to approve the license as an open-source license due to redundancy with the Fair License.[2]

The WTFPL version 2 is an accepted Copyfree license.[14] It is also accepted by Fedora as a free license and GPL-compatible.[15]

Some software authors have said that the license is not very serious;[16] forks have tried to address wording ambiguity and liability concerns.[17][18] OSI founding president Eric S. Raymond interpreted the license as written satire against the restrictions of the GPL and other software licenses;[19] WTFPL version 2 author Sam Hocevar later confirmed that the WTFPL is a parody of the GPL.[20] Free-culture activist Nina Paley said she considered the WTFPL a free license for cultural works.[21]

Google does not allow its employees to contribute to projects under public domain equivalent licenses like the WTFPL (and Unlicense CC0), while allowing contributions to 0BSD licensed and US government PD projects.[22]

See also

- 0BSD a public domain equivalent license used by Toybox and explicitly allowed for Android
- Unlicense
- Beerware
- CC0
- License-free software
- Public domain software
- Software using the WTFPL (category)
- Public-domain-equivalent license
- Fair License

References

1. "Various Licenses and Comments about Them" (https://www.gnu.org/licenses/license-list.en.html#WTFPL). *GNU Operating System*. Free Software Foundation.
2. "OSI Board Meeting Minutes, Wednesday, March 4, 2009" (https://web.archive.org/web/20160316070255/https://opensource.org/minutes20090304). Open Source Initiative. 2009-03-04. Archived from the original (http://www.opensource.org/minutes20090304) on 2016-03-16. Retrieved 2013-04-03. "[...] the following licenses to be discussed and approved/disapproved by the Board. [...] WTFPL Submission: [...] Comments: It's no different from dedication to the public domain. Author has submitted license approval request -- author is free to make public domain dedication. Although he agrees with the recommendation, Mr. Michlmayr notes that public domain doesn't exist in Europe. Recommend: Reject. [...] Mr. Michlmayr did not agree with the reasons cited for possible rejection of the WFTPL license since public domain doesn't exist in Europe. [...] Mr. Michlmayr moved that we reject the WFTPL as redundant to the Fair License."
3. Sam Hocevar (2012-12-27). "Frequently Asked Questions" (https://web.archive.org/web/20131225073947/http://www.wtfpl.net/faq/). WTFPL – Do What the Fuck You Want to Public License. Archived from the original (http://www.wtfpl.net/faq/) on 2013-12-25. Retrieved 2016-07-19.
4. Sam Hocevar (2012-12-26). "WTFPL version 2" (https://web.archive.org/web/20160715014738/http://www.wtfpl.net/about/). Archived from the original (http://www.wtfpl.net/about) on 2016-07-15. Retrieved 2016-07-19.
5. Kreutzer, Till. "Validity of the Creative Commons Zero 1.0 Universal Public Domain Dedication and its usability for bibliographic metadata from the perspective of German Copyright Law" (https://web.archive.org/web/20170525023540/https://www.rd-alliance.org/sites/default/files/cc0-analysis-kreuzer.pdf) (PDF). Büro für informationsrechtliche Expertise. Archived from the original (https://rd-alliance.org/sites/default/files/cc0-analysis-kreuzer.pdf) (PDF) on 2017-05-25. Retrieved 2015-12-29.
6. "The MIT License" (https://opensource.org/licenses/MIT). *Open Source Initiative*.
7. "Top Open Source Licenses" (https://web.archive.org/web/20160510102832/https://www.blackducksoftware.com/top-open-source-licenses). Black Duck Software. Archived from the original (https://www.blackducksoftware.com/top-open-source-licenses) on 2016-05-10.
8. "LICENCE.txt" (https://git.openstreetmap.org/potlatch2.git/blob/HEAD:/LICENCE.txt). *Potlatch 2*. GitHub. December 2004. Retrieved 2012-01-16.

9. "license.txt" (http://liero.be/licenses/135/license.txt). *Liero official website*. 2013-09-03. Retrieved 2016-07-12. "The original Liero data and binary files are copyright 1998 Joosa Riekkinen
They are, unless otherwise stated, available under the WTFPL license: http://sam.zoy.org/wtfpl/"
10. "incomplete iOS 10.2 jailbreak for 64 bit devices by qwertyoruiopz and marcograssi" (https://github.com/kpwn/yalu102). *GitHub*. 2017-02-02. Retrieved 2022-10-06.
11. "Category:WTFPL licensed extensions" (https://www.mediawiki.org/wiki/Category:WTFPL_licensed_extensions). MediaWikiWiki. 2016. Retrieved 2016-07-19.
12. "Category:WTFPL" (https://commons.wikimedia.org/wiki/Category:WTFPL). Wikimedia Commons. 2025. Retrieved 2025-04-19.
13. "Build software better, together" (https://github.com/search?l=&o=desc&q=license:wtfpl&s=&type=Repositories). *GitHub*. Retrieved 2025-03-13.
14. "Copyfree Licenses" (http://copyfree.org/standard/licenses). *Copyfree*. The Copyfree Initiative.
15. Callaway, Tom (2016-05-17). "Licensing:Main" (https://fedoraproject.org/wiki/Licensing:Main?rd=Licensing). *Fedora Project Wiki*. Retrieved 2016-07-10.
16. Suder, Kuba (2011-01-15). "On Open Source licensing" (https://mackuba.eu/2011/01/15/on-open-source-licensing/). *Apples & Rubies* (Blog). "There are at least two not very serious licenses which have essentially the same meaning as public domain. I'm talking about the Beerware license and WTFPL ('Do What The Fuck You Want To' license). I really like these because they pretty well represent my opinion about the legalese bullshit that most licenses are so full of."
17. theiostream (2012-03-24). "Introducing WTFPL v3" (https://web.archive.org/web/20140705065219/http://theiostream.tumblr.com/post/19850980363/introducing-wtfpl-v3) (Blog). tumblr. Archived from the original (https://theiostream.tumblr.com/post/19850980363/introducing-wtfpl-v3) on July 5, 2014. Retrieved 2016-07-19.
18. Ben McGinnes (2013-10-01). "Do What The Fuck You Want To But It's Not My Fault Public License v1 (WTFNMFPL-1.0)" (https://web.archive.org/web/20160423225128/https://tldrlegal.com/license/do-what-the-fuck-you-want-to-but-it's-not-my-fault-public-license-v1-(wtfnmfpl-1.0)#fulltext). *tl;drLegal*. Archived from the original (https://tldrlegal.com/license/do-what-the-fuck-you-want-to-but-it's-not-my-fault-public-license-v1-(wtfnmfpl-1.0)) on 2016-04-23. Retrieved 2016-07-19. "… with a CYA clause …."
19. Eric S. Raymond (2010-05-19). "Software licenses as conversation" (http://esr.ibiblio.org/?p=2000) (Blog). esr.iblio.org. Retrieved 2016-07-19. "It's even clearer that the Do What the Fuck You Want To Public License is a satire. The author is one of those who thinks the Free Software Foundation has traduced the word 'free' by hedging the GNU General Public License about with restrictions and boobytraps in the name of 'freedom' – and he's got an issue or two with BSD as well. He is poking fun at both camps, not gently at all. His page about the WTFPL is funny-because-it's-true hilarious, and I admit that I feel a sneaking temptation to start using it myself."
20. Sam Hocevar (2015-09-21). "Should I change the name of the WTFPL?" (http://programmers.stackexchange.com/questions/149050/should-i-change-the-name-of-the-wtfpl#comment617907_161949). *Programmers Stack Exchange* (User comment). Retrieved 2016-07-19. "The WTFPL is a parody of the GPL, which has a similar copyright header and list of permissions to modify (i.e. none), see for instance gnu.org/licenses/gpl-3.0.en.html. The purpose of the WTFPL wording is to give more freedom than the GPL does."
21. Nina Paley (2011-06-24). "How To Free Your Work" (https://web.archive.org/web/20160820001500/http://questioncopyright.org/how_to_free_your_work). *QuestionCopyright.org*. Archived from the original (http://questioncopyright.org/how_to_free_your_work) on 2016-08-20. Retrieved 2016-07-19.

22. "Open Source Patching" (https://opensource.google/documentation/reference/patching#forbidden). Retrieved 2020-09-29.

External links

- Official website (https://www.wtfpl.net)

Wikipedia
The Free Encyclopedia

zlib License

The **zlib license** is a permissive software license which defines the terms under which the zlib software library can be distributed. It is also used by many other open-source packages. The libpng library uses a similar license, **libpng license**, sometimes referred interchangeably as *zlib/libpng license*.

The zlib license has been approved by the Free Software Foundation (FSF) as a free software license,[2] and by the Open Source Initiative (OSI) as an open source license.[3] It is compatible with the GNU General Public License.[2]

zlib license

Author	Jean-loup Gailly and Mark Adler
Latest version	1.2.11[1]
Publisher	Jean-loup Gailly and Mark Adler
Published	15 April 1995
SPDX identifier	Zlib
Debian FSG compatible	Yes
FSF approved	Yes[2]
OSI approved	Yes
GPL compatible	Yes[2]
Copyleft	No
Linking from code with a different licence	Yes

Terms

The license only has the following points to be accounted for:

- Software is used on 'as-is' basis. Authors are not liable for any damages arising from its use.
- The distribution of a modified version of the software is subject to the following restrictions:
  1. The authorship of the original software must not be misrepresented,
  2. Altered source versions must not be misrepresented as being the original software, and
  3. The license notice must not be removed from source distributions.

The license does not require source code to be made available if distributing binary code.

Text

The license terms are as follows:

```
Copyright (c) <year> <copyright holders>

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.
```

See also

Free and open-source software portal

- Comparison of free and open-source software licenses
- Software using the zlib license (category)

References

1. "Zlib License" (http://zlib.net/zlib_license.html).
2. "Various Licenses and Comments about Them" (https://www.gnu.org/licenses/license-list.html#ZLib). Retrieved 17 June 2007. "License of ZLib [...] is a free software license, and compatible with the GPL."
3. "The zlib/libpng License (Zlib)" (http://opensource.org/licenses/Zlib). Retrieved 16 January 2013.

External links

- License text on the zlib homepage (http://zlib.net/zlib_license.html)

Comparison of free and open-source software licenses

This comparison only covers software licenses which have a linked Wikipedia article for details and which are approved by at least one of the following expert groups: the Free Software Foundation, the Open Source Initiative, the Debian Project and the Fedora Project. For a list of licenses not specifically intended for software, see List of free-content licences.

FOSS licenses

FOSS stands for "Free and Open Source Software". There is no one universally agreed-upon definition of FOSS software and various groups maintain approved lists of licenses. The Open Source Initiative (OSI) is one such organization keeping a list of open-source licenses.[1] The Free Software Foundation (FSF) maintains a list of what it considers free.[2] FSF's free software and OSI's open-source licenses together are called FOSS licenses. There are licenses accepted by the OSI which are not free as per the Free Software Definition. The Open Source Definition allows for further restrictions like price, type of contribution and origin of the contribution, e.g. the case of the NASA Open Source Agreement, which requires the code to be "original" work.[3][4] The OSI does not endorse FSF license analysis (interpretation) as per their disclaimer.[5]

The FSF's Free Software Definition focuses on the user's unrestricted rights to use a program, to study and modify it, to copy it, and to redistribute it for any purpose, which are considered by the FSF the *four essential freedoms*.[6][7] The OSI's open-source criteria focuses on the availability of the source code and the advantages of an unrestricted and community driven development model.[8] Yet, many FOSS licenses, like the Apache License, and all Free Software licenses allow commercial use of FOSS components.[9]

General comparison

For a simpler comparison across the most common licenses see free-software license comparison.

The following table compares various features of each license and is a general guide to the terms and conditions of each license, based on seven subjects or categories. Recent tools like the European Commissions' Joinup Licensing Assistant,[10] makes possible the licenses selection and comparison based on more than 40 subjects or categories, with access to their SPDX identifier and full text. The table below lists the permissions and limitations regarding the following subjects:

- **Linking** - linking of the licensed code with code licensed under a different license (e.g. when the code is provided as a library)
- **Distribution** - distribution of the code to third parties
- **Modification** - modification of the code by a licensee
- **Patent grant** - protection of licensees from patent claims made by code contributors regarding their contribution, and protection of contributors from patent claims made by licensees
- **Private use** - whether modification to the code must be shared with the community or may be used privately (e.g. internal use by a corporation)
- **Sublicensing** - whether modified code may be licensed under a different license (for example a copyright) or must retain the same license under which it was provided
- **TM grant** - use of trademarks associated with the licensed code or its contributors by a licensee

In this table, "permissive" means the software has minimal restrictions on how it can be used, modified, and redistributed, usually including a warranty disclaimer. "Copyleft" means the software requires that its source code be made publicly available and that all provisions in the license be preserved in derivative works.

License	Author	Latest version	Publication date	Linking	Distribution	Modification	Patent grant	Private use	Sublicensing
Academic Free License[11]	Lawrence E. Rosen	3.0	2002	Permissive	Permissive	Permissive	Yes	Yes	Permissive
Affero General Public License	Affero Inc	2.0	2007	Copylefted[12]	Copyleft except for the GNU AGPL[13]	Copyleft[13]	?	Yes[13]	?
Apache License	Apache Software Foundation	2.0	2004	Permissive[14]	Permissive[14]	Permissive[14]	Yes[14]	Yes[14]	Permissive[14]
Apple Public Source License	Apple Computer	2.0	August 6, 2003	Permissive	?	Limited	?	?	?
Artistic License	Larry Wall	2.0	2000	With restrictions	With restrictions	With restrictions	No	Permissive	With restriction
Beerware	Poul-Henning Kamp	42	1998[a]	Permissive	Permissive	Permissive	No	Permissive	Permissive
BSD License	Regents of the University of California	3.0	?	Permissive[15]	Permissive[15]	Permissive[15]	Manually[15]	Yes[15]	Permissive[15]
Boost Software License	Devin Smith[16]	1.0	August 17, 2003	Permissive	Permissive	Permissive	No	Permissive	With restriction
Creative Commons Zero	Creative Commons	1.0	2009	Public Domain[17][18]	Public Domain	Public Domain	No	Public Domain	Public Domair
CC BY	Creative Commons	4.0	2002	Permissive[19]	Permissive	Permissive	No	Yes	Permissive
CC BY-SA	Creative Commons	4.0	2002	Copylefted[19]	Copylefted	Copylefted	No	Yes	Copylefted[20]
CeCILL	CEA / CNRS / INRIA	2.1	June 21, 2013	Permissive	Permissive	Permissive	No	Permissive	With restriction
Common Development and Distribution License	Sun Microsystems	1.0	December 1, 2004	Permissive	?	Limited	?	?	?
Common Public License	IBM	1.0	May 2001	Permissive	?	Copylefted	?	?	?
Cryptix General License	Cryptix Foundation	—	1995	Permissive	Permissive	Permissive	Manually	Yes	?
Eclipse Public License	Eclipse Foundation	2.0	August 24, 2017	Permissive[21]	Copylefted[21][22]	Copylefted[21]	Yes[21]	Yes[21]	Copylefted[21]
Educational Community License	Indiana University[23]	1.0	2007	Permissive	?	Permissive	?	?	?
European Union Public Licence	European Commission	1.2	May 2017	Permissive, according to EU law (Recitals 10 & 15 Directive 2009/24/EC)	Copylefted, with an explicit compatibility list[24]	Copylefted, with an explicit compatibility list[24]	Yes[25]	Yes[25]	Copylefted, wit an explicit compatibility list[24]
FreeBSD	The FreeBSD project	—	April 1999	Permissive[26]	Permissive[26]	Permissive[26]	Manually[26]	Permissive[26]	Permissive[26]
GNU Affero General Public License	Free Software Foundation	3.0	2007	GNU GPLv3 only[27]	Copylefted[28]	Copylefted[28]	Yes[29]	Network usage is not considered private use[29]	Copylefted[28]
GNU General Public License	Free Software Foundation	3.0	June 2007	GPLv3 compatible only[30][31]	Copylefted[28]	Copylefted[28]	Yes[32]	Yes[32]	Copylefted[28]
GNU Lesser General Public License	Free Software Foundation	3.0	June 2007	With restrictions[33]	Copylefted[28]	Copylefted[28]	Yes[34]	Yes	Copylefted[28]
IBM Public License	IBM	1.0	August 1999	Copylefted	?	Copylefted	?	?	?
ISC license	Internet Systems Consortium	—	June 2003	Permissive	Permissive	Permissive	Manually	Permissive	Permissive
LaTeX Project Public License	LaTeX project	1.3c	?	Permissive	?	Permissive	?	?	?

Microsoft Public License	Microsoft	—	?	Copylefted	Copylefted	Copylefted	No	Permissive	?
MIT license / X11 license	MIT	—	1988	Permissive[35]	Permissive[35]	Permissive[35]	Manually[35]	Yes[35]	Permissive[35]
Mozilla Public License	Mozilla Foundation	2.0	January 3, 2012	Permissive[36]	Copylefted[36]	Copylefted[36]	Yes[36]	Yes[36]	Copylefted[36]
Netscape Public License	Netscape	1.1	?	Limited	?	Limited	?	?	?
Open Software License[11]	Lawrence Rosen	3.0	2005	Permissive	Copylefted	Copylefted	Yes	Yes	Copylefted
OpenSSL license	OpenSSL Project	—	?	Permissive	?	Permissive	?	?	?
PHP License[37]	PHP Group	3.01	2019	With restrictions	With restrictions	With restrictions	Yes	Yes	With restriction
Python Software Foundation License	Python Software Foundation	3.9.1	May 10, 2020	Permissive	Permissive	Permissive	Yes	Permissive	Permissive
Q Public License	Trolltech	?	?	Limited	?	Limited	?	?	?
Sleepycat License	Sleepycat Software	—	1996	Permissive	With restrictions	Permissive	No	Yes	No
Unlicense	unlicense.org	1	December 2010	Permissive/Public domain	Permissive/Public domain	Permissive/Public domain	?	Permissive/Public domain	Permissive/Pub domain
W3C Software Notice and License	W3C	20021231	December 31, 2002	Permissive	?	Permissive	?	?	?
Do What The Fuck You Want To Public License (WTFPL)	Banlu Kemiyatorn, Sam Hocevar	2	December 2004	Permissive/Public domain	Permissive/Public domain	Permissive/Public domain	No	Yes	Yes
XCore Open Source License also separate "Hardware License Agreement"	XMOS	?	February 2011	Permissive	Permissive	Permissive	Manually	Yes	Permissive
XFree86 1.1 License	The XFree86 Project, Inc	?	?	Permissive	?	Permissive	?	?	?
zlib/libpng license	Jean-Loup Gailly and Mark Adler	—	April 15, 1995	Permissive	Permissive	Permissive	Manually	Yes	Permissive

Other licenses that don't have information:

license	Author	Latest version	Publication date
Eiffel Forum License	NICE	2	2002
Intel Open Source License	Intel Corporation	—	?
RealNetworks Public Source License	RealNetworks	?	?
Reciprocal Public License	Scott Shattuck	1.5	2007
Sun Industry Standards Source License	Sun Microsystems	?	?
Sun Public License	Sun Microsystems	?	?
Sybase Open Watcom Public License	Open Watcom	—	2003-01-28
Zope Public License	Zope Foundation	2.1	?
Server Side Public License	MongoDB	1.0	2018-10-16

Approvals

This table lists for each license what organizations from the FOSS community have approved it – be it as a "free software" or as an "open source" license – , how those organizations categorize it, and the license compatibility between them for a combined or mixed derivative work. Organizations usually approve specific versions of software licenses. For instance, a FSF approval means that the Free Software Foundation (FSF) considers a license to be free-software license. The FSF recommends at least "Compatible with GPL" and preferably copyleft. The OSI recommends a mix of permissive and copyleft licenses, the Apache License 2.0, 2- & 3-clause BSD license, GPL, LGPL, MIT license, MPL 2.0, CDDL and EPL.

License and version	FSF approval [38]	GPL (v3) compatibility [39][40][41][42][43]	OSI approval [44]	Debian approval [45][46]	Fedora approval [47]
Academic Free License	Yes	No	Yes	No	Yes
Apache License 1.x	Yes	No	Yes	Yes	Yes
Apache License 2.0	Yes	GPLv3 only[48]	Yes	Yes	Yes
Apple Public Source License 1.x	No[49]	No	Yes	No	No
Apple Public Source License 2.0	Yes	No	Yes	No	Yes
Artistic License 1.0	No[note 1]	No	Yes	Yes	No
Artistic License 2.0	Yes	Yes	Yes	Yes	Yes
Beerware License	see "Informal license" section[50]	see "Informal license" section[50]	No	No	Yes[51]
Original BSD license	Yes	No	No[52]	Yes	Yes
Revised BSD license	Yes	Yes	Yes	Yes	Yes
Simplified BSD license	Yes	Yes	Yes	Yes	Yes
Zero-Clause BSD License	Yes	Yes	Yes[53]	*?*	Yes
Boost Software License	Yes	Yes	Yes	Yes	Yes
CeCILL	Yes	Yes	Yes	Yes	Yes
Common Development and Distribution License	Yes	GPLv3 (GPLv2 disputed)[54][55][56][57][58][59]	Yes	Yes	Yes
Common Public License	Yes	No	Yes	Yes	Yes
Creative Commons Zero	Yes[60]	Yes[60]	No[61]	Yes	Yes[62]
Creative Commons BY-SA 4.0	Yes	GPLv3[63]	*?*	Yes	*?*
Cryptix General License	Yes	Yes	Yes	Yes	Yes
Eclipse Public License	Yes	No	Yes	Yes	Yes
Educational Community License	Yes	Yes[64]	Yes	No	Yes
Eiffel Forum License 2	Yes	Yes	Yes	Yes	Yes
European Union Public Licence	Yes	Yes[24]	Yes	Yes	Yes
GNU Affero General Public License	Yes	Yes[27][65]	Yes	Yes	Yes
GNU General Public License v2	Yes	No[note 2][66]	Yes	Yes	Yes
GNU General Public License v3	Yes	Yes[note 3][66]	Yes	Yes	Yes
GNU Lesser General Public License	Yes	Yes	Yes	Yes	Yes
GNU Free Documentation License	Yes	No[67]	Yes[68]	No[69]	No
IBM Public License	Yes	No	Yes	Yes	Yes
Intel Open Source License	Yes	Yes	Yes	No	No
ISC license	Yes[70]	Yes	Yes	Yes	Yes
LaTeX Project Public License	Yes	No	Yes	Yes	Yes
Microsoft Public License	Yes	No	Yes	No	Yes
Microsoft Reciprocal License	Yes	No	Yes	No	Yes
MIT license / X11 license	Yes	Yes	Yes	Yes	Yes
MIT No Attribution License	Yes	Yes	Yes	*?*	Yes
Mozilla Public License 1.1	Yes	No	Yes	Yes	Yes
Mozilla Public License 2.0	Yes	Yes[note 4][71]	Yes	Yes	Yes
NASA Open Source Agreement	No	No	Yes	*?*	No
Netscape Public License	Yes	No	No	No	Yes
Open Software License	Yes	No	Yes	No	Yes
OpenSSL license	Yes	No	No	Yes	Yes
PHP License	Yes	No	Yes	Yes	Yes
Python Software Foundation License 2.0.1; 2.1.1 and newer	Yes	Yes	Yes	Yes	Yes
Q Public License	Yes	No	Yes	No	Yes
Reciprocal Public License 1.5	No	No	Yes	No	No
Sleepycat License	Yes	Yes	Yes	Yes	Yes
Sun Industry Standards Source License	Yes	No	Yes	No	Yes
Sun Public License	Yes	No	Yes	No	Yes
Sybase Open Watcom Public License	No	No	Yes	No	No

Unlicense	Yes[72]	Yes[60]	Yes[73]	?	Yes[62]
W3C Software Notice and License	Yes	Yes	Yes	Yes	Yes
Do What The Fuck You Want To Public License (WTFPL)	Yes[note 5]	Yes	No[74]	Yes	Yes
XFree86 1.1 License	Yes	Yes[75]	No	No	No
zlib/libpng license	Yes	Yes	Yes	Yes	Yes
Zope Public License 1.0	Yes	No	No	No	Yes
Zope Public License 2.0	Yes	Yes	Yes	No	Yes

1. The original version of the Artistic License is defined as non-free because it is overly vague, not because of the substance of the license. The FSF encourages projects to use the Clarified Artistic License (https://www.gnu.org/licenses/license-list.html#ArtisticLicense) instead.
2. But can be made compatible by upgrading to GPLv3 via the optional "or later" clause added in most GPLv2 license texts.
3. But not with GPLv2 without "or later" clause.
4. MPL 2.0 is GPL compatible unless marked "Incompatible with Secondary Licenses".
5. Listed as WTFPL.

See also

Free and open-source software portal

- Free software
- Free-software license
- List of free and open-source software packages
- List of open-source hardware projects
- List of open-source programming languages
- List of open-source video games
- Open-source license
- Open-source software
- Source-available software
- SPDX-License-identifier
- License proliferation
- European Union Public Licence

Notes

a. See footnote of the Beerware article

References

1. Open source licenses - Licenses by Name (http://opensource.org/licenses/alphabetical) on opensource.org
2. "Various Licenses and Comments about Them" (https://www.gnu.org/licenses/license-list.html). *Free Software Foundation*. Retrieved August 8, 2011.
3. "Various Licenses and Comments about Them: NASA Open Source Agreement" (https://www.gnu.org/licenses/license-list.en.html#NASA). *Free Software Foundation*.
4. "Licenses by Name" (https://opensource.org/licenses/alphabetical). *Open Source Initiative*. 16 September 2022.
5. "Other Resources & Disclaimer" (https://opensource.org/node/778). *Open Source Initiative*. "*While the OSI acknowledges these as potentially helpful resources for the community, it does not endorse any content, contributors or license interpretations from these websites.[...]The OSI does not promote or exclusively favor any of the above resources, but instead mentions them as a neutral, separate third-party.*"
6. "Relationship between the Free Software movement and Open Source movement" (https://www.gnu.org/philosophy/free-software-for-freedom.html), Free Software Foundation, Inc
7. "What is Free Software" (https://www.gnu.org/philosophy/free-sw.html), Free Software Foundation, Inc
8. opensource.org/about (https://opensource.org/about) "Open source is a development method for software that harnesses the power of distributed peer review and transparency of process. The promise of open source is better quality, higher reliability, more flexibility, lower cost, and an end to predatory vendor lock-in."
9. Popp, Dr. Karl Michael (2015). *Best Practices for commercial use of open source software*. Norderstedt, Germany: Books on Demand. ISBN 978-3738619096.
10. "Joinup Licensing Assistant" (https://joinup.ec.europa.eu/solution/joinup-licensing-assistant/joinup-licensing-assistant-jla). Retrieved 31 March 2020.
11. "OSL 3.0 Explained" (https://rosenlaw.com/OSL3.0-explained.htm). *rosenlaw.com*.
12. "affero.org: Affero General Public License version 1 (AGPLv1)" (https://web.archive.org/web/20191123101252/http://www.affero.org:80/oagpl.html). Archived from the original (http://www.affero.org/oagpl.html) on November 23, 2019.
13. "affero.org: Affero General Public License version 2 (AGPLv2)" (https://web.archive.org/web/20191123102313/http://www.affero.org:80/agpl2.html). Archived from the original (http://www.affero.org/agpl2.html) on November 23, 2019.
14. "Apache License, Version 2.0" (https://www.apache.org/licenses/LICENSE-2.0.html#redistribution). *www.apache.org*.
15. "BSD license" (http://opensource.org/licenses/BSD-3-Clause). 22 May 2011.
16. "Boost Software License" (https://web.archive.org/web/20250424015917/https://www.boost.org/users/license.html). Boost (C++ libraries). December 3, 2003. Archived from the original (https://www.boost.org/users/license.html) on April 24, 2025. Retrieved April 21, 2025.
17. "Using CC0 for public domain software" (https://web.archive.org/web/20110514163106/https://creativecommons.org/weblog/entry/27081). Creative Commons. April 15, 2011. Archived from the original (https://creativecommons.org/weblog/entry/27081) on May 14, 2011. Retrieved May 10, 2011.
18. "Various Licenses and Comments about Them" (https://www.gnu.org/licenses/license-list.html). GNU Project. Retrieved April 4, 2015.

19. cc-by-4-0-and-cc-by-sa-4-0-added-to-our-list-of-free-licenses (https://www.fsf.org/blogs/licensing/cc-by-4-0-and-cc-by-sa-4-0-added-to-our-list-of-free-licenses) (2015)
20. "Compatible Licenses" (https://creativecommons.org/share-your-work/licensing-considerations/compatible-licenses/). Creative Commons.
21. Beaton, Wayne. "Eclipse Public License 2.0 (EPL) | The Eclipse Foundation" (https://www.eclipse.org/legal/epl-2.0/). *Eclipse Foundation*.
22. "Explanation of Common Open Source Licenses" (https://www.capitalone.com/tech/open-source/open-source-licenses-explained/). *Capital One*.
23. Greenstein, Daniel; Wheeler, Brad (1 March 2007). "Open Source Collaboration in Higher Education: Guidelines and Report of the Licensing and Policy Framework Summit for Software Sharing in Higher Education" (https://scholarworks.iu.edu/dspace/handle/2022/3076) – via scholarworks.iu.edu.
24. "Matrix of EUPL compatible open source licences | Interoperable Europe Portal" (https://interoperable-europe.ec.europa.eu/collection/eupl/matrix-eupl-compatible-open-source-licences). *interoperable-europe.ec.europa.eu*.
25. "EUPL text (EUPL-1.2) | Interoperable Europe Portal" (https://interoperable-europe.ec.europa.eu/collection/eupl/eupl-text-eupl-12). *interoperable-europe.ec.europa.eu*.
26. "The FreeBSD Copyright" (https://www.freebsd.org/copyright/freebsd-license/). *The FreeBSD Project*.
27. https://www.gnu.org/licenses/agpl.html : section 13 of the GNU AGPLv3 license
28. https://www.gnu.org/copyleft/gpl-howto.html : GNU licenses copyleft
29. "GNU Affero General Public License - GNU Project - Free Software Foundation" (https://www.gnu.org/licenses/agpl-3.0.html).
30. https://www.gnu.org/licenses/gpl-faq.html#IfLibraryIsGPL : If library is under GPLv3
31. https://www.gnu.org/licenses/gpl-faq.html#LinkingWithGPL : Linking with the GNU GPLv3
32. "gnu.org" (https://www.gnu.org/licenses/gpl.html). *www.gnu.org*.
33. https://www.gnu.org/licenses/lgpl.html : the section 4 of the GNU Lesser General Public License version 3
34. "GNU Lesser General Public License v3.0 - GNU Project - Free Software Foundation" (https://www.gnu.org/licenses/lgpl.html).
35. "MIT License" (http://opensource.org/licenses/MIT). 31 October 2006.
36. "Mozilla Public License, version 2.0" (https://www.mozilla.org/en-US/MPL/2.0/). *www.mozilla.org*.
37. "PHP License 3.01" (https://www.php.net/license/3_01.txt).
38. Free Software Foundation. "Various Licenses and Comments about Them" (https://www.gnu.org/licenses/license-list.html). *Licenses*. Free Software Foundation.
39. Free Software Foundation. "To be GPL-Compatible has to be compatible with Licenses GNU GPLv3 and GNU GPLv2 – Free Software Foundation" (https://www.gnu.org/licenses/license-list.html#Introduction). *Software Licenses*. Free Software Foundation.
40. Free Software Foundation. "GPL-Compatible Free Software Licenses – Free Software Foundation" (https://www.gnu.org/licenses/license-list.html#GPLCompatibleLicenses). *Software Licenses*. Free Software Foundation.
41. Free Software Foundation. "GPL-Incompatible Free Software Licenses – Free Software Foundation" (https://www.gnu.org/licenses/license-list.html#GPLIncompatibleLicenses). *Software Licenses*. Free Software Foundation.
42. Free Software Foundation. "GPL-compatible Definition by FSF – Free Software Foundation" (http://gplv3.fsf.org/wiki/index.php/FAQ_Update#What_does_it_mean_to_say_a_license_is_.22compatible_with_the_GPL.22.3F). *GPL-compatible Definition*. Free Software Foundation.
43. Free Software Foundation. "GPL-compatible Definition previous version by FSF – Free Software Foundation" (https://www.gnu.org/licenses/gpl-faq.html#WhatDoesCompatMean). *GPL-compatible Definition*. Free Software Foundation.
44. Open Source Initiative (16 September 2022). "The Approved Licenses" (http://www.opensource.org/licenses/). *License Information*. Open Source Initiative.
45. Debian. "Debian – License information" (http://www.debian.org/legal/licenses/). *Licenses*. Debian.
46. "The DFSG and Software Licenses" (http://wiki.debian.org/DFSGLicenses). Debian wiki.
47. Fedora. "Allowed Licenses" (https://docs.fedoraproject.org/en-US/legal/allowed-licenses/). Fedora Project.
48. Free Software Foundation. "Apache License, Version 2.0" (https://www.gnu.org/philosophy/license-list.html#apache2). *Licenses*. Free Software Foundation.
49. "Apple Public Source License (APSL), version 1.x" (https://www.gnu.org/licenses/license-list.html#apsl1). Retrieved 2013-08-07.
50. "Various Licenses and Comments about Them" (https://www.gnu.org/licenses/license-list.html#informal). Free Software Foundation. 2016-01-05. Retrieved 2016-01-05.
51. "Licensing/Beerware" (https://fedoraproject.org/wiki/Licensing/Beerware). Fedora Project. Retrieved 2015-03-10.
52. "The BSD License:Licensing" (https://web.archive.org/web/20091129081849/http://www.opensource.org/licenses/bsd-license.php). Open Source Initiative. Archived from the original (http://www.opensource.org/licenses/bsd-license.php) on 29 November 2009. Retrieved 1 February 2021.
53. "[License-review] Please rename "Free Public License-1.0.0" to 0BSD" (http://lists.opensource.org/pipermail/license-review_lists.opensource.org/2018-November/003830.html). Open Source Initiative. Retrieved 2019-02-11.
54. "Various Licenses and Comments About Them - Common Development and Distribution License" (https://www.gnu.org/licenses/license-list.html#CDDL). Free Software Foundation. Retrieved 2006-12-31.
55. Michael Larabel (6 October 2015). "Ubuntu Is Planning To Make The ZFS File-System A "Standard" Offering" (https://www.phoronix.com/scan.php?page=news_item&px=Ubuntu-ZFS-Standard-Plans). Phoronix.
56. Dustin Kirkland (18 February 2016). "ZFS Licensing and Linux" (https://insights.ubuntu.com/2016/02/18/zfs-licensing-and-linux/). *Ubuntu Insights*. Canonical.
57. Are GPLv2 and CDDL incompatible? (http://blog.hansenpartnership.com/are-gplv2-and-cddl-incompatible/) on hansenpartnership.com by James E.J. Bottomley *"What the above analysis shows is that even though we presumed combination of GPLv2 and CDDL works to be a technical violation, there's no way actually to prosecute such a violation because we can't develop a convincing theory of harm resulting. Because this makes it impossible to take the case to court, effectively it must be concluded that the combination of GPLv2 and CDDL, provided you're following a GPLv2 compliance regime for all the code, is allowable."* (23 February 2016)
58. Moglen, Eben; Choudhary, Mishi (26 February 2016). "The Linux Kernel, CDDL and Related Issues" (https://softwarefreedom.org/resources/2016/linux-kernel-cddl.html).
59. GPL Violations Related to Combining ZFS and Linux (https://sfconservancy.org/blog/2016/feb/25/zfs-and-linux/) on sfconservancy.org by Bradley M. Kuhn and Karen M. Sandler (February 25, 2016)
60. "Various Licenses and Comments about Them - GNU Project - Free Software Foundation" (https://www.gnu.org/licenses/license-list.en.html#CC0). *www.gnu.org*.
61. "Frequently Answered Questions" (http://opensource.org/faq#cc-zero). opensource.org. 21 October 2007. "CC0 was not explicitly rejected, but the License Review Committee was unable to reach consensus that it should be approved"
62. "License Approval" (https://docs.fedoraproject.org/en-US/legal/license-approval/). *Fedora Docs*.
63. "Creative Commons BY-SA 4.0 declared one-way compatible with GNU GPL version 3 — Free Software Foundation — Working together for free software" (https://www.fsf.org/blogs/licensing/creative-commons-by-sa-4-0-declared-one-way-compatible-with-gnu-gpl-version-3).
64. Free Software Foundation. "Educational Community License 2.0" (https://www.gnu.org/philosophy/license-list.html#ECL2.0). *Licenses*. Free Software Foundation.
65. https://www.gnu.org/licenses/ : "We use only licenses that are compatible with the GNU GPL for GNU software."
66. "Frequently Asked Questions about the GNU Licenses – Is GPLv3 compatible with GPLv2?" (https://www.gnu.org/licenses/gpl-faq.html#v2v3Compatibility). gnu.org. Retrieved 3 June 2014. "*No. Some of the requirements in GPLv3, such as the requirement to provide Installation Information, do not exist in GPLv2. As a result, the licenses are not compatible: if you tried to combine code released under both these licenses, you would violate section 6 of GPLv2. However, if code is released under GPL "version 2 or later," that is compatible with GPLv3 because GPLv3 is one of the options it permits.*"
67. "Re: Proposed statement WRT GNU FDL" (https://lists.debian.org/debian-legal/2003/04/msg00258.html).

68. "SPDX License List | Software Package Data Exchange (SPDX)" (https://spdx.org/licenses/).
69. "General Resolution: Why the GNU Free Documentation License is not suitable for Debian main" (https://www.debian.org/vote/2006/vote_001). *www.debian.org*.
70. Free Software Foundation. "A Quick Guide to GPLv3" (http://www.fsf.org/licensing/licenses/quick-guide-gplv3.html#new-compatible-licenses). *Licenses*. Free Software Foundation.
71. Mozilla Foundation. "MPL 2.0 FAQ" (https://www.mozilla.org/MPL/2.0/FAQ.html). *Licenses*. Mozilla Foundation.
72. "Various Licenses and Comments about Them - GNU Project - Free Software Foundation" (https://www.gnu.org/licenses/license-list.en.html#Unlicense). *www.gnu.org*.
73. "[License-review] Request for legacy approval: The Unlicense" (http://lists.opensource.org/pipermail/license-review_lists.opensource.org/2020-June/004890.html). *lists.opensource.org*.
74. "OSI Board Meeting Minutes, Wednesday, March 4, 2009" (https://web.archive.org/web/20160316070255/https://opensource.org/minutes20090304). 4 May 2009. Archived from the original (http://www.opensource.org/minutes20090304) on 16 March 2016. Retrieved 11 June 2010.
75. Free Software Foundation. "XFree86 1.1 License" (https://www.gnu.org/philosophy/license-list.html#XFree861.1License). *Licenses*. Free Software Foundation.

Contributor License Agreement

A **Contributor License Agreement** (**CLA**) defines the terms under which intellectual property has been contributed to a company/project, typically software under an open source license.

Rationale

CLAs can be used to enable vendors to easily pursue legal resolution in the case of copyright disputes,[1] or to relicense products to which contributions have been received from third parties.[2] CLAs are important especially for corporate open source projects under a copyleft license, since without a CLA the contribution would restrict the guardian as well.

The purpose of a CLA is to ensure that the guardian of a project's outputs has the necessary ownership or grants of rights over all contributions to allow them to distribute under the chosen license, often by granting an irrevocable license to allow the project maintainer to use the contribution; if copyright is actually transferred, the agreement is more normally known as a Copyright Transfer Agreement. CLAs also have roles in raising awareness of IPR issues within a project.[3]

Relicensing controversy

When a CLA requires a contributor to assign unrestricted republishing rights to the project, contributed code can be relicensed at the discretion of the project, even when the CLA does not assign copyright to the project. Prominent open source advocates regard CLAs as dangerous to open source rights.[4]

Examples

In 2019 MongoDB used these rights granted by its CLA to achieve a move to a non-open-source license.[5]

In January 2021, the Elasticsearch project used such rights to move the project to a non-open-source license [6] arguing Amazon had been "misleading and confusing the community".

> [...] Our license change is aimed at preventing companies from taking our Elasticsearch and Kibana products and providing them directly as a service without collaborating with us.
>
> Our license change comes after years of what we believe to be Amazon/AWS misleading and confusing the community - enough is enough.
>
> We’ve tried every avenue available including going through the courts, but with AWS’s ongoing behavior, we have decided to change our license so that we can focus on building products and innovating rather than litigating [...] [7]

Drew DeVault, a lead developer with a number of open source projects such as Sway, regards this move as a loophole. Both these projects were licensed under a copyleft license, which uses copyright to protect contributions, yet the CLA negates the usefulness of copyright in achieving this protection:

> Elasticsearch belongs to its 1,573 contributors, who retain their copyright, and granted Elastic a license to distribute their work without restriction. This is the loophole which Elastic exploited when they decided that Elasticsearch would no longer be open source [...][8]

CLAs which restrict relicensing

Project Harmony

Project Harmony was established by Canonical in 2010 to optionally avoid the problems discussed above. It provides a CLA template-builder. Based on choices made, the CLA will allow the contributor to keep copyright and assign rights to the project (as above) but with various restrictions on relicensing: using the template requires choosing one of the mutually-exclusive options, which range in restrictiveness.[9] A real-world example is the Ubuntu project. The CLA leaves copyright with the contributor and allows the project to relicense the code but with a restriction based on the license the contribution was made under:

> 2.3 Outbound License Based on the grant of rights in Sections 2.1 and 2.2, if We include Your Contribution in a Material, We may license the Contribution under any license, including copyleft, permissive, commercial, or proprietary licenses. As a condition on the exercise of this right, We agree to also license the Contribution under the terms of the license or licenses which We are using for the Material on the Submission Date.[10]

Fedora

The Fedora Project formerly required contributors to sign a CLA, either as an organization or as an individual.[11] However, this was retired in 2011[12] and instead contributors must agree to the Fedora Project Contributor Agreement (https://docs.fedoraproject.org/en-US/legal/fpca/), which is not a license agreement and does not include assignment of copyright.

Users

Companies and projects that use CLAs include:

- .NET Foundation[13]
- Apache Software Foundation[14]
- Canonical Ltd[15]
- Clojure[16]
- Cloud Native Computing Foundation[17]
- CyanogenMod[18]

- Diaspora[19]
- Digia/Qt Project[20]
- Discourse[21]
- Django[22]
- Dojo Toolkit[23]
- eBay Software Foundation, LLC[24][25] Subsidiary of eBay
- Eclipse[26]
- Elastic[27]
- Facebook[28]
- Go[29]
- Google[30]
- HashiCorp[31]
- Home Assistant[32]
- InfluxDB[33]
- Joomla[34]
- jQuery[35]
- Kubernetes[36]
- OpenBMC[37]
- Python[38]
- Meteor[39]
- Microsoft[40]
- MuseScore[41]
- OpenMediaVault[42]
- OpenStack[43]
- Puppet[44]
- Salesforce[45]
- Sangoma[46]
- TiddlyWiki[47]
- TLDR Pages[48]
- W3C[49]
- Zend Technologies for Zend Framework (1.x series only)

KDE

KDE uses Free Software Foundation Europe's Fiduciary Licence Agreement[50] of which (FLA-1.2) states in section 3.3:

> FSFE shall only exercise the granted rights and licences in accordance with the principles of Free Software as defined by the Free Software Foundations. FSFE guarantees to use the rights and licences transferred in strict accordance with the regulations imposed by Free Software licences, including, but not limited to, the GNU General Public Licence (GPL) or the GNU

> Lesser General Public Licence (LGPL) respectively. In the event FSFE violates the principles of Free Software, all granted rights and licences shall automatically return to the Beneficiary and the licences granted hereunder shall be terminated and expire.[51]

However, it is optional and every contributor is allowed not to assign their copyright to KDE e.V.

See also

- Developer Certificate of Origin
- Know your customer

References

1. "Licenses" (https://www.apache.org/licenses/). *www.apache.org*.
2. "Contributing FAQ" (https://web.archive.org/web/20071209175149/http://forge.mysql.com/wiki/ContributingFAQ). *mysql.com*. Archived from the original (http://forge.mysql.com/wiki/ContributingFAQ) on 2007-12-09. Retrieved 2007-12-22.
3. "Contributor Licence Agreements" (http://www.oss-watch.ac.uk/resources/cla.xml). Oss-watch.ac.uk. Retrieved 2014-01-24.
4. comments, 28 Feb 2019 Richard FontanaFeed 291up 2. "Why CLAs aren't good for open source" (https://opensource.com/article/19/2/cla-problems). *Opensource.com*. Retrieved 2021-01-26.
5. Corbet, Jonathan (2018-10-18). "Making the GPL more scary" (https://lwn.net/Articles/768670/). *LWN.net*. Retrieved 2023-04-29.
6. "Upcoming licensing changes to Elasticsearch and Kibana" (https://web.archive.org/web/20210114141243/https://www.elastic.co/blog/licensing-change). *www.elastic.co*. 2021-01-14. Archived from the original (https://www.elastic.co/blog/licensing-change) on 2021-01-14. Retrieved 2023-04-29.
7. "Amazon: NOT OK - why we had to change Elastic licensing" (https://web.archive.org/web/20230330011349/https://www.elastic.co/blog/why-license-change-aws). *www.elastic.co*. 2021-01-14. Archived from the original (https://www.elastic.co/blog/why-license-change-aws) on 2023-03-30. Retrieved 2023-04-29.
8. DeVault, Drew (2021-01-19). "Elasticsearch does not belong to Elastic" (https://drewdevault.com/2021/01/19/Elasticsearch-does-not-belong-to-Elastic.html). *drewdevault.com*. Retrieved 2023-04-29.
9. "Guide to the CAs | Harmony Agreements" (http://harmonyagreements.org/guide.html). *harmonyagreements.org*. Retrieved 2021-01-26.
10. "Contributor agreement form | Contributors" (https://ubuntu.com/legal/contributors/agreement). *Ubuntu*. Retrieved 2021-01-26.
11. "Legal:Licenses/CLA - Fedora Project Wiki" (https://fedoraproject.org/wiki/Legal:Licenses/CLA). *fedoraproject.org*. Retrieved 2023-01-29.
12. "Meeting:Board meeting 2011-04-05 - Fedora Project Wiki" (https://fedoraproject.org/wiki/Meeting:Board_meeting_2011-04-05#FPCA_Update). *fedoraproject.org*. Retrieved 2023-01-29.
13. "Contribution License Agreement" (http://cla.dotnetfoundation.org/). .NET Foundation.org. Retrieved 2015-01-11.
14. "Licenses" (https://www.apache.org/licenses/#clas). Apache.org. Retrieved 2014-01-24.

15. "Canonical contributor license agreement" (http://www.canonical.com/contributors). *Canonical Ubuntu*. Retrieved 2014-01-24.
16. "Clojure - Contributor Agreement" (https://clojure.org/dev/contributor_agreement).
17. "CNCF Linux Foundation ID CLA signing FAQ" (https://identity.linuxfoundation.org/projects/cncf). Retrieved 2020-10-19.
18. "Individual Contributor License Grant" (https://web.archive.org/web/20160308140334/http://review.cyanogenmod.org/static/cla_individual.html). CyanogenMod. Archived from the original (http://review.cyanogenmod.org/static/cla_individual.html) on 2016-03-08.
19. "diaspora/diaspora · GitHub" (https://github.com/diaspora/diaspora). Github.com. Retrieved 2014-01-24.
20. "Legal Aspects" (https://qt-project.org/legal.html). Qt Project. Retrieved 2014-01-24.
21. "CLA assistant" (https://www.discourse.org/cla).
22. "Contributor License Agreements | Django" (https://www.djangoproject.com/foundation/cla/). Djangoproject.com. Retrieved 2014-01-24.
23. "CLA" (http://dojofoundation.org/about/cla). The Dojo Foundation. Retrieved 2014-01-24.
24. "SeLion : Individual Contributor License Agreement (CLA)" (https://docs.google.com/forms/d/1t_Rqly6-qVP536O8ESltIRCHM0qBsmahWUdJda0oo3U/viewform). Retrieved 2015-02-23.
25. "LIST OF SUBSIDIARIES AS OF DECEMBER 31, 2014" (https://www.sec.gov/Archives/edgar/data/1065088/000106508815000054/ebay123114_10-kex2101.htm). Retrieved 2015-02-23.
26. "Eclipse Foundation Contributor License Agreement" (http://www.eclipse.org/legal/CLA.php). Eclipse.org. 2013-06-17. Retrieved 2014-01-24.
27. "Contributor Agreement Elastic" (https://www.elastic.co/contributor-agreement). elastic.co.
28. "Contributing to Facebook Projects" (https://code.facebook.com/cla). facebook.com. Retrieved 2015-07-08.
29. "CONTRIBUTORS - The Go Programming Language" (https://golang.org/CONTRIBUTORS). Golang.org. Retrieved 2014-01-24.
30. "Google Individual Contributor License Agreement, v1.1" (https://developers.google.com/open-source/cla/individual?csw=1). Google Inc. Retrieved 2014-06-09.
31. Hashicorp. " "Contributor License Agreement" " (https://www.hashicorp.com/cla). *Hashicorp: Infrastructure Enables Innovation*. Retrieved 2021-08-09.
32. Schoutsen, Paulus (2017-01-21). "Home Assistant Governance [updated]" (https://www.home-assistant.io/blog/2017/01/21/home-assistant-governance/). *Home Assistant*. Retrieved 2024-03-04.
33. "InfluxDB Contributor License Agreement" (http://influxdb.com/community/cla.html). Retrieved 25 August 2014.
34. "Contribute" (http://developer.joomla.org/contributor-agreements.html). Developer.joomla.org. Retrieved 2014-01-24.
35. jQuery Foundation - jquery.org. "jQuery Foundation Contributor License Agreement | Contribute to jQuery" (http://contribute.jquery.org/CLA/). Contribute.jquery.org. Retrieved 2014-01-24.
36. Cloud Native Computing Foundation. "Kubernetes CLA" (https://github.com/kubernetes/community/blob/master/CLA.md). CNCF. Retrieved 2019-02-19.
37. *GitHub - openbmc/docs* (https://github.com/openbmc/docs), openbmc, 2019-02-20, retrieved 2019-02-21
38. "Contributing to the PSF" (https://www.python.org/psf/contrib/). Python Software Foundation.
39. "Contributing to Meteor" (https://contribute.meteor.com/). Contribute.meteor.com. Retrieved 2014-01-24.

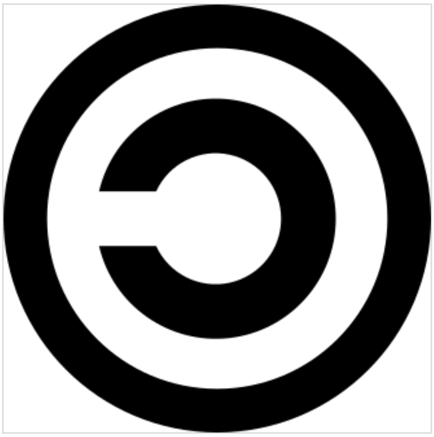
40. "Contributing to Microsoft Visual Studio Code" (https://github.com/Microsoft/vscode/wiki/Contributor-License-Agreement). Microsoft. Retrieved 2016-04-22.
41. "Contributor License Agreement | MuseScore" (https://musescore.org/en/cla). *musescore.org*. Retrieved 2019-08-30.
42. "Contributor Agreement" (http://openmediavault.org/contributoragreement.html). OpenMediaVault. Retrieved 2014-01-24.
43. "OpenStack Contributors License Agreement" (https://wiki.openstack.org/wiki/How_To_Contribute#Contributors_License_Agreement). OpenStack. Retrieved 2014-01-24.
44. "Puppet Contributor License Agreement" (https://cla.puppet.com/cla-text). Retrieved 2016-02-01.
45. "Salesforce Contributor License Agreement" (https://web.archive.org/web/20200728073423/https://cla.salesforce.com/status/salesforce/observable-membrane/pull/48). Archived from the original (https://cla.salesforce.com/status/salesforce/observable-membrane/pull/48) on 2020-07-28. Retrieved 2020-05-21.
46. "CONTRIBUTOR LICENSE AGREEMENT" (https://web.archive.org/web/20230429235353/https://oss-cla.sangoma.com/asterisk/asterisk). *www.sangoma.com*. 2023-04-26. Archived from the original (https://oss-cla.sangoma.com/asterisk/asterisk) on 2023-04-29. Retrieved 2023-04-29.
47. "License" (https://en.wikipedia.org/wiki/TiddlyWiki#License). *TiddlyWiki*. 19 December 2022.
48. "TLDR-pages/TLDR" (https://github.com/tldr-pages/tldr/blob/master/CONTRIBUTING.md). *GitHub*. 26 February 2022.
49. "W3C Community Contributor License Agreement (CLA)" (https://www.w3.org/community/about/process/cla/). Community and Business Groups. *W3C*. Retrieved 2024-05-11.
50. "FSFE welcomes KDE's adoption of the Fiduciary Licence Agreement (FLA)" (http://ev.kde.org/announcements/2008-08-22-fsfe-welcomes-fla.php). 2008-08-22.
51. "Fiduciary Licence Agreement (Version 1.2)" (http://www.fsfe.org/projects/ftf/FLA.en.pdf) (PDF). Retrieved 2010-12-31.

External links

- Contributor Licence Agreements (http://www.oss-watch.ac.uk/resources/cla.xml) from OSS Watch
- ContributorAgreements.org (http://contributoragreements.org)
- CLA assistant (https://cla-assistant.io) — enables contributors to sign CLAs from within a pull request

WIKIPEDIA
The Free Encyclopedia

Copyleft


Copyleft symbol

Copyleft is the legal technique of granting certain freedoms over copies of copyrighted works with the requirement that the same rights be preserved in derivative works. In this sense, *freedoms* refers to the use of the work for any purpose, and the ability to modify, copy, share, and redistribute the work, with or without a fee. Licenses which implement copyleft can be used to maintain copyright conditions for works ranging from computer software, to documents, art, and scientific discoveries, and similar approaches have even been applied to certain patents.[1]

Copyleft software licenses are considered *protective* or *reciprocal* in contrast with permissive free software licenses,[2] and require that information necessary for reproducing and modifying the work must be made available to recipients of the software program, which are often distributed as executables. 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. Free and open copyleft licenses ensure everyone's rights to freely use the product but it prohibits owning, registering copyright and earning royalties from copyright. In contrast to those commercial and non free copyleft licenses restrict certain usage scenarios.

Notable free and open 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;[3] the Mozilla Public License; the Free Art License;[4] and the Creative Commons share-alike license condition[5]—with the last two being intended for non-software works, such as documents and pictures, both academic or artistic in nature. Wikipedia is copyleft under the Creative Commons Attribution-ShareAlike license.

Examples for commercial and non-free copyleft licenses are the Server Side Public License and Open Token Compensation License.[6]

History

Li-Chen Wang's Palo Alto Tiny BASIC for the Intel 8080 appeared in *Dr. Dobb's Journal* in May 1976. The listing begins with the title, author's name, and date, but also has "@COPYLEFT ALL WRONGS RESERVED".[7]

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's motivation was that a few years earlier he had worked 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 worn out.[8][9]

As Stallman deemed it impractical in the short term to eliminate current copyright law and the wrongs he perceived it to perpetuate, he decided to work within the framework of existing law; in 1985,[10] he created his own copyright license, the Emacs General Public License,[11] 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.[12] 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", which is a pun on the common copyright disclaimer "all rights reserved".[12]

In France, a series of meetings taking place in 2000 under the title "Copyleft Attitude" gave birth to the Free Art License (FAL),[13] theoretically valid in any jurisdiction bound by the Berne Convention and recommended by Stallman's own Free Software Foundation.[14] Shortly thereafter, a separate, unrelated initiative in the United States yielded the Creative Commons license, available since 2001 in both permissive (BY) and copyleft (BY-SA) variants and more specifically tailored to U.S. law.

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 of the work the freedom to carry out all of these activities. These freedoms (from the Free Software Definition) include:[8][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.[8] 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.[8]

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.[8]

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-source code and then adding extra restrictions to it, an action prevented by copyleft-licensing the software. Some creators, such as Elastic,[17] feel that preventing commercial enterprises from using and then selling their product under a proprietary license is also an incentive.

Furthermore, the open-source culture of programming has been described as a gift economy, where social power is determined by an individual's contributions.[18] 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.[19]

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.[20] 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 forgo copyleft. The Free Software Foundation recommends that when "widespread use of the code is vital for advancing the cause of free software",[21] 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 exact or modified versions of that work, with or without a fee.[22][23]

Unlike similar permissive licenses that also grant these freedoms, copyleft licenses also ensure that any modified versions of a work covered by a copyleft license must also grant these freedoms. Thus, copyleft licenses have conditions: that modifications of any work licensed under a copyleft license 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 source licensed under a copyleft license for works they invent is free to choose any other license (or none at all) provided they meet the fair use standard.[24]

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.

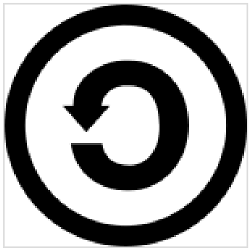
Some laws used for copyleft licenses vary from one country to another, and may also be granted in terms that vary from country to country. For example, in some countries, it is acceptable to sell a software product without warranty, in standard GNU General Public License style, while in most European 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 is specified in most European copyleft licenses, for example, the European Union Public Licence (EUPL),[25] or the CeCILL license,[26] a license that allows one to use GNU GPL in combination with a limited warranty.

For projects which will be run over a network, a variation of the GNU GPL, called the Affero General Public License (GNU AGPL), ensures that the source code is available to users of software over a network.

Types and relation to other licenses

	Free			Non-free			
	Public domain & equivalents	**Permissive license**	**Copyleft (protective license)**	**Noncommercial license**	**Proprietary license**	**Trade secret**	**Commercial Copyleft**
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; certain rights may or may not be granted	No information made public	Grants use rights, but restricts them to certain use cases
For software	PD, Unlicense, CC0	BSD, MIT, Apache	GPL, AGPL	JRL, AFPL	Proprietary software, no public license	Private, internal software	SSPL, OCTL[6]
For other creative works	PD, CC0	CC BY	CC BY-SA, FAL	CC BY-NC	Copyright, no public license, CC BY-ND	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 proprietization) or just the initial recipient (freedom to proprietize). However, current copyright law and the availability of both types of licenses, copyleft and permissive, allow authors to choose the type under which to license the works they invent.



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

For documents, art, and other works other than software and code, the Creative Commons share-alike licensing system and the GNU Free Documentation License (GFDL) allow authors to apply limitations to certain sections of their work, exempting some parts of the work 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 strength of the copyleft license governing a work is determined by the extent to which its provisions can be imposed on all kinds of derivative works. Thus, the term "weak copyleft" refers to licenses where not all derivative works inherit the copyleft license; whether a derivative work inherits or not often depends on how it was derived.

"Weak copyleft" licenses are often used to cover software libraries. This allows other software to link to the library and be redistributed without the requirement for the linking software to also be licensed under the same terms. Only changes to the software licensed under a "weak copyleft" license become subject itself 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.[27] Free-software licenses that use "weak" copyleft include the GNU Lesser General Public License and the Mozilla Public License.

The GNU General Public License is an example of a license implementing strong copyleft. An even stronger copyleft license is the AGPL, which requires the publishing of the source code for software as a service use cases.[28][29][30][31]

The Sybase Open Watcom Public License is one of the strongest copyleft licenses, as this license closes the so-called "private usage" loophole of the GPL, and requires the publishing of source code in any use case. For this reason, the license is considered non-free by the Free Software Foundation, the GNU Project, and the Debian project.[32] However, the license is accepted as open source by the OSI.

The Design Science License (DSL) is a strong copyleft license that applies to any work, not only software or documentation, but also literature, artworks, music, photography, and video. DSL was written by Michael Stutz after he took an interest in applying GNU-style copyleft to non-software works, which later came to be called libre works. In the 1990s, it was used on music recordings, visual art, and even novels. It is not considered compatible with the GNU GPL by the Free Software Foundation.[33]

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 author but that the author will be recognized and ensure the same or compatible license terms cover the changed code.[34] Some Creative Commons licenses are examples of share-alike copyleft licenses.

Permissive licenses

Those licenses 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, Expat license, and the various BSD licenses.

Debate and controversy

It has been suggested that copyleft has become a divisive issue in the ideological strife between the Open Source Initiative and the free software movement.[35] 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.[36][33]
- 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.[37]
- Some of the software programs of the GNU Project are published under non-copyleft licenses.[38]
- Stallman has endorsed the use of non-copyleft licenses in certain circumstances, for example in the case of the Ogg Vorbis relicensing.[39]

"Viral" licensing

"**Viral license**" is a pejorative name for copyleft licenses.[40][41][42][43][44] 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.[45][46][47] The name 'viral license' refers to the fact that any works derived from a copyleft work must preserve the copyleft permissions when distributed.

Some advocates of the various BSD Licenses 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.[48][49][50] 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."[51] 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".[52]

In response to Microsoft's attacks on the GPL, several prominent free-software developers and advocates released a joint statement supporting the license.[53] According to FSF compliance engineer David Turner, the term "viral license" creates a misunderstanding and a fear of using copylefted free software.[54] While a person can catch a virus without active action, license conditions take effect upon effective usage or adoption.[55] 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."[56] 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."[57]

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.[58] 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 in ways such as these. This allowed communication may or may not include reusing libraries or routines via dynamic linking – some commentators say it does,[59] 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 works licensed under terms that are not compatible with the LGPL, with works covered by the LGPL.[60]

Symbol

The copyleft symbol is a mirrored version of the copyright symbol, ©: a reversed C in a circle.[61] A 2016 proposal to add the symbol to a future version of Unicode was accepted by the Unicode Technical Committee.[62][63] The code point U+1F12F 🄯 COPYLEFT SYMBOL was added in Unicode 11.[63][64] The copyleft symbol has no legal status.[65]

As of 2024, the symbol is generally provided as standard in the system fonts of most current operating systems, but if need be it may be approximated with character U+2184 ↄ LATIN SMALL LETTER REVERSED C between parenthesis (ↄ).

🄯 🄯

Copyleft symbol

In Unicode	U+1F12F 🄯 COPYLEFT SYMBOL Alternative symbol: (ↄ)
Different from	
Different from	U+00A9 © COPYRIGHT SIGN

Typing the character

On modern computer systems, the character U+1F12F 🄯 COPYLEFT SYMBOL can be generated using one of these methods (keyboard shortcuts):

- ChromeOS: Ctrl+⇧ Shift+u, 1F12F, then ↵ Enter or Space.[66]
- HTML: `&#127279 or &#x1F12F`
- Linux: Ctrl+⇧ Shift+u, 1F12F, then ↵ Enter or Space
- Mac: Code point: U+1F12F
- Windows: Alt+1F12F

See also

- List of copyleft software licenses
- All rights reversed – Pun indicating a release under copyleft licensing status
- Anti-copyright notice – Statement encouraging the unrestricted distribution of a work
- Commercial use of copyleft works – Ethical dilemma
- Comparison of open-source and closed-source software
- Copyfraud – False copyright claims to public-domain content
- Copyright abolition – Movement advocating to abolish copyright
- Copyright alternatives – Compensation systems for digital copying
- Criticism of intellectual property – Ownership of creative expressions and processes
- Internet freedom – Digital rights, freedom of information and the right to Internet access and speech
- Free content – Nonrestrictive creative work
- Free Culture movement – Social movement concerning creative works

- Free software movement – Social movement
- History of free and open-source software
- Criticism of copyright – Dissenting views of copyright law
- Public copyright license – Type of license
- Public domain – Works outside the scope of copyright law
- Free video – Video content that is free to use for any purpose
- *Steal This Film*
- In Praise of Copying – 2010 book by Marcus Boon

Free and open-source software portal

References

1. 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).
2. Troan, Larry (2006). *Open Source from a Proprietary Perspective* (https://web.archive.org/web/20140122163130/https://www.redhat.com/f/summitfiles/presentation/May31/Open%20Source%20Dynamics/Troan_OpenSourceProprietyPersp.pdf) (PDF). Red Hat Summit. Nashville: Red Hat. Archived from the original (https://www.redhat.com/f/summitfiles/presentation/May31/Open%20Source%20Dynamics/Troan_OpenSourceProprietyPersp.pdf) (PDF) on 22 January 2014.
3. Stallman, Richard (29 June 2007). "GNU General Public License" (https://www.gnu.org/copyleft/gpl.html). *GNU Project*. Retrieved 1 May 2017.
4. "Free Art License 1.3" (https://artlibre.org/licence/lal/en/). *Copyleft Attitude*. Retrieved 6 October 2021.
5. "Attribution-ShareAlike 4.0 International (CC BY-SA 4.0)" (https://creativecommons.org/licenses/by-sa/4.0/). *Creative Commons*. Retrieved 14 August 2015.
6. "Open Token Compensation License" (https://license-token.com/). *Open Token Compensation License homepage*. Retrieved 27 March 2025.
7. 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.
8. Carver, Brian W. (5 April 2005). "Share and Share Alike: Understanding and Enforcing Open Source and Free Software Licenses" (https://lawcat.berkeley.edu/record/1119855?ln=en). *Berkeley Technology Law Journal*. **20** (1): 443. doi:10.15779/Z388T19 (https://doi.org/10.15779%2FZ388T19). S2CID 153900462 (https://api.semanticscholar.org/CorpusID:153900462). SSRN 1586574 (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1586574). Retrieved 27 June 2023.
9. 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.
10. Moody, Glyn (2002). *Rebel Code*. p. 26.
11. "Emacs General Public License" (http://www.free-soft.org/gpl_history/emacs_gpl.html). 5 July 2001. Retrieved 23 August 2008.
12. Stallman, Richard (21 January 2008). "About the GNU Project" (https://www.gnu.org/gnu/thegnuproject.html). Free Software Foundation. Retrieved 1 May 2017.
13. "Free Art License – Frequently Asked Questions" (https://artlibre.org/faq_eng/). *Copyleft Attitude*. Retrieved 6 October 2021.
14. "Licenses" (https://www.gnu.org/licenses/licenses.html). *GNU Project*. Retrieved 6 October 2021.: "*We don't take the position that artistic or entertainment works must be free, but if you want to make one free, we recommend the Free Art License.*"
15. "What is free software?" (https://www.gnu.org/philosophy/free-sw.html). *GNU*. 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. The New York Times (https://www.nytimes.com/2019/12/15/technology/amazon-aws-cloud-competition.html). Retrieved 23 April 2022
18. 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.
19. 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.
20. 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.
21. "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.
22. "What is Copyleft?" (https://www.gnu.org/copyleft/). GNU Project.
23. "Scheibner, James – "What price freedom (of software)? A guide for Australian legal practitioners on open source licensing" [2017] PrecedentAULA 23; (2017) 139 Precedent 39" (http://classic.austlii.edu.au/au/journals/PrecedentAULA/2017/23.html). *classic.austlii.edu.au*. Retrieved 27 November 2021.
24. 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.
25. "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.
26. "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.
27. 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*.
28. 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.
29. 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.
30. "Why the Affero GPL" (https://www.gnu.org/licenses/why-affero-gpl.html). GNU Project. 10 May 2015. Retrieved 1 May 2017.
31. 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*".
32. "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. 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."
33. "Various Licenses and Comments about Them" (https://www.gnu.org/licenses/license-list.html#dsl). GNU Project.
34. Engelfriet, Arnoud (2010). "Choosing an Open Source License". *IEEE Software*. **27**. Athens University of Economics and Business: 48–49. doi:10.1109/MS.2010.5 (https://doi.org/10.1109%2FMS.2010.5). S2CID 36712745 (https://api.semanticscholar.org/CorpusID:36712745).
35. Biancuzzi, Federico (30 June 2005). "ESR: "We Don't Need the GPL Anymore" " (https://web.archive.org/web/20180306105402/http://www.onlamp.com/pub/a/onlamp/2005/06/30/esr_interview.html). ONLamp.com. Archived from the original (http://www.onlamp.com/pub/a/onlamp/2005/06/30/esr_interview.html) on 6 March 2018. Retrieved 23 August 2008.
36. Tiemann, Michael (18 September 2006). "Licenses by Name" (http://www.opensource.org/licenses/alphabetical). Retrieved 23 August 2008.
37. Raymond, Eric Steven (9 November 2002). "Licensing HOWTO" (http://www.catb.org/~esr/Licensing-HOWTO.html#choosing). Retrieved 23 August 2008.
38. "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.

39. 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."
40. "Microsoft license spurns open source" (http://news.cnet.com/2100-1001-268889.html). *CNET*. CBS Interactive.
41. "Some rights reserved: the alternatives to copyright" (https://www.wired.co.uk/news/archive/2011-12/16/alternatives-to-copyright). *Wired UK*.
42. "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.
43. 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.
44. 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.
45. 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.
46. "General Public Virus" (http://catb.org/esr/jargon/oldversions/jarg221.txt). *Jargon File 2.2.1*. 15 December 1990. Retrieved 29 April 2007.
47. 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.
48. 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.
49. 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.
50. Griffis, L. Adrian (15 July 2000). "The GNU Public Virus" (https://web.archive.org/web/20171109004114/http://themes.freshmeat.net/articles/view/172#comment-5548). Archived from the original (http://themes.freshmeat.net/articles/view/172#comment-5548) on 9 November 2017. Retrieved 29 April 2007.
51. 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.
52. 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)
53. *Free Software Leaders Stand Together* (https://en.wikisource.org/wiki/Free_Software_Leaders_Stand_Together) – via Wikisource.
54. 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.
55. "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*. 3 February 2012. Retrieved 22 August 2018.
56. McGowan, David (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.
57. 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.
58. "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.
59. Raymond, Eric Steven (9 November 2002). "Licensing HOWTO" (http://www.catb.org/~esr/Licensing-HOWTO.html#id2789302). Retrieved 21 March 2010.
60. 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.
61. "The Unicode Standard, Version 15.0: Enclosed Alphanumeric Supplement" (https://www.unicode.org/charts/PDF/U1F100.pdf) (PDF). *unicode.org*.

62. "Proposal to add the Copyleft Symbol to Unicode" (https://www.unicode.org/L2/L2016/16059-copyleft.pdf) (PDF).
63. "Proposed New Characters: Pipeline Table" (http://unicode.org/alloc/Pipeline.html). *Unicode Character Proposals*. Unicode Consortium. Retrieved 18 April 2017.
64. "Announcing The Unicode Standard, Version 11.0" (http://blog.unicode.org/2018/06/announcing-unicode-standard-version-110.html). *The Unicode Blog*. Retrieved 6 June 2018.
65. 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.
66. "Unicode Character "🄯" (U+1F12F)" (https://www.compart.com/en/unicode/U+1F12F). *www.compart.com*. Retrieved 12 September 2024.

External links

- Media related to Copyleft at Wikimedia Commons
- The dictionary definition of *copyleft* at Wiktionary

The Open Source Definition

(Redirected from Debian Free Software Guidelines)

The Open Source Definition (**OSD**) is a document published by the Open Source Initiative. Derived from Bruce Perens' Debian Free Software Guidelines, the definition is the most common standard for open-source software. The definition has ten criteria, such as requiring freely accessed source code and granting the open-source rights to everyone who receives a copy of the program. Covering both copyleft and permissive licenses, it is effectively identical to the definition of free software, but motivated by more pragmatic and business-friendly considerations. The Open Source Initiative's board votes on proposals of licenses to certify that they are compliant with the definition, and maintains a list of compliant licenses on its website. The definition has been adapted into the Open Knowledge Foundation's Open Definition for open knowledge and into open hardware definitions.

History

There have been several attempts to define open source and free software. Amongst the earliest was Free Software Foundation's Free Software Definition, which then defined as the three freedoms of Free Software (Freedom Zero was added later). Published versions of FSF's Free Software Definition existed as early as 1986, having been published in the first edition of the (now defunct) GNU's Bulletin.[1]

Debian Free Software Guidelines

The **Debian Free Software Guidelines** (**DFSG**) was first published together with the first version of the Debian Social Contract in July 1997.[2] The primary author was Bruce Perens, with input from the Debian developers during a month-long discussion on a private mailing list, as part of the larger Debian Social Contract. Perens was copied to an email discussion between Ean Schuessler (then of Debian) and Donnie Barnes of Red Hat, in which Schuessler accused Red Hat of never elucidating its social contract with the Linux community. Perens realized that Debian did not have any formal social contract either, and immediately started creating one. The (then) Three Freedoms, which preceded the drafting and promulgation of the DFSG, were unknown to its authors.[3]

The guidelines were:

1. Free redistribution.
2. Inclusion of source code.
3. Allowing for modifications and derived works.
4. Integrity of the author's source code (as a compromise).
5. No discrimination against persons or groups.
6. No discrimination against fields of endeavor, like commercial use.
7. The license needs to apply to all to whom the program is redistributed.
8. License must not be specific to a product.
9. License must not restrict other software.

10. Example licenses: The GNU GPL, BSD, and Artistic licenses are examples of licenses considered free.[2][4]

Open source

As Netscape released the open-source Mozilla browser in 1998, Bruce Perens again drafted a set of open-source guidelines to go with the release.[5] It has been claimed that the Open Source Definition was created by re-titling the exact text of the DFSG.

A modified version of this definition was adopted by the Open Source Initiative (OSI) as the Open Source Definition.[6][7] The OSI uses the label "open source", rather than "free software", because it felt that the latter term had undesirable ideological and political freight, and it wanted to focus on the pragmatic and business-friendly arguments for open-source software.[6] It adopted a closed rather than membership-driven organizational model in order to draft the definition and work together with a wider variety of stakeholders than other free or open-source projects.[6]

Once the DFSG became the Open Source Definition, Richard Stallman saw the need to differentiate free software from open source and promoted the Free Software Definition.[8]

Debian diverges

In November 1998, Ian Jackson and others proposed several changes in a draft versioned 1.4, but the changes were never made official. Jackson stated[9] that the problems were "loose wording" and the patch clause.

The Debian General Resolution 2004-003,[10] titled "Editorial amendments to the social contract", modified the Social Contract. The proposer Andrew Suffield stated:[11]

> "The rule is 'this resolution only changes the letter of the law, not the spirit'. Mostly it changes the wording of the social contract to better reflect what it is supposed to mean, and this is mostly in light of issues that were not considered when it was originally written."

However, the change of the sentence "We promise to keep the Debian GNU/Linux Distribution entirely free software" into "We promise that the Debian system and all its components will be free" resulted in the release manager, Anthony Towns, making a practical change:[12]

> "As [SC #1] is no longer limited to 'software', and as this decision was made by developers after and during discussion of how we should consider non-software content such as documentation and firmware, I don't believe I can justify the policy decisions to exempt documentation, firmware, or content any longer, as the Social Contract has been amended to cover all these areas."

This prompted another General Resolution, 2004–004,[13] in which the developers voted overwhelmingly against immediate action, and decided to postpone those changes until the next release (whose development started a year later, in June 2005).

Criteria

Providing access to the source code is not enough for software to be considered "open-source".[14] The Open Source Definition requires that ten criteria be met:[15][6]

1. Free redistribution[15]
2. Source code must be accessible and the license must permit redistribution in the form of source code (rather than object code).[15] In order to modify the software, access to source code is required.[16]
3. Derivative works must be allowed and able to be redistributed under the same licensing terms as the open-source product[15]
4. The license may require that the original software be distributed intact, but only if modifications are able to be distributed as patches without restriction.[15][16]
5. No discrimination between users[15]
6. No discrimination between uses, including commercial use[15]
7. Everyone who receives a copy of the program is granted all the open-source rights[15]
8. The license must cover all the code, not a particular product or distribution.[15][16]
9. There may not be restrictions on other software distributed at the same time[15]
10. Technological neutrality—cannot restrict use to any particular technology.[15] For example, a license that requires a user to click a box agreeing to it is not free because the work cannot be distributed as a paper copy.[16]

The Open Source Definition is available under a Creative Commons (CC BY 4.0) license.[17] It covers both copyleft—where redistribution and derivative works must be released under a free license—and permissive licenses—where derivative works can be released under any license. It is part of the open source movement rather than the free software movement, and seeks to promote the availability of open-source software for anyone seeking to reuse it, even the makers of proprietary software.[6][18][16] It does not address warranty disclaimers, although these are very common in open-source software.[16] The definition does not specify a governance structure for open-source projects.[6]

Compliant licenses

The criteria are used by the OSI to approve certain licenses as compatible with the definition, and maintain a list of compliant licenses. New licenses have to submit a formal proposal that is discussed by the OSI mailing list before it is approved or rejected by the OSI board. Seven approved licenses are particularly recommended by the OSI as "popular, widely used, or having strong communities":[19]

- Apache License 2.0
- BSD 3-Clause and BSD 2-Clause Licenses
- All versions of the GNU General Public License
- All versions of the GNU Lesser Public License
- MIT License
- Mozilla Public License 2.0
- Common Development and Distribution License (CDDL)
- Eclipse Public License version 2.0

Application

Software

Most discussions about the DFSG happen on the *debian-legal* mailing list. When a Debian Developer first uploads a package for inclusion in Debian, the *ftpmaster* team checks the software licenses and determines whether they are in accordance with the social contract. The team sometimes confers with the debian-legal list in difficult cases.

Non-"software" content

The DFSG is focused on software, but the word itself is unclear—some apply it to everything that can be expressed as a stream of bits, while a minority considers it to refer to just computer programs. Also, the existence of PostScript, executable scripts, sourced documents, etc., greatly muddies the second definition. Thus, to break the confusion, in June 2004 the Debian project decided to explicitly apply the same principles to software documentation, multimedia data and other content. The non-program content of Debian began to comply with the DFSG more strictly in Debian 4.0 (released in April 2007) and subsequent releases.

GFDL

Much documentation written by the GNU Project, the Linux Documentation Project and others licensed under the GNU Free Documentation License contain invariant sections, which do not comply with the DFSG. This assertion is the end result of a long discussion and the General Resolution 2006-001.[20]

Due to the GFDL invariant sections, content under this license must be separately contained in an additional "non-free" repository which is not officially considered part of Debian.

Multimedia files

It can be sometimes hard to define what constitutes the "source" for multimedia files, such as whether an uncompressed image file is the source of a compressed image and whether the 3D model before ray tracing is the source for its resulting image.

debian-legal tests for DFSG compliance

The *debian-legal* mailing list subscribers have created some tests to check whether a license violates the DFSG. The common tests (as described in the draft DFSG FAQ)[21] are the following:

- "The Desert Island test". Imagine a castaway on a desert island with a solar-powered computer. This would make it impossible to fulfill any requirement to make changes publicly available or to send patches to some particular place. This holds even if such requirements are only upon request, as the castaway might be able to receive messages but be unable to send them. To be free, software must be modifiable by this unfortunate castaway, who must also be able to legally share modifications with friends on the island.

- "The Dissident test". Consider a dissident in a totalitarian state who wishes to share a modified bit of software with fellow dissidents, but does not wish to reveal the identity of the modifier, or directly reveal the modifications themselves, or even possession of the program, to the government. Any requirement for sending source modifications to anyone other than the recipient of the modified binary—in fact, any forced distribution at all, beyond giving source to those who receive a copy of the binary—would put the dissident in danger. For Debian to consider software free it must not require any such excess distribution.
- "The Tentacles of Evil test". Imagine that the author is hired by a large evil corporation and, now in their thrall, attempts to do the worst to the users of the program: to make their lives miserable, to make them stop using the program, to expose them to legal liability, to make the program non-free, to discover their secrets, etc. The same can happen to a corporation bought out by a larger corporation bent on destroying free software in order to maintain its monopoly and extend its evil empire. To be free, the license cannot allow even the author to take away the required freedoms.

Reception

The Open Source Definition is the most widely used definition for open-source software,[22] and is often used as a standard for whether a project is open source.[17] It and the official definitions of free software by the Free Software Foundation (FSF) essentially cover the same software licenses.[6][23] Nevertheless, there is a values difference between the free software and open source movements: the former is more based on ethics and values, the latter on pragmatism.[6]

Derived definitions

The Open Knowledge Foundation's Open Definition is substantially derivative of the Open Source Definition.[24]

The Open Source Hardware Statement of Principles is adapted from the Open Source Definition.[25][22]

See also

Free and open-source software portal

- Comparison of free and open-source software licenses
- History of free and open-source software
- The Free Software Definition

References

1. Richard M. Stallman, *What is the Free Software Foundation?*, GNU's Bulletin, Volume 1, No.1, February 1986 (https://www.gnu.org/bulletins/bull1.txt)
2. Bruce Perens (1997-07-04). "Debian's "Social Contract" with the Free Software Community" (http://lists.debian.org/debian-announce/debian-announce-1997/msg00017.html). *debian-announce mailing list*.

3. Bruce Perens: "when I had to write license guidelines for Debian, the Four Freedoms document was unknown. (http://news.slashdot.org/comments.pl?sid=1129863&cid=26875815)"
4. "Debian Social Contract" (http://www.debian.org/social_contract#guidelines). Debian. 2004-04-26.
5. Overly, Michael R. (2003). *The Open Source Handbook*. Pike & Fischer. p. 5. ISBN 978-0-937275-12-2.
6. Gardler, Ross; Walli, Stephen R (2022). "Evolving Perspective on Community and Governance". *Open Source Law, Policy and Practice*. Oxford University PressOxford. p. 47–48, 52. doi:10.1093/oso/9780198862345.003.0002 (https://doi.org/10.1093%2Foso%2F9780198862345.003.0002). ISBN 978-0-19-886234-5.
7. Katz, Andrew (2022). "Everything Open" (https://academic.oup.com/book/44727/chapter/378969640). *Open Source Law, Policy and Practice*. Oxford University Press. p. 521. ISBN 978-0-19-260687-7.
8. Richard Stallman. "Why "Open Source" misses the point of Free Software" (https://www.gnu.org/philosophy/open-source-misses-the-point.html). *GNU website*.
9. Ian Jackson: Draft new DFSG (http://lists.debian.org/debian-devel/1998/11/msg01944.html), debian-devel mailing list
10. General Resolution: Editorial amendments to the social contract (http://www.debian.org/vote/2004/vote_003)
11. Andrew Suffield: Re: Candidate social contract amendments (part 1: editorial) (3rd draft) (http://lists.debian.org/debian-vote/2004/01/msg00692.html), debian-vote mailing list
12. Anthony Towns: Social Contract GR's effect on Sarge (http://lists.debian.org/debian-devel/2004/04/msg01929.html), debian-devel mailing list
13. General Resolution: Sarge Release Schedule in view of GR 2004-003 (http://www.debian.org/vote/2004/vote_004)
14. Greenleaf, Graham; Lindsay, David (2018). *Public Rights: Copyright's Public Domains*. Cambridge University Press. p. 485. ISBN 978-1-107-13406-5.
15. Erlich, Zippy (2007). "Open Source Software". *Handbook of Research on Open Source Software*. IGI Global. pp. 187–188. ISBN 978-1591409991.
16. Laurent, Andrew M. St (2004). *Understanding Open Source and Free Software Licensing: Guide to Navigating Licensing Issues in Existing & New Software*. O'Reilly Media, Inc. pp. 9–11. ISBN 978-0-596-55395-1.
17. Mertic, John (2023). *Open Source Projects - Beyond Code: A blueprint for scalable and sustainable open source projects*. Packt Publishing Ltd. p. 5. ISBN 978-1-83763-385-2.
18. Meeker, Heather J. (2008). *The Open Source Alternative: Understanding Risks and Leveraging Opportunities*. John Wiley & Sons. pp. 21–22. ISBN 978-0-470-25581-0.
19. Smith, P McCoy (2022). "Copyright, Contract, and Licensing in Open Source". *Open Source Law, Policy and Practice*. Oxford University PressOxford. pp. 108–111. doi:10.1093/oso/9780198862345.003.0003 (https://doi.org/10.1093%2Foso%2F9780198862345.003.0003). ISBN 978-0-19-886234-5.
20. General Resolution: Why the GNU Free Documentation License is not suitable for Debian main (http://www.debian.org/vote/2006/vote_001)
21. The Debian Free Software FAQ (http://people.debian.org/~bap/dfsg-faq.html)
22. De Maria, Carmelo; Díaz Lantada, Andrés; Di Pietro, Licia; Ravizza, Alice; Ahluwalia, Arti (2022). "Open-Source Medical Devices: Concept, Trends, and Challenges Toward Equitable Healthcare Technology". *Engineering Open-Source Medical Devices*. Cham: Springer International Publishing. p. 4. doi:10.1007/978-3-030-79363-0_1 (https://doi.org/10.1007%2F978-3-030-79363-0_1). ISBN 978-3-030-79362-3.

23. Kelty, Christpher M. (2008). "The Cultural Significance of free Software – Two Bits" (http://twobits.net/pub/Kelty-TwoBits.pdf) (PDF). Duke University Press. p. 99. Archived (https://web.archive.org/web/20160304070704/http://twobits.net/pub/Kelty-TwoBits.pdf) (PDF) from the original on 2016-03-04. Retrieved 2016-02-24.
24. Martin, Victoria (2022). *The Complete Guide to Open Scholarship*. Bloomsbury Publishing. p. 27. ISBN 979-8-216-06415-2.
25. Bonvoisin, Jérémy; Mies, Robert; Boujut, Jean-François; Stark, Rainer (2017). "What is the "Source" of Open Source Hardware?" (https://account.openhardware.metajnl.com/index.php/up-j-joh/article/view/joh.7). *Journal of Open Hardware*. **1** (1). doi:10.5334/joh.7 (https://doi.org/10.5334%2Fjoh.7). ISSN 2514-1708 (https://search.worldcat.org/issn/2514-1708).

External links

- The Open Source Definition (https://opensource.org/docs/osd)
- The Open Source Definition by Bruce Perens (http://www.oreilly.com/openbook/opensources/book/perens.html), Open Sources: Voices from the Open Source Revolution, January 1999, ISBN 1-56592-582-3
- Debian Social Contract and Free Software Guidelines (https://www.debian.org/social_contract#guidelines)
- debian-legal list, with archives from previous discussions (http://lists.debian.org/debian-legal/)
- Draft DFSG FAQ (https://people.debian.org/~bap/dfsg-faq.html)
- Section A.1.3 of *Why OSS/FS? Look at the Numbers!* (https://dwheeler.com/oss_fs_why.html#other-license-issues) identifies some of the major issues discussed by debian-legal.
- List of software licenses currently found in Debian (https://www.debian.org/legal/licenses/)
- The DFSG and Software Licenses (https://wiki.debian.org/DFSGLicenses) Debian wiki

Definition of Free Cultural Works

The **Definition of Free Cultural Works** evaluates and recommends compatible free content licenses.

Definition of Free Cultural Works logo, selected in a logo contest in 2006[1]

History

The Open Content Project by David A. Wiley in 1998 was a predecessor project which defined open content. In 2003, Wiley joined the Creative Commons as "Director of Educational Licenses" and announced the Creative Commons and their licenses as successors to his Open Content Project.[2][3]

Therefore, Creative Commons' Erik Möller[4] in collaboration with Richard Stallman, Lawrence Lessig, Benjamin Mako Hill,[4] Angela Beesley,[4] and others started in 2006 the Free Cultural Works project for defining free content. The first draft of the *Definition of Free Cultural Works* was published 2 April 2006.[5] The 1.0 and 1.1 versions were published in English and translated into several languages.[6]

The *Definition of Free Cultural Works* is used by the Wikimedia Foundation.[7] In 2008, the Attribution and Attribution-ShareAlike Creative Commons licenses were marked as "Approved for Free Cultural Works".[8]

Following in June 2009, Wikipedia migrated to use two licenses: the Creative Commons Attribution-ShareAlike as main license, additionally to the previously used GNU Free Documentation License (which was made compatible[9]).[10] An improved license compatibility with the greater free content ecosystem was given as reason for the license change.[11][12]

In October 2014, the Open Knowledge Foundation's Open Definition 2.0 for *Open Works* and *Open Licenses* described "open" as synonymous to the definition of free in the "Definition of Free Cultural Works" (and also the Open Source Definition and Free Software Definition).[13] A distinct difference is the focus given to the public domain and that it focuses also on the accessibility ("open access") and the readability ("open formats"). The same three creative commons licenses are recommended for open content (CC BY, CC BY-SA, and CC0[14][15][16]) as additionally three for open data intended own licenses, the Open Data Commons Public Domain Dedication and Licence (PDDL), the Open Data Commons Attribution License (ODC-BY) and the Open Data Commons Open Database License (ODbL).

"Free cultural works" approved licenses

- Against DRM
- BSD-like non-copyleft licenses
- CERN Open Hardware License

- CC0
- Creative Commons Attribution (CC BY)
- Creative Commons Attribution ShareAlike (CC BY-SA)
- Design Science License
- Free Art License
- FreeBSD Documentation License
- GNU Free Documentation License (without invariant sections)[17]
- GNU General Public License
- MirOS Licence
- MIT License
- Open Publication License

References

1. Logo contest (http://freedomdefined.org/Logo_contest) on freedomdefined.org (2006)
2. OpenContent is officially closed. And that's just fine. (https://web.archive.org/web/20030802222546/http://opencontent.org/) on opencontent.org (30 June 2003, archived)
3. Creative Commons Welcomes David Wiley as Educational Use License Project Lead (https://web.archive.org/web/20030806102812/http://creativecommons.org/press-releases/entry/3733) by matt (June 23rd, 2003)
4. "History - Definition of Free Cultural Works" (http://freedomdefined.org/History). Freedomdefined.org. Retrieved 2012-11-14.
5. "Revision history of "Definition" - Definition of Free Cultural Works" (https://freedomdefined.org/?title=Definition&offset=20070213&action=history). Freedomdefined.org. Retrieved 2012-11-14.
6. "Definition of Free Cultural Works" (http://freedomdefined.org/Definition). Freedomdefined.org. 2008-12-01. Retrieved 2012-11-14.
7. "Resolution:Licensing policy" (https://wikimediafoundation.org/wiki/Resolution:Licensing_policy). Wikimedia Foundation. Retrieved 2012-11-14.
8. "Approved for Free Cultural Works" (https://creativecommons.org/weblog/entry/8051). Creative Commons. 2009-07-24. Retrieved 2012-11-14.
9. "FDL 1.3 FAQ" (https://www.gnu.org/licenses/fdl-1.3-faq.html). Gnu.org. Retrieved 2011-11-07.
10. "Resolution:Licensing update approval - Wikimedia Foundation" (https://wikimediafoundation.org/wiki/Resolution:Licensing_update_approval).
11. Wikipedia + CC BY-SA = Free Culture Win! (https://creativecommons.org/weblog/entry/15411) on creativecommons.org by Mike Linksvayer, June 22nd, 2009
12. Licensing update rolled out in all Wikimedia wikis (https://blog.wikimedia.org/2009/06/30/licensing-update-rolled-out-in-all-wikimedia-wikis/) on wikimedia.org by Erik Moeller on June 30th, 2009 *"Perhaps the most significant reason to choose CC-BY-SA as our primary content license was to be compatible with many of the other admirable endeavors out there to share and develop free knowledge"*
13. Open Definition 2.1 (http://opendefinition.org/od/2.1/en/) on opendefinition.org
14. licenses (http://opendefinition.org/licenses/) on opendefinition.com
15. Creative Commons 4.0 BY and BY-SA licenses approved conformant with the Open Definition (https://blog.creativecommons.org/2013/12/27/creative-commons-4-0-by-and-by-sa-licenses-approved-conformant-with-the-open-definition/) by Timothy Vollmer on creativecommons.org (December 27th, 2013)

16. Open Definition 2.0 released (https://blog.creativecommons.org/2014/10/07/open-definition-2-0-released/) by Timothy Vollmer on creativecommons.org (October 7th, 2014)
17. licenses (http://freedomdefined.org/Licenses) on freedomdefined.org

External links

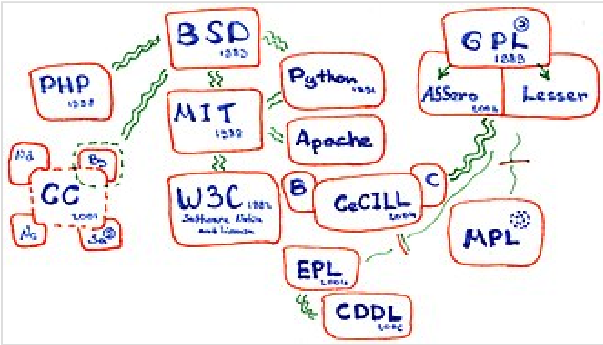
- Definition of Free Cultural Works on freedomdefined.org
- 2006 Announcement on freedomdefined.org
- Understanding Free Cultural Works (https://creativecommons.org/share-your-work/public-domain/freeworks) on creativecommons.org
- Free content defined (https://wikieducator.org/Wikieducator_tutorial/What_is_free_content/Free_content_defined) on WikiEducator
- FreeCulturalWorks (https://www.deviantart.com/freeculturalworks) on DeviantArt

WIKIPEDIA
The Free Encyclopedia

Free license

A **free license** or **open license** is a license that allows copyrighted work to be reused, modified, and redistributed. These uses are normally prohibited by copyright, patent or other Intellectual property (IP) laws. The term broadly covers free content licenses and open-source licenses, also known as free software licenses.

History



Network of licenses (and years of license creation)

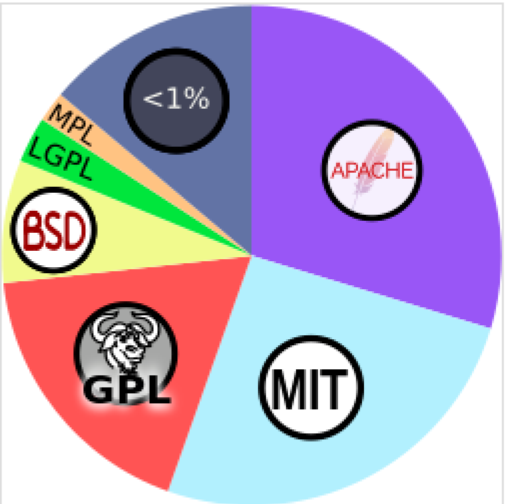
The invention of the term "free license" and the focus on the rights of users were connected to the sharing traditions of the hacker culture of the 1970s public domain software ecosystem, the social and political free software movement (since 1980) and the open source movement (since the 1990s).[1] These rights were codified by different groups and organizations for different domains in Free Software Definition, Open Source Definition, Debian Free Software Guidelines, Definition of Free Cultural Works and The Open Definition.[2] These definitions were then transformed into licenses, using the copyright as legal mechanism. Ideas of free/open licenses have since spread into different spheres of society.

Open source, free culture (unified as free and open-source movement), anticopyright, Wikimedia Foundation projects, public domain advocacy groups and pirate parties are connected with free and open licenses.

Free software license

Free software licenses, also known as open-source licenses, are software licenses that allow content to be used, modified, and shared.[3] They facilitate free and open-source software (FOSS) development.[4] Intellectual property (IP) laws restrict the modification and sharing of creative works.[5] Free and open-source licenses use these existing legal structures for an inverse purpose.[6] They grant the recipient the rights to use the software, examine the source code, modify it, and distribute the modifications. These criteria are outlined in the Open Source Definition and The Free Software Definition.[7]

After 1980, the United States began to treat software as a literary work covered by copyright law.[8] Richard Stallman founded the free software movement in response to the rise of proprietary software.[9] The term "open source" was used by the Open Source Initiative (OSI), founded by free software developers Bruce Perens and Eric S. Raymond.[10][11] "Open source" is alternative label that emphasizes the strengths of the open development model rather than software freedoms.[12] While the goals behind the terms are different, open-source licenses and free software licenses describe the same type of licenses.[13]



Popular free and open source licenses include the Apache License, the MIT License, the GNU General Public License (GPL), the BSD Licenses, the GNU Lesser General Public License (LGPL) and the Mozilla Public License (MPL).

The two main categories of free and open-source licenses are permissive and copyleft.[14] Both grant permission to change and distribute software. Typically, they require attribution and disclaim liability.[15][16] Permissive licenses come from academia.[17] Copyleft licenses come from the free software movement.[18] Copyleft licenses require derivative works to be distributed with the source code and under a similar license.[15][16] Since the mid-2000s, courts in multiple countries have upheld the terms of both types of license.[19] Software developers have filed cases as copyright infringement and as breaches of contract.[20]

Free content license

According to the current definition of open content on the OpenContent website, any general, royalty-free copyright license would qualify as an open license because it 'provides users with the right to make more kinds of uses than those normally permitted under the law. These permissions are granted to users free of charge.' However, the narrower definition used in the Open Definition effectively limits open content to libre content. Any free content license, defined by the Definition of Free Cultural Works, would qualify as an open content license.


Definition of Free Cultural Works logo, selected in a logo contest in 2006[21]

Licenses

By type of license

- Public domain licenses
  - Creative Commons CC0
  - WTFPL
  - Unlicense
  - Public Domain Dedication and License (PDDL)[22]
- Permissive licenses
  - Apache License
  - BSD License
  - MIT License
  - Mozilla Public License (file-based permissive copyleft)

    - Creative Commons Attribution
- Copyleft & patentleft licenses
    - GNU GPL, LGPL (weaker copyleft), AGPL (stronger copyleft)
    - Creative Commons Attribution Share-Alike
    - Mozilla Public License
    - Common Development and Distribution License
    - GFDL (without invariant sections)
    - Free Art License

By type of content

- Open-source software
    - The Open Source Definition
- Open Content
    - Open Content License
    - Open Publication License
- Open-source hardware
- Open database
    - Creative Commons v4
    - Open Database License

By authors

- Creative Commons
- Free Software Foundation
- Open Source Initiative
- Microsoft
    - Microsoft Public License
    - Microsoft Reciprocal License
- Open Content Project
- Open Data Commons from Open Knowledge Foundation
    - Public Domain Dedication and License (PDDL)
    - Attribution License (ODC-By)
    - Open Database License (ODC-ODbL)
- European Union
    - European Union Public Licence

See also

- License compatibility
- License proliferation

Notes

1. Kelty, Christpher M. (2018). "The Cultural Significance of free Software - Two Bits" (http://twobits.net/pub/Kelty-TwoBits.pdf) (PDF). Duke University press - durham and london. p. 99. "*Prior to 1998, Free Software referred either to the Free Software Foundation (and the watchful, micromanaging eye of Stallman) or to one of thousands of different commercial, avocational, or university-research projects, processes, licenses, and ideologies that had a variety of names: sourceware, freeware, shareware, open software, public domain software, and so on. The term Open Source, by contrast, sought to encompass them all in one movement.*"
2. Open Definition 2.1 (http://opendefinition.org/od/2.1/en/) on opendefinition.org *"This essential meaning matches that of “open” with respect to software as in the Open Source Definition and is synonymous with “free” or “libre” as in the Free Software Definition and Definition of Free Cultural Works."*
3. Coleman 2004, "Political Agnosticism".
4. Rosen 2005, pp. 73–90.
5. Rosen 2005, pp. 22–23.
6. Rosen 2005, pp. 103–106.
7. Greenbaum 2016, pp. 1304–1305.
8. Oman 2018, pp. 641–642.
9. Williams 2002, ch. 1.
10. Carver 2005, pp. 448–450.
11. Greenbaum 2016, § I.A.
12. Brock 2022, § 16.3.4.
13. Byfield 2008.
14. Smith 2022, § 3.2.
15. Sen, Subramaniam & Nelson 2008, pp. 211–212.
16. Meeker 2020, 16:13.
17. Rosen 2005, p. 69.
18. Joy 2022, pp. 990–992.
19. Smith 2022, § 3.4.1.
20. Smith 2022, § 3.4.
21. Logo contest (http://freedomdefined.org/Logo_contest) on freedomdefined.org (2006)
22. PDDL 1.0 (http://opendatacommons.org/licenses/pddl/1-0/) on opendatacommons.org

References

- Brock, Amanda, ed. (2022). *Open Source Law, Policy and Practice* (Second ed.). Oxford University Press. doi:10.1093/oso/9780198862345.001.0001 (https://doi.org/10.1093%2Foso%2F9780198862345.001.0001). ISBN 978-0-19-886234-5.
  - Smith, P McCoy. "Copyright, Contract, and Licensing in Open Source (https://doi.org/10.1093/oso/9780198862345.003.0003)". In Brock (2022).
- Byfield, Bruce (March 4, 2008). " "Free" and "Open Source" Software: Navigating the Shibboleths" (https://www.datamation.com/open-source/free-and-open-source-software-navigating-the-shibboleths/). *Datamation*. Retrieved June 6, 2024.

- Carver, Brian W. (2005). "Share and Share Alike: Understanding and Enforcing Open Source and Free Software Licenses" (https://www.jstor.org/stable/24117523). *Berkeley Technology Law Journal*. **20** (1): 443–481. ISSN 1086-3818 (https://search.worldcat.org/issn/1086-3818). JSTOR 24117523 (https://www.jstor.org/stable/24117523).
- Coleman, Gabriella (2004). "The Political Agnosticism of Free and Open Source Software and the Inadvertent Politics of Contrast" (https://www.jstor.org/stable/3318232). *Anthropological Quarterly*. **77** (3): 507–519. doi:10.1353/anq.2004.0035 (https://doi.org/10.1353%2Fanq.2004.0035). hdl:10524/1583 (https://hdl.handle.net/10524%2F1583). ISSN 0003-5491 (https://search.worldcat.org/issn/0003-5491). JSTOR 3318232 (https://www.jstor.org/stable/3318232).
- Greenbaum, Eli (April 2016). "The Non-Discrimination Principle in Open Source Licensing" (http://cardozolawreview.com/wp-content/uploads/2018/08/GREENBAUM.37.4.pdf) (PDF). *Cardoza Law Review*. **37** (4).
- Joy, Reagan (2022). "The Tragedy of the Creative Commons: An Analysis of How Overlapping Intellectual Property Rights Undermine the Use of Permissive Licensing". *Case Western Reserve Law Review*. **72** (4): 977–1013.
- Meeker, Heather (January 2020). *Open Source Software Licensing Basics for Corporate Users* (https://www.youtube.com/watch?v=gF4b1TA5Q5w&list=PLAVikl6VpxPeBtplWOnfzNmiUz529AYAy). *Open Source Software Licensing*. Retrieved December 7, 2023.
- Oman, Ralph (Spring 2018). "Computer Software as Copyrightable Subject Matter: Oracle V. Google, Legislative Intent, and the Scope of Rights in Digital Works". *Harvard Journal of Law & Technology*. **31** (2): 639–652.
- Rosen, Lawrence (2005). *Open Source Licensing: Software Freedom and Intellectual Property Law* (https://www.rosenlaw.com/oslbook.htm) (Paperback ed.). Upper Saddle River, NJ: Prentice Hall. ISBN 978-0-13-148787-1. Archived (https://web.archive.org/web/20221219063515/https://rosenlaw.com/open-source-licensing-software-freedom-and-intellectual-property-law/) from the original on December 19, 2022. Retrieved January 21, 2023.
- Sen, Ravi; Subramaniam, Chandrasekar; Nelson, Matthew L. (Winter 2008). "Determinants of the Choice of Open Source Software License". *Journal of Management Information Systems*. **25** (3): 207–239. doi:10.2753/MIS0742-1222250306 (https://doi.org/10.2753%2FMIS0742-1222250306).
- Williams, Sam (2002). *Free as in Freedom: Richard Stallman's Crusade for Free Software* (https://www.oreilly.com/openbook/freedom/) (First ed.). Sebastopol, California : Farnham: O'Reilly Media. ISBN 978-0-596-00287-9. Archived (https://web.archive.org/web/20230207051646/https://en.wikisource.org/wiki/Free_as_in_Freedom_(2002)) from the original on February 7, 2023. Retrieved February 6, 2023.

External links

- Open software licenses (https://opensource.org/licenses/)
- Open licenses (https://opendefinition.org/licenses/)
- Various Licenses and Comments about Them – GNU Project – Free Software Foundation (https://www.gnu.org/licenses/license-list.en.html)
- Licenses - Definition of Free Cultural Works (http://freedomdefined.org/Licenses)
- proposed Open Source Hardware (OSHW) Statement of Principles and Definition v1.0 (http://freedomdefined.org/OSHW)

The Free Software Definition

The Free Software Definition, written by Richard Stallman and published by the Free Software Foundation (FSF), defines free software as being software that ensures that the users have freedom in using, studying, sharing and modifying that software. The term "free" is used in the sense of "free speech," not of "free of charge."[1] The earliest-known publication of the definition was in the February 1986 edition[2] of the now-discontinued *GNU's Bulletin* publication by the FSF. The canonical source for the document is in the philosophy section of the GNU Project website. As of April 2008, it is published in 39 languages.[3] The FSF publishes a list of licences that meet this definition.

The Four Essential Freedoms of Free Software

The definition published by the FSF in February 1986 had two points:[2]

> The word "free" in our name does not refer to price; it refers to freedom. First, the freedom to copy a program and redistribute it to your neighbors, so that they can use it as well as you. Second, the freedom to change a program, so that you can control it instead of it controlling you; for this, the source code must be made available to you.

In 1996, when the gnu.org website was launched, "free software" was defined referring to "three levels of freedom" by adding an explicit mention of the freedom to study the software (which could be read in the two-point definition as being part of the freedom to change the program).[4][5] Stallman later avoided the word "levels", saying that all of the freedoms are needed, so it is misleading to think in terms of levels.

Finally, another freedom was added, to explicitly say that users should be able to run the program. The existing freedoms were already numbered one to three, but this freedom should come before the others, so it was added as "freedom zero".[6][7]

The modern definition defines free software by whether or not the recipient has the following four freedoms:[8]

- The freedom to run the program as you wish, for any purpose (freedom 0).
- The freedom to study how the program works, and change it so it does your computing as you wish (freedom 1). Access to the source code is a precondition for this.
- The freedom to redistribute copies so you can help your neighbor (freedom 2).
- The freedom to distribute copies of your modified versions to others (freedom 3). 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.

Freedoms 1 and 3 require source code to be available because studying and modifying software without its source code is highly impractical.

Later definitions

In July 1997, Bruce Perens published the Debian Free Software Guidelines.[9] A definition based on the DFSG was also used by the Open Source Initiative (OSI) under the name "*The Open Source Definition*".

Comparison with *The Open Source Definition*

Despite the philosophical differences between the free software movement and the open-source-software movement, the official definitions of free software by the FSF and of open-source software by the OSI basically refer to the same software licences, with a few minor exceptions. While stressing these philosophical differences, the Free Software Foundation comments:

> The term "open source" software is used by some people to mean more or less the same category as free software. It is not exactly the same class of software: they accept some licences that we consider too restrictive, and there are free software licences they have not accepted. However, the differences in extension of the category are small: nearly all free software is open source, and nearly all open source software is free.
>
> —Free Software Foundation[10]

See also

Free and open-source software portal

- Free software movement (FSM)
- The *GNU Manifesto*
- Definition of Free Cultural Works
- Debian Free Software Guidelines
- *The Open Source Definition*

References

1. "What is free software? - GNU Project - Free Software Foundation" (https://www.gnu.org/philosophy/free-sw.html.en). Gnu.org. 2013-06-18. Retrieved 2013-10-03.
2. Stallman, Richard M. (February 1986). "GNU's Bulletin, Volume 1 Number 1" (https://www.gnu.org/bulletins/bull1.txt). Gnu.org. p. 8. Retrieved 2019-02-08.
3. "The Free Software Definition - Translations of this page" (https://www.gnu.org/philosophy/free-sw.html.en#translations). Free Software Foundation Inc. Retrieved 2013-10-03.
4. "What is Free Software? - GNU Project - Free Software Foundation (FSF)" (http://www.ru.j-npcs.org/usoft/WWW/www_gnu.org/philosophy/free-sw.html.en). Ru.j-npcs.org. 1997-03-20. Retrieved 2013-10-03.

5. "What is Free Software? - GNU Project - Free Software Foundation (FSF)" (https://web.archive.org/web/19980126185518/https://www.gnu.org/philosophy/free-sw.html). Archived from the original (https://www.gnu.org/philosophy/free-sw.html.en) on January 26, 1998. Retrieved 2013-10-03.
6. Free Software Foundation (2018-07-21). "What is free software? - GNU Project - Free Software Foundation (Footnote)" (https://www.gnu.org/philosophy/free-sw.html.en#f1). "The reason they are numbered 0, 1, 2 and 3 is historical. Around 1990 there were three freedoms, numbered 1, 2 and 3. Then we realized that the freedom to run the program needed to be mentioned explicitly. It was clearly more basic than the other three, so it properly should precede them. Rather than renumber the others, we made it freedom 0."
7. "The Four Freedoms" (http://ma.tt/2014/01/four-freedoms/). 23 January 2014. "I [Matt Mullenweg] originally thought Stallman started counting with zero instead of one because he's a geek. He is, but that wasn't the reason. Freedoms one, two, and three came first, but later he wanted to add something to supersede all of them. So: freedom zero. The geekness is a happy accident."
8. Stallman, Richard. "The Free Software Definition" (https://www.gnu.org/philosophy/free-sw.html.en). Free Software Foundation. Retrieved 2013-10-15.
9. Bruce Perens. "Debian's "Social Contract" with the Free Software Community" (http://lists.debian.org/debian-announce/debian-announce-1997/msg00017.html). *debian-announce mailing list*.
10. "Categories of Free and Nonfree Software - GNU Project - Free Software Foundation" (https://www.gnu.org/philosophy/categories.html.en).

External links

- The Free Software Definition (https://www.gnu.org/philosophy/free-sw.en.html)

The Open Source Definition

The Open Source Definition (**OSD**) is a document published by the Open Source Initiative. Derived from Bruce Perens' Debian Free Software Guidelines, the definition is the most common standard for open-source software. The definition has ten criteria, such as requiring freely accessed source code and granting the open-source rights to everyone who receives a copy of the program. Covering both copyleft and permissive licenses, it is effectively identical to the definition of free software, but motivated by more pragmatic and business-friendly considerations. The Open Source Initiative's board votes on proposals of licenses to certify that they are compliant with the definition, and maintains a list of compliant licenses on its website. The definition has been adapted into the Open Knowledge Foundation's Open Definition for open knowledge and into open hardware definitions.

History

There have been several attempts to define open source and free software. Amongst the earliest was Free Software Foundation's Free Software Definition, which then defined as the three freedoms of Free Software (Freedom Zero was added later). Published versions of FSF's Free Software Definition existed as early as 1986, having been published in the first edition of the (now defunct) GNU's Bulletin.[1]

Debian Free Software Guidelines

The **Debian Free Software Guidelines** (**DFSG**) was first published together with the first version of the Debian Social Contract in July 1997.[2] The primary author was Bruce Perens, with input from the Debian developers during a month-long discussion on a private mailing list, as part of the larger Debian Social Contract. Perens was copied to an email discussion between Ean Schuessler (then of Debian) and Donnie Barnes of Red Hat, in which Schuessler accused Red Hat of never elucidating its social contract with the Linux community. Perens realized that Debian did not have any formal social contract either, and immediately started creating one. The (then) Three Freedoms, which preceded the drafting and promulgation of the DFSG, were unknown to its authors.[3]

The guidelines were:

1. Free redistribution.
2. Inclusion of source code.
3. Allowing for modifications and derived works.
4. Integrity of the author's source code (as a compromise).
5. No discrimination against persons or groups.
6. No discrimination against fields of endeavor, like commercial use.
7. The license needs to apply to all to whom the program is redistributed.
8. License must not be specific to a product.
9. License must not restrict other software.
10. Example licenses: The GNU GPL, BSD, and Artistic licenses are examples of licenses considered free.[2][4]

Open source

As Netscape released the open-source Mozilla browser in 1998, Bruce Perens again drafted a set of open-source guidelines to go with the release.[5] It has been claimed that the Open Source Definition was created by re-titling the exact text of the DFSG.

A modified version of this definition was adopted by the Open Source Initiative (OSI) as the Open Source Definition.[6][7] The OSI uses the label "open source", rather than "free software", because it felt that the latter term had undesirable ideological and political freight, and it wanted to focus on the pragmatic and business-friendly arguments for open-source software.[6] It adopted a closed rather than membership-driven organizational model in order to draft the definition and work together with a wider variety of stakeholders than other free or open-source projects.[6]

Once the DFSG became the Open Source Definition, Richard Stallman saw the need to differentiate free software from open source and promoted the Free Software Definition.[8]

Debian diverges

In November 1998, Ian Jackson and others proposed several changes in a draft versioned 1.4, but the changes were never made official. Jackson stated[9] that the problems were "loose wording" and the patch clause.

The Debian General Resolution 2004-003,[10] titled "Editorial amendments to the social contract", modified the Social Contract. The proposer Andrew Suffield stated:[11]

> "The rule is 'this resolution only changes the letter of the law, not the spirit'. Mostly it changes the wording of the social contract to better reflect what it is supposed to mean, and this is mostly in light of issues that were not considered when it was originally written."

However, the change of the sentence "We promise to keep the Debian GNU/Linux Distribution entirely free software" into "We promise that the Debian system and all its components will be free" resulted in the release manager, Anthony Towns, making a practical change:[12]

> "As [SC #1] is no longer limited to 'software', and as this decision was made by developers after and during discussion of how we should consider non-software content such as documentation and firmware, I don't believe I can justify the policy decisions to exempt documentation, firmware, or content any longer, as the Social Contract has been amended to cover all these areas."

This prompted another General Resolution, 2004–004,[13] in which the developers voted overwhelmingly against immediate action, and decided to postpone those changes until the next release (whose development started a year later, in June 2005).

Criteria

Providing access to the source code is not enough for software to be considered "open-source".[14] The Open Source Definition requires that ten criteria be met:[15][6]

1. Free redistribution[15]

2. Source code must be accessible and the license must permit redistribution in the form of source code (rather than object code).[15] In order to modify the software, access to source code is required.[16]
3. Derivative works must be allowed and able to be redistributed under the same licensing terms as the open-source product[15]
4. The license may require that the original software be distributed intact, but only if modifications are able to be distributed as patches without restriction.[15][16]
5. No discrimination between users[15]
6. No discrimination between uses, including commercial use[15]
7. Everyone who receives a copy of the program is granted all the open-source rights[15]
8. The license must cover all the code, not a particular product or distribution.[15][16]
9. There may not be restrictions on other software distributed at the same time[15]
10. Technological neutrality—cannot restrict use to any particular technology.[15] For example, a license that requires a user to click a box agreeing to it is not free because the work cannot be distributed as a paper copy.[16]

The Open Source Definition is available under a Creative Commons (CC BY 4.0) license.[17] It covers both copyleft—where redistribution and derivative works must be released under a free license—and permissive licenses—where derivative works can be released under any license. It is part of the open source movement rather than the free software movement, and seeks to promote the availability of open-source software for anyone seeking to reuse it, even the makers of proprietary software.[6][18][16] It does not address warranty disclaimers, although these are very common in open-source software.[16] The definition does not specify a governance structure for open-source projects.[6]

Compliant licenses

The criteria are used by the OSI to approve certain licenses as compatible with the definition, and maintain a list of compliant licenses. New licenses have to submit a formal proposal that is discussed by the OSI mailing list before it is approved or rejected by the OSI board. Seven approved licenses are particularly recommended by the OSI as "popular, widely used, or having strong communities":[19]

- Apache License 2.0
- BSD 3-Clause and BSD 2-Clause Licenses
- All versions of the GNU General Public License
- All versions of the GNU Lesser Public License
- MIT License
- Mozilla Public License 2.0
- Common Development and Distribution License (CDDL)
- Eclipse Public License version 2.0

Application

Software

Most discussions about the DFSG happen on the *debian-legal* mailing list. When a Debian Developer first uploads a package for inclusion in Debian, the *ftpmaster* team checks the software licenses and determines whether they are in accordance with the social contract. The team sometimes confers with the debian-legal list in difficult cases.

Non-"software" content

The DFSG is focused on software, but the word itself is unclear—some apply it to everything that can be expressed as a stream of bits, while a minority considers it to refer to just computer programs. Also, the existence of PostScript, executable scripts, sourced documents, etc., greatly muddies the second definition. Thus, to break the confusion, in June 2004 the Debian project decided to explicitly apply the same principles to software documentation, multimedia data and other content. The non-program content of Debian began to comply with the DFSG more strictly in Debian 4.0 (released in April 2007) and subsequent releases.

GFDL

Much documentation written by the GNU Project, the Linux Documentation Project and others licensed under the GNU Free Documentation License contain invariant sections, which do not comply with the DFSG. This assertion is the end result of a long discussion and the General Resolution 2006-001.[20]

Due to the GFDL invariant sections, content under this license must be separately contained in an additional "non-free" repository which is not officially considered part of Debian.

Multimedia files

It can be sometimes hard to define what constitutes the "source" for multimedia files, such as whether an uncompressed image file is the source of a compressed image and whether the 3D model before ray tracing is the source for its resulting image.

debian-legal tests for DFSG compliance

The *debian-legal* mailing list subscribers have created some tests to check whether a license violates the DFSG. The common tests (as described in the draft DFSG FAQ)[21] are the following:

- "The Desert Island test". Imagine a castaway on a desert island with a solar-powered computer. This would make it impossible to fulfill any requirement to make changes publicly available or to send patches to some particular place. This holds even if such requirements are only upon request, as the castaway might be able to receive messages but be unable to send them. To be free, software must be modifiable by this unfortunate castaway, who must also be able to legally share modifications with friends on the island.

- "The Dissident test". Consider a dissident in a totalitarian state who wishes to share a modified bit of software with fellow dissidents, but does not wish to reveal the identity of the modifier, or directly reveal the modifications themselves, or even possession of the program, to the government. Any requirement for sending source modifications to anyone other than the recipient of the modified binary—in fact, any forced distribution at all, beyond giving source to those who receive a copy of the binary—would put the dissident in danger. For Debian to consider software free it must not require any such excess distribution.
- "The Tentacles of Evil test". Imagine that the author is hired by a large evil corporation and, now in their thrall, attempts to do the worst to the users of the program: to make their lives miserable, to make them stop using the program, to expose them to legal liability, to make the program non-free, to discover their secrets, etc. The same can happen to a corporation bought out by a larger corporation bent on destroying free software in order to maintain its monopoly and extend its evil empire. To be free, the license cannot allow even the author to take away the required freedoms.

Reception

The Open Source Definition is the most widely used definition for open-source software,[22] and is often used as a standard for whether a project is open source.[17] It and the official definitions of free software by the Free Software Foundation (FSF) essentially cover the same software licenses.[6][23] Nevertheless, there is a values difference between the free software and open source movements: the former is more based on ethics and values, the latter on pragmatism.[6]

Derived definitions

The Open Knowledge Foundation's Open Definition is substantially derivative of the Open Source Definition.[24]

The Open Source Hardware Statement of Principles is adapted from the Open Source Definition.[25][22]

See also

foss ***Free and open-source software portal***

- Comparison of free and open-source software licenses
- History of free and open-source software
- The Free Software Definition

References

1. Richard M. Stallman, *What is the Free Software Foundation?*, GNU's Bulletin, Volume 1, No.1, February 1986 (https://www.gnu.org/bulletins/bull1.txt)
2. Bruce Perens (1997-07-04). "Debian's "Social Contract" with the Free Software Community" (http://lists.debian.org/debian-announce/debian-announce-1997/msg00017.html). *debian-announce mailing list*.

3. Bruce Perens: "when I had to write license guidelines for Debian, the Four Freedoms document was unknown. (http://news.slashdot.org/comments.pl?sid=1129863&cid=26875815)"
4. "Debian Social Contract" (http://www.debian.org/social_contract#guidelines). Debian. 2004-04-26.
5. Overly, Michael R. (2003). *The Open Source Handbook*. Pike & Fischer. p. 5. ISBN 978-0-937275-12-2.
6. Gardler, Ross; Walli, Stephen R (2022). "Evolving Perspective on Community and Governance". *Open Source Law, Policy and Practice*. Oxford University PressOxford. p. 47–48, 52. doi:10.1093/oso/9780198862345.003.0002 (https://doi.org/10.1093%2Foso%2F9780198862345.003.0002). ISBN 978-0-19-886234-5.
7. Katz, Andrew (2022). "Everything Open" (https://academic.oup.com/book/44727/chapter/378969640). *Open Source Law, Policy and Practice*. Oxford University Press. p. 521. ISBN 978-0-19-260687-7.
8. Richard Stallman. "Why "Open Source" misses the point of Free Software" (https://www.gnu.org/philosophy/open-source-misses-the-point.html). *GNU website*.
9. Ian Jackson: Draft new DFSG (http://lists.debian.org/debian-devel/1998/11/msg01944.html), debian-devel mailing list
10. General Resolution: Editorial amendments to the social contract (http://www.debian.org/vote/2004/vote_003)
11. Andrew Suffield: Re: Candidate social contract amendments (part 1: editorial) (3rd draft) (http://lists.debian.org/debian-vote/2004/01/msg00692.html), debian-vote mailing list
12. Anthony Towns: Social Contract GR's effect on Sarge (http://lists.debian.org/debian-devel/2004/04/msg01929.html), debian-devel mailing list
13. General Resolution: Sarge Release Schedule in view of GR 2004-003 (http://www.debian.org/vote/2004/vote_004)
14. Greenleaf, Graham; Lindsay, David (2018). *Public Rights: Copyright's Public Domains*. Cambridge University Press. p. 485. ISBN 978-1-107-13406-5.
15. Erlich, Zippy (2007). "Open Source Software". *Handbook of Research on Open Source Software*. IGI Global. pp. 187–188. ISBN 978-1591409991.
16. Laurent, Andrew M. St (2004). *Understanding Open Source and Free Software Licensing: Guide to Navigating Licensing Issues in Existing & New Software*. O'Reilly Media, Inc. pp. 9–11. ISBN 978-0-596-55395-1.
17. Mertic, John (2023). *Open Source Projects - Beyond Code: A blueprint for scalable and sustainable open source projects*. Packt Publishing Ltd. p. 5. ISBN 978-1-83763-385-2.
18. Meeker, Heather J. (2008). *The Open Source Alternative: Understanding Risks and Leveraging Opportunities*. John Wiley & Sons. pp. 21–22. ISBN 978-0-470-25581-0.
19. Smith, P McCoy (2022). "Copyright, Contract, and Licensing in Open Source". *Open Source Law, Policy and Practice*. Oxford University PressOxford. pp. 108–111. doi:10.1093/oso/9780198862345.003.0003 (https://doi.org/10.1093%2Foso%2F9780198862345.003.0003). ISBN 978-0-19-886234-5.
20. General Resolution: Why the GNU Free Documentation License is not suitable for Debian main (http://www.debian.org/vote/2006/vote_001)
21. The Debian Free Software FAQ (http://people.debian.org/~bap/dfsg-faq.html)
22. De Maria, Carmelo; Díaz Lantada, Andrés; Di Pietro, Licia; Ravizza, Alice; Ahluwalia, Arti (2022). "Open-Source Medical Devices: Concept, Trends, and Challenges Toward Equitable Healthcare Technology". *Engineering Open-Source Medical Devices*. Cham: Springer International Publishing. p. 4. doi:10.1007/978-3-030-79363-0_1 (https://doi.org/10.1007%2F978-3-030-79363-0_1). ISBN 978-3-030-79362-3.

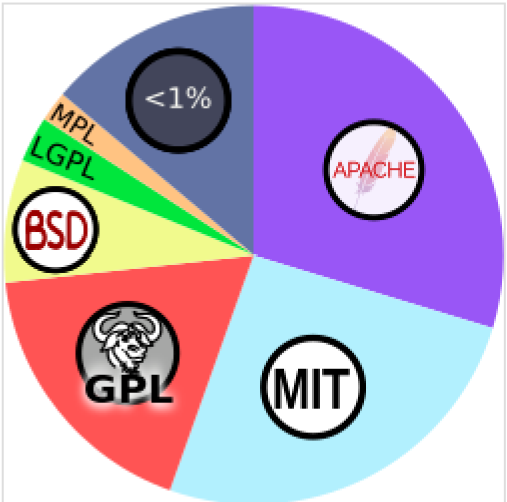
23. Kelty, Christpher M. (2008). "The Cultural Significance of free Software – Two Bits" (http://twobits.net/pub/Kelty-TwoBits.pdf) (PDF). Duke University Press. p. 99. Archived (https://web.archive.org/web/20160304070704/http://twobits.net/pub/Kelty-TwoBits.pdf) (PDF) from the original on 2016-03-04. Retrieved 2016-02-24.
24. Martin, Victoria (2022). *The Complete Guide to Open Scholarship*. Bloomsbury Publishing. p. 27. ISBN 979-8-216-06415-2.
25. Bonvoisin, Jérémy; Mies, Robert; Boujut, Jean-François; Stark, Rainer (2017). "What is the "Source" of Open Source Hardware?" (https://account.openhardware.metajnl.com/index.php/up-j-joh/article/view/joh.7). *Journal of Open Hardware*. **1** (1). doi:10.5334/joh.7 (https://doi.org/10.5334%2Fjoh.7). ISSN 2514-1708 (https://search.worldcat.org/issn/2514-1708).

External links

- The Open Source Definition (https://opensource.org/docs/osd)
- The Open Source Definition by Bruce Perens (http://www.oreilly.com/openbook/opensources/book/perens.html), Open Sources: Voices from the Open Source Revolution, January 1999, ISBN 1-56592-582-3
- Debian Social Contract and Free Software Guidelines (https://www.debian.org/social_contract#guidelines)
- debian-legal list, with archives from previous discussions (http://lists.debian.org/debian-legal/)
- Draft DFSG FAQ (https://people.debian.org/~bap/dfsg-faq.html)
- Section A.1.3 of *Why OSS/FS? Look at the Numbers!* (https://dwheeler.com/oss_fs_why.html#other-license-issues) identifies some of the major issues discussed by debian-legal.
- List of software licenses currently found in Debian (https://www.debian.org/legal/licenses/)
- The DFSG and Software Licenses (https://wiki.debian.org/DFSGLicenses) Debian wiki

WIKIPEDIA
The Free Encyclopedia

Open-source license



Popular open source licenses include the Apache License, the MIT License, the GNU General Public License (GPL), the BSD Licenses, the GNU Lesser General Public License (LGPL) and the Mozilla Public License (MPL).

Open-source licenses are software licenses that allow content to be used, modified, and shared. They facilitate free and open-source software (FOSS) development. Intellectual property (IP) laws restrict the modification and sharing of creative works. Free and open-source licenses use these existing legal structures for an inverse purpose. They grant the recipient the rights to use the software, examine the source code, modify it, and distribute the modifications. These criteria are outlined in the Open Source Definition.

After 1980, the United States began to treat software as a literary work covered by copyright law. Richard Stallman founded the free software movement in response to the rise of proprietary software. The term "open source" was used by the Open Source Initiative (OSI), founded by free software developers Bruce Perens and Eric S. Raymond. "Open source" emphasizes the strengths of the open development model rather than software freedoms. While the goals behind the terms are different, open-source licenses and free software licenses describe the same type of licenses.[1]

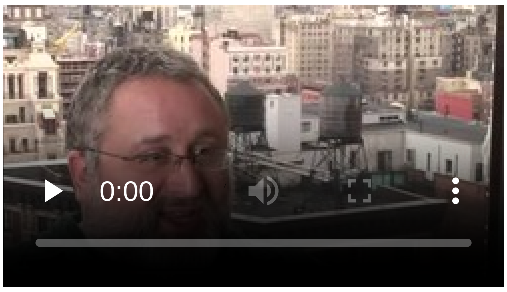
The two main categories of open-source licenses are permissive and copyleft. Both grant permission to change and distribute software. Typically, they require attribution and disclaim liability. Permissive licenses come from academia. Copyleft licenses come from the free software movement. Copyleft licenses require derivative works to be distributed with the source code and under a similar license. Since the mid-2000s, courts in multiple countries have upheld the terms of both types of license. Software developers have filed cases as copyright infringement and as breaches of contract.

Background

Intellectual property (IP) is a legal category that treats creative output as property, comparable to private property.[2] Legal systems grant the owner of an IP the right to restrict access in many ways.[3] Owners can sell, lease, gift, or license their properties.[4] Multiple types of IP law cover software including trademarks, patents, and copyrights.[4]

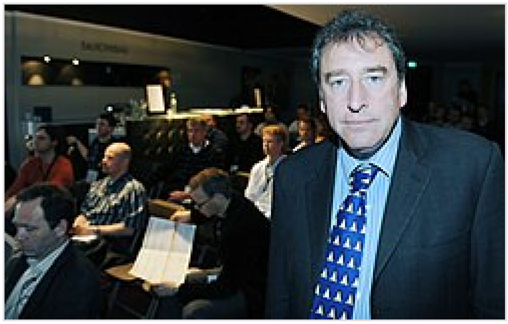
Most countries, including the United States (US), have created copyright laws in line with the Berne Convention with slight variations.[5] These laws assign a copyright whenever a work is released in any fixed format.[6] Under US copyright law, the initial release is considered an original work.[7] The creator, or their employer, holds the copyright to this original work and therefore has the exclusive right to make copies, release modified versions, distribute copies, perform publicly, or display the work publicly.

Modified versions of the original work are derivative works.[8] When a creator modifies an existing work, they hold the copyright to their modifications.[9] Unless the original work was in the public domain, a derivative work can only be distributed with the permission of every copyright holder.[10]



Legal scholar Eben Moglen on the history of copyright

In 1980, the US government amended the law to treat software as a literary work. Software released after this point was restricted by IP laws.[11] At that time, American activist and programmer Richard Stallman was working as a graduate student at the MIT Computer Science and Artificial Intelligence Laboratory. Stallman witnessed fragmentation among software developers. He blamed the spread of proprietary software and closed models of development. To push back against these trends, Stallman founded the free software movement.[12] Throughout the 1980s, he started the GNU Project to create a free operating system, wrote essays on freedom, founded the Free Software Foundation (FSF), and wrote several free software licenses.[13] The FSF used existing intellectual property laws for the opposite of their intended goal of restriction. Instead of imposing restrictions, free software explicitly provided freedoms to the recipient.[14]


Bruce Perens, author of the Open Source Definition

In the 90s, the term "open source" was coined as an alternative label for free software, and specific criteria were laid out to determine which licenses covered free and open-source software.[15][16] Two active members of the free software community, Bruce Perens and Eric S. Raymond, founded the Open Source Initiative (OSI).[17] At Debian, Perens had proposed the Debian Free Software Guidelines (DFSG).[18] The DFSG were drafted to provide a more specific and objective standard for the FOSS that Debian would host in their repositories.[19] The OSI adopted the DSFG and used them as the basis for their Open Source Definition.[20] The Free Software Foundation maintains a rival set of criteria, the Free Software Definition.[21] Historically, these three organizations and their sets of criteria have been the notable authorities in determining whether a license covers free and open-source software.[22] There is significant diversity among individual licenses but little difference between the rival definitions.[16] The three definitions each require that people receiving covered software must be able to use, modify, and redistribute the covered work.[23]

Eric S. Raymond was a proponent of the term "open source" over "free software". He viewed open source as more appealing to businesses and more reflective of the tangible advantages of FOSS development. One of Raymond's goals was to expand the existing hacker community to include large commercial developers.[24] In *The Cathedral and the Bazaar*, Raymond compared open-source development to the bazaar, an open-air public market.[25] He argued that aside from ethics, the open model provided advantages that proprietary software could not replicate.[26][27] Raymond focused heavily on feedback, testing, and bug reports.[28] He contrasted the proprietary model where small pools of secretive workers carried out this work with the development of Linux where the pool of testers included potentially the entire world.[29] He summarized this strength as "Given enough eyeballs, all bugs are shallow."[30] The

OSI succeeded in bringing open-source development to corporate developers including Sun Microsystems, IBM, Netscape, Mozilla, Apache, Apple Inc., Microsoft, and Nokia. These companies released code under existing licenses and drafted their own to be approved by the OSI.[31][32]

Types

Open-source licenses are categorized as copyleft or permissive.[33] Copyleft licenses require derivative works to include source code under a similar license. Permissive licenses do not, and therefore the code can be used within proprietary software. Copyleft can be further divided into strong and weak depending on whether they define derivative works broadly or narrowly.[34][35]

Licenses focus on copyright law, but code is also covered by other forms of IP.[36] Major open-source licenses written since the late 1990s contain patent grants. These open-source patent grants cover the patents held by the developers.[37] Software patents cover ideas and, rather than a specific implementation, cover *any* implementation of a claim. Patent claims give the holder the right to exclude others from making, using, selling, or importing products based on the idea. Because patents grant the right to exclude rather than the right to create, it is possible to have a patent on an idea but still be unable to legally implement it if the invention relies on another patented idea. Thus, open-source patent grants can offer permission only from covered patents. They cannot guarantee that a third party has not patented any concepts embodied in the code.[36] The older permissive licenses do not discuss patents directly and offer only implicit patent grants in their offers to use or sell covered material.[38] Newer copyleft licenses and the 2004 Apache License offer explicit patent grants and limited protection from patent litigation.[39] These patent retaliation clauses protect developers by terminating grants for any party who initiates a patent lawsuit regarding covered software.[39]

Trademarks are the only form of IP not shared by free and open-source software. Trademarks on FOSS function the same as any other trademark.[40] A trademark is a design that identifies the distinct source of a product. Because they distinguish products, the same designs can be used in different fields where there is no risk of confusing similar sources.[41] To give up control of a trademark would result in the loss of that trademark. Therefore, no open-source license freely offers the use of a trademark.[42]

Trademark restrictions can overlap copyrights and affect material otherwise freely available.[43] The US Supreme Court described using trademark law to restrict public domain content as "mutant copyright".[44] In *Dastar Corp. v. Twentieth Century Fox Film Corp.*, the court "caution[ed] against misuse or over-extension of trademark" law without providing a firm decision on those mutant copyrights.[45][46] Trademark overlap can leave open-source and free content projects vulnerable to a "hostile takeover" if outside parties file for trademarks on derivative works.[47] Notably, Andrey Duskin applied for trademarks on the SCP Foundation, a collaborative writing project, when creating derivative works based on SCP stories.[48]

Permissive

Permissive licenses, also known as academic licenses,[49] allow recipients to use, modify, and distribute software with no obligation to provide source code. Institutions created these licenses to distribute their creations to the public.[49] Permissive licenses are usually short, often less than a page of text. They impose few conditions. Most include disclaimers of warranty and obligations to credit authors. A few include explicit provisions for patents, trademarks, and other forms of intellectual property.[50]

Permissive licenses generally originate in academic institutions like the Massachusetts Institute of Technology.

The University of California, Berkeley created the first open-source license when they began distributing their Berkeley Software Distribution (BSD) operating system. The BSD license and its later variations permit modification and distribution of the covered software. The BSD licenses brought the concept of academic freedom of ideas to computing. Early academic software authors had shared code based on implied promises. Berkeley made these concepts explicit with clear disclaimers for liability and warranty along with conditions, or clauses, for redistribution. The original had four clauses, but subsequent versions have further reduced the restrictions. As a result, it is common to specify if the covered software uses a 2-clause or 3-clause version.[51][52]

The Massachusetts Institute of Technology (MIT) created an academic license based on the BSD original. The MIT license clarified the conditions by making them more explicit.[53] For example, the MIT license describes the right to sublicense.[54] One of the strengths of open-source development is the continual process where developers can build on the derivative works of each other and combine their projects into collective works. Explicitly making covered code sublicensable provides a legal advantage when tracking the chain of authorship.[53] The BSD and MIT are template licenses that can be adapted to any project. They are widely adapted and used by many FOSS projects.[51]

The Apache License is more comprehensive and explicit. The Apache Software Foundation wrote it for their Apache HTTP Server. Version 2, published in 2004, offers legal advantages over simple licenses and provides similar grants.[55] While the BSD and MIT licenses offer an implicit patent grant,[56] the Apache License includes a section on patents with an explicit grant from contributors.[57] Additionally, it is one of the few permissive licenses with a patent retaliation clause.[58] Patent retaliation, or patent suspension, clauses take effect if a licensee initiates patent infringement litigation on covered code. In that situation, the patent grants are revoked. These clauses protect against patent trolling.[59]

Copyleft

The Copyleft sticker from an envelope Don Hopkins mailed to Richard Stallman in 1984

Copyleft licenses require source code to be distributed with software and require the source code to be made available under a similar license.[34][60] Like the permissive licenses, most copyleft licenses require attribution.[61] Most, including the GPL, disclaim implied warranties.[62]

Copyleft uses the restrictions of IP law—contrary to their usual purpose—to mandate that the code remain open.[63] The term and its related slogan, "All rights reversed", had been previously used in a playful manner by the *Principia Discordia* and Tiny BASIC; the modern usage begins with Richard Stallman's efforts to create a free operating system. In 1984, programmer Don Hopkins mailed a manual to Stallman with a "Copyleft Ⓛ"

sticker. Stallman, who was working on the GNU operating system, adopted the term.[64] An early version of copyleft licensing was used for the 1985 release of GNU Emacs.[14][65] The term became associated with the FSF's later reciprocal licenses, notably the GNU General Public License (GPL).[66]

Traditional, proprietary software licenses are written with the goal of increasing profit, but Stallman wrote the GPL to increase the body of available free software. His reciprocal licenses offer the rights to use, modify, and distribute the work on the condition that people must release derivative works under a license offering these same freedoms. Software built on a copyleft base must come with the source code, and the source code must be available under the same or a similar license. This offers protection against proprietary software consuming code without giving back.[67][68] Richard Stallman stated that "the central idea of copyleft is to use copyright law, but flip it over to serve the opposite of its usual purpose: instead of a means of privatizing software, [copyright] becomes a means of keeping software free."[69] Free software licenses are also open-source software licenses.[70] The separate terms free software and open-source software reflect different values rather than a legal difference.[71] Both movements and their formal definitions require the covered work to be made available with source code and with permission for modification and redistribution.[16] There are occasional edge cases where only one of the FSF or the OSI accept a license, but the popular free software licenses are open source, including the GPL.[72]

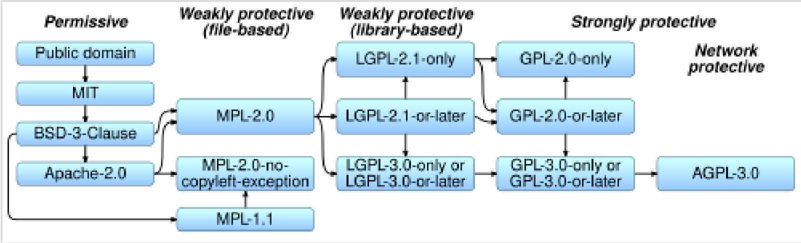
Mitchell Baker drafted the Mozilla Public License while on Netscape's legal team.[73]

Practical benefits to copyleft licenses have attracted commercial developers. Corporations have used and written reciprocal licenses with a narrower scope than the GPL.[74] For example, Netscape drafted their own copyleft terms after rejecting permissive licenses for the Mozilla project.[32] The GPL remains the most popular license of this type, but there are other significant examples. The FSF has crafted the Lesser General Public License (LGPL) for libraries. Mozilla uses the Mozilla Public License (MPL) for their releases, including Firefox. IBM drafted the Common Public License (CPL) and later adopted the Eclipse Public License (EPL). A difference between the GPL and other reciprocal licenses is how they define derivative works covered by the reciprocal provisions. The GPL, and the Affero License (AGPL) based on it, use a broad scope to describe affected works. The AGPL extends the reciprocal obligation in the GPL to cover software made available over a network.[74][32] They are called strong copyleft in contrast to the weaker copyleft licenses often used by corporations. Weak copyleft uses narrower, explicit definitions of derivative works.[75][35] The MPL uses a file-based definition, the CPL and EPL use a module-based definition, and the FSF's own LGPL refers to software libraries.[76]

Compatibility

License compatibility determines how code with different licenses can be distributed together. The goal of open-source licensing is to make the work freely available, but this becomes complicated when working with multiple terminologies imposing different requirements.[77] There are many uncommonly used licenses and some projects write their own bespoke agreements. As a result, this causes more confusion than other legal aspects. When releasing a collection of applications, each license can be considered separately. However, when attempting to combine software, code from another project can only be in-licensed if the project uses compatible terms and conditions.[78]

When combining code bases, the original licenses can be maintained for separate components, and the larger work released under a compatible license.[79] This compatibility is often one-way. Public domain content can be used anywhere as there is no copyright claim, but code acquired under any almost any set of terms cannot be waved to the public domain. Permissive licenses can be used within copyleft works, but copyleft material cannot be released under a permissive license. Some weak copyleft licenses can be used under the GPL and are said to be GPL-compatible. GPL software can only be used under the GPL or AGPL.[77] Permissive licenses are broadly compatible because they can cover separate parts of a project. Multiple licenses including the GPL and Apache License have been revised to enhance compatibility.[80]



Open-source software licenses and how they interact

Translation issues, ambiguity in licensing terms, and incompatibility of some licenses with the law in certain jurisdictions compound the problem of license compatibility.[81] Downloading an open-source module is straightforward, but complying with the licensing terms can be more difficult.[82] Because of the amount of software dependencies, engineers working on complex projects often rely on license management software to achieve compliance with the licensing terms of open-source components.[83] Many open-source software files do not unambiguously state the license, increasing the difficulties of compliance.[82]

Enforcement

Free and open-source software licenses have been successfully enforced in civil court since the mid-2000s.[85] In a pair of early lawsuits—Jacobsen v. Katzer in the United States and Welte v. Sitecom in Germany—defendants argued that open-source licenses were invalid.[86][87] Sitecom and Katzer separately argued that the licenses were unenforceable. Both the US and German courts rejected these claims. They ruled that the defendants could not have legally distributed the software if the licenses were unenforceable.[85][84]

Early legal victories by programmer Harald Welte established a precedent for open-source software litigation in Germany.[84]

Courts have found that distributing software indicates acceptance of the license's terms.[88] Physical software releases can obtain the consumer's assent with notices placed on shrinkwrap. Online distribution can use clickwrap, a digital equivalent where the user must click to accept.[89] Open-source software has an additional acceptance mechanism. Without permission from the copyright holder, the law prohibits redistribution.[90] Therefore, courts treat redistribution as acceptance of the license terms. These can include attribution provisions or source code provisions for copyleft licenses.[91][92]

Developers typically achieve compliance without lawsuits. Social pressures, like the potential for community backlash, are often sufficient.[93] Cease and desist letters are a common method to bring companies back into compliance, especially in Germany.[94] A standard process has developed in the German legal system. FOSS developers present companies with a cease and desist letter. These letters outline how to come back into compliance from a violation. German judges can issue a court-mandated cease and desist order to unresponsive companies. Civil cases proceed if these first steps fail. The German procedural laws are clear and favorable to claimants.[95]

Uncertainties remain in how different courts will handle certain aspects of licensing.[96] For software in general, there are debates about what can be patented and what can be copyrighted. Regarding an application programming interface (API), the European Court of Justice noted in the 2012 *SAS Institute* case that "ideas and principles which underlie [computer program] interfaces are not protected by copyright".[97] In a similar 2021 case, the US Supreme Court permitted the recreation of an API in a transformative product under fair use.[98]

A long-debated subject within the FOSS community is whether open-source licenses are "bare licenses" or contracts.[96] A bare license is a set of conditions under which actions otherwise restricted by IP laws are permitted.[85] Under the bare license interpretation, advocated by the FSF, a case is brought to court by the copyright holder as copyright infringement.[85] Under the contract interpretation, a case can be brought to court by an involved party as a breach of contract.[99] US and French courts have tried cases under both interpretations.[100] Non-profit organizations like FSF and the Software Freedom Conservancy offer to hold the rights to developers' projects to enforce compliance.[95]

Public domain software

When a copyright expires, the work enters the public domain, and is freely available to anyone.[102] Some creative works are not covered by copyright and enter directly into the public domain. In the early history of computing, this applied to software.[11] Early computer software was often given away with hardware.[103] Developed initially at MIT, the pioneering video game Spacewar! was used to market and test the PDP-1 computer.[104]

Early computer programs like the pioneering video game Spacewar! are in the public domain.[101]

According to attorney Lawrence Rosen, copyright laws were not written with the expectation that creators would place their work into the public domain. Thus intellectual property laws lack clear paths to waive a copyright. Highly permissive licenses described as "public domain" may legally function as unilateral contracts that offer something but impose no terms.[105][106]

A public-domain-equivalent license, like the Creative Commons CC0, provides a waiver of copyright claims into the public domain along with a permissive software license as a fallback. In jurisdictions that do not accept a public domain waiver, the permissive license takes effect.[107] Public domain waivers share limitations with simple academic licenses. This creates the possibility that an outside party could attempt to control a public domain work via patent or trademark law.[108] Public domain waivers handle warranties differently from any type of license. Even very permissive ones, like the MIT license, disclaim warranty and liability. Anyone using the free software must accept this disclaimer as a condition. Because public domain content is available to everyone, the copyright waiver cannot impose a disclaimer.[102]

Use in proprietary software

Open-source licenses allow other businesses to commercialize covered software.[109] Work released under a permissive license can be incorporated into proprietary software.[110] Permissive licenses permit the addition of new terms, including proprietary ones.[111][112] Proprietary software has heavily integrated open-source code released under the Apache, BSD, and MIT licenses.[113] Open core is a business model where developers release a core piece of software as open source and monetize a product containing it as proprietary software.[114] The strong copyleft GPL is written to prevent distribution within proprietary software.[115][116] Weak copyleft licenses impose specific requirements on derivative works that may allow the covered code to be distributed within proprietary software in certain circumstances.[77]

Cloud computing relies on free and open-source software and avoids the distribution that triggers most licenses. Cloud software is hosted rather than distributed.[117] A vendor hosts the software online, and their end users do not have to download, access, or even know about the code in use.[118] The copyleft GNU Affero General Public License (AGPL) is triggered when covered code is hosted or distributed.[119] Some developers have adopted the AGPL, and others have switched to proprietary licenses with features of open-source licensing.[120] For example, open-core developer Elastic switched from the Apache license to the "source-available" Server Side Public License.[121] Source-available software comes with source code as a reference.[122]

Since 2010, the cloud model has grown in prominence.[117] Developers have criticized cloud companies that profit from hosting open-source software without contributing money or code upstream, comparing the practice to strip mining.[123] Cloud computing leader Amazon Web Services has stated they comply with licenses and act in their customers' best interests.[123][124]

See also

- Beerware
- Comparison of free and open-source software licenses
- List of free-content licenses
- Multi-licensing
- Software Composition Analysis
- List of free and open-source software licenses
- List of copyleft software licenses
- List of open-source programming languages
- List of permissive software licenses

Notes

1. Byfield 2008.

2. Rosen 2005, p. 22.
3. Rosen 2005, pp. 22–23.
4. Rosen 2005, p. 15.
5. Smith 2022, sec. 3.1.2.
6. Fagundes & Perzanowski 2020, p. 529.
7. Rosen 2005, p. 17.
8. Rosen 2005, pp. 27–28.
9. Rosen 2005, pp. 28–29.
10. Rosen 2005, p. 28.
11. Oman 2018, pp. 641–642.
12. Williams 2002, ch. 1.
13. Williams 2002, ch. 7.
14. Williams 2002, ch. 9.
15. Greenbaum 2016, sec. I.A.
16. Maracke 2019, sec. 2.2.
17. Carver 2005, pp. 448–450.
18. Perens 1999.
19. Greenbaum 2016, pp. 1302–1303.
20. Greenbaum 2016, pp. 1304–1305.
21. Greenbaum 2016, p. 1305.
22. Fontana 2010, p. 2.
23. Coleman 2004, "Political Agnosticism".
24. Raymond 1999, "Memes and Mythmaking".
25. Meeker 2020, 2:33–3:06.
26. Raymond 2001, "The Cathedral and the Bazaar".
27. Brock 2022, sec. 16.3.4.
28. Raymond 2001.
29. Raymond 2001, "The Social Context of Open-Source Software".
30. Raymond 2001, p. 19.
31. Onetti & Verma 2009, p. 69.
32. Hammerly, Paquin & Walton 1999.
33. Smith 2022, sec. 3.2.
34. Sen, Subramaniam & Nelson 2008, pp. 211–212.
35. Meeker 2020, 16:13.
36. Rosen 2005, pp. 22–24.
37. Bain & Smith 2022, sec. 10.4.3.
38. Bain & Smith 2022, sec. 10.4.2.
39. Bain & Smith 2022, sec. 10.4.4.
40. Chestek 2022, p. 30.
41. Chestek 2022, pp. 184–185.
42. Rosen 2005, p. 38.
43. Joy 2022, p. 986.
44. Joy 2022, p. 989.
45. *Dastar Corp. v. Twentieth Century Fox Film Corp.*, 539 U.S. 23 (https://scholar.google.com/scholar_case?case=13445605668854417212#p34), 34 (2003).

46. Joy 2022, pp. 987–988.
47. Joy 2022, pp. 1004–1006.
48. Joy 2022, pp. 979, 1002.
49. Rosen 2005, p. 69.
50. Rosen 2005, pp. 101–102.
51. Smith 2022, sec. 3.2.1.1.
52. OSI 2023.
53. Rosen 2005, pp. 73–90.
54. OSI 2023, "The MIT License".
55. Smith 2022, sec. 3.2.1.2.
56. Bain & Smith 2022, sec. 10.4.2.
57. OSI 2023, "Apache License, Version 2.0".
58. Bain & Smith 2022, ch. 10.
59. Bain & Smith 2022, sec. 10.4.4.
60. St. Laurent 2004, pp. 38–39.
61. Ballhausen 2019, p. 86.
62. Rosen 2005, p. 135.
63. Rosen 2005, pp. 103–106.
64. Keats 2010, p. 64.
65. "Full Text of GNU Emacs Copying Permission Notice" (https://www.tech-insider.org/free-software/research/1985/0715-b.txt). 1985.
66. Keats 2010, pp. 63–67.
67. Rosen 2005, pp. 103–109.
68. Meeker 2020, 6:00–7:22.
69. Joy 2022, pp. 990–992.
70. Onetti & Verma 2009, p. 71.
71. St. Laurent 2004, pp. 81–83, 114.
72. Ballhausen 2019, p. 82.
73. St. Laurent 2004, pp. 68, 75.
74. Tsai 2008, pp. 564–570.
75. Sen, Subramaniam & Nelson 2008, pp. 212–213.
76. Rosen 2005, refer to corresponding chapters.
77. Smith 2022, sec. 3.3.
78. Rosen 2005, pp. 243–247.
79. St. Laurent 2004, pp. 159–163.
80. See Smith 2022, p. 102 for: Apache License version 2.0 in 2004, GPL version 3 in 2007, LGPL version 3 in 2007, and AGPL version 3 in 2007. See Smith 2022, pp. 95–101 for: MPL version 2.0 in 2012 and EPL version 2 in 2017.
81. Bernelin 2020, pp. 100, 102.
82. Ombredanne 2020, p. 105.
83. Ombredanne 2020, p. 106.
84. Ballhausen 2022, sec. 5.3.
85. Smith 2022, sec. 3.4.1.
86. *Jacobsen v. Katzer*, 535 F.3d 1373 (https://scholar.google.com/scholar_case?case=17776182574171214893) (Fed. Cir. 2008).

87. *Welte v. Sitecom* (District Court of Munich 2004), No. 21 O 6123/04 (https://www.ifross.org/Fremdartikel/judgment_dc_munich_gpl.pdf).
88. Smith 2022, p. 106.
89. Rosen 2005, p. 137.
90. Rosen 2005, p. 138.
91. Rosen 2005, ch. 6.
92. Meeker 2020, 17:04.
93. St. Laurent 2004, pp. 158–159.
94. Ballhausen 2022, p. 127.
95. Ballhausen 2022, sec. 5.4.
96. Walden 2022, sec. 1.1.
97. Smith 2022, sec. 3.1.3.
98. *Google LLC v. Oracle America, Inc.*, 593 U.S. (https://scholar.google.com/scholar_case?case=4050620474768552042#p1203), 1203 (2021).
99. Smith 2022, sec. 3.4.2.
100. Smith 2022, sec. 3.4.
101. Ross 2021, "Spacewar: End of Development".
102. Rosen 2005, p. 36.
103. Walden 2022, p. 3.
104. Smith 2019, pp. 55–56.
105. Rosen 2005, pp. 74–77.
106. St. Laurent 2004, p. 98.
107. Fagundes & Perzanowski 2020, p. 524.
108. Joy 2022, pp. 1008–1010.
109. Brock 2022, sec. 16.3.3.
110. St. Laurent 2004, p. 14.
111. St. Laurent 2004, p. 22.
112. Onetti & Verma 2009, p. 81.
113. St. Laurent 2004, p. 30.
114. Brock 2022, sec. 16.4.2.3.
115. Tsai 2008, p. 550.
116. St. Laurent 2004, p. 39.
117. Brock 2022, sec. 16.5.2.
118. Brock 2022, sec. 16.4.2.8.
119. Brock 2022, sec. 16.4.2.2.
120. Brock 2022, sec. 16.5.3.
121. Brock 2022, sec. 16.5.3.8.
122. Kunert 2022.
123. Wakabayashi 2019.
124. Brock 2022, sec. 16.5.3.2.

References

- Ballhausen, Miriam (June 2019). "Free and Open Source Software Licenses Explained" (https://www.computer.org/csdl/magazine/co/2019/06/08728094/1axaDABEhXy). *Computer*. **52**

(6): 82–86. doi:10.1109/MC.2019.2907766 (https://doi.org/10.1109%2FMC.2019.2907766).

- Bernelin, Margo (2020). "The Compatibility of Open/Free Licences: a Legal Imbroglio". *International Journal of Law and Information Technology*. **28** (2): 93–111. doi:10.1093/ijlit/eaaa010 (https://doi.org/10.1093%2Fijlit%2Feaaa010).
- Brock, Amanda, ed. (2022). *Open Source Law, Policy and Practice* (Second ed.). Oxford University Press. doi:10.1093/oso/9780198862345.001.0001 (https://doi.org/10.1093%2Foso%2F9780198862345.001.0001). ISBN 978-0-19-886234-5.
  - Bain, Malcom; Smith, P McCoy. "Patents and the Defensive Response (https://doi.org/10.1093/oso/9780198862345.003.0010)". In Brock (2022).
  - Ballhausen, Miriam. "Copyright Enforcement (https://doi.org/10.1093/oso/9780198862345.003.0005)". In Brock (2022).
  - Chestek, Pamela. "Trademarks (https://doi.org/10.1093/oso/9780198862345.003.0009)". In Brock (2022).
  - Smith, P McCoy. "Copyright, Contract, and Licensing in Open Source (https://doi.org/10.1093/oso/9780198862345.003.0003)". In Brock (2022).
  - Walden, Ian. "Open Source as Philosophy, Methodology, and Commerce: Using Law with Attitude (https://doi.org/10.1093/oso/9780198862345.003.0001)". In Brock (2022).
- Byfield, Bruce (March 4, 2008). " "Free" and "Open Source" Software: Navigating the Shibboleths" (https://www.datamation.com/open-source/free-and-open-source-software-navigating-the-shibboleths/). *Datamation*. Retrieved June 6, 2024.
- Carver, Brian W. (2005). "Share and Share Alike: Understanding and Enforcing Open Source and Free Software Licenses" (https://www.jstor.org/stable/24117523). *Berkeley Technology Law Journal*. **20** (1): 443–481. ISSN 1086-3818 (https://search.worldcat.org/issn/1086-3818). JSTOR 24117523 (https://www.jstor.org/stable/24117523).
- Coleman, Gabriella (2004). "The Political Agnosticism of Free and Open Source Software and the Inadvertent Politics of Contrast" (https://www.jstor.org/stable/3318232). *Anthropological Quarterly*. **77** (3): 507–519. doi:10.1353/anq.2004.0035 (https://doi.org/10.1353%2Fanq.2004.0035). hdl:10524/1583 (https://hdl.handle.net/10524%2F1583). ISSN 0003-5491 (https://search.worldcat.org/issn/0003-5491). JSTOR 3318232 (https://www.jstor.org/stable/3318232).
- DiBona, Chris; Stone, Mark; Ockman, Sam, eds. (1999). *Open Sources: Voices from the Open Source Revolution* (https://www.oreilly.com/openbook/opensources/book/). Sebastopol, California: O'Reilly Media. Archived (https://web.archive.org/web/20230127172901/https://www.oreilly.com/openbook/opensources/book/) from the original on January 27, 2023. Retrieved January 21, 2023.
  - Hammerly, Jim; Paquin, Tom; Walton, Susan. "Freeing the Source: The Story of Mozilla (https://www.oreilly.com/openbook/opensources/book/netrev.html)". In DiBona, Stone & Ockman (1999).
  - Perens, Bruce. "The Open Source Definition (https://www.oreilly.com/openbook/opensources/book/perens.html)". In DiBona, Stone & Ockman (1999).
  - Raymond, Eric S. "The Revenge of the Hackers (https://www.oreilly.com/openbook/opensources/book/raymond2.html)". In DiBona, Stone & Ockman (1999).
- Fagundes, Dave; Perzanowski, Aaron (November 2020). "Abandoning Copyright". *William & Mary Law Review*. **62** (2): 487–569.
- Fontana, Richard E. (April 2010). "Open Source License Enforcement and Compliance". *The Computer and Internet Lawyer*. **27** (4). Aspen.
- Greenbaum, Eli (April 2016). "The Non-Discrimination Principle in Open Source Licensing" (http://cardozolawreview.com/wp-content/uploads/2018/08/GREENBAUM.37.4.pdf) (PDF). *Cardoza Law Review*. **37** (4).

- Joy, Reagan (2022). "The Tragedy of the Creative Commons: An Analysis of How Overlapping Intellectual Property Rights Undermine the Use of Permissive Licensing". *Case Western Reserve Law Review*. **72** (4): 977–1013.
- Keats, Jonathon (2010). *Virtual Words: Language on the Edge of Science and Technology* (https://doi.org/10.1093/oso/9780195398540.003.0017). Oxford University Press. doi:10.1093/oso/9780195398540.003.0017 (https://doi.org/10.1093%2Foso%2F9780195398540.003.0017). ISBN 978-0-19-539854-0. Archived (https://web.archive.org/web/20230326192756/https://academic.oup.com/book/40770/chapter-abstract/348692229?redirectedFrom=fulltext) from the original on March 26, 2023. Retrieved January 28, 2023.
- Kunert, Paul (September 8, 2022). "Open Source Biz Shifts Akka to Business Source License" (https://www.theregister.com/2022/09/08/open_source_biz_sick_of/). Archived (https://web.archive.org/web/20221031061942/https://www.theregister.com/2022/09/08/open_source_biz_sick_of/) from the original on October 31, 2022. Retrieved January 25, 2023.
- Maracke, Catharina (July 2019). "Free and Open Source Software and FRAND-Based Patent Licenses: How to Mediate Between Standard Essential Patent and Free and Open Source Software" (https://doi.org/10.1111%2Fjwip.12114). *The Journal of World Intellectual Property*. **22** (3–4): 78–102. doi:10.1111/jwip.12114 (https://doi.org/10.1111%2Fjwip.12114).
- Meeker, Heather (January 2020). *Open Source Software Licensing Basics for Corporate Users* (https://www.youtube.com/watch?v=gF4b1TA5Q5w&list=PLAVikl6VpxPeBtplWOnfzNmiUz529AYAy). *Open Source Software Licensing*. Retrieved December 7, 2023.
- Oman, Ralph (Spring 2018). "Computer Software as Copyrightable Subject Matter: Oracle V. Google, Legislative Intent, and the Scope of Rights in Digital Works". *Harvard Journal of Law & Technology*. **31** (2): 639–652.
- Ombredanne, Philippe (2020). "Free and Open Source Software License Compliance: Tools for Software Composition Analysis". *Computer*. **53** (10): 105–109. doi:10.1109/MC.2020.3011082 (https://doi.org/10.1109%2FMC.2020.3011082).
- Onetti, Alberto; Verma, Sameer (May 1, 2009). "Open Source Licensing and Business Models". *ICFAI Journal of Knowledge Management*. **7** (1): 68–94.
- OSI (February 2023). "OSI Approved Licenses" (https://opensource.org/licenses/). *opensource.org*. Open Source Initiative. Retrieved December 23, 2023.
- Raymond, Eric S. (2001). *The Cathedral & the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary* (http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/) (Revised ed.). Sebastopol, California: O'Reilly Media. ISBN 978-0-596-00108-7. Archived (https://web.archive.org/web/20030424100429/http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/) from the original on April 24, 2003. Retrieved January 28, 2023.
- Rosen, Lawrence (2005). *Open Source Licensing: Software Freedom and Intellectual Property Law* (https://www.rosenlaw.com/oslbook.htm) (Paperback ed.). Upper Saddle River, NJ: Prentice Hall. ISBN 978-0-13-148787-1. Archived (https://web.archive.org/web/20221219063515/https://rosenlaw.com/open-source-licensing-software-freedom-and-intellectual-property-law/) from the original on December 19, 2022. Retrieved January 21, 2023.
- Ross, Heather (January 4, 2021). "Spacewar - Guide, History, Origin and More" (https://history-computer.com/spacewar-guide/). *History-Computer*. Retrieved December 23, 2023.
- Sen, Ravi; Subramaniam, Chandrasekar; Nelson, Matthew L. (Winter 2008). "Determinants of the Choice of Open Source Software License". *Journal of Management Information Systems*. **25** (3): 207–239. doi:10.2753/MIS0742-1222250306 (https://doi.org/10.2753%2FMIS0742-1222250306).
- Smith, Alexander (2019). *They Create Worlds: The Story of the People and Companies That Shaped the Video Game Industry*. Vol. 1: 1971 – 1982. Boca Raton, Florida: CRC Press. ISBN 978-1-138-38990-8.

- St. Laurent, Andrew M. (2004). *Understanding Open Source and Free Software Licensing* (https://freecomputerbooks.com/Understanding-Open-Source-and-Free-Software-Licensing.html). Sebastopol, California: O'Reilly Media. ISBN 978-0596005818.
- Tsai, John (2008). "For Better or Worse: Introducing the Gnu General Public License Version 3". *Berkeley Technology Law Journal*. **23** (1): 547–581.
- Wakabayashi, Daisuke (December 15, 2019). "Prime Leverage: How Amazon Wields Power in the Technology World" (https://archive.today/20231030070154/https://www.nytimes.com/2019/12/15/technology/amazon-aws-cloud-competition.html). *The New York Times*. Archived from the original (https://www.nytimes.com/2019/12/15/technology/amazon-aws-cloud-competition.html) on October 30, 2023. Retrieved May 24, 2024.
- Williams, Sam (2002). *Free as in Freedom: Richard Stallman's Crusade for Free Software* (https://www.oreilly.com/openbook/freedom/) (First ed.). Sebastopol, California : Farnham: O'Reilly Media. ISBN 978-0-596-00287-9. Archived (https://web.archive.org/web/20230207051646/https://en.wikisource.org/wiki/Free_as_in_Freedom_(2002)) from the original on February 7, 2023. Retrieved February 6, 2023.

Wikipedia
The Free Encyclopedia

Permissive software license

A **permissive software license**, sometimes also called **BSD-like** or **BSD-style** license,[1] is a free-software license which instead of copyleft protections, carries only minimal restrictions on how the software can be used, modified, and redistributed, usually including a warranty disclaimer. Examples include the GNU All-permissive License, MIT License, BSD licenses, Apple Public Source License and Apache license. As of 2016, the most popular free-software license is the permissive MIT license.[2][3]

Comparison table

	Public domain & equivalents	Permissive license	Copyleft (protective license)	Noncommercial license	Proprietary license	Trade secret
Description	Grants all rights	Grants use rights, forbids almost nothing (allows proprietization, license compatibility)	Grants use rights, forbids proprietization	Grants rights for noncommercial use only. May not be combined with copyleft.	Traditional use of copyright; no rights need be granted	No information made public
Software	PD, CC0	BSD, MIT, Apache	GPL, AGPL	JRL, AFPL	proprietary software, no public license	private, internal software
Other creative works	PD, CC0	CC BY	CC BY-SA, Free Art License	CC BY-NC, CC BY-NC-SA	Copyright, no public license	unpublished

Example

The following is the full text of the simple GNU All-permissive License:

> Copyright <YEAR>, <AUTHORS>
>
> Copying and distribution of this file, with or without modification, are permitted in any medium without royalty, provided the copyright notice and this notice are preserved. This file is offered as-is, without any warranty.
>
> —GNU All-permissive License[4][5]

Definitions

The Open Source Initiative defines a permissive software license as a "non-copyleft license that guarantees the freedoms to use, modify and redistribute".[6] GitHub's *choosealicense* website describes the permissive MIT license as "[letting] people do anything they want with your code as long as they provide attribution back to you and don't hold you liable."[7] California Western School of Law's newmediarights.com defined them as follows: "The 'BSD-like' licenses such as the BSD, MIT and Apache licenses are extremely permissive, requiring little more than attributing the original portions of the licensed code to the original developers in your own code and/or documentation."[1]

Comparison to copyleft

Copyleft licenses generally require the reciprocal publication of the source code of any modified versions under the original work's copyleft license.[8][9] Permissive licenses, in contrast, do not try to guarantee that modified versions of the software will remain free and publicly available, generally requiring only that the original copyright notice be retained.[1] As a result, derivative works, or future versions, of permissively-licensed software can be released as proprietary software.[10]

Defining how liberal a license is, however, is not something easily quantifiable, and often depends on the goals of the final users. If the latter are developers, for some it might be valuable to have the right to modify and exploit source code written by others and possibly incorporate it into proprietary code and make money with it (and therefore these see permissive licenses as offering them a "right"),[11] while for other developers it might be more valuable to know that nobody will ever capitalize what has mostly been their work (and therefore these see copyleft licenses as offering them a "right"). Furthermore, the final users might not be developers at all, and in this case copyleft licenses offer them the everlasting right to access a software as free software, ensuring that it will never become closed source – while permissive licenses offer no rights at all to non-developer final users, and software released with a permissive license could theoretically become from one day to another a closed source malware without the user even knowing it.

Permissive licenses offer more extensive license compatibility than copyleft licenses, which cannot generally be freely combined and mixed, because their reciprocity requirements conflict with each other.[12][13][14][15][16]

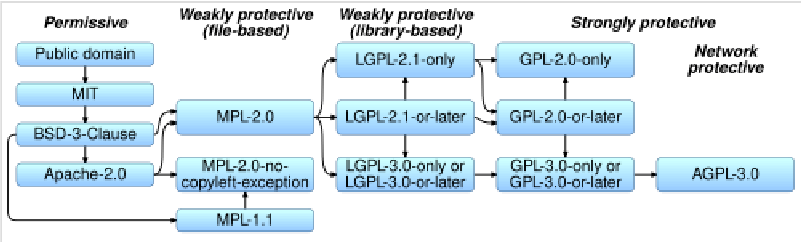
Comparison to public domain

Computer Associates Int'l v. Altai used the term "public domain" to refer to works that have become widely shared and distributed under permission, rather than work that was deliberately put into the public domain. However, permissive licenses are not actually equivalent to releasing a work into the public domain.

Permissive licenses often do stipulate some limited requirements, such as that the original authors must be credited (*attribution*). If a work is truly in the public domain, this is usually not legally required, but a United States copyright registration requires disclosing material that has been previously published,[17] and attribution may still be considered an ethical requirement in academia.

Advocates of permissive licenses often recommend against attempting to release software to the public domain, on the grounds that this can be legally problematic in some jurisdictions.[18][19] Public-domain-equivalent licenses are an attempt to solve this problem, providing a fallback permissive license for cases where renunciation of copyright is not legally possible, and sometimes also including a disclaimer of warranties similar to most permissive licenses.

License compatibility



License compatibility between common free and open-source software (FOSS) licenses according to David A. Wheeler (2007): the vector arrows denote a one directional compatibility, therefore better compatibility on the left side ("permissive licenses") than on the right side ("copyleft licenses").[20]

In general permissive licenses have good license compatibility with most other software licenses in most situations.[12][13]

Due to their non-restrictiveness, most permissive software licenses are even compatible with copyleft licenses, which are incompatible with most other licenses. Some older permissive licenses, such as the 4-clause BSD license, the PHP License, and the OpenSSL License, have clauses requiring advertising materials to credit the copyright holder, which made them incompatible with copyleft licenses. Popular modern permissive licenses, however, such as the MIT License, the 3-clause BSD license and the zlib license, don't include advertising clauses and are generally compatible with copyleft licenses.

Some licenses do not allow derived works to add a restriction that says a redistributor cannot add more restrictions. Examples include the CDDL and MsPL. However such restrictions also make the license incompatible with permissive free-software licenses.

Reception and adoption

While they have been in use since the mid-1980s,[21] several authors noted an increase in the popularity of permissive licenses during the 2010s.[22][23][24][25]

As of 2015, the MIT License, a permissive license, is the most popular free software license, followed by GPLv2.[2][3]

Other terms

Non-copyleft

> A "permissive" license is simply a non-copyleft open source license.
>
> —Open Source Initiative, [26]

Sometimes the word "permissive" is considered too ambiguous, because all free software licenses are "permissive", in the sense that they all allow to modify and redistribute the source code. In most cases the real opposition is between copyleft licenses and non-copyleft ones, thus some authors prefer to use the term "non-copyleft" instead of "permissive".[27][28][26]

Copycenter

> Berkeley had what we called "copycenter," which is "take it down to the copy center and make as many copies as you want."
>
> —Marshall Kirk McKusick, [29]

Copycenter is a term originally used to explain the modified BSD license, a permissive free-software license. The term was presented by computer scientist and Berkeley Software Distribution (BSD) contributor Marshall Kirk McKusick at a BSD conference in 1999. It is a word play on copyright, copyleft and copy center.[29][30]

Pushover license

> We call them "pushover licenses" because they can't say "no" when one user tries to deny freedom to others.."
>
> —Richard Stallman, founder of the GNU operating system[31]

In the Free Software Foundation's guide to license compatibility and relicensing, Richard Stallman defines permissive licenses as "pushover licenses", comparing them to those people who "can't say no", because they are seen as granting a right to "deny freedom to others."[31] The Foundation recommends pushover licenses only for small programs, below 300 lines of code, where "the benefits provided by copyleft are usually too small to justify the inconvenience of making sure a copy of the license always accompanies the software".[32]

See also

foss ***Free and open-source software***

portal

- List of permissive software licenses
- License-free software
- Public domain equivalent license
- Free-software license
- Comparison of free and open-source software licenses
- Free Software Foundation

References

1. New Media Rights (2008-09-12). "Open Source Licensing Guide" (http://www.newmediarights.org/open_source/new_media_rights_open_source_licensing_guide). *California Western School of Law*.
2. "Top 20 licenses" (https://web.archive.org/web/20160719043600/https://www.blackducksoftware.com/top-open-source-licenses). Black Duck Software. 19 November 2015. Archived from the original (http://www.blackducksoftware.com/resources/data/top-20-licenses) on 19 July 2016. Retrieved 19 November 2015. "*1. MIT license 24%, 2. GNU General Public License (GPL) 2.0 23%, 3. Apache License 16%, 4. GNU General Public License (GPL) 3.0 9%, 5. BSD License 2.0 (3-clause, New or Revised) License 6%, 6. GNU Lesser General Public License (LGPL) 2.1 5%, 7. Artistic License (Perl) 4%, 8. GNU Lesser General Public License (LGPL) 3.0 2%, 9. Microsoft Public License 2%, 10. Eclipse Public License (EPL) 2%*"
3. Balter, Ben (2015-03-09). "Open source license usage on GitHub.com" (https://github.com/blog/1964-license-usage-on-github-com). github.com. Retrieved 2015-11-21. " "1 MIT 44.69%, 2 Other 15.68%, 3 GPLv2 12.96%, 4 Apache 11.19%, 5 GPLv3 8.88%, 6 BSD 3-clause 4.53%, 7 Unlicense 1.87%, 8 BSD 2-clause 1.70%, 9 LGPLv3 1.30%, 10 AGPLv3 1.05%"
4. Free Software Foundation, Various Licenses and Comments about Them, GNU All-permissive License (https://www.gnu.org/licenses/license-list.html#GNUAllPermissive)
5. Information for Maintainers of GNU Software, License Notices for Other Files (https://www.gnu.org/prep/maintain/html_node/License-Notices-for-Other-Files.html)
6. permissive (https://opensource.org/faq#permissive) on opensource.org "A "permissive" license is simply a non-copyleft open-source license – one that guarantees the freedoms to use, modify and redistribute, but that permits proprietary derivatives."
7. Choosing an open-source license doesn't need to be scary (https://choosealicense.com/) on choosealicense.com "Which of the following best describes your situation? – I want it simple and permissive."
8. "What is Copyleft" (https://www.gnu.org/copyleft/copyleft.html). GNU. Retrieved 21 April 2011.
9. "Categories of free and nonfree software" (https://www.gnu.org/philosophy/categories.html). gnu.org.
10. Amadeo, Ron (21 July 2018). "Google's iron grip on Android: Controlling open source by any means necessary" (https://arstechnica.com/gadgets/2018/07/googles-iron-grip-on-android-controlling-open-source-by-any-means-necessary/). *Ars Technica*.

11. With this in mind, the FreeBSD project advocates permissive licenses for companies and commercial use-cases: they say that they place only *"minimal restrictions on future behavior"* and argue that copyleft licenses are *"legal time-bombs"*. See Montague, Bruce (2013-11-13). "Why you should use a BSD style license for your Open Source Project" (http://www.freebsd.org/doc/en_US.ISO8859-1/articles/bsdl-gpl/article.html). FreeBSD. Retrieved 2015-11-28. "9. GPL Advantages and Disadvantages [..] 12. Conclusion
In contrast to the GPL, which is designed to prevent the proprietary commercialization of open-source code, the BSD license places minimal restrictions on future behavior. This allows BSD code to remain open source or become integrated into commercial solutions, as a project's or company's needs change. In other words, the BSD license does not become a legal time-bomb at any point in the development process.

In addition, since the BSD license does not come with the legal complexity of the GPL or LGPL licenses, it allows developers and companies to spend their time creating and promoting good code rather than worrying if that code violates licensing."
12. "Licence Compatibility" (https://web.archive.org/web/20150617130550/https://joinup.ec.europa.eu/software/page/licence_compatibility_and_interoperability). *European Union Public Licence*. joinup.ec.europa.eu. Archived from the original (https://joinup.ec.europa.eu/software/page/licence_compatibility_and_interoperability) on 2015-06-17. Retrieved 2015-05-30. "*The licenses for distributing free or open source software (FOSS) are divided in two families: permissive and copyleft. Permissive licenses (BSD, MIT, X11, Apache, Zope) are generally compatible and interoperable with most other licenses, tolerating to merge, combine or improve the covered code and to re-distribute it under many licenses (including non-free or "proprietary").*"
13. Hanwell, Marcus D. (2014-01-28). "Should I use a permissive license? Copyleft? Or something in the middle?" (http://opensource.com/business/14/1/what-license-should-i-use-open-source-project). opensource.com. Retrieved 2015-05-30. "*Permissive licensing simplifies things One reason the business world, and more and more developers [...], favor permissive licenses is in the simplicity of reuse. The license usually only pertains to the source code that is licensed and makes no attempt to infer any conditions upon any other component, and because of this there is no need to define what constitutes a derived work. I have also never seen a license compatibility chart for permissive licenses; it seems that they are all compatible.*"
14. "Frequently Asked Questions about the GNU Licenses – Is GPLv3 compatible with GPLv2?" (https://www.gnu.org/licenses/gpl-faq.html#v2v3Compatibility). gnu.org. Retrieved 2014-06-03. "*No. Some of the requirements in GPLv3, such as the requirement to provide Installation Information, do not exist in GPLv2. As a result, the licenses are not compatible: if you tried to combine code released under both these licenses, you would violate section 6 of GPLv2. However, if code is released under GPL "version 2 or later," that is compatible with GPLv3 because GPLv3 is one of the options it permits.*"
15. Landley, Rob. "CELF 2013 Toybox talk" (http://landley.net/talks/celf-2013.txt). *landley.net*. Retrieved 2013-08-21. "*GPLv3 broke "the" GPL into incompatible forks that can't share code.*"
16. "Interpreting, enforcing and changing the GNU GPL, as applied to combining Linux and ZFS" (https://www.fsf.org/licensing/zfs-and-linux). fsf.org. Retrieved 2020-06-08.
17. US Copyright Office Form CO (https://www.copyright.gov/forms/notice.html); see also *Ashton-Tate v. Fox*
18. "OpenBSD Copyright Policy" (https://www.openbsd.org/policy.html). The OpenBSD project. Retrieved 2020-06-09. "In some jurisdictions, it is doubtful whether voluntarily placing one's own work into the public domain is legally possible. For that reason, to make any substantial body of code free, it is preferable to state the copyright and put it under an ISC or BSD license instead of attempting to release it into the public domain."

19. Hipp, D. Richard. "Why SQLite succeeded as a database" (https://changelog.com/podcast/201#transcript-215). The Changelog. "Also at the time I did not realize, having lived my whole life in the United States, which is, you know, under British common law, where the public domain is something that's recognized. I did not realize that there were a lot of jurisdictions in the world where it's difficult or impossible for someone to place their works in the public domain. I didn't know. So that's a complication."
20. The Free-Libre / Open Source Software (FLOSS) License Slide (https://dwheeler.com/essays/floss-license-slide.html) by David A. Wheeler on October 4, 2021
21. Haff, Gordon. "The mysterious history of the MIT License" (https://opensource.com/article/19/4/history-mit-license). opensource.com. Retrieved 2020-06-08. "*[There's] a good argument to be made that the MIT License, also called the X Consortium or X11 License at the time, crystallized with X11 in 1987, and that's the best date to use. You could argue it was created in 1985 with possible adjustments over the next couple of years.*"
22. Vaughan-Nichols, Steven J. "The fall of GPL and the rise of permissive open-source licenses" (https://www.zdnet.com/article/the-fall-of-gpl-and-the-rise-of-permissive-open-source-licenses/). ZDNet. Retrieved 2015-11-28. "*The GPL is still the world's most popular open-source license but it's declining in use, while permissive licenses are gaining more fans, and some developers are choosing to release code without any license at all.*"
23. Ronacher, Armin (2013-07-23). "Licensing in a Post Copyright World" (http://lucumr.pocoo.org/2013/7/23/licensing/). lucumr.pocoo.org. Retrieved 2015-11-18.
24. Aslett, Matthew (2011-06-06). "The trend towards permissive licensing" (https://web.archive.org/web/20151013085257/https://blogs.the451group.com/opensource/2011/06/06/the-trend-towards-permissive-licensing/). the451group.com. Archived from the original (https://blogs.the451group.com/opensource/2011/06/06/the-trend-towards-permissive-licensing/) on 2015-10-13. Retrieved 2015-11-28.
25. Does your code need a license? (http://opensource.com/law/13/5/does-your-code-need-license) Posted 02 May 2013 by Jason Hibbets "Q: Are there software-development companies favoring a certain open-source license over another? What is the trend in the community? A: We're definitely seeing some trends away from copyleft licenses—mostly towards permissive licenses"
26. "Frequently Asked Questions | Open Source Initiative" (https://opensource.org/faq#permissive). *Open Source Initiative*. Retrieved 2022-08-09. "A 'permissive' license is simply a non-copyleft open source license"
27. "Copyleft versus non-copyleft licenses in free / open source software" (https://kbdeveloper.qoppa.com/copyleft-versus-non-copyleft-licenses-in-free-open-source-software/). *Qoppa Software*. 2014-11-21. Retrieved 2022-08-09.
28. Sen, Ravi; Subramaniam, Chandrasekar; Nelson, Matthew L. (2011). "Open Source Software Licenses: Strong-Copyleft, Non-Copyleft, or Somewhere in Between?" (https://mays.tamu.edu/research/open-source-software-licenses-strong-copyleft-non-copyleft-or-somewhere-in-between/). *Decision Support Systems*. **52** (1): 199–206. doi:10.1016/j.dss.2011.07.004 (https://doi.org/10.1016%2Fj.dss.2011.07.004). ISSN 0167-9236 (https://search.worldcat.org/issn/0167-9236). Retrieved 2022-08-09.
29. "Add Kirk's comment about "copycenter"; it's just too good to pass up" (https://github.com/freebsd/freebsd/commit/60a677e4faa8053b9d87688d1a5575d1331fc8ce). Historical FreeBSD fortune(6) database. Retrieved 2020-06-08.
30. Raymond, Eric S. "copycenter" (http://www.catb.org/jargon/html/C/copycenter.html). The Jargon File.

31. Stallman, Richard (2016-02-08). "License Compatibility and Relicensing" (https://www.gnu.org/licenses/license-compatibility.html). Free Software Foundation. Retrieved 2019-09-29. "*In general, lax permissive licenses (modified BSD, X11, Expat, Apache, Python, etc.) are compatible with each other. That's because they have no requirements about other code that is added to the program. They even permit putting the entire program (perhaps with changes) into a proprietary software product; thus, we call them "pushover licenses" because they can't say "no" when one user tries to deny freedom to others.*"
32. How to choose a license for your own work – Free Software Foundation (https://www.gnu.org/licenses/license-recommendations.html#small)

External links

Public domain

The **public domain** (**PD**) consists of all the creative work to which no exclusive intellectual property rights apply. Those rights may have expired,[1] been forfeited,[2] expressly waived, or may be inapplicable.[3] Because no one holds the exclusive rights, anyone can legally use or reference those works without permission.[3][4]

As examples, the works of William Shakespeare, Ludwig van Beethoven, Miguel de Cervantes, Zoroaster, Lao Zi, Confucius, Aristotle, L. Frank Baum, Leonardo da Vinci and Georges Méliès are in the public domain either by virtue of their having been created before copyright existed, or by their copyright term having expired.[1] Some works are not covered by a country's copyright laws, and are therefore in the public domain; for example, in the United States, items excluded from copyright include the formulae of Newtonian physics and cooking recipes.[5] Other works are actively dedicated by their authors to the public domain (see waiver); examples include reference implementations of cryptographic algorithms.[6] The term *public domain* is not normally applied to situations where the creator of a work retains residual rights, in which case use of the work is referred to as "under license" or "with permission".

As rights vary by country and jurisdiction, a work may be subject to rights in one country and be in the public domain in another. Some rights depend on registrations on a country-by-country basis, and the absence of registration in a particular country, if required, gives rise to public-domain status for a work in that country. The term *public domain* may also be interchangeably used with other imprecise or undefined terms such as the *public sphere* or *commons*, including concepts such as the "commons of the mind", the "intellectual commons", and the "information commons".[7]

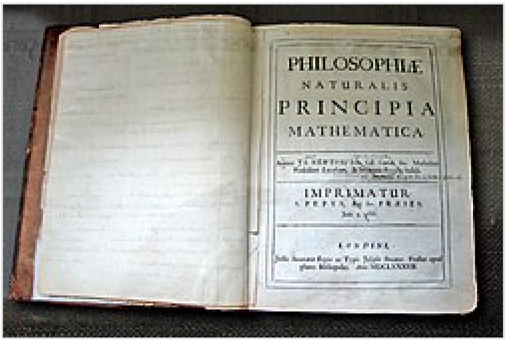
History

Although the term *domain* did not come into use until the mid-18th century, the concept can be traced back to the ancient Roman law, "as a preset system included in the property right system".[8] The Romans had a large proprietary rights system where they defined "many things that cannot be privately owned"[8] as *res nullius*, *res communes*, *res publicae* and *res universitatis*.[9] The term *res nullius* was defined as things not yet appropriated.[10] The term *res communes* was defined as "things that could be commonly enjoyed by mankind, such as air, sunlight and ocean."[8] The term *res publicae* referred to things that were shared by all citizens, and the term *res universitatis* meant things that were owned by the municipalities of Rome.[8] When looking at it from a historical perspective, one could say the construction of the idea of "public domain" sprouted from the concepts of *res communes*, *res publicae*, and *res universitatis* in early Roman law.[8]

When the first early copyright law was originally established in Britain with the Statute of Anne in 1710, public domain did not appear. However, similar concepts were developed by British and French jurists in the 18th century. Instead of "public domain", they used terms such as *publici juris* or *propriété publique* to describe works that were not covered by copyright law.[11]

The phrase "fall in the public domain" can be traced to mid-19th-century France to describe the end of copyright term. The French poet Alfred de Vigny equated the expiration of copyright with a work falling "into the sink hole of public domain"[12] and if the public domain receives any attention from intellectual property lawyers it is still treated as little more than that which is left when intellectual property rights, such as copyright, patents, and trademarks, expire or are abandoned.[7] In this historical context Paul Torremans describes copyright as a "little coral reef of private right jutting up from the ocean of the public domain".[13] Copyright law differs by country, and the American legal scholar Pamela Samuelson has described the public domain as being "different sizes at different times in different countries".[14]

Definition



Newton's own copy of his *Principia*, with hand-written corrections for the second edition

Definitions of the boundaries of the public domain in relation to copyright, or intellectual property more generally, regard the public domain as a negative space; that is, it consists of works that are no longer in copyright term or were never protected by copyright law.[15] According to James Boyle this definition underlines common usage of the term *public domain* and equates the public domain to public property and works in copyright to private property. However, the usage of the term *public domain* can be more granular, including for example uses of works in copyright permitted by copyright exceptions. Such a definition regards work in copyright as private property subject to fair use rights and limitation on ownership.[1] A conceptual definition comes from Lange, who focused on what the public domain should be: "it should be a place of sanctuary for individual creative expression, a sanctuary conferring affirmative protection against the forces of private appropriation that threatened such expression".[15] Patterson and Lindberg described the public domain not as a "territory", but rather as a concept: "[T]here are certain materials – the air we breathe, sunlight, rain, space, life, creations, thoughts, feelings, ideas, words, numbers – not subject to private ownership. The materials that compose our cultural heritage must be free for all living to use no less than matter necessary for biological survival."[16] The term *public domain* may also be interchangeably used with other imprecise or undefined terms such as the *public sphere* or *commons*, including concepts such as the "commons of the mind", the "intellectual commons", and the "information commons".[7]

Public domain by medium

Books

A public-domain book is a book with no copyright, a book that was created without a license, or a book where its copyrights expired[17] or have been forfeited.[18]

In most countries the term of protection of copyright expires on the first day of January, 70 years after the death of the latest living author. The longest copyright term is in Mexico, which has life plus 100 years for all deaths since July 1928.[19]

A notable exception is the United States, where every book and tale published before 1930 is in the public domain; US copyrights last for 95 years for books originally published between 1930 and 1978 if the copyright was properly registered and maintained.[20]

For example: the works of Jane Austen, Lewis Carroll, Machado de Assis, Olavo Bilac and Edgar Allan Poe are in the public domain worldwide as they all died over 100 years ago.[21]

Project Gutenberg, the Internet Archive and Wikisource make tens of thousands of public domain books available online as ebooks.[22][23][24]

Music

People have been creating music for millennia. The first musical notation system, the Music of Mesopotamia system, was created 4,000 years ago. Guido of Arezzo introduced Latin musical notation in the 10th century.[25] This laid the foundation for the preservation of global music in the public domain, a distinction formalized alongside copyright systems in the 17th century. Musicians copyrighted their publications of musical notation as literary writings, but performing copyrighted pieces and creating derivative works were not restricted by early copyright laws. Copying was widespread, in compliance with the law, but expansions of those laws intended to benefit literary works and responding to commercial music recording technology's reproducibility have led to stricter rules. Relatively recently, a normative view that copying in music is not desirable and lazy has become popular among professional musicians.

US copyright laws distinguish between musical compositions and sound recordings, the former of which refers to melody, notation or lyrics created by a composer or lyricist, including sheet music, and the latter referring to a recording performed by an artist, including a CD, LP, or digital sound file.[26] Musical compositions fall under the same general rules as other works, and anything published before 1925 is considered public domain. Sound recordings, on the other hand, are subject to different rules and are not eligible for public domain status until 2021–2067, depending on the date and location of publishing, unless explicitly released beforehand.[20]

The Musopen project records music in the public domain for the purposes of making the music available to the general public in a high-quality audio format. Online musical archives preserve collections of classical music recorded by Musopen and offer them for download/distribution as a public service.

Films

A public-domain film is a film that was never under copyright, was released to public domain by its author, or whose copyright has expired. All films released in the United States before 1 January 1930 have been entered in the public domain in that country.

Value

Pamela Samuelson has identified eight "values" that can arise from information and works in the public domain.[27]

Possible values include:

1. Building blocks for the creation of new knowledge, examples include data, facts, ideas, theories, and scientific principle.
2. Access to cultural heritage through information resources such as ancient Greek texts and Mozart's symphonies.
3. Promoting education, through the spread of information, ideas, and scientific principles.
4. Enabling follow-on innovation, through for example expired patents and copyright.
5. Enabling low cost access to information without the need to locate the owner or negotiate rights clearance and pay royalties, through for example expired copyrighted works or patents, and non-original data compilation.[28]
6. Promoting public health and safety, through information and scientific principles.
7. Promoting the democratic process and values, through news, laws, regulation, and judicial opinion.
8. Enabling competitive imitation, through for example expired patents and copyright, or publicly disclosed technologies that do not qualify for patent protection.[27]:22

The 1968 horror film *Night of the Living Dead* is public domain in the United States because its theatrical distributor failed to place a copyright indication on the prints, as would have been required to obtain a copyright at that time.

Relationship with derivative works

Derivative works include translations, musical arrangements, and dramatizations of a work, as well as other forms of transformation or adaptation.[29] Copyrighted works may not be used for derivative works without permission from the copyright owner,[30] while public domain works can be freely used for derivative works without permission.[31][32] Artworks that are public domain may also be reproduced photographically or artistically or used as the basis of new, interpretive works.[33] Works derived from public domain works can be copyrighted.[34]

Once works enter into the public domain, derivative works such as adaptations in book and film may increase noticeably, as happened with Frances Hodgson Burnett's novel *The Secret Garden*, which became public domain in the US in 1977 and most of the rest of the world in 1995.[35] By 1999, the plays of Shakespeare, all public domain, had been used in more than 420 feature-length films.[36][37] In addition to straightforward adaptation, they have been used as the launching point for transformative retellings such as Tom Stoppard's *Rosencrantz and Guildenstern Are Dead* and Troma Entertainment's *Tromeo and Juliet*.[38][39][40] Marcel Duchamp's *L.H.O.O.Q.* is a derivative of Leonardo da Vinci's *Mona Lisa*, one of thousands of derivative works based on the public domain painting.[31] The 2018 film *A Star is Born* is a remake of the 1937 film of the same name, which is in the public domain due to an unrenewed copyright.[41]

Rights in public domain reproduction

Courts in different jurisdictions have come to different conclusions as to whether the reproduction of a public domain work gains its own rights protection, or whether it too is in the public domain. In a German 2016 case, the Reiss-Engelhorn-Museen, an art museum, sued Wikimedia Commons over photographs uploaded to the database depicting pieces of art in the museum. The museum claimed that the photos were taken by their staff, and that photography within the museum by visitors was prohibited. Therefore, photos taken by the museum, even of material that itself had fallen into the public domain, were protected by copyright law and would need to be removed from the Wikimedia image repository. The court ruled that the photographs taken by the museum would be protected under the German Copyright Act, stating that since the photographer needed to make practical decisions about the photograph that it was protected material.[42] In contrast, in the 1999 US case *Bridgeman Art Library v. Corel Corp.*, the court ruled that exact photographic copies of public domain images could not be protected by copyright in the United States because the copies lack originality.[43]

Perpetual copyright

In some countries, certain works may never fully lapse into the public domain. In the United Kingdom, for example, there is a perpetual crown copyright for the Authorized King James Version of the Bible.[44]

While the copyright has expired for the Peter Pan works by J. M. Barrie (the play *Peter Pan, or the Boy Who Wouldn't Grow Up* and the novel *Peter and Wendy*) in the United Kingdom, it was granted a special exception under the Copyright, Designs and Patents Act 1988 (Schedule 6)[45] that requires royalties to be paid for commercial performances, publications and broadcasts of the story of Peter Pan within the UK, as long as Great Ormond Street Hospital (to whom Barrie gave the copyright) continues to exist.

In a paying public domain regime, works that have entered the public domain after their copyright has expired, or traditional knowledge and traditional cultural expressions that have never been subject to copyright, are still subject to royalties payable to the state or to an authors' association. The user does not have to seek permission to copy, present or perform the work, but does have to pay the fee. Typically the royalties are directed to support of living artists.[46]

Public domain mark

In 2010, The Creative Commons proposed the Public Domain Mark (PDM) as symbol to indicate that a work is free of known copyright restrictions and therefore in the public domain.[47][48] The public domain mark is a combination of the copyright symbol, which acts as copyright notice, with the international 'no' symbol. The Europeana databases use it, and for instance on the Wikimedia Commons in February 2016 2.9 million works (~10% of all works) are listed with the mark.[49]

Creative Commons' Public Domain Mark

Application to copyrightable works

Works not covered by copyright law

The underlying idea that is expressed or manifested in the creation of a work generally cannot be the subject of copyright law (see idea–expression divide). Mathematical formulae will therefore generally form part of the public domain, to the extent that their expression in the form of software is not covered by copyright.[50]

Works created before the existence of copyright and patent laws also form part of the public domain. For example, the Bible and the inventions of Archimedes are in the public domain. However, translations or new formulations of these works may be copyrighted in themselves.[51]

Expiration of copyright

Determination of whether a copyright has expired depends on an examination of the copyright in its source country.

In most countries that are signatories to the Berne Convention, copyright term is based on the life of the author, and extends to 50 or 70 years beyond the death of the author. (See List of copyright terms of countries.)

In the United States, determining whether a work has entered the public domain or is still under copyright depends upon what the law or regulation was at creation, and whether new regulations have grandfathered in certain older works. Because copyright terms shifted over the course of the 20th century from a fixed-term based on first publication, with a possible renewal term, to a term extending to 50, then 70, years after the death of the author. The claim that "pre-1930 works are in the public domain" is correct only for published works; unpublished works are under federal copyright for at least the life of the author plus 70 years.

Legal traditions differ on whether a work in the public domain can have its copyright restored. In the European Union, the Copyright Duration Directive was applied retroactively, restoring and extending the terms of copyright on material previously in the public domain. Term extensions by the US and Australia generally have not removed works from the public domain, but rather delayed the addition of works to it. However, the United States moved away from that tradition with the Uruguay Round Agreements Act, which removed from the public domain many foreign-sourced works that had previously not been in copyright in the US for failure to comply with US-based formalities requirements. Consequently, in the US, foreign-sourced works and US-sourced works are now treated differently, with foreign-sourced works remaining under copyright regardless of compliance with formalities, while domestically sourced works may be in the public domain if they failed to comply with then-existing formalities requirements—a situation described as odd by some scholars, and unfair by some US-based rightsholders.[52]

Government works

Works of various governments around the world may be excluded from copyright law and may therefore be considered to be in the public domain in their respective countries.[53] They may also be in the public domain in other countries as well. Material in the public domain is still considered so when included as part of larger copyrighted creations.[54]

In the United States, work created by the federal government is not subject to copyright law, placing it within the public domain. However, the government may own and use copyrighted materials that were not initially created by them.[55] Alternatively, materials created by the United Kingdom's government are not automatically in public domain but instead placed under the Open Government Licence.[56] The status of government creations vary depending on the country they are in.

Dedicating works to the public domain

Release without copyright notice

Before 1 March 1989, in the US, works could be easily given into the public domain by just releasing it without an explicit copyright notice. With the Berne Convention Implementation Act of 1988 (and the earlier Copyright Act of 1976, which went into effect in 1978), all works were by default copyright protected and needed to be actively given into public domain by a waiver statement/anti-copyright can call notice.[57][58] Not all legal systems have processes for reliably donating works to the public domain, e.g. civil law of continental Europe. This may even "effectively prohibit any attempt by copyright owners to surrender rights automatically conferred by law, particularly moral rights".[59]

Public-domain-like licenses

An alternative is for copyright holders to issue a license which irrevocably grants as many rights as possible to the general public. Real public domain makes licenses unnecessary, as no owner/author is required to grant permission ("Permission culture"). There are multiple licenses which aim to release works into the public domain. In 2000 the WTFPL was released as a public domain like software license.[60] Creative Commons (created in 2002 by Lawrence Lessig, Hal Abelson, and Eric Eldred) has introduced several public-domain-like licenses, called Creative Commons licenses. These give authors of works (that would qualify for copyright) the ability to decide which protections they would like to place on their material. As copyright is the default license for new material, Creative Commons licenses offer authors a variety of options to designate their work under whichever license they wish, as long as this does not violate standing copyright law.[61] For example, a CC BY license allows for re-users to distribute, remix, adapt, and build upon material, while also agreeing to provide attribution to the author in any of these cases.[62] In 2009 the Creative Commons released the CC0, which was created for compatibility with law domains which have no concept of *dedicating into public domain*. This is achieved by a public domain waiver statement and a fallback all-permissive license, in case the waiver is not possible.[63][64] Unlike in the US, where author's moral rights are generally not specifically regulated, in some countries where moral rights are protected separately in law it is not possible to waive those rights, but only the rights related to the exploitation of the work. A solution to this issue (as found in the Creative Commons Zero dedication) is to interpret the license by setting "three different layers of action. First, the right holder waives any copyright and related rights that can be waived in accordance with the applicable law. Secondly, if there are rights that the right holder cannot waive under applicable law, they are licensed in a way that mirrors as closely as possible the legal effect of a waiver. And finally, if there are any rights that the right holders cannot waive or license, they affirm that they will not exercise them and they will not assert any claim with respect to the use of the work, once again within the limits of applicable law. (...) In countries where moral rights exist but where they can be waived or not asserted, they are waived if asserted (e.g. the UK). In countries where they cannot be waived they will remain into full effect in accordance to the applicable law (think of France, Spain or Italy where moral rights cannot be waived)."[65] The same occurs in Switzerland.

The Unlicense, published around 2010, has a focus on an anti-copyright message. The Unlicense offers a public domain waiver text with a fallback public domain-like license inspired by permissive licenses but without attribution.[66][67] Another option is the Zero Clause BSD license, released in 2006 and aimed at software.[68]

In October 2014, the Open Knowledge Foundation recommends the Creative Commons CC0 license to dedicate content to the public domain,[69][70] and the Open Data Commons Public Domain Dedication and License (PDDL) for data.[71]

Patents

In most countries, the term of rights for patents is 20 years, after which the invention becomes part of the public domain. In the United States, the contents of patents are considered valid and enforceable for 20 years from the date of filing within the United States or 20 years from the earliest date of filing if under 35 USC 120, 121, or 365(c).[72] However, the *text* and any *illustration* within a patent, provided the illustrations are essentially line drawings and do not in any substantive way reflect the "personality" of the person drawing them, are not subject to copyright protection.[73] This is separate from the *patent* rights just mentioned.

Trademarks

A trademark registration may remain in force indefinitely, or expire without specific regard to its age. For a trademark registration to remain valid, the owner must continue to use it. In some circumstances, such as disuse, failure to assert trademark rights, or common usage by the public without regard for its intended use, it could become generic, and therefore part of the public domain.

Because trademarks are registered with governments, some countries or trademark registries may recognize a mark, while others may have determined that it is generic and not allowable as a trademark in that registry. For example, the drug acetylsalicylic acid (2-acetoxybenzoic acid) is better known as *aspirin* in the United States—a generic term. In Canada, however, *Aspirin*, with an uppercase A, is still a trademark of the German company Bayer, while aspirin, with a lowercase "a", is not. Bayer lost the trademark in the United States, the UK and France after World War I, as part of the Treaty of Versailles. So many copycat products entered the marketplace during the war that it was deemed generic just three years later.

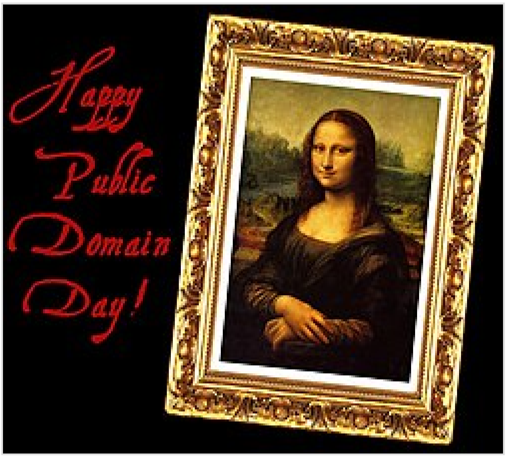
Informal uses of trademarks are not covered by trademark protection. For example, Hormel, producer of the canned meat product Spam, does not object to informal use of the word "spam" in reference to unsolicited commercial email.[74] However, it has fought attempts by other companies to register names including the word 'spam' as a trademark in relation to computer products, despite that Hormel's trademark is only registered in reference to food products (a trademark claim is made within a particular field). Such defences have failed in the United Kingdom.[75]

Public Domain Day

Public Domain Day is an observance of when copyrighted works expire and works enter into the public domain.[76] This legal transition of copyright works into the public domain usually happens every year on 1 January based on the individual copyright laws of each country.[76]

The observance of a "Public Domain Day" was initially informal; the earliest known mention was in 2004 by Wallace McLean (a Canadian public domain activist),[77] with support for the idea echoed by Lawrence Lessig.[78] As of 1 January 2010,[79] there is as Public Domain Day website lists the authors whose works are entering the public domain. There are activities in countries around the world by various organizations all under the banner Public Domain Day, this can help people around the world celebrate works written a while ago.

An English logo of the 2023/2024 Public Domain Day



Visual created for Public Domain Day. Features Leonardo da Vinci's *Mona Lisa*, as it is famously part of the public domain

See also

- Public records
- Center for the Study of the Public Domain
- Copyfraud
- Copyleft
- Copyright status of works by the federal government of the United States
- Copyright Term Extension Act
- *Eldred v. Ashcroft*
- Fair dealing
- Free-culture movement
- Free software
- Freedom of panorama
- Limitations and exceptions to copyright
- List of copyright terms of countries
- List of films in the public domain in the United States
- *Millar v Taylor*
- Orphan works
- Paying public domain
- Protection of Classics
- Public Domain Enhancement Act
- Public domain image resources
- Public domain in the United States
- Public domain software
- Rule of the shorter term

References

1. Boyle, James (2008). *The Public Domain: Enclosing the Commons of the Mind* (https://books.google.com/books?id=Fn1Pl9Gv_EMC&q=public+domain). CSPD. p. 38. ISBN 978-0-300-13740-8. Archived (https://web.archive.org/web/20150214065428/http://www.google.com/books?id=Fn1Pl9Gv_EMC&dq=public+domain&source=gbs_navlinks_s) from the original on 14 February 2015.

2. Graber, Christoph B.; Nenova, Mira B. (2008). *Intellectual Property and Traditional Cultural Expressions in a Digital Environment* (https://web.archive.org/web/20141220043810/http://www.google.com/books?id=gK6OI0hrANsC&dq=%22public+domain%22+intellectual+property&lr=&as_brr=3&source=gbs_navlinks_s). Edward Elgar Publishing. p. 173. ISBN 978-1-84720-921-4. Archived from the original (https://books.google.com/books?id=gK6OI0hrANsC&q=%22public+domain%22+intellectual+property) on 20 December 2014. Retrieved 27 October 2016.
3. "Works Unprotected by Copyright Law" (https://web.archive.org/web/20160302173206/http://www.bitlaw.com/copyright/unprotected.html). *bitlaw.com*. Archived from the original (http://www.bitlaw.com/copyright/unprotected.html) on 2 March 2016.
4. "What Is the Public Domain?" (https://www.copyrightlaws.com/what-is-the-public-domain/). *Copyrightlaws.com: Copyright courses and education in plain English*. 7 March 2023. Retrieved 9 October 2024.
5. "Copyright Protection Not Available for Names, Titles, or Short Phrases" (https://web.archive.org/web/20160405033421/http://copyright.gov/circs/circ34.pdf) (PDF). *copyright.gov*. Archived from the original (https://copyright.gov/circs/circ34.pdf) (PDF) on 5 April 2016. "Listings of ingredients, as in recipes, labels, or formulas. When a recipe or formula is accompanied by an explanation or directions, the text directions may be copyrightable, but the recipe or formula itself remains uncopyrightable."
6. "SERPENT - A Candidate Block Cipher for the Advanced Encryption Standard" (https://web.archive.org/web/20130113104543/http://www.cl.cam.ac.uk/~rja14/serpent.html). 1999. Archived from the original (http://www.cl.cam.ac.uk/~rja14/serpent.html) on 13 January 2013. "Serpent is now completely in the public domain, and we impose no restrictions on its use. This was announced on 21 August at the First AES Candidate Conference."
7. Ronan 2006, p. 103 (https://books.google.com/books?id=dMYXq9V1JBQC&q=statute+of+anne+copyright).
8. Huang 2009.
9. Foures-Diop, Anne-Sophie (2011). "Revue juridique de l'Ouest, 2011-1: Les choses communes (Première partie)" (https://www.persee.fr/doc/juro_0990-1027_2011_num_24_1_4336) [Common things (Part I)]. *Revue Juridique de l'Ouest* (in French). **24** (1). Ecole de Avocats du Grand Quest de Rennes: 59–112. doi:10.3406/juro.2011.4336 (https://doi.org/10.3406%2Fjuro.2011.4336).
10. Rose, Carol M. (2003). "Romans, Roads, and Romantic Creators: Traditions of Public Property in the Information Age" (http://www.jstor.org/stable/20059173). *Law and Contemporary Problems*. **66** (1/2). Duke University School of Law: 89–110. ISSN 0023-9186 (https://search.worldcat.org/issn/0023-9186). JSTOR 20059173 (https://www.jstor.org/stable/20059173).
11. Torremans 2007, pp. 134-135 (https://books.google.com/books?id=wHJBemWuPT4C&q=%22perpetual+copyright%22).
12. Torremans 2007, p. 154 (https://books.google.com/books?id=wHJBemWuPT4C&q=%22perpetual+copyright%22).
13. Torremans 2007, p. 137 (https://books.google.com/books?id=wHJBemWuPT4C&q=%22perpetual+copyright%22).
14. Ronan 2006, p. 102 (https://books.google.com/books?id=dMYXq9V1JBQC&q=statute+of+anne+copyright).
15. Ronan 2006, p. 104 (https://books.google.com/books?id=dMYXq9V1JBQC&q=statute+of+anne+copyright).
16. Ronan 2006, p. 105 (https://books.google.com/books?id=dMYXq9V1JBQC&q=statute+of+anne+copyright).
17. Boyle, James (1 January 2008). *The Public Domain: Enclosing the Commons of the Mind* (https://archive.org/details/publicdomainencl0000boyl). Yale University Press. ISBN 978-0-300-13740-8. Retrieved 30 December 2016 – via Internet Archive. "public domain."

18. Graber, Christoph Beat; Nenova, Mira Burri (1 January 2008). *Intellectual Property and Traditional Cultural Expressions in a Digital Environment* (https://books.google.com/books?id=gK6OI0hrANsC&q=%22public%20domain%22%20intellectual%20property). Edward Elgar Publishing. ISBN 978-1-84844-391-4. Retrieved 30 December 2016 – via Google Books.
19. Kopel, Matthew. "LibGuides: Copyright Services: Copyright Term and the Public Domain" (https://guides.library.cornell.edu/copyright/publicdomain). *guides.library.cornell.edu*. Retrieved 9 October 2024.
20. "Copyright Term and the Public Domain in the United States" (https://copyright.cornell.edu/publicdomain). Cornell University. Retrieved 15 October 2018.
21. "Public domain | Definition, Meaning, Examples, Years, & Copyright | Britannica" (https://www.britannica.com/topic/public-domain). *www.britannica.com*. Retrieved 9 October 2024.
22. Preston, Sherry (9 October 2023). "Cover to Cover: Access thousands of books on Project Gutenberg" (https://starherald.com/news/community/gering/cover-to-cover-access-thousands-of-books-on-project-gutenberg/article_0a968c7a-6706-11ee-b8b5-1bb3f4ad78df.html). *Star-Herald*. Retrieved 5 July 2024.
23. Robertson, Adi (31 March 2020). "The National Emergency Library is offering free ebooks — but is it lending or piracy?" (https://www.theverge.com/2020/3/31/21200108/internet-archive-national-emergency-library-authors-guild-piracy-cdl). *The Verge*. Retrieved 5 July 2024.
24. "What is Wikisource?" (https://www.ed.ac.uk/information-services/help-consultancy/is-skills/wikimedia/wikisource/what-is-wikisource). *The University of Edinburgh*. 26 August 2021. Retrieved 5 July 2024.
25. Otten, J. "The Catholic Encyclopedia" (https://www.newadvent.org/cathen/07065a.htm). *New Advent*. Robert Appleton Company. Retrieved 6 January 2022.
26. "Copyright Registration of Musical Compositions and Sound Recordings" (https://www.copyright.gov/circs/circ56a.pdf) (PDF). United States Copyright Office. Retrieved 15 October 2018.
27. Guibault, Lucy; Hugenholtz, Bernt (2006). *The future of the public domain: identifying the commons in information law* (https://books.google.com/books?id=KJmNGglq0nwC&q=public+domain). Kluwer Law International. ISBN 978-9-0411-24357. Archived (https://web.archive.org/web/20141218080732/http://www.google.com/books?id=KJmNGglq0nwC&dq=public+domain&lr=&as_brr=3&source=gbs_navlinks_s) from the original on 18 December 2014.
28. March Perry; Thomas Margoni (2010). "From music tracks to Google maps: who owns Computer Generated Works?". SSRN 1647584 (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1647584).
29. Stern, Richard H. (2001). "L.H.O.O.Q. Internet related Derivative Works" (https://web.archive.org/web/20180819013902/http://docs.law.gwu.edu/facweb/claw/lhooq0.htm). *Supplemental material Computer Law 484*. The George Washington University Law School. Archived from the original (http://docs.law.gwu.edu/facweb/claw/Lhooq0.htm) on 19 August 2018. Retrieved 23 May 2010.
30. Leaffer, Marshall A. (1995). *Understanding copyright law* (https://archive.org/details/understandingcop00leaf/page/46). Legal text series; Contemporary Casebook Series (2nd ed.). M. Bender. p. 46 (https://archive.org/details/understandingcop00leaf/page/46). ISBN 0-256-16448-7.
31. *Introduction to intellectual property: theory and practice* (https://web.archive.org/web/20150406230406/http://www.google.com/books?id=n7DkfPpwLbEC&dq=adaptation+public+domain+disney&lr=&source=gbs_navlinks_s). World Intellectual Property Organisation, Kluwer Law International. 1997. p. 313. ISBN 978-90-411-0938-5. Archived from the original (https://books.google.com/books?id=n7DkfPpwLbEC&q=adaptation+public+domain+disney) on 6 April 2015.

32. Fishman, Stephen (September 2008). *The copyright handbook: what every writer needs to know* (https://archive.org/details/copyrighthandboo0000fish). Nolo. p. 178. ISBN 978-1-4133-0893-8. Retrieved 1 June 2010.
33. Fishman, Stephen (2008). *Public domain: how to find and use copyright-free writings, music, art and more* (https://archive.org/details/publicdomainhowt00fish_3). Nolo. pp. 124–125. ISBN 978-1-4133-0858-7.
34. Stim, Rich (4 April 2013). "Public Domain Trouble Spots" (https://fairuse.stanford.edu/overview/public-domain/trouble-spots/). *Stanford Copyright and Fair Use Center*. Public Domain Works That Are Modified. Archived (https://web.archive.org/web/20160518231739/http://fairuse.stanford.edu/overview/public-domain/trouble-spots) from the original on 18 May 2016.
35. Lundin, Anne H. (2 August 2004). *Constructing the canon of children's literature: beyond library walls and ivory towers* (https://books.google.com/books?id=72flittye58C&pg=PA138). Routledge. p. 138. ISBN 978-0-8153-3841-3. Retrieved 1 June 2010.
36. Young, Mark (ed.). *The Guinness Book of Records 1999*. Bantam Books. p. 358.
37. Voigts-Virchow, Eckartm (2004). *Janespotting and Beyond: British Heritage Retrovisions Since the Mid-1990s*. Gunter Narr Verlag. p. 92.
38. Homan, Sidney (2004). *Directing Shakespeare: a scholar onstage* (https://books.google.com/books?id=EM0n1ueBa_sC&pg=PT101). Ohio University Press. p. 101. ISBN 978-0-8214-1550-4. Retrieved 1 June 2010.
39. Kossak, Saskia (2005). *"Frame my face to all occasions": Shakespeare's Richard III on screen* (https://books.google.com/books?id=-PNZAAAAMAAJ). Braumüller. p. 17. ISBN 978-3-7003-1492-9. Retrieved 1 June 2010.
40. Cartmell, Deborah; Whelehan, Imelda (2007). *The Cambridge companion to literature on screen* (https://books.google.com/books?id=fhJNFc1f0DAC&pg=PA69). Cambridge University Press. p. 69. ISBN 978-0-521-61486-3. Retrieved 1 June 2010.
41. Lenker, Maureen Lee (10 July 2022). "Everything you need to know about the 1937 version of 'A Star Is Born' " (https://ew.com/movies/2018/10/07/a-star-is-born-1937-everything-to-know/). *Entertainment Weekly*. Retrieved 12 August 2020.
42. Michel, Stefan (1 June 2019). "Digitisation of art in the public domain–museum urges Wikimedia to take down reproductions of out-of-protection artworks". *Journal of Intellectual Property Law & Practice*. **14** (6): 427–429. doi:10.1093/jiplp/jpz042 (https://doi.org/10.1093%2Fjiplp%2Fjpz042). ISSN 1747-1532 (https://search.worldcat.org/issn/1747-1532).
43. Simon Stokes (2001). *Art and copyright*. Hart Publishing. pp. 103–104. ISBN 978-1-84113-225-9.
44. Metzger, Bruce M. (2006). *The Oxford companion to the Bible* (https://archive.org/details/isbn_9780195046458/page/618). Oxford: Oxford Univ. Press. pp. 618 (https://archive.org/details/isbn_9780195046458/page/618). ISBN 978-0-1950-46458.
45. "Copyright, Designs and Patents Act 1988 (c. 48)" (https://web.archive.org/web/20080601225021/http://www.opsi.gov.uk/acts/acts1988/Ukpga_19880048_en_28.htm). *Office of Public Sector Information*. 1988. p. 28. Archived from the original (http://www.opsi.gov.uk/acts/acts1988/Ukpga_19880048_en_28.htm) on 1 June 2008. Retrieved 2 September 2008.
46. WIPO Secretariat (24 November 2010), *Note on the Meanings of the Term "Public Domain" in the Intellectual Property System with special reference to the Protection of Traditional Knowledge and Traditional Cultural Expressions/Expressions of Folklore* (http://www.wipo.int/edocs/mdocs/tk/en/wipo_grtkf_ic_17/wipo_grtkf_ic_17_inf_8.doc), Intergovernmental Committee on Intellectual Property and Genetic Resources, Traditional Knowledge and Folklore: Seventeenth Session, retrieved 28 November 2018
47. "Creative Commons announces the Public Domain Mark" (https://web.archive.org/web/20101016044229/http://www.h-online.com/open/news/item/Creative-Commons-announces-the-Public-Domain-Mark-1106052.html). *The H Open*. The H. 12 October 2010. Archived from the original (http://www.h-online.com/open/news/item/Creative-Commons-announces-the-Public-Domain-Mark-1106052.html) on 16 October 2010. Retrieved 12 October 2010.

48. Peters, Diane (11 October 2010). "Improving Access to the Public Domain: the Public Domain Mark" (https://creativecommons.org/weblog/entry/23830). Creative Commons. Archived (https://web.archive.org/web/20101014022853/http://creativecommons.org/weblog/entry/23830) from the original on 14 October 2010. Retrieved 12 October 2010.
49. Category:CC-PD-Mark (https://commons.wikimedia.org/wiki/Category:CC-PD-Mark) Archived (https://web.archive.org/web/20160312115401/https://commons.wikimedia.org/wiki/Category%3ACC-PD-Mark) 12 March 2016 at the Wayback Machine in February 2016
50. "Ideas, Methods, or Systems" (https://www.copyright.gov/circs/circ31.pdf) (PDF). *United States Copyright Office*. Circular 31. January 2012. Archived (https://web.archive.org/web/20240914140801/https://www.copyright.gov/circs/circ31.pdf) (PDF) from the original on 14 September 2024. Retrieved 29 November 2024.
51. Hibbler, Elliott. "Libraries: Copyright and Scholarship: Public Domain" (https://libguides.bc.edu/copyright/publicdomain). *Research Guides - Libraries at Boston College*. Archived (https://web.archive.org/web/20240827231603/https://libguides.bc.edu/copyright/publicdomain) from the original on 27 August 2024. Retrieved 29 November 2024.
52. Karjala, Dennis (2008). "Judicial Oversight of Copyright Legislation". *Northern Kentucky Law Review*. **35**: 253.
53. "Copyright Office Basics" (https://web.archive.org/web/20090225235545/http://www.copyright.gov/circs/circ01.pdf) (PDF). Archived from the original (https://www.copyright.gov/circs/circ01.pdf) (PDF) on 25 February 2009.
54. Nimmer, Melville B., and David Nimmer (1997). *Nimmer on Copyright*, section 13.03(F)(4). Albany: Matthew Bender.
55. *Copyright Law of the United States (Title 17) and Related Laws Contained in Title 17 of the United States Code* (https://www.copyright.gov/title17/). United States of America: U.S. Copyright Office. December 2024. p. 16.
56. "Image copyright standards for GOV.UK – Inside GOV.UK" (https://insidegovuk.blog.gov.uk/gov-uk-standards-and-guidelines/image-copyright-standards-for-gov-uk/). *insidegovuk.blog.gov.uk*. Retrieved 28 April 2025.
57. "Copyright Term and the Public Domain in the United States" (http://copyright.cornell.edu/resources/publicdomain.cfm). *Cornell University Copyright Information Center*. Archived (https://web.archive.org/web/20100926203654/http://www.copyright.cornell.edu/resources/publicdomain.cfm) from the original on 26 September 2010. Retrieved 30 December 2016.
58. "Copyright Notice (https://www.copyright.gov/circs/circ03.pdf)", U.S. Copyright Office Circular 3, 2008. Archived (https://web.archive.org/web/20120926110554/http://www.copyright.gov/circs/circ03.pdf) 26 September 2012 at the Wayback Machine.
59. "About CC0 — "No Rights Reserved" " (https://creativecommons.org/about/cc0). Creative Commons. Archived (https://web.archive.org/web/20130428010038/http://creativecommons.org/about/cc0) from the original on 28 April 2013. Retrieved 23 April 2013.
60. "do What The Fuck you want to Public License Version 1.0" (https://web.archive.org/web/20130602181949/http://anonscm.debian.org/viewvc/pkg-wmaker/trunk/COPYING.WTFPL?revision=2&view=markup). *anonscm - Debian*. March 2000. Archived from the original (http://anonscm.debian.org/viewvc/pkg-wmaker/trunk/COPYING.WTFPL?revision=2&view=markup) on 2 June 2013.
61. Boyle, J. (2008). "A Creative Commons". *The Public Domain: Enclosing the Commons of the Mind*. Yale University Press. pp. 179–204. ISBN 978-0-300-14275-4. JSTOR j.ctt1npvzg.12 (https://www.jstor.org/stable/j.ctt1npvzg.12).
62. "About CC Licenses" (https://creativecommons.org/about/cclicenses/). *Creative Commons*. Retrieved 23 November 2020.
63. "11/17: Lulan Artisans Textile Competition" (https://creativecommons.org/2009/06/17/1117-lulan-artisans-textile-competition/). 17 June 2009. Archived (https://web.archive.org/web/20161231080413/https://creativecommons.org/2009/06/17/1117-lulan-artisans-textile-competition/) from the original on 31 December 2016. Retrieved 30 December 2016.

64. Kreutzer, Till (2011). "Validity of the Creative Commons Zero 1.0 Universal Public Domain Dedication and its usability for bibliographic metadata from the perspective of German Copyright Law" (https://web.archive.org/web/20170525023540/https://www.rd-alliance.org/sites/default/files/cc0-analysis-kreuzer.pdf) (PDF). Archived from the original (https://rd-alliance.org/sites/default/files/cc0-analysis-kreuzer.pdf) (PDF) on 25 May 2017.
65. Labastida, Ignasi; Margoni, Thomas (2020). "Licensing FAIR Data for Reuse". *Data Intelligence*. **2** (1–2): 199–207. doi:10.1162/dint_a_00042 (https://doi.org/10.1162%2Fdint_a_00042). hdl:2445/150542 (https://hdl.handle.net/2445%2F150542). ISSN 2641-435X (https://search.worldcat.org/issn/2641-435X).
66. Brockmeier, Joe (24 February 2016). "The Unlicense: A License for No License" (https://web.archive.org/web/20160324154539/http://ostatic.com/blog/the-unlicense-a-license-for-no-license). *ostatic.com*. Archived from the original (http://ostatic.com/blog/the-unlicense-a-license-for-no-license) on 24 March 2016. Retrieved 29 November 2024.
67. Bendiken, Arto (4 January 2010). "Unlicense Yourself: Set Your Code Free" (https://unlicense.org/#what-is-the-unlicense). *Unlicense.org*.
68. "BSD 0-Clause License (0BSD) Explained in Plain English" (https://www.tldrlegal.com/license/bsd-0-clause-license). *tl;drLegal*. Archived (https://web.archive.org/web/20241118233052/https://www.tldrlegal.com/license/bsd-0-clause-license) from the original on 18 November 2024. Retrieved 29 November 2024.
69. "Conformant Licenses" (https://web.archive.org/web/20160301222106/http://opendefinition.org/licenses/). *opendefinition.com*. Archived from the original (http://opendefinition.org/licenses/) on 1 March 2016.
70. Vollmer, Timothy; Vézina, Brigitte; Miyara, Jocelyn; Benedict, Connor; Wetzler, Jennryn; Hollich, Shanna; Murray, Corrine (27 December 2013). "Creative Commons 4.0 BY and BY-SA licenses approved conformant with the Open Definition" (https://creativecommons.org/2013/12/27/creative-commons-4-0-by-and-by-sa-licenses-approved-conformant-with-the-open-definition/). *Creative Commons*. Archived (https://web.archive.org/web/20160304102501/https://blog.creativecommons.org/2013/12/27/creative-commons-4-0-by-and-by-sa-licenses-approved-conformant-with-the-open-definition/) from the original on 4 March 2016.
71. "Open Data Commons Public Domain Dedication and License (PDDL)" (https://opendatacommons.org/licenses/pddl/). *Open Data Commons: legal tools for open data*.
72. Manual of Patent Examining Procedure available at "MPEP" (http://www.uspto.gov/web/offices/pac/mpep/s2701.html). Archived (https://web.archive.org/web/20150418103411/http://www.uspto.gov/web/offices/pac/mpep/s2701.html) from the original on 18 April 2015. Retrieved 26 April 2015.
73. Officer, Office of the Chief Communications. "Terms of Use for USPTO Websites" (http://www.uspto.gov/main/ccpubguide.htm). Archived (https://web.archive.org/web/20090925103511/http://www.uspto.gov/main/ccpubguide.htm) from the original on 25 September 2009. Retrieved 30 December 2016.
74. "SPAM® Brand and the Internet" (http://arquivo.pt/wayback/20091013192409/http://www.spam.com/about/internet.aspx). Hormel Foods. Archived from the original (http://www.spam.com/about/internet.aspx) on 13 October 2009.
75. McCarthy, Kieren (31 January 2005). "Hormel Spam trademark case canned" (https://www.theregister.co.uk/2005/01/31/spam_ruling/). *The Register*. Archived (https://web.archive.org/web/20080707001848/https://www.theregister.co.uk/2005/01/31/spam_ruling/) from the original on 7 July 2008. Retrieved 2 September 2008.
76. Richmond, Shane (1 January 2010). "Happy Public Domain Day! Here's to many more" (https://web.archive.org/web/20120515214612/http://blogs.telegraph.co.uk/technology/shanerichmond/100004467/happy-public-domain-day-heres-to-many-more/). Telegraph Blogs. Archived from the original (http://blogs.telegraph.co.uk/technology/shanerichmond/100004467/happy-public-domain-day-heres-to-many-more/) on 15 May 2012. Retrieved 24 December 2011.

77. McLean, Wallace J. (1 January 2004). "Happy Public Domain Day!" (https://web.archive.org/web/20170603192213/http://www3.wcl.american.edu/cni/0401/35315.html). American University. Archived from the original (http://www3.wcl.american.edu/cni/0401/35315.html) on 3 June 2017. Retrieved 9 March 2016.
78. Lessig, Lawrence (1 January 2004). "Public domain day - in Canada (Lessig Blog)" (https://web.archive.org/web/20111107083145/http://lessig.org/blog/2004/01/public_domain_day_in_canada.html). Lessig.org. Archived from the original (http://lessig.org/blog/2004/01/public_domain_day_in_canada.html) on 7 November 2011. Retrieved 25 December 2011.
79. "Public Domain Day 2010" (https://web.archive.org/web/20121014091006/http://www.metafilter.com/87938/Public-Domain-Day-2010). *MetaFilter*. Archived from the original (http://www.metafilter.com/87938/Public-Domain-Day-2010) on 14 October 2012.

Works cited

- Huang, Hui (2009). "On public domain in copyright law" (https://link.springer.com/article/10.1007/s11463-009-0011-6). *Frontiers of Law in China*. **4** (2): 178–195. doi:10.1007/s11463-009-0011-6 (https://doi.org/10.1007%2Fs11463-009-0011-6). S2CID 153766621 (https://api.semanticscholar.org/CorpusID:153766621) – via Springer.
- Ronan, Deazley (2006). *Rethinking copyright: history, theory, language* (https://books.google.com/books?id=dMYXq9V1JBQC). Edward Elgar Publishing. p. 102. ISBN 978-1-84542-282-0. Archived (https://web.archive.org/web/20111119042246/https://www.google.com/books?id=dMYXq9V1JBQC&dq=statute%20of%20anne%20copyright&lr=&as_brr=3&source=gbs_navlinks_s) from the original on 19 November 2011 – via Google Books.
- Torremans, Paul, ed. (2007). *Copyright Law: A Handbook of Contemporary Research* (https://books.google.com/books?id=wHJBemWuPT4C). Research handbooks in intellectual property. Cheltenham, UK; Northampton, MA: Edward Elgar Publishing. ISBN 978-1-84844-021-0 – via Google Books.

External links

- Center for the Study of the Public Domain (https://web.law.duke.edu/cspd/), Duke University
- Communia (https://reform.communia-association.org/), internet association on the digital public domain

Wikipedia
The Free Encyclopedia

Digital rights management

Digital rights management (**DRM**) is the management of legal access to digital content. Various tools or **technological protection measures**,[1] such as access control technologies, can restrict the use of proprietary hardware and copyrighted works.[2] DRM technologies govern the use, modification and distribution of copyrighted works (e.g. software, multimedia content) and of systems that enforce these policies within devices.[3] DRM technologies[4] include licensing agreements[5] and encryption.[6]

Laws in many countries criminalize the circumvention of DRM, communication about such circumvention, and the creation and distribution of tools used for such circumvention. Such laws are part of the United States' Digital Millennium Copyright Act (DMCA),[7] and the European Union's Information Society Directive[8] – with the French DADVSI an example of a member state of the European Union implementing that directive.[9]

Copyright holders argue that DRM technologies are necessary to protect intellectual property, just as physical locks prevent personal property from theft.[1] For examples, they can help the copyright holders for maintaining artistic controls,[10] and supporting licenses' modalities such as rentals.[11] Industrial users (i.e. industries) have expanded the use of DRM technologies to various hardware products, such as Keurig's coffeemakers,[12][13] Philips' light bulbs,[14][15] mobile device power chargers,[16][17][18] and John Deere's tractors.[19] For instance, tractor companies try to prevent farmers from making repairs via DRM.[20][21]

DRM is controversial. There is an absence of evidence about the DRM capability in preventing copyright infringement, some complaints by legitimate customers for caused inconveniences, and a suspicion of stifling innovation and competition.[22] Furthermore, works can become permanently inaccessible if the DRM scheme changes or if a required service is discontinued.[23] DRM technologies have been criticized for restricting individuals from copying or using the content legally, such as by fair use or by making backup copies. DRM is in common use by the entertainment industry (*e.g.*, audio and video publishers).[24] Many online stores such as OverDrive use DRM technologies, as do cable and satellite service operators. Apple removed DRM technology from iTunes around 2009.[25] Typical DRM also prevents lending materials out through a library, or accessing works in the public domain.[1]

Introduction

The rise of digital media and analog-to-digital conversion technologies has increased the concerns of copyright-owners, particularly within the music and video industries. While analog media inevitably lose quality with each copy generation and during normal use, digital media files may be duplicated without limit with no degradation. Digital devices make it convenient for consumers to convert (rip) media originally in a physical, analog or broadcast form into a digital form for portability or later use. Combined with the Internet and file-sharing tools, made unauthorized distribution of copyrighted content (digital piracy) much easier.

History

DRM became a major concern with the growth of the Internet in the 1990s, as piracy crushed CD sales and online video became popular. It peaked in the early 2000s as various countries attempted to respond with legislation and regulations and dissipated in the 2010s as social media and streaming services largely replaced piracy and content providers elaborated next-generation business models.

Early efforts

In 1983, the Software Service System (SSS) devised by the Japanese engineer Ryuichi Moriya was the first example of DRM technology. It was subsequently refined under the name superdistribution. The SSS was based on encryption, with specialized hardware that controlled decryption and enabled payments to be sent to the copyright holder. The underlying principle was that the physical distribution of encrypted digital products should be completely unrestricted and that users of those products would be encouraged to do so.[26]

An early DRM protection method for computer and Nintendo Entertainment System games was when the game would pause and prompt the player to look up a certain page in a booklet or manual that came with the game; if the player lacked access to the material, they would not be able to continue.

An early example of a DRM system is the Content Scramble System (CSS) employed by the DVD Forum on DVD movies. CSS uses an encryption algorithm to encrypt content on the DVD disc. Manufacturers of DVD players must license this technology and implement it in their devices so that they can decrypt the content. The CSS license agreement includes restrictions on how the DVD content is played, including what outputs are permitted and how such permitted outputs are made available. This keeps the encryption intact as the content is displayed.

In May 1998, the Digital Millennium Copyright Act (DMCA) passed as an amendment to US copyright law. It had controversial (possibly unintended) implications. Russian programmer Dmitry Sklyarov was arrested for alleged DMCA infringement after a presentation at DEF CON. The DMCA has been cited as chilling to legitimate users;[27] such as security consultants including Niels Ferguson, who declined to publish vulnerabilities he discovered in Intel's secure-computing scheme due to fear of arrest under DMCA; and blind or visually impaired users of screen readers or other assistive technologies.[28]

In 1999, Jon Lech Johansen released DeCSS, which allowed a CSS-encrypted DVD to play on a computer running Linux, at a time when no compliant DVD player for Linux had yet been created. The legality of DeCSS is questionable: one of its authors was sued, and reproduction of the keys themselves is subject to restrictions as illegal numbers.[29]

More modern examples include ADEPT, FairPlay, Advanced Access Content System.

The World Intellectual Property Organization Copyright Treaty (WCT) was passed in 1996. The US Digital Millennium Copyright Act (DMCA), was passed in 1998. The European Union enacted the Information Society Directive. In 2006, the lower house of the French parliament adopted such legislation as part of the controversial DADVSI law, but added that protected DRM techniques should be made interoperable, a move which caused widespread controversy in the United States. The Tribunal de grande instance de Paris concluded in 2006, that the complete blocking of any possibilities of making private copies was an impermissible behaviour under French copyright law.

2000s

The broadcast flag concept was developed by Fox Broadcasting in 2001, and was supported by the MPAA and the U.S. Federal Communications Commission (FCC). A ruling in May 2005 by a United States courts of appeals held that the FCC lacked authority to impose it on the US TV industry. It required that all HDTVs obey a stream specification determining whether a stream can be recorded. This could block instances of fair use, such as time-shifting. It achieved more success elsewhere when it was adopted by the Digital Video Broadcasting Project (DVB), a consortium of about 250 broadcasters, manufacturers, network operators, software developers, and regulatory bodies from about 35 countries involved in attempting to develop new digital TV standards.

In January 2001, the Workshop on Digital Rights Management of the World Wide Web Consortium was held.[30]

On 22 May 2001, the European Union passed the Information Society Directive, with copyright protections.

In 2003, the European Committee for Standardization/Information Society Standardization System (CEN/ISSS) DRM Report was published.[31]

In 2004, the Consultation process of the European Commission, and the DG Internal Market, on the Communication COM(2004)261 by the European Commission on "Management of Copyright and Related Rights" closed.[32]

In 2005, DRM Workshops of Directorate-General for Information Society and Media (European Commission), and the work of the High Level Group on DRM were held.[33]

In 2005, Sony BMG installed DRM software on users' computers without clearly notifying the user or requiring confirmation. Among other things, the software included a rootkit, which created a security vulnerability. When the nature of the software was made public much later, Sony BMG initially minimized the significance of the vulnerabilities, but eventually recalled millions of CDs, and made several attempts to patch the software to remove the rootkit. Class action lawsuits were filed, which were ultimately settled by agreements to provide affected consumers with a cash payout or album downloads free of DRM.[34]

Microsoft's media player Zune released in 2006 did not support content that used Microsoft's PlaysForSure DRM scheme.[35]

Windows Media DRM, reads instructions from media files in a rights management language that states what the user may do with the media.[36] Later versions of Windows Media DRM implemented music subscription services that make downloaded files unplayable after subscriptions are cancelled, along with the ability for a regional lockout.[37] Tools like FairUse4WM strip Windows Media of DRM restrictions.[38]

The Gowers Review of Intellectual Property by the British Government from Andrew Gowers was published in 2006 with recommendations regarding copyright terms, exceptions, orphaned works, and copyright enforcement.

DVB (DVB-CPCM) is an updated variant of the broadcast flag. The technical specification was submitted to European governments in March 2007. As with much DRM, the CPCM system is intended to control use of copyrighted material by the end-user, at the direction of the copyright holder. According to Ren Bucholz of the Electronic Frontier Foundation (EFF), "You won't even know ahead of time whether and how you will be able to record and make use of particular programs or devices".[39] The normative sections were approved for publication by the DVB Steering Board, and formalized by ETSI as a formal European Standard (TS 102 825-X) where X refers to the Part number. Nobody has yet stepped forward to provide a Compliance and Robustness regime for the standard, so it is not presently possible to fully implement a system, as no supplier of device certificates has emerged.

In December 2006, the industrial-grade Advanced Access Content System (AACS) for HD DVD and Blu-ray Discs, a process key was published by hackers, which enabled unrestricted access to AACS-protected content.[40][41]

In January 2007, EMI stopped publishing audio CDs with DRM, stating that "the costs of DRM do not measure up to the results."[42] In March, Musicload.de, one of Europe's largest internet music retailers, announced their position strongly against DRM. In an open letter, Musicload stated that three out of every four calls to their customer support phone service are as a result of consumer frustration with DRM.[43]

Apple Inc. made music DRM-free after April 2007[44] and labeled all music as "DRM-Free" after 2008.[45] Other works sold on iTunes such as apps, audiobooks, movies, and TV shows are protected by DRM.[46]

A notable DRM failure happened in November 2007, when videos purchased from Major League Baseball prior to 2006 became unplayable due to a change to the servers that validate the licenses.[47]

In 2007, the European Parliament supported the EU's direction on copyright protection.

Asus released a soundcard which features a function called "Analog Loopback Transformation" to bypass the restrictions of DRM. This feature allows the user to record DRM-restricted audio via the soundcard's built-in analog I/O connection.[48][49]

Digital distributor GOG.com (formerly Good Old Games) specializes in PC video games and has a strict non-DRM policy.[50]

Baen Books and O'Reilly Media, dropped DRM prior to 2012, when Tor Books, a major publisher of science fiction and fantasy books, first sold DRM-free e-books.[51]

The Axmedis project completed in 2008. It was a European Commission Integrated Project of the FP6, has as its main goal automating content production, copy protection, and distribution, to reduce the related costs, and to support DRM at both B2B and B2C areas, harmonizing them.

The INDICARE project was a dialogue on consumer acceptability of DRM solutions in Europe that completed in 2008.

In mid-2008, the Windows version of *Mass Effect* marked the start of a wave of titles primarily making use of SecuROM for DRM and requiring authentication with a server. The use of the DRM scheme in 2008's *Spore* led to protests, resulting in searches for an unlicensed version. This backlash against the activation limit led *Spore* to become the most pirated game in 2008, topping the top 10 list compiled by TorrentFreak.[52][53] However, *Tweakguides* concluded that DRM does not appear to increase video game piracy, noting that other games on the list, such as *Call of Duty 4* and *Assassin's Creed*, use DRM without limits or online activation. Additionally, other video games that use DRM, such as *BioShock*, *Crysis Warhead*, and *Mass Effect*, do not appear on the list.[54]

Many mainstream publishers continued to rely on online DRM throughout the later half of 2008 and early 2009, including Electronic Arts, Ubisoft, Valve, and Atari, *The Sims 3* being a notable exception in the case of Electronic Arts.[55] Ubisoft broke with the tendency to use online DRM in late 2008, with the release of *Prince of Persia* as an experiment to "see how truthful people really are" regarding the claim that DRM was inciting people to use illegal copies.[56] Although Ubisoft has not commented on the results of the "experiment", Tweakguides noted that two torrents on Mininova had over 23,000 people downloading the game within 24 hours of its release.[57]

In 2009, Amazon remotely deleted purchased copies of George Orwell's *Animal Farm* (1945) and *Nineteen Eighty-Four* (1949) from customers' Amazon Kindles after refunding the purchase price.[58] Commentators described these actions as Orwellian and compared Amazon to Big Brother from *Nineteen Eighty-Four*.[59][60][61][62] Amazon CEO Jeff Bezos then issued a public apology. FSF wrote that this was an example of the excessive power Amazon has to remotely censor content, and called upon Amazon to drop DRM.[63] Amazon then revealed the reason behind its deletion: the e-books in question were unauthorized reproductions of Orwell's works, which were not within the public domain and that the company that published and sold on Amazon's service had no right to do so.[64]

2010 – present

Ubisoft formally announced a return to online authentication on 9 February 2010, through its Uplay online game platform, starting with *Silent Hunter 5*, *The Settlers 7*, and *Assassin's Creed II*.[65] *Silent Hunter 5* was first reported to have been compromised within 24 hours of release,[66] but users of the cracked version soon found out that only early parts of the game were playable.[67] The Uplay system works by having the installed game on the local PCs incomplete and then continuously downloading parts of the game code from Ubisoft's servers as the game progresses.[68] It was more than a month after the PC release in the first week of April that software was released that could bypass Ubisoft's DRM in *Assassin's Creed II*. The software did this by emulating a Ubisoft server for the game. Later that month, a real crack was released that was able to remove the connection requirement altogether.[69][70]

In March 2010, Uplay servers suffered a period of inaccessibility due to a large-scale DDoS attack, causing around 5% of game owners to become locked out of playing their game.[71] The company later credited owners of the affected games with a free download, and there has been no further downtime.[72]

In 2011, comedian Louis C.K. released his concert film *Live at the Beacon Theater* as an inexpensive (US$5), DRM-free download. The only attempt to deter unlicensed copies was a letter emphasizing the lack of corporate involvement and direct relationship between artist and viewer. The film was a commercial success, turning a profit within 12 hours of its release. The artist suggested that piracy rates were lower than normal as a result, making the release an important case study for the digital marketplace.[73][74][75]

In 2012, the EU Court of Justice ruled in favor of reselling copyrighted games.[76]

In 2012, India implemented digital rights management protection.[77][78][79]

In 2012, webcomic *Diesel Sweeties* released a DRM-free PDF e-book.[80][81][82] He followed this with a DRM-free iBook specifically for the iPad[83] that generated more than 10,000 downloads in three days.[84] That led Stevens to launch a Kickstarter project – "ebook stravaganza 3000" – to fund the conversion of 3,000 comics, written over 12 years, into a single "humongous" e-book to be released both for free and through the iBookstore; launched 8 February 2012, with the goal of raising $3,000 in 30 days. The "payment optional" DRM-free model in this case was adopted on Stevens' view that "there is a class of webcomics reader who would prefer to read in large chunks and, even better, would be willing to spend a little money on it."[84]

In February 2012, Double Fine asked for crowdfunding for an upcoming video game, *Double Fine Adventure*, on Kickstarter and offered the game DRM-free for backers. This project exceeded its original goal of $400,000 in 45 days, raising in excess of $2 million.[85] Crowdfunding acted as a pre-order or alternatively as a subscription. After the success of *Double Fine Adventure*, many games were crowd-funded and many offered a DRM-free version.[86][87][88]

Websites – such as library.nu (shut down by court order on 15 February 2012), BookFi, BookFinder, Library Genesis, and Sci-Hub – allowed e-book downloading by violating copyright.[89][90][91][92]

As of 2013, other developers, such as Blizzard Entertainment put most of the game logic is on the "side" or taken care of by the servers of the game maker. Blizzard uses this strategy for its game *Diablo III* and Electronic Arts used this same strategy with their reboot of *SimCity*, the necessity of which has been questioned.[93]

In 2014, the EU Court of Justice ruled that circumventing DRM on game devices was legal under some circumstances.[94][95]

In 2014, digital comic distributor Comixology allowed rights holders to provide the option of DRM-free downloads. Publishers that allow this include Dynamite Entertainment, Image Comics, Thrillbent, Top Shelf Productions, and Zenescope Entertainment.[96]

In February 2022, Comixology, which was later under the ownership of Amazon, ended the option of downloading DRM-free downloads on all comics, although any comics previously purchased prior to the date will have the option to download comics without DRM.[97][98]

Technologies

Verification

Product keys

A product key, typically an alphanumerical string, can represent a license to a particular copy of software. During the installation process or software launch, the user is asked to enter the key; if the key is valid (typically via internal algorithms), the key is accepted, and the user can continue. Product keys can be combined with other DRM practices (such as online "activation"), to prevent cracking the software to run without a product key, or using a keygen to generate acceptable keys.

Activation limits

DRM can limit the number of devices on which a legal user can install content.[99] This restriction typically support 3-5 devices. This affects users who have more devices than the limit. Some allow one device to be replaced with another. Without this software and hardware upgrades may require an additional purchase.

Persistent online DRM

Always-on DRM checks and rechecks authorization while the content is in use by interacting with a server operated by the copyright holder. In some cases, only part of the content is actually installed, while the rest is downloaded dynamically during use.

Encryption

Encryption alters content in a way that means that it cannot be used without first decrypting it.[99] Encryption can ensure that other restriction measures cannot be bypassed by modifying software, so DRM systems typically rely on encryption in addition to other techniques.

Copy restriction

Error message on a Nokia 6810 warning that a file is "copyright protected"

Microsoft PlayReady prevents illicit copying of multimedia and other files.[100]

Restrictions can be applied to electronic books and documents, in order to prevent copying, printing, forwarding, and creating backup copies. This is common for both e-publishers and enterprise Information Rights Management. It typically integrates with content management system software.[101]

While some commentators claim that DRM complicates e-book publishing,[102] it has been used by organizations such as the British Library in its secure electronic delivery service to permit worldwide access to rare documents which, for legal reasons, were previously only available to authorized individuals actually visiting the Library's document centre.[103][104][105]

Four main e-book DRM schemes are in common use, from Adobe, Amazon, Apple, and the Marlin Trust Management Organization (MTMO).

- Adobe's DRM is applied to EPUBs and PDFs, and can be read by several third-party e-book readers, as well as Adobe Digital Editions (ADE) software. Barnes & Noble uses DRM technology provided by Adobe, applied to EPUBs and the older PDB (Palm OS) format e-books.
- Amazon's DRM is an adaption of the original Mobipocket encryption and is applied to Amazon's `.azw4`, KF8, and Mobipocket format e-books. Topaz format e-books have their own encryption system.[106]
- Apple's FairPlay DRM is applied to EPUBs and can be read only by Apple's iBooks app on iOS devices and Mac OS computers.
- The Marlin DRM was developed and is maintained by open industry group Marlin Developer Community (MDC) and is licensed by MTMO. (Marlin was founded by Intertrust, Panasonic, Philips, Samsung, and Sony.) Online textbook publisher Kno uses Marlin to protect EPUB books. These books can be read on the Kno App for iOS and Android.

Runtime restrictions

Windows Vista contains a DRM system called Protected Media Path, which contains Protected Video Path (PVP).[107] PVP tries to stop DRM-restricted content from playing while unsigned software is running, in order to prevent the unsigned software from accessing the content. Additionally, PVP can encrypt information during transmission to the monitor or the graphics card, which makes it more difficult to make unauthorized recordings.

Bohemia Interactive have used a form of technology since *Operation Flashpoint: Cold War Crisis*, wherein if the game copy is suspected of being unauthorized, annoyances like guns losing their accuracy or the players turning into a bird are introduced.[108] Croteam's *Serious Sam 3: BFE* causes a special invincible foe in the game to appear and constantly attack the player until they are killed.[109][110]

Regional lockout

Regional lockout (or region coding) prevents the use of a certain product or service, except in a specific region or territory. Lockout may be enforced through physical means, through technological means such as inspecting the user's IP address or using an identifying code, or through unintentional means introduced by devices that support only region-specific technologies (such as video formats, i.e., NTSC and PAL).

Tracking

Watermarks

Digital watermarks can be steganographically embedded within audio or video data. They can be used for recording the copyright owner, the distribution chain or identifying the purchaser. They are not complete DRM mechanisms in their own right, but are used as part of a system for copyright enforcement, such as helping provide evidence for legal purposes, rather than enforcing restrictions.[111]

Some audio/video editing programs may distort, delete, or otherwise interfere with watermarks. Signal/modulator-carrier chromatography may separate watermarks from the recording or detect them as glitches. Additionally, comparison of two separately obtained copies of audio using basic algorithms can reveal watermarks.

Metadata

Sometimes, metadata is included in purchased media which records information such as the purchaser's name, account information, or email address. Also included may be the file's publisher, author, creation date, download date, and various notes. This information is not embedded in the content, as a watermark is. It is kept separate from the content, but within the file or stream.

As an example, metadata is used in media purchased from iTunes for DRM-free as well as DRM-restricted content. This information is included as MPEG standard metadata.[112][113]

Hardware

US Cable television set-top boxes require a specific piece of hardware to operate. The CableCard standard is used to restrict content to services to which the customer is subscribed. Content has an embedded broadcast flag that the card examines to decide whether the content can be viewed by a specific user.

Implementations

- Analog Protection System (Macrovision)
- DCS Copy Protection
- B-CAS
- CableCARD
- Broadcast flag
- DVB-CPCM
- Conditional-access module
- Copy Control Information
- ISDB#Copy-protection technology
- FairPlay
- Extended Copy Protection (XCP)
- Content Scramble System (CSS)
- ARccOS protection
- Advanced Access Content System (AACS)
- Content Protection for Recordable Media (CPRM)
- Digital Transmission Content Protection
- High-bandwidth Digital Content Protection (HDCP)
- Protected Media Path
- Trusted Platform Module#Uses
- Intel Management Engine#Design
- Cinavia
- HTML video Encrypted Media Extensions (HTML EME, often implemented with Widevine)
- Denuvo
- StarForce
- SafeDisc
- SecuROM
- SafetyNet
- Google Play Integrity

In addition, platforms such as Steam may include DRM mechanisms. Most of the mechanisms above are copy protection mechanisms rather than DRM mechanisms per se.

Laws

The World Intellectual Property Organization supports the World Intellectual Property Organization Copyright Treaty (WCT) which requires nations to enact laws against DRM circumvention. The WIPO Internet Treaties do not mandate criminal sanctions, merely requiring "effective legal remedies".[114]

Australia

Australia prohibits circumvention of "access control technical protection measures" in Section 116 of the Copyright Act. The law currently imposes penalties for circumvention of such measures[115] as well as the manufacturing[116] and distribution[117] of tools to enable it.

DRM may be legally circumvented under a few distinct circumstances which are named as exceptions in the law:

1. permission of the rightsholder
2. enabling interoperability with copyrighted software
3. encryption research
4. security testing
5. disabling access to private information (circumvention only)
6. national security or law enforcement
7. library acquisition decisions (circumvention only)
8. acts prescribed by regulation (circumvention only)

A person circumventing the access control bears the burden of proof that one of these exceptions apply.

Penalties for violation of the anti-circumvention laws include an injunction, monetary damages, and destruction of enabling devices.[118]

China

China's copyright law was revised in 2001 (https://en.wikisource.org/wiki/Copyright_Law_of_the_People%27s_Republic_of_China_(2001)) and included a prohibition on "intentionally circumventing or destroying the technological measures taken by a right holder for protecting the copyright or copyright-related rights in his work, sound recording or video recording, without the permission of the copyright owner, or the owner of the copyright-related rights". However, the Chinese government had faced backlash from Nintendo over the heavy burden on law enforcement action against circumvention devices, stating that the police only view game copiers as infringing Nintendo's trademark, not as infringing copyright. In response, Nintendo obtained copyright registration for its software in 2013 to make it easier to make law enforcement against game copiers and other circumvention devices.[119]

European Union

The EU operates under its Information Society Directive, its WIPO implementation. The European Parliament then directed member states to outlaw violation of international copyright for commercial purposes. Punishments range from fines to imprisonment. It excluded patent rights and copying for personal, non-commercial purposes. Copyrighted games can be resold.[76] Circumventing DRM on game devices is legal under some circumstances; protections cover only technological measures the interfere with prohibited actions.[94][95]

India

India acceded to the WIPO Copyright Treaty and the WIPO Performances and Phonograms Treaty on July 4, 2018,[120] after a 2012 amendment to the Copyright Act criminalized the circumvention of technical protections. Fair use is not explicitly addressed, but the anti-circumvention provisions do not prohibit circumventing for non-infringing purposes.[77][78][79]

Israel

Israel is not a signatory to the WIPO Copyright Treaty. Israeli law does not expressly prohibit the circumvention of technological protection measures.[121]

Japan

Japan outlawed circumvention of technological protection measures on June 23, 1999 through an amendment of its 1970 copyright law.[122] The private copying exception does not apply if it has become available due to circumvention of TPMs,[123] and circumvention of a TPM is deemed as copyright infringement. However, circumvention is allowed for research purposes or if it otherwise does not harm the rightsholder's interests.[124]

Pakistan

Pakistan is not a signatory to the WIPO Copyright Treaty or the WIPO Performances and Phonograms Treaty. Pakistani law does not criminalize the circumvention of technological protection measures.[125]

As of January 2022, Pakistan's Intellectual Property Office intended to accede to the WIPO Copyright Treaty and WIPO Performances and Phonograms Treaty. However, there has been no major progress for Pakistan to accede to the treaties,[126] and the timeline of the enactments of amendments to the Copyright Ordinance is unclear.[127] As of February 2023, Pakistan's Intellectual Property Office was currently finalizing draft amendments to its Copyright Ordinance.[128]

United States

US protections are governed by the Digital Millennium Copyright Act (DMCA). It criminalizes the production and dissemination of technology that lets users circumvent copy-restrictions. Reverse engineering is expressly permitted, providing a safe harbor where circumvention is necessary to interoperate with other software.

Open-source software that decrypts protected content is not prohibited per se. Decryption done for the purpose of achieving interoperability of open source operating systems with proprietary systems is protected. Dissemination of such software for the purpose of violating or encouraging others to violate copyrights is prohibited.

DMCA has been largely ineffective.[129] Cirumvention software is widely available. However, those who wish to preserve the DRM systems have attempted to use the Act to restrict the distribution and development of such software, as in the case of DeCSS. DMCA contains an exception for research, although the exception is subject to qualifiers that created uncertainty in that community.

Cryptanalytic research may violate the DMCA, although this is unresolved.

Notable lawsuits

- *DVD Copy Control Association, Inc. v. Bunner*
- *DVD Copy Control Association, Inc. v. Kaleidescape, Inc.*
- *RealNetworks, Inc. v. DVD Copy Control Association, Inc.*
- *Universal v. Reimerdes*

Opposition

DRM faces widespread opposition. John Walker[130] and Richard Stallman are notable critics.[131][132] Stallman has claimed that using the word "rights" is misleading and suggests that the word "restrictions", as in "Digital Restrictions Management", replace it.[133] This terminology has been adopted by other writers and critics.[134][135][136]

Other prominent critics include Ross Anderson, who headed a British organization that opposes DRM and similar efforts in the UK and elsewhere, and Cory Doctorow.[137] EFF and organizations such as FreeCulture.org are opposed to DRM.[22] The Foundation for a Free Information Infrastructure criticized DRM's effect as a trade barrier from a free market perspective.[138]

Bruce Schneier argues that digital copy prevention is futile: "What the entertainment industry is trying to do is to use technology to contradict that natural law. They want a practical way to make copying hard enough to save their existing business. But they are doomed to fail."[139] He described trying to make digital files uncopyable as like "trying to make water not wet".[140]

The creators of *StarForce* stated that "The purpose of copy protection is not making the game uncrackable – it is impossible."[141]

Bill Gates spoke about DRM at 2006 CES, saying that DRM causes problems for legitimate consumers.[142]

The Norwegian consumer rights organization "Forbrukerrådet" complained to Apple in 2007 about the company's use of DRM, accusing it of unlawfully restricting users' access to their music and videos, and of using EULAs that conflict with Norwegian consumer legislation. The complaint was supported by consumers' ombudsmen in Sweden and Denmark, and was reviewed in the EU in 2014. The United States Federal Trade Commission held hearings in March 2009, to review disclosure of DRM limitations to customers' use of media products.[143]

Valve president Gabe Newell stated, "most DRM strategies are just dumb" because they only decrease the value of a game in the consumer's eyes. Newell suggested that the goal should instead be "[creating] greater value for customers through service value". Valve operates Steam, an online store for PC games, as well as a social networking service and a DRM platform.[144]

Defective by Design member protesting DRM on 25 May 2007

At the 2012 Game Developers Conference, the CEO of CD Projekt Red, Marcin Iwinski, announced that the company would not use DRM. Iwinski stated of DRM, "It's just over-complicating things... the game... is cracked in two hours." Iwinski added "DRM does not protect your game. If there are examples that it does, then people maybe should consider it, but then there are complications with legit users."[145]

The Association for Computing Machinery and the Institute of Electrical and Electronics Engineers opposed DRM, naming AACS as a technology "most likely to fail" in an issue of *IEEE Spectrum*.[146]

Public licenses

The GNU General Public License version 3, as released by the Free Software Foundation, has a provision that "strips" DRM of its legal value, so people can break the DRM on GPL software without breaking laws such as the DMCA. In May 2006, FSF launched a "Defective by Design" campaign against DRM.[147][148]

Creative Commons provides licensing options that encourage creators to work without the use of DRM.[149] Creative Commons licenses have anti-DRM clauses, making the use of DRM by a licensee a breach of the licenses' Baseline Rights.[150]

DRM-free works

Many publishers and artists label their works "DRM-free". Major companies that have done so include Apple, GOG.com, Tor Books and Vimeo on Demand. Comixology once had DRM-free works available for sale until 2022 when its parent company, Amazon, removed the option to buy DRM-free works as part of their migration to Amazon's website, although previous purchases remained DRM-free.[151]

Label proposed by the Free Software Foundation for DRM-free works

Shortcomings

Availability

Many DRM systems require online authentication. Whenever the server goes down, or a territory experiences an Internet outage, it locks out people from registering or using the material.[152] This is especially true for products that require a persistent online connection, where, for example, a successful DDoS attack on the server essentially makes the material unusable.

Usability

Compact discs (CDs) with DRM schemes are not standards-compliant, and are labeled CD-ROMs. CD-ROMs cannot be played on all CD players or personal computers.[153]

Performance

Certain DRM systems have been associated with reduced performance: some games implementing Denuvo Anti-Tamper performed better without DRM.[154][155] However, in March 2018, *PC Gamer* tested *Final Fantasy XV* for the performance effects of Denuvo, which was found to cause no negative gameplay impact despite a little increase in loading time.[156]

Robustness

DRM copy-prevention schemes can never be wholly secure since the logic needed to decrypt the content is present either in software or hardware and implicitly can be hacked. An attacker can extract this information, decrypt and copy the content, bypassing the DRM.[137]

Satellite and cable systems distribute their content widely and rely on hardware DRM systems. Such systems can be hacked by reverse engineering the protection scheme.

Analog hole

Audio and visual material (excluding interactive materials, *e.g.*, video games) are subject to the analog hole, namely that in order to view the material, the digital signal must be turned into an analog signal. Post-conversion, the material can be then be copied and reconverted to a digital format.

The analog hole cannot be filled without externally imposed restrictions, such as legal regulations, because the vulnerability is inherent to all analog presentation.[157] The conversion from digital to analog and back reduces recording quality. The HDCP attempt to plug the analog hole was largely ineffective.[158][159]

Consumer rights

Ownership restrictions

DRM opponents argue that it violates private property rights and restricts a range of normal and legal user activities. A DRM component such as that found on a digital audio player restricts how it acts with regard to certain content, overriding user's wishes (for example, preventing the user from copying a copyrighted song to CD as part of a compilation). Doctorow described this as "the right to make up your own copyright laws".[160]

Windows Vista disabled or degraded content play that used a Protected Media Path.[161] DRM restricts the right to make personal copies, provisions lend copies to friends, provisions for service discontinuance, hardware agnosticism, software and operating system agnosticism,[162] lending library use, customer protections against contract amendments by the publisher, and whether content can pass to the owner's heirs.[163]

Obsolescence

When standards and formats change, DRM-restricted content may become obsolete.

When a company undergoes business changes or bankruptcy, its previous services may become unavailable. Examples include MSN Music,[164] Yahoo! Music Store,[165] Adobe Content Server 3 for Adobe PDF,[166] and Acetrax Video on Demand.[167]

Piracy

DRM laws are widely flouted: according to Australia Official Music Chart Survey, copyright infringements from all causes are practised by millions of people.[168] According to the EFF, "in an effort to attract customers, these music services try to obscure the restrictions they impose on you with clever marketing."[169]

Economic implication

Trade-offs between control and sales

Jeff Raikes, ex-president of the Microsoft Business Division, stated: "If they're going to pirate somebody, we want it to be us rather than somebody else".[170] An analogous argument was made in an early paper by Kathleen Conner and Richard Rummelt.[171] A subsequent study of digital rights management for e-books by Gal Oestreicher-Singer and Arun Sundararajan showed that relaxing some forms of DRM can be beneficial to rights holders because the losses from piracy are outweighed by the increase in value to legal buyers. Even if DRM were unbreakable, pirates still might not be willing to purchase, so sales might not increase.[172]

Piracy can be beneficial to some content providers by increase consumer awareness, spreading and popularizing content. This can also increase revenues via other media, such as live performances.

Mathematical models suggest that DRM schemes can fail to do their job on multiple levels.[173] The biggest failure is that the burden that DRM poses on a legitimate customer reduces the customer's willingness to buy. An ideal DRM would not inconvenience legal buyers. The mathematical models are strictly applicable to the music industry.

Alternatives

Several business models offer DRM alternatives.[174]

Subscription

Streaming services have created profitable business models by signing users to monthly subscriptions in return for access to the service's library. This model has worked for music (such as Spotify, Apple Music, etc.) and video (such as Netflix, Disney+, Hulu, etc.).

"Easy and cheap"

Accessing a pirated copy can be illegal and inconvenient. Businesses that charge acceptable fees for doing so tend to attract customers. A business model that dissuades illegal file sharing is to make legal content downloading easy and cheap. Pirate websites often host malware which attaches itself to the files served.[175] If content is provided on legitimate sites and is reasonably priced, consumers are more likely to purchase media legally.[174]

Crowdfunding or pre-order

Crowdfunding has been used as a publishing model for digital content.[85]

Promotion for traditional products

Many artists give away individual tracks to create awareness for a subsequent album.[174]

Artistic Freedom Voucher

The Artistic Freedom Voucher (AFV) introduced by Dean Baker is a way for consumers to support "creative and artistic work". In this system, each consumer receives a refundable tax credit of $100 to give to any artist of creative work. To restrict fraud, the artists must register with the government. The voucher prohibits any artist that receives the benefits from copyrighting their material for a certain length of time. Consumers would be allowed to obtain music for a certain amount of time easily and the consumer would decide which artists receive the $100. The money can either be given to one artist or to many, and this distribution is up to the consumer.[176]

See also

- Anti-tamper software
- Closed platform
- Digital asset management
- License manager
- ODRL
- Right to repair
- Software metering
- Software protection dongle
- Secure Digital Music Initiative
- Trusted Computing
- Web Environment Integrity

References

1. "The pros, cons, and future of DRM" (https://www.cbc.ca/news/science/the-pros-cons-and-future-of-drm-1.785237). Cbc.ca. 7 August 2009. Archived (https://web.archive.org/web/20130819222032/http://www.cbc.ca/news/technology/story/2009/08/06/tech-digital-locks-drm-tpm-rights-management-protection-measures-copyright-copy-protection.html) from the original

on 19 August 2013. Retrieved 7 January 2012. "Digital locks – also known as digital rights management (DRM) technologies or technological protection measures (TPM)"

2. *Computer Forensics: Investigating Network Intrusions and Cybercrime* (https://books.google.com/books?id=0rVfRwcIPYgC&q=DRM+%22access+control+technology%22&pg=SA9-PA26). Cengage Learning. 16 September 2009. pp. 9–26. ISBN 978-1435483521.
3. "Fact Sheet: Digital Rights Management and have to do: Technical Protection Measures" (https://web.archive.org/web/20160414002554/http://www.priv.gc.ca/resource/fs-fi/02_05_d_32_e.asp). *Priv.gc.ca*. 24 November 2006. Archived from the original (https://www.priv.gc.ca/resource/fs-fi/02_05_d_32_e.asp) on 14 April 2016. Retrieved 29 July 2013.
4. "Digital rights management" (https://www.thomsonreuters.com/en-gb/help/highq/files-module/files-admin/drm.html). *www.thomsonreuters.com*. Retrieved 7 December 2024.
5. "Digital Rights Management Systems and Copy Protection Schemes" (https://web.archive.org/web/20110209160908/http://w2.eff.org/IP/DRM/). eff.org. Archived from the original (https://w2.eff.org/IP/DRM/) on 9 February 2011. Retrieved 13 February 2011.
6. Kranich, Nancy. "Chap 1(pg.8)." The Information Commons. Creative Commons, 2004. Print.
7. "Public Law 105 – 304 – Digital Millennium Copyright Act" (https://www.gpo.gov/fdsys/pkg/PLAW-105publ304/html/PLAW-105publ304.htm). *U. S. Government Publishing Office*. Archived (https://web.archive.org/web/20180716140639/https://www.gpo.gov/fdsys/pkg/PLAW-105publ304/html/PLAW-105publ304.htm) from the original on 16 July 2018. Retrieved 26 July 2015.
8. "Directive 2001/29/EC of the European Parliament and of the Council of 22 May 2001 on the harmonisation of certain aspects of copyright and related rights in the information society" (http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32001L0029&qid=1437963741051). *Official Journal of the European Union*. 22 June 2001. Archived (https://web.archive.org/web/20151222085135/http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32001L0029&qid=1437963741051) from the original on 22 December 2015. Retrieved 26 July 2015.
9. "LOI n° 2006-961 du 1er août 2006 relative au droit d'auteur et aux droits voisins dans la société de l'information" (http://legifrance.gouv.fr/affichTexte.do?cidTexte=JORFTEXT000000266350&dateTexte=&categorieLien=id). *Journal officiel de la République française* (in French). 3 August 2006. Archived (https://web.archive.org/web/20150529175139/http://www.legifrance.gouv.fr/affichTexte.do?cidTexte=JORFTEXT000000266350&dateTexte=&categorieLien=id) from the original on 29 May 2015. Retrieved 26 July 2015.
10. "Images and the Internet" (http://www.artistscope.com/protection.asp). Archived (https://web.archive.org/web/20180710121521/http://www.artistscope.com/protection.asp) from the original on 10 July 2018. Retrieved 16 February 2009.
11. Christopher Levy (3 February 2003). "Making Money with Streaming Media" (https://web.archive.org/web/20060514024614/http://www.streamingmedia.com/r/printerfriendly.asp?id=8306). streamingmedia.com. Archived from the original (http://www.streamingmedia.com/r/printerfriendly.asp?id=8306) on 14 May 2006. Retrieved 28 August 2006.
12. Bode, Karl (3 March 2014). "Keurig Will Use DRM in New Coffee Maker To Lock Out Refill Market" (https://www.techdirt.com/articles/20140227/06521826371/keurig-will-use-drm-new-coffee-maker-to-lock-out-refill-market.shtml). techdirt.com. Archived (https://web.archive.org/web/20150503063944/https://www.techdirt.com/articles/20140227/06521826371/keurig-will-use-drm-new-coffee-maker-to-lock-out-refill-market.shtml) from the original on 3 May 2015. Retrieved 3 May 2015.
13. Chris Welch (28 August 2014). "Keurig's coffee brewer 'DRM' has already been defeated" (https://www.theverge.com/2014/8/28/6079711/keurig-coffee-pod-drm-has-been-defeated). Archived (https://web.archive.org/web/20170708011845/https://www.theverge.com/2014/8/28/6079711/keurig-coffee-pod-drm-has-been-defeated) from the original on 8 July 2017. Retrieved 30 August 2017.

14. Philips pushes lightbulb firmware update that locks out third-party bulbs (https://boingboing.net/2015/12/14/philips-pushes-lightbulb-firmw.html) Archived (https://web.archive.org/web/20151216231732/http://boingboing.net/2015/12/14/philips-pushes-lightbulb-firmw.html) 16 December 2015 at the Wayback Machine by Cory Doctorow on Boing Boing (14 December 2015)
15. Light Bulb DRM: Philips Locks Purchasers Out Of Third-Party Bulbs With Firmware Update (https://www.techdirt.com/articles/20151214/07452133070/lightbulb-drm-philips-locks-purchasers-out-third-party-bulbs-with-firmware-update.shtml) Archived (https://web.archive.org/web/20151217130712/https://www.techdirt.com/articles/20151214/07452133070/lightbulb-drm-philips-locks-purchasers-out-third-party-bulbs-with-firmware-update.shtml) 17 December 2015 at the Wayback Machine on techdirt.com (14 December 2015)
16. Another Thing You Need: 'DRM For Chargers' (https://www.techdirt.com/articles/20070724/094533.shtml) Archived (https://web.archive.org/web/20160402044236/https://www.techdirt.com/articles/20070724/094533.shtml) 2 April 2016 at the Wayback Machine on techdirt.com by Carlo Longino (25 July 2007)
17. Ed Felten (26 July 2007). "DRM for Chargers: Possibly Good for Users" (https://freedom-to-tinker.com/blog/felten/drm-chargers-possibly-good-users/). freedom-to-tinker.com. Archived (https://web.archive.org/web/20160401220532/https://freedom-to-tinker.com/blog/felten/drm-chargers-possibly-good-users/) from the original on 1 April 2016. Retrieved 20 March 2016. "*Apple has filed a patent application on a technology for tethering rechargeable devices (like iPods) to particular chargers. The idea is that the device will only allow its batteries to be recharged if it is connected to an authorized charger. Whether this is good for consumers depends on how a device comes to be authorized. If "authorized" just means "sold or licensed by Apple" then consumers won't benefit – the only effect will be to give Apple control of the aftermarket for replacement chargers.*"
18. hacking-dell-laptop-charger-identification (http://hackaday.com/2014/03/03/hacking-dell-laptop-charger-identification/) Archived (https://web.archive.org/web/20210504142145/https://hackaday.com/2014/03/03/hacking-dell-laptop-charger-identification/) 4 May 2021 at the Wayback Machine on hackaday.com (3 March 2014)
19. Wiens, Kyle (21 April 2015). "We Can't Let John Deere Destroy the Very Idea of Ownership" (https://www.wired.com/2015/04/dmca-ownership-john-deere/). *Wired*. wired.com. Archived (https://web.archive.org/web/20210127060843/https://www.wired.com/2015/04/dmca-ownership-john-deere/) from the original on 27 January 2021. Retrieved 3 May 2015.
20. Sydell, Laura (17 August 2015). "DIY Tractor Repair Runs Afoul of Copyright Law" (https://www.npr.org/sections/alltechconsidered/2015/08/17/432601480/diy-tractor-repair-runs-afoul-of-copyright-law). npr.com. Archived (https://web.archive.org/web/20210114214322/https://www.npr.org/sections/alltechconsidered/2015/08/17/432601480/diy-tractor-repair-runs-afoul-of-copyright-law) from the original on 14 January 2021. Retrieved 31 August 2015.
21. Newman, Lily Hay. "A New Jailbreak for John Deere Tractors Rides the Right-to-Repair Wave" (https://www.wired.com/story/john-deere-tractor-jailbreak-defcon-2022/). *Wired*. ISSN 1059-1028 (https://search.worldcat.org/issn/1059-1028). Retrieved 12 August 2024.
22. "DRM" (https://www.eff.org/issues/drm). Electronic Frontier Foundation. Archived (https://web.archive.org/web/20180705015859/https://www.eff.org/issues/drm) from the original on 5 July 2018. Retrieved 7 January 2012.
23. "The DRM graveyard: A brief history of digital rights management in music" (http://opensource.com/life/11/11/drm-graveyard-brief-history-digital-rights-management-music). opensource.com. 3 November 2011. Archived (https://web.archive.org/web/20120606125646/http://opensource.com/life/11/11/drm-graveyard-brief-history-digital-rights-management-music) from the original on 6 June 2012. Retrieved 7 January 2012.

24. "QuickPlay Distributes TV Over Mobile Wi-Fi" (http://www.mediapost.com/publications/article/117244/quickplay-distributes-tv-over-mobile-wi-fi.html). MediaDaily News. 11 November 2009. Archived (https://web.archive.org/web/20150109012043/http://www.mediapost.com/publications/article/117244/quickplay-distributes-tv-over-mobile-wi-fi.html) from the original on 9 January 2015. Retrieved 28 December 2014.
25. Bobbie Johnson, San Francisco (6 January 2009). "Apple drops copy protection from iTunes" (https://www.theguardian.com/technology/2009/jan/06/apple-drops-itunes-copy-protection). *Guardian*. Archived (https://web.archive.org/web/20140204034137/http://www.theguardian.com/technology/2009/jan/06/apple-drops-itunes-copy-protection) from the original on 4 February 2014. Retrieved 6 June 2012.
26. Patent application 58-186100 (http://v3.espacenet.com/publicationDetails/biblio?DB=EPODOC&adjacent=true&locale=en_EP&FT=D&date=19850501&CC=JP&NR=60077218A&KC=A) Archived (https://web.archive.org/web/20220216045220/https://worldwide.espacenet.com/publicationDetails/biblio?DB=EPODOC&adjacent=true&locale=en_EP&FT=D&date=19850501&CC=JP&NR=60077218A&KC=A) 16 February 2022 at the Wayback Machine (Publication #60-077218), *Software Control System*, Japan Patent Office, 5 October 1983, Ryoichi Mori, applicant. Reported by Industrial Property Digital Library.
27. "RIAA challenges SDMI attack" (https://www.usenix.org/legacy/publications/library/proceedings/sec01/craver.pdf) (PDF). 7 January 2002. Archived (https://web.archive.org/web/20201031040253/https://www.usenix.org/legacy/publications/library/proceedings/sec01/craver.pdf) (PDF) from the original on 31 October 2020. Retrieved 13 January 2016.
28. "Joint Comments of the American Council of the Blind and the American Foundation for the Blind, DOCKET NO. RM 2011-7" (https://www.copyright.gov/1201/2011/initial/american_foundation_blind.pdf) (PDF). American Council of the Blind and American Foundation for the Blind. 2011. Archived (https://web.archive.org/web/20150226130028/http://www.copyright.gov/1201/2011/initial/american_foundation_blind.pdf) (PDF) from the original on 26 February 2015. Retrieved 22 December 2014.
29. "Memorandum Order, in MPAA v. Reimerdes, Corley and Kazan" (https://web.archive.org/web/20161213144005/https://w2.eff.org/IP/Video/MPAA_DVD_cases/?f=20000202_ny_memorandum_order.html). *Electronic Frontier Foundation*. New York. 2 February 2000. Archived from the original (https://w2.eff.org/IP/Video/MPAA_DVD_cases/?f=20000202_ny_memorandum_order.html) on 13 December 2016. Retrieved 27 July 2015.
30. "W3C Workshop – Digital Rights Management for the Web" (http://www.w3.org/2000/12/drm-ws/Overview.html). W3.org. Archived (https://web.archive.org/web/20100211145937/http://www.w3.org/2000/12/drm-ws/Overview.html) from the original on 11 February 2010. Retrieved 31 August 2010.
31. "Digital Rights Management" (https://web.archive.org/web/20061005073900/http://www.cenorm.be/cenorm/businessdomains/businessdomains/isss/activity/drm_fg.asp). European Committee for Standardization (CEN). Archived from the original (http://www.cenorm.be/cenorm/businessdomains/businessdomains/isss/activity/drm_fg.asp) on 5 October 2006.
32. "The address you requested is obsolete" (https://web.archive.org/web/20060422013330/http://europa.eu.int/comm/internal_market/copyright/management/management_en.htm). Europa.eu.int. 23 February 2009. Archived from the original (http://europa.eu.int/comm/internal_market/copyright/management/management_en.htm) on 22 April 2006. Retrieved 31 August 2010.
33. "eEurope 2005 Action Plan" (https://web.archive.org/web/20060520194647/http://ec.europa.eu/information_society/eeurope/2005/all_about/action_plan/index_en.htm). *European Commission – Information Society – eEurope 2005*. 2005. Archived from the original (http://ec.europa.eu/information_society/eeurope/2005/all_about/action_plan/index_en.htm) on 20 May 2006. Retrieved 28 June 2018.

34. McMillan, Robert (23 May 2006). "Settlement Ends Sony Rootkit Case" (https://web.archive.org/web/20070930190832/http://www.pcworld.com/article/id,125838-page,1-c,unresolvedtechstandards/article.html). *PC World*. Archived from the original (http://www.pcworld.com/article/id,125838-page,1-c,unresolvedtechstandards/article.html) on 30 September 2007. Retrieved 8 April 2007.
35. "Zune Won't Play Old DRM Infected Files" (http://yro.slashdot.org/article.pl?sid=06/09/19/1342256). slashdot.org. 19 September 2006. Archived (https://web.archive.org/web/20220216045215/https://slashdot.org/story/06/09/19/1342256/zune-wont-play-old-drm-infected-files) from the original on 16 February 2022. Retrieved 19 September 2007.
36. Ross J. Anderson (2001). *Security Engineering* (https://archive.org/details/securityengineer00ande/page/705). Wiley. p. 705 (https://archive.org/details/securityengineer00ande/page/705). ISBN 0-471-38922-6.
37. "22: Copyright and DRM" (http://www.cl.cam.ac.uk/~rja14/book.html). *Security Engineering*. WILEY. Archived (https://web.archive.org/web/20110226114349/http://www.cl.cam.ac.uk/~rja14/book.html) from the original on 26 February 2011. Retrieved 12 May 2013.
38. "Engadget FairUse4WM strips Windows Media DRM!" (https://www.engadget.com/2006/08/25/fairuse4wm-strips-windows-media-drm). 25 August 2006. Archived (https://web.archive.org/web/20060831134248/http://www.engadget.com/2006/08/25/fairuse4wm-strips-windows-media-drm/) from the original on 31 August 2006. Retrieved 25 August 2006.
39. "Who Controls Your Television?" (https://web.archive.org/web/20080129123215/http://w2.eff.org/IP/DVB/dvb_briefing_paper.php). Electronic Frontier Foundation. Archived from the original (https://w2.eff.org/IP/DVB/dvb_briefing_paper.php) on 29 January 2008. Retrieved 1 January 2008.
40. Xeni Jardin (28 December 2006). "Report: HD-DVD copy protection defeated" (https://www.boingboing.net/2006/12/28/report-hddvd-copy-pr.html). *BoingBoing*. Archived (https://web.archive.org/web/20071225075834/http://www.boingboing.net/2006/12/28/report-hddvd-copy-pr.html) from the original on 25 December 2007. Retrieved 1 January 2008.
41. Cory Doctorow (30 May 2007). "New AACS processing key leaks onto the net" (https://web.archive.org/web/20090124055806/http://www.boingboing.net/2007/05/30/new-aacs-processing-.html). *BoingBoing*. Archived from the original (https://www.boingboing.net/2007/05/30/new-aacs-processing-.html) on 24 January 2009.
42. Marechal, Sander (9 January 2007). "DRM on audio CDs abolished" (http://lxer.com/module/newswire/view/78008/index.html). Archived (https://web.archive.org/web/20111108144136/http://lxer.com/module/newswire/view/78008/index.html) from the original on 8 November 2011. Retrieved 9 January 2007.
43. Ken Fisher (18 March 2007). "Musicload: 75% of customer service problems caused by DRM" (https://arstechnica.com/news.ars/post/20070318-75-percent-customer-problems-caused-by-drm.html). Ars Technica. Archived (https://web.archive.org/web/20070320204358/http://arstechnica.com/news.ars/post/20070318-75-percent-customer-problems-caused-by-drm.html) from the original on 20 March 2007. Retrieved 20 March 2007.
44. "Apple Unveils Higher Quality DRM-Free Music on the iTunes Store" (https://web.archive.org/web/20070408155651/http://www.apple.com/pr/library/2007/04/02itunes.html). Archived from the original (https://www.apple.com/pr/library/2007/04/02itunes.html) on 8 April 2007. Retrieved 13 September 2008.
45. "Apple hides account info in DRM-free music, too" (https://arstechnica.com/news.ars/post/20070530-apple-hides-account-info-in-drm-free-music-too.html). 30 May 2007. Archived (https://web.archive.org/web/20080908050250/http://arstechnica.com/news.ars/post/20070530-apple-hides-account-info-in-drm-free-music-too.html) from the original on 8 September 2008. Retrieved 13 September 2008.

46. "Apple announces all music on iTunes to go DRM-free – no word on movies, TV shows, games, audiobooks and applications" (https://www.defectivebydesign.org/itunes-drm-free). 8 January 2009. Archived (https://web.archive.org/web/20111125144927/http://www.defectivebydesign.org/itunes-drm-free) from the original on 25 November 2011. Retrieved 25 July 2009.
47. "MLB Fans Who Bought DRM Videos Get Hosed" (http://yro.slashdot.org/article.pl?sid=07/11/07/2014253). slashdot.org. 7 November 2007. Archived (https://web.archive.org/web/20220216045215/https://news.slashdot.org/story/07/11/07/2014253/mlb-fans-who-bought-drm-videos-get-hosed) from the original on 16 February 2022. Retrieved 8 November 2007.
48. "Computex 2007: ASUS Showcases New Generation Audio Card" (http://www.techpowerup.com/forums/threads/computex-2007-asus-showcases-new-generation-audio-card.32449/). *www.techpowerup.com*. 6 June 2007. Archived (https://web.archive.org/web/20150109002358/http://www.techpowerup.com/forums/threads/computex-2007-asus-showcases-new-generation-audio-card.32449/) from the original on 9 January 2015. Retrieved 28 December 2014.
49. "PC Pro Product Reviews Asus Xonar D2" (https://web.archive.org/web/20090531023636/http://www.pcpro.co.uk/reviews/121497/asus-xonar-d2.html). Pcpro.co.uk. 1 August 2007. Archived from the original (http://www.pcpro.co.uk/reviews/121497/asus-xonar-d2.html) on 31 May 2009. Retrieved 20 February 2012.
50. Caron, Frank (9 September 2008). "First look: GOG revives classic PC games for download age" (https://arstechnica.com/gaming/news/2008/09/first-look-gog-revives-classic-pc-games-for-download-age.ars). Ars Technica. Archived (https://web.archive.org/web/20211220150816/https://arstechnica.com/gaming/2008/09/first-look-gog-revives-classic-pc-games-for-download-age/) from the original on 20 December 2021. Retrieved 27 December 2012. "[...] [Good Old Games] focuses on bringing old, time-tested games into the downloadable era with low prices and no DRM."
51. "Tor/Forge E-book Titles to Go DRM-Free" (http://www.tor.com/blogs/2012/04/torforge-e-book-titles-to-go-drm-free). Tor.com. 24 April 2012. Archived (https://web.archive.org/web/20180309172101/https://www.tor.com/2012/04/24/torforge-e-book-titles-to-go-drm-free/) from the original on 9 March 2018. Retrieved 24 April 2012.
52. Ernesto (4 December 2008). "Top 10 Most Pirated Games of 2008" (http://torrentfreak.com/top-10-most-pirated-games-of-2008-081204/). TorrentFreak. Archived (https://web.archive.org/web/20090220140559/http://torrentfreak.com/top-10-most-pirated-games-of-2008-081204/) from the original on 20 February 2009. Retrieved 24 November 2011.
53. Andy Greenberg; Mary Jane Irwin (12 September 2008). "Spore's Piracy Problem" (https://www.forbes.com/2008/09/12/spore-drm-piracy-tech-security-cx_ag_mji_0912spore.html). *Forbes*. Archived (https://web.archive.org/web/20081107032515/https://www.forbes.com/2008/09/12/spore-drm-piracy-tech-security-cx_ag_mji_0912spore.html) from the original on 7 November 2008. Retrieved 22 October 2011.
54. Koroush Ghazi (14 December 2008). "PC Game Piracy Examined: Page 4" (https://web.archive.org/web/20191028160743/https://www.tweakguides.com/Piracy_4.html). Tweakguides. Archived from the original (http://www.tweakguides.com/Piracy_4.html) on 28 October 2019. Retrieved 24 November 2011.
55. "The Sims 3 Will Not Use DRM: News from" (https://archive.today/20120719012143/http://www.1up.com/do/newsStory?cId=3173495). 1UP.com. 29 March 2009. Archived from the original (http://www.1up.com/do/newsStory?cId=3173495) on 19 July 2012. Retrieved 31 August 2010.
56. Kuchera, Ben (12 December 2008). "PC Prince of Persia contains no DRM. It's a trap!" (https://arstechnica.com/gaming/news/2008/12/pc-prince-of-persia-contains-no-drm-its-a-trap.ars). Arstechnica.com. Archived (https://web.archive.org/web/20100703115740/http://arstechnica.com/gaming/news/2008/12/pc-prince-of-persia-contains-no-drm-its-a-trap.ars) from the original on 3 July 2010. Retrieved 31 August 2010.

57. Koroush Ghazi (11 December 2008). "PC Game Piracy Examined: Page 8" (https://web.archive.org/web/20191105161132/https://www.tweakguides.com/Piracy_8.html). Tweakguides. Archived from the original (http://www.tweakguides.com/Piracy_8.html) on 5 November 2019. Retrieved 3 December 2012.
58. Stone, Brad (18 July 2009). "Amazon Erases Orwell Books From Kindle Devices" (https://www.nytimes.com/2009/07/18/technology/companies/18amazon.html). *New York Times*. Archived (https://web.archive.org/web/20100410001949/http://www.nytimes.com/2009/07/18/technology/companies/18amazon.html) from the original on 10 April 2010. Retrieved 1 May 2010.
59. David Pogue (17 July 2009). "Some E-Books Are More Equal Than Others" (https://pogue.blogs.nytimes.com/2009/07/17/some-e-books-are-more-equal-than-others/). *New York Times*. Archived (https://web.archive.org/web/20110709143707/http://pogue.blogs.nytimes.com/2009/07/17/some-e-books-are-more-equal-than-others/) from the original on 9 July 2011. Retrieved 25 July 2009.
60. Pete Cashmore (17 July 2009). "Big Brother: Amazon Remotely Deletes 1984 From Kindles" (http://mashable.com/2009/07/17/amazon-kindle-1984/). *Mashable*. Archived (https://web.archive.org/web/20111103001444/http://mashable.com/2009/07/17/amazon-kindle-1984/) from the original on 3 November 2011. Retrieved 25 July 2009.
61. Mark Frauenfelder (17 July 2009). "Amazon zaps purchased copies of Orwell's 1984 and Animal Farm from Kindles" (https://boingboing.net/2009/07/17/amazon-zaps-purchase.html). Archived (https://web.archive.org/web/20090720140146/http://www.boingboing.net/2009/07/17/amazon-zaps-purchase.html) from the original on 20 July 2009. Retrieved 25 July 2009.
62. Ina Fried (17 July 2009). "Amazon recalls (and embodies) Orwell's '1984' " (https://web.archive.org/web/20110929143324/http://news.cnet.com/8301-13860_3-10289983-56.html). Archived from the original (http://news.cnet.com/8301-13860_3-10289983-56.html) on 29 September 2011. Retrieved 25 July 2009.
63. Free Software Foundation (23 July 2009). "Amazon's CEO Jeff Bezos apologizes for Kindle ebook deletion. Free Software Foundation calls upon Amazon to free the ebook reader" (http://www.fsf.org/news/amazon-apologizes). Archived (https://web.archive.org/web/20090726012920/http://www.fsf.org/news/amazon-apologizes) from the original on 26 July 2009. Retrieved 25 July 2009.
64. "Why Amazon went Big Brother on some Kindle e-books" (https://arstechnica.com/tech-policy/news/2009/07/amazon-sold-pirated-books-raided-some-kindles.ars). arstechnica.com. 17 July 2009. Archived (https://web.archive.org/web/20090720060850/http://arstechnica.com/tech-policy/news/2009/07/amazon-sold-pirated-books-raided-some-kindles.ars) from the original on 20 July 2009. Retrieved 14 June 2017.
65. Ubisoft (9 February 2010). "Ubisoft Press Release" (https://www.ubisoftgroup.com/en-US/press/detail.aspx?cid=tcm:99-30202-16&ctid=tcm:95-27313-32). Ubisoft. Archived (https://web.archive.org/web/20150109004559/https://www.ubisoftgroup.com/en-US/press/detail.aspx?cid=tcm:99-30202-16&ctid=tcm:95-27313-32) from the original on 9 January 2015. Retrieved 4 March 2010.
66. Nic Simmonds (4 March 2010). "Ubisoft's contentious DRM scheme already hacked" (http://www.mygaming.co.za/news/pc/5515-Ubisofts-contentious-DRM-scheme-already-hacked.html). MyGaming. Archived (https://web.archive.org/web/20100307182638/http://www.mygaming.co.za/news/pc/5515-Ubisofts-contentious-DRM-scheme-already-hacked.html) from the original on 7 March 2010. Retrieved 4 March 2010.
67. Andy Chalk (4 March 2010). "Ubisoft Denies Launch Day Crack for Silent Hunter 5 DRM" (http://www.escapistmagazine.com/news/view/98843-Ubisoft-Denies-Launch-Day-Crack-for-Silent-Hunter-5-DRM). Escapist Magazine. Archived (https://web.archive.org/web/20100408121141/http://www.escapistmagazine.com/news/view/98843-Ubisoft-Denies-Launch-Day-Crack-for-Silent-Hunter-5-DRM) from the original on 8 April 2010. Retrieved 27 March 2010.

68. Ben Kuchera (18 February 2010). "Official explanation of controversial Assassin's Creed 2 DRM" (https://arstechnica.com/gaming/news/2010/02/ubisoft-details-drm.ars). Escapist Magazine. Archived (https://web.archive.org/web/20120303211937/http://arstechnica.com/gaming/news/2010/02/ubisoft-details-drm.ars) from the original on 3 March 2012. Retrieved 27 March 2010.
69. "Ubisoft's DRM for Assassin's Creed II is Cracked" (http://www.tomshardware.com/news/assassins-creed-crack-hack-drm-ac2,10260.html). Tomshardware.com. 23 April 2010. Archived (https://archive.today/20120919152151/http://www.tomshardware.com/news/assassins-creed-crack-hack-drm-ac2,10260.html) from the original on 19 September 2012. Retrieved 31 August 2010.
70. Lowensohn, Josh (21 April 2010). "Ubisoft's controversial 'always on' PC DRM hacked" (http://news.cnet.com/8301-27076_3-20003120-248.html). News.cnet.com. Archived (https://web.archive.org/web/20101201123105/http://news.cnet.com/8301-27076_3-20003120-248.html) from the original on 1 December 2010. Retrieved 31 August 2010.
71. John Leyden (8 March 2010). "Ubisoft undone by anti-DRM DDoS storm" (https://www.theregister.co.uk/2010/03/08/ubisoft_anti_drm_hack_attack/). *The Register*. Archived (https://web.archive.org/web/20100312223742/http://www.theregister.co.uk/2010/03/08/ubisoft_anti_drm_hack_attack/) from the original on 12 March 2010. Retrieved 27 March 2010.
72. Andre Yoskowitz (8 March 2010). "Ubisoft apologizes to users affected by 'always on' DRM" (http://www.afterdawn.com/news/article.cfm/2010/03/26/ubisoft_apologizes_to_users_affected_by_always_on_drm). Afterdawn. Archived (https://web.archive.org/web/20100328235520/http://www.afterdawn.com/news/article.cfm/2010/03/26/ubisoft_apologizes_to_users_affected_by_always_on_drm) from the original on 28 March 2010. Retrieved 27 March 2010.
73. "Louis CK: Live at the Beacon Theater" (https://web.archive.org/web/20120516115209/http://buy.louisck.net/news/a-statement-from-louis-c-k). Buy.louisck.net. 13 December 2011. Archived from the original (https://buy.louisck.net/news/a-statement-from-louis-c-k) on 16 May 2012. Retrieved 7 January 2012.
74. Ingram, Mathew (14 December 2011). "What Louis C.K. Knows That Most Media Companies Don't" (https://www.bloomberg.com/news/articles/2011-12-14/what-louis-c-dot-k-dot-knows-that-most-media-companies-dont). *Businessweek*. Archived (https://web.archive.org/web/20170103161314/https://www.bloomberg.com/news/articles/2011-12-14/what-louis-c-dot-k-dot-knows-that-most-media-companies-dont) from the original on 3 January 2017. Retrieved 7 January 2012.
75. chris hannay (6 September 2012). "Why Louis C.K.'s big payday proves the Internet has ethics" (https://www.theglobeandmail.com/technology/tech-news/why-louis-cks-big-payday-proves-the-internet-has-ethics/article4181150/). *The Globe and Mail*. Archived (https://web.archive.org/web/20121103202631/http://www.theglobeandmail.com/technology/tech-news/why-louis-cks-big-payday-proves-the-internet-has-ethics/article4181150/) from the original on 3 November 2012. Retrieved 7 January 2012.
76. "European Court of Justice rules on the right to sell your digital games and licenses" (http://www.pcgamer.com/2012/07/03/european-court-of-justice-rules-on-the-right-to-sell-your-digital-games-and-licenses/). *PC Gamer*. 3 July 2012. Archived (https://web.archive.org/web/20140531104922/http://www.pcgamer.com/2012/07/03/european-court-of-justice-rules-on-the-right-to-sell-your-digital-games-and-licenses/) from the original on 31 May 2014. Retrieved 8 October 2014.
77. AGARWAL, DEVIKA AGARWAL & RADHIKA (4 May 2016). "Needless pressure to change copyright laws" (https://www.thehindubusinessline.com/opinion/needless-pressure-to-change-copyright-laws/article8557036.ece). *@businessline*. Archived (https://web.archive.org/web/20180926190436/https://www.thehindubusinessline.com/opinion/needless-pressure-to-change-copyright-laws/article8557036.ece) from the original on 26 September 2018. Retrieved 16 May 2018.

78. Thomas, Zakir (2012), "Overview of Changes to Indian Copyright Law" (https://web.archive.org/web/20190819052522/http://nopr.niscair.res.in/bitstream/123456789/14460/1/JIPR%2017(4)%20324-334.pdf) (PDF), *Journal of Intellectual Property Rights*, **17**: 324–334, 332, archived from the original (http://nopr.niscair.res.in/bitstream/123456789/14460/1/JIPR%2017(4)%20324-334.pdf) (PDF) on 19 August 2019
79. Scaria, Arul George (2012), "Does India Need Digital Rights Management Provisions or Better Digital Business Management Strategies?" (https://web.archive.org/web/20180516174600/http://nopr.niscair.res.in/bitstream/123456789/14771/1/JIPR%2017%285%29%20463-477.pdf) (PDF), *Journal of Intellectual Property Rights*, **17**: 463–477, 465, archived from the original (http://nopr.niscair.res.in/bitstream/123456789/14771/1/JIPR%2017%285%29%20463-477.pdf) (PDF) on 16 May 2018
80. "A New DRM-free Experiment: Diesel Sweeties" (http://news.cnet.com/8301-30685_3-57359936-264/a-new-drm-free-experiment-diesel-sweeties/). news.cnet.com. 17 January 2012. Archived (https://web.archive.org/web/20120311124558/http://news.cnet.com/8301-30685_3-57359936-264/a-new-drm-free-experiment-diesel-sweeties/) from the original on 11 March 2012. Retrieved 17 February 2012.
81. "Diesel Sweeties Webcomics Ebook" (http://www.dieselsweeties.com/ebooks//). www.dieselsweeties.com. Archived (https://web.archive.org/web/20120221105815/http://www.dieselsweeties.com/ebooks/) from the original on 21 February 2012. Retrieved 17 February 2012.
82. "Free PDF of the first Diesel Sweeties comic collection" (https://boingboing.net/2012/01/17/free-pdf-of-the-first-diesel-s.html). boingboing.net. 17 January 2012. Archived (https://web.archive.org/web/20120226183646/http://boingboing.net/2012/01/17/free-pdf-of-the-first-diesel-s.html) from the original on 26 February 2012. Retrieved 17 February 2012.
83. "An Experiment in iBookery" (https://web.archive.org/web/20120407180255/http://www.dieselsweeties.com/blog/?p=737). dieselsweeties.com. 23 January 2012. Archived from the original (http://www.dieselsweeties.com/blog/?p=737) on 7 April 2012. Retrieved 17 February 2012.
84. "iBooks Follow-up" (https://web.archive.org/web/20120130163103/http://www.dieselsweeties.com/blog/?p=740). dieselsweeties.com. 26 January 2012. Archived from the original (http://www.dieselsweeties.com/blog/?p=740) on 30 January 2012. Retrieved 17 February 2012.
85. "Double Fine's adventure game will be DRM free for backers" (https://www.videogamer.com/news/double_fines_adventure_game_will_be_drm_free_for_backers.html). videogamer.com. 16 February 2012. Archived (https://web.archive.org/web/20180818150748/https://www.videogamer.com/news/double_fines_adventure_game_will_be_drm_free_for_backers.html) from the original on 18 August 2018. Retrieved 17 February 2012.
86. drmfree-torment-is-the-most-richly-backed-game-on-kickstarter (https://web.archive.org/web/20130411075739/http://www.theinquirer.net/inquirer/news/2259790/drmfree-torment-is-the-most-richly-backed-game-on-kickstarter) on theinquirer.net
87. civitas-the-drm-free-alternative-to-simcity-hits-kickstarter (http://www.techspot.com/news/51884-civitas-the-drm-free-alternative-to-simcity-hits-kickstarter.html) Archived (https://web.archive.org/web/20180818214529/https://www.techspot.com/news/51884-civitas-the-drm-free-alternative-to-simcity-hits-kickstarter.html) 18 August 2018 at the Wayback Machine on techspot.com
88. Kitaru Kickstarter supporters get DRM-free copy of the game (http://www.technologytell.com/gaming/93169/kitaru-kickstarter-supporters-get-drm-free-copy-of-the-game/) Archived (https://web.archive.org/web/20170626193112/http://www.technologytell.com/gaming/93169/kitaru-kickstarter-supporters-get-drm-free-copy-of-the-game/) 26 June 2017 at the Wayback Machine on technologytell.com

89. Kelty, Christopher M. (1 March 2012). "The disappearing virtual library" (http://www.aljazeera.com/indepth/opinion/2012/02/2012227143813304790.html). *Al Jazeera*. Archived (https://web.archive.org/web/20130314204623/http://www.aljazeera.com/indepth/opinion/2012/02/2012227143813304790.html) from the original on 14 March 2013. Retrieved 22 March 2013.
90. "Court Orders Shutdown of Libgen, Bookfi, and Sci-Hub" (https://torrentfreak.com/court-orders-shutdown-of-libgen-bookfi-and-sci-hub-151102). *torrentfreak.com*. 2 November 2015. Archived (https://web.archive.org/web/20151104105150/https://torrentfreak.com/court-orders-shutdown-of-libgen-bookfi-and-sci-hub-151102/) from the original on 4 November 2015. Retrieved 5 November 2015.
91. Schiermeier, Quirin (2015). "Pirate research-paper sites play hide-and-seek with publishers" (http://www.nature.com/news/pirate-research-paper-sites-play-hide-and-seek-with-publishers-1.18876). *Nature*. doi:10.1038/nature.2015.18876 (https://doi.org/10.1038%2Fnature.2015.18876). S2CID 188158277 (https://api.semanticscholar.org/CorpusID:188158277). Archived (https://web.archive.org/web/20151206032713/http://www.nature.com/news/pirate-research-paper-sites-play-hide-and-seek-with-publishers-1.18876) from the original on 6 December 2015. Retrieved 6 December 2015.
92. "Sci-hub, bookfi and libgen resurface after being shut down" (https://torrentfreak.com/sci-hub-and-libgen-resurface-after-being-shut-down-151121/). TorrentFreak. 21 November 2015. Archived (https://web.archive.org/web/20200504205320/https://torrentfreak.com/sci-hub-and-libgen-resurface-after-being-shut-down-151121/) from the original on 4 May 2020. Retrieved 7 January 2016.
93. Summer lWalker (20 March 2013). "SimCity Modder Catches EA Lying" (https://web.archive.org/web/20130325231454/http://www.gamebreaker.tv/pc-games/simcity-modder-catches-ea-lying/). Gamebreaker TV. Archived from the original (http://www.gamebreaker.tv/pc-games/simcity-modder-catches-ea-lying/) on 25 March 2013. Retrieved 28 March 2013.
94. "Europe's Highest Court Says DRM Circumvention May Be Lawful in Certain Circumstances" (https://www.techdirt.com/articles/20140123/08532725967/europes-highest-court-says-drm-circumvention-may-be-lawful-certain-circumstances.shtml). *Techdirt*. 23 January 2014. Archived (https://web.archive.org/web/20141016165933/https://www.techdirt.com/articles/20140123/08532725967/europes-highest-court-says-drm-circumvention-may-be-lawful-certain-circumstances.shtml) from the original on 16 October 2014. Retrieved 8 October 2014.
95. "DRM Circumvention May Be Legal, European Union Court Rules" (http://www.escapistmagazine.com/news/view/131597-DRM-Circumvention-May-Be-Legal-European-Union-Court-Rules). *The Escapist*. 23 January 2014. Archived (https://web.archive.org/web/20141012104236/http://www.escapistmagazine.com/news/view/131597-DRM-Circumvention-May-Be-Legal-European-Union-Court-Rules) from the original on 12 October 2014. Retrieved 8 October 2014.
96. "Amazon's comixology introduces DRM-free downloads" (https://www.theguardian.com/technology/2014/jul/25/amazon-comixology-drm-free-downloads-walking-dead-sex-criminals-darkness-comics). *The Guardian*. 25 July 2014. Archived (https://web.archive.org/web/20140729184150/http://www.theguardian.com/technology/2014/jul/25/amazon-comixology-drm-free-downloads-walking-dead-sex-criminals-darkness-comics) from the original on 29 July 2014. Retrieved 30 July 2014.
97. @comiXology (11 February 2022). "15/. For now, we're retiring DRM-free downloads. Previous purchases with downloadable content will continue to be downloadable from http://amazon.com, so no need to rush. Your DRM-free downloads can now be found here: https://amazon.com/comixology/account" (https://x.com/comiXology/status/1492222215139459072) (Tweet) – via Twitter.
98. Espósito, Filipe (17 February 2023). "Comixology app gets major update, but users don't seem to like it" (https://9to5mac.com/2022/02/17/comixology-app-gets-major-update-but-users-dont-seem-to-like-it/). *9To5Mac*.

99. Coates, Shannon Kathleen; Abroshan, Hossein (29 November 2023). "Guideline for the Production of Digital Rights Management (DRM)" (https://aircconline.com/ijsptm/V12N4/12423ijsptm03.pdf) (PDF). *International Journal of Security, Privacy and Trust Management*. **12** (3/4): 31–45. doi:10.5121/ijsptm.2023.12403 (https://doi.org/10.5121%2Fijsptm.2023.12403).
100. Kim, Daniel (30 August 2019). "PlayReady DRM - 5 Things to Know About DRM Technology" (https://pallycon.com/blog/5-things-you-need-to-know-about-multi-drm-technology-part-1/). *PallyCon*. Retrieved 11 March 2021.
101. "NASCA is a DRM solution..." (http://www.sds.samsung.com/popup/solution/epoint.jsp) Archived (https://web.archive.org/web/20131115034406/http://www.sds.samsung.com/popup/solution/epoint.jsp) from the original on 15 November 2013. Retrieved 16 July 2013.
102. TinHat (June 2006). "eBooks and Digital Rights Management (DRM), for ePublishers" (http://www.tinhat.com/ebooks_epublishing/epublishers_drm.html). tinhat.com. Archived (https://web.archive.org/web/20080609112125/http://www.tinhat.com/ebooks_epublishing/epublishers_drm.html) from the original on 9 June 2008. Retrieved 28 May 2008.
103. Braid, Andrew (22 June 2005). "The use of a Digital Rights Management System for Document Supply" (https://web.archive.org/web/20160611154507/http://202.206.214.254:8089/GYZJ/NEPAPER/papers/096e-Braid.pdf) (PDF). *71th IFLA General Conference and Council*. IFLA World Library and Information Congress. Archived from the original (http://202.206.214.254:8089/GYZJ/NEPAPER/papers/096e-Braid.pdf) (PDF) on 11 June 2016.
104. "Practical problems for libraries distributing ebooks & secure electronic delivery" (https://www.locklizard.com/libraries-secure-electronic-delivery/). LockLizard. 2016. Archived (https://web.archive.org/web/20160603175051/http://www.locklizard.com/libraries-secure-electronic-delivery/) from the original on 3 June 2016. Retrieved 11 May 2016.
105. "How to open your on Demand order" (https://www.bl.uk/help/how-to-open-your-on-demand-order). British Library. 1 April 2014. Archived (https://web.archive.org/web/20160603192046/http://www.bl.uk/help/how-to-open-your-on-demand-order) from the original on 3 June 2016. Retrieved 11 May 2016.
106. "MobileRead Wiki – Topaz" (https://wiki.mobileread.com/wiki/Topaz). *wiki.mobileread.com*. Archived (https://web.archive.org/web/20190204174456/https://wiki.mobileread.com/wiki/Topaz) from the original on 4 February 2019. Retrieved 4 February 2019.
107. drewbatgit; v-kents; DCtheGeek; msatranjr (3 March 2021). "Protected Media Path - Win32 apps" (https://learn.microsoft.com/en-us/windows/win32/medfound/protected-media-path). *learn.microsoft.com*. Retrieved 5 November 2022.
108. Nathan Grayson (17 November 2011). "Interview: Bohemia Interactive's CEO on fighting piracy, creative DRM" (http://www.pcgamer.com/2011/11/17/interview-bohemia-interactives-ceo-on-fighting-piracy-creative-drm/). pcgamer.com. Archived (https://web.archive.org/web/20120107120010/http://www.pcgamer.com/2011/11/17/interview-bohemia-interactives-ceo-on-fighting-piracy-creative-drm) from the original on 7 January 2012. Retrieved 5 January 2012.
109. John Walker (7 December 2011). "Serious Sam's DRM Is A Giant Pink Scorpion" (http://www.rockpapershotgun.com/2011/12/07/serious-sams-drm-is-a-giant-pink-scorpion/). rockpapershotgun.com. Archived (https://web.archive.org/web/20111208162958/http://www.rockpapershotgun.com/2011/12/07/serious-sams-drm-is-a-giant-pink-scorpion/) from the original on 8 December 2011. Retrieved 9 December 2011.
110. socketboy (8 December 2011). "Serious Sam 3 Non-Traditional DRM Solutions Part 2" (https://web.archive.org/web/20120216132324/http://ve3d.ign.com/articles/news/63236/Serious-Sam-3-Non-Traditional-DRM-Solutions-Part-2). ign.com. Archived from the original (http://ve3d.ign.com/articles/news/63236/Serious-Sam-3-Non-Traditional-DRM-Solutions-Part-2) on 16 February 2012. Retrieved 9 December 2011.

111. Wolf, Patrick; Steinebach, Martin; Diener, Konstantin (27 February 2007). Yagüe, Mariemma (ed.). "Complementing DRM with digital watermarking: mark, search, retrieve". *Online Information Review*. **31** (1): 10–21. doi:10.1108/14684520710731001 (https://doi.org/10.1108%2F14684520710731001). ISSN 1468-4527 (https://search.worldcat.org/issn/1468-4527).
112. "Tunes Plus DRM-free, not free of annoying glitches" (https://www.engadget.com/2007/05/31/itunes-plus-drm-free-not-free-of-annoying-glitches/). Engadget. 31 May 2007. Archived (https://web.archive.org/web/20180811032526/https://www.engadget.com/2007/05/31/itunes-plus-drm-free-not-free-of-annoying-glitches/) from the original on 11 August 2018. Retrieved 26 August 2017.
113. "Watermarked iTunes files" (https://www.macworld.com/article/1058178/ituneswatermark.html). *Macworld*. Archived (https://web.archive.org/web/20200729214840/https://www.macworld.com/article/1058178/ituneswatermark.html) from the original on 29 July 2020. Retrieved 17 April 2020.
114. Urs Gasser, 'Legal Frameworks and Technological Protection Measures: Moving towards a Best Practices Model' Research Publication No. 2006-04 at Berkman Klein Center for Internet and Society (Harvard) <available at: https://courses.edx.org/c4x/HarvardX/HLS1.1x/asset/Urs_Gasser.pdf Archived (https://web.archive.org/web/20170809032213/https://courses.edx.org/c4x/HarvardX/HLS1.1x/asset/Urs_Gasser.pdf) 9 August 2017 at the Wayback Machine last accessed 17 May 2018>
115. "COPYRIGHT ACT 1968 - SECT 116AN Circumventing an access control technological protection measure" (https://www8.austlii.edu.au/cgi-bin/viewdoc/au/legis/cth/consol_act/ca1968133/s116an.html).
116. "COPYRIGHT ACT 1968 - SECT 116AO Manufacturing etc. a circumvention device for a technological protection measure" (https://www8.austlii.edu.au/cgi-bin/viewdoc/au/legis/cth/consol_act/ca1968133/s116ao.html).
117. "COPYRIGHT ACT 1968 - SECT 116AP Providing etc. a circumvention service for a technological protection measure" (https://www8.austlii.edu.au/cgi-bin/viewdoc/au/legis/cth/consol_act/ca1968133/s116ap.html).
118. "COPYRIGHT ACT 1968 - SECT 116AQ Remedies in actions under this Subdivision" (https://www8.austlii.edu.au/cgi-bin/viewdoc/au/legis/cth/consol_act/ca1968133/s116aq.html).
119. Nintendo of America. "2014 Special 301 Comments on Piracy of Nintendo Video Game Products" (https://downloads.regulations.gov/USTR-2013-0040-0034/attachment_1.pdf) (PDF).
120. "India Joins WIPO Internet Treaties: A Major Step Forward" (https://www.mpa-apac.org/2018/07/india-joins-wipo-internet-treaties-a-major-step-forward/). *mpaa-apac.org*. Retrieved 11 November 2024.
121. "Israel Technology Law Blog" (https://israeltechnologylaw.wordpress.com/2014/02/16/no-digital-rights-management-here/). *israeltechnologylaw.wordpress.com*. Archived (https://web.archive.org/web/20210429231827/https://israeltechnologylaw.wordpress.com/2014/02/16/no-digital-rights-management-here/) from the original on 29 April 2021. Retrieved 16 February 2022.
122. Art. 1, [https://www.japaneselawtranslation.go.jp/en/laws/view/3379#je_s21 Supplementary Provisions (Act No. 77 of June 23, 1999)
123. Art. 30 (ii), Copyright Act (Act No. 48 of May 6, 1970) (https://www.japaneselawtranslation.go.jp/en/laws/view/3379#je_ch8at3)
124. Art. 113, Copyright Act (Act No. 48 of May 6, 1970) (https://www.japaneselawtranslation.go.jp/en/laws/view/3379#je_ch8at3)
125. "301's Flaws" (https://www.eff.org/special-404). *eff.org*. Retrieved 17 September 2023.
126. "2022 Investment Climate Statements: Pakistan" (https://www.state.gov/reports/2022-investment-climate-statements/pakistan/). *www.state.gov*. Retrieved 17 September 2023.

127. 2023 Special 301 Report (https://ustr.gov/sites/default/files/2023-04/2023%20Special%20301%20Report.pdf) (PDF) (Report). Office of the United States Trade Representative. April 2023. p. 81. Retrieved 31 May 2024.
128. "2023 Investment Climate Statements: Pakistan" (https://www.state.gov/reports/2023-investment-climate-statements/pakistan/). *www.state.gov*. Retrieved 17 September 2023.
129. Doctorow, Cory (24 March 2007). "DMCA's author says the DMCA is a failure, blames record industry" (https://web.archive.org/web/20110623191635/http://boingboing.net/2007/03/24/dmcas_author_says_th.html). Archived from the original (https://boingboing.net/2007/03/24/dmcas_author_says_th.html) on 23 June 2011. Retrieved 12 January 2011.
130. Walker, John (13 September 2003). "The Digital Imprimatur: How big brother and big media can put the Internet genie back in the bottle" (http://www.fourmilab.ch/documents/digital-imprimatur/). Archived (https://web.archive.org/web/20111105014841/http://www.fourmilab.ch/documents/digital-imprimatur/) from the original on 5 November 2011. Retrieved 4 February 2004.
131. Richard Stallman. "The Right to Read" (https://www.gnu.org/philosophy/right-to-read.html). Archived (https://web.archive.org/web/20140420074747/http://www.gnu.org/philosophy/right-to-read.html) from the original on 20 April 2014. Retrieved 18 April 2014.
132. O'Riordan, Ciaran (16 January 2006). "Transcript of Opening session of first international GPLv3 conference" (http://www.ifso.ie/documents/gplv3-launch-2006-01-16.html). Archived (https://web.archive.org/web/20111021033558/http://www.ifso.ie/documents/gplv3-launch-2006-01-16.html) from the original on 21 October 2011. Retrieved 21 February 2006.
133. "Opposing Digital Rights Mismanagement (Or Digital Restrictions Management, as we now call it)?" (https://www.gnu.org/philosophy/opposing-drm.html). Archived (https://web.archive.org/web/20140706043116/http://www.gnu.org/philosophy/opposing-drm.html) from the original on 6 July 2014. Retrieved 29 May 2014.
134. David Berlind (18 November 2005). "Sony rootkit: The untold story" (https://www.zdnet.com/article/sony-rootkit-the-untold-story/). *ZDNet*. Archived (https://web.archive.org/web/20150611005913/http://www.zdnet.com/article/sony-rootkit-the-untold-story/) from the original on 11 June 2015. Retrieved 20 August 2015.
135. Grassmuck, Volker (2003), "Vom PC zum TC: Trusted Computing und Digital Restrictions Management" (https://web.archive.org/web/20160305103403/http://waste.informatik.hu-berlin.de/Grassmuck/Texts/TCG.pdf) (PDF), *Trusted Computing*, Kommunikation & Recht, Frankfurt am Main: Verlag Recht und Wirtschaft, archived from the original (http://waste.informatik.hu-berlin.de/Grassmuck/Texts/TCG.pdf) (PDF) on 5 March 2016, retrieved 20 August 2015
136. Jamali, Hamid R.; Nicholas, David; Rowlands, Ian (2009). "Scholarly e-books: the views of 16,000 academics: Results from the JISC National E-Book Observatory". *ASLIB Proceedings*. Emerald Insight. pp. 33–47. doi:10.1108/00012530910932276 (https://doi.org/10.1108%2F00012530910932276).
137. Doctorow, Cory (17 June 2004). "Microsoft Research DRM Talk" (https://web.archive.org/web/20100102045736/http://changethis.com/4.DRM). craphound.com. Archived from the original (http://www.changethis.com/4.DRM) (PDF) on 2 January 2010. Retrieved 17 August 2007. "At the end of the day, all DRM systems share a common vulnerability: they provide their attackers with ciphertext, the cipher and the key. At this point, the secret isn't a secret anymore."
138. Pavlik, John (2008). *Media in the Digital Age*. New York: Columbia University Press. p. 181. ISBN 978-0231142090.
139. The Futility of Digital Copy Prevention (http://www.schneier.com/crypto-gram-0105.html#3) Archived (https://web.archive.org/web/20111115145148/http://www.schneier.com/crypto-gram-0105.html#3) 15 November 2011 at the Wayback Machine Crypto-Gram Newsletter, 15 May 2001

140. Schneier, Bruce (7 September 2005). "Quickest Patch Ever" (https://web.archive.org/web/20100103095236/http://www.wired.com/politics/security/commentary/securitymatters/2006/09/71738). *Wired*. Condé Nast. Archived from the original (https://www.wired.com/politics/security/commentary/securitymatters/2006/09/71738) on 3 January 2010.
141. "Official words of StarForce on DRM" (http://www.glop.org/starforce/images/starforce-official-words.png). Archived (https://web.archive.org/web/20100926064848/http://glop.org/starforce/images/starforce-official-words.png) from the original on 26 September 2010. Retrieved 8 September 2010.
142. "Bill Gates On The Future Of DRM" (https://techcrunch.com/2006/12/14/bill-gates-on-the-future-of-drm/). 14 December 2006. Archived (https://web.archive.org/web/20220216045221/https://techcrunch.com/2006/12/14/bill-gates-on-the-future-of-drm/) from the original on 16 February 2022. Retrieved 16 February 2022.
143. "FTC Town Hall to Address Digital Rights Management Technologies" (https://web.archive.org/web/20141029070751/http://www.ftc.gov/news-events/press-releases/2008/12/ftc-town-hall-address-digital-rights-management-technologies). Federal Trade Commission. 23 December 2008. Archived from the original (https://www.ftc.gov/news-events/press-releases/2008/12/ftc-town-hall-address-digital-rights-management-technologies) on 29 October 2014.
144. "Gabe Newell Says DRM Strategies 'Are Just Dumb' " (https://web.archive.org/web/20160204142323/http://www.1up.com/news/gabe-newell-drm-strategies-dumb). 2 December 2008. Archived from the original on 4 February 2016. Retrieved 2 December 2008.
145. Kain, Eric. " 'Witcher 2' Developer: 'We Will Never Use Any DRM Anymore' " (https://www.forbes.com/sites/erikkain/2012/03/08/witcher-2-developer-we-will-never-use-any-drm-anymore/). *Forbes*. Archived (https://web.archive.org/web/20120317091733/http://www.forbes.com/sites/erikkain/2012/03/08/witcher-2-developer-we-will-never-use-any-drm-anymore/) from the original on 17 March 2012. Retrieved 30 March 2012.
146. Perry, Tekla S. (1 January 2005). "Loser: DVD Copy Protection, Take 2" (https://spectrum.ieee.org/loser-dvd-copy-protection-take-2). *IEEE Spectrum*. Retrieved 11 August 2024.
147. "Frequently Asked Questions about the GNU Licenses – GNU Project – Free Software Foundation" (http://www.fsf.org/licensing/licenses/gpl-faq.html#DRMProhibited). FSF. 28 July 2010. Archived (https://web.archive.org/web/20100203231849/http://www.fsf.org/licensing/licenses/gpl-faq.html#DRMProhibited) from the original on 3 February 2010. Retrieved 31 August 2010.
148. "The Campaign to Eliminate DRM" (https://www.defectivebydesign.org/). DefectiveByDesign.org. Archived (https://web.archive.org/web/20180123173030/https://www.defectivebydesign.org/) from the original on 23 January 2018. Retrieved 31 August 2010.
149. "Frequently Asked Questions – CcWiki" (http://wiki.creativecommons.org/FAQ#Is_Creative_Commons_involved_in_digital_rights_management_.28DRM.29.3F). 9 November 2006. Archived (https://web.archive.org/web/20130726210512/http://wiki.creativecommons.org/FAQ#Is_Creative_Commons_involved_in_digital_rights_management_.28DRM.29.3F) from the original on 26 July 2013. Retrieved 22 November 2006.
150. "Baseline Rights – Creative Commons Wiki" (http://wiki.creativecommons.org/Baseline_Rights). 13 July 2007. Archived (https://web.archive.org/web/20150401094821/https://wiki.creativecommons.org/Baseline_Rights) from the original on 1 April 2015. Retrieved 23 December 2007.
151. Hector, Hamish (17 January 2022). "Friendly reminder: soon new ComiXology purchases will not be DRM-free" (https://www.techradar.com/news/friendly-reminder-soon-new-comixology-purchases-will-not-be-drm-free). *TechRadar*. Future US.

152. Taylor, Mollie (8 November 2021). "A great day for DRM as Denuvo lapse renders tons of games temporarily unplayable" (https://www.pcgamer.com/a-great-day-for-drm-as-denuvo-lapse-renders-tons-of-games-temporarily-unplayable/). *PC Gamer*. Archived (https://web.archive.org/web/20211109012336/https://www.pcgamer.com/a-great-day-for-drm-as-denuvo-lapse-renders-tons-of-games-temporarily-unplayable/) from the original on 9 November 2021. Retrieved 9 November 2021.
153. Lewis, Rita (8 January 2008). "What is DRM and Why Should I Care?" (https://web.archive.org/web/20080614083008/http://firefox.org/news/articles/1045/1/What-is-DRM-and-why-should-I-care/Page1.html). Firefox News. Archived from the original (http://firefox.org/news/articles/1045/1/What-is-DRM-and-why-should-I-care/Page1.html) on 14 June 2008. Retrieved 10 July 2008.
154. Machkovech, Sam (26 December 2018). "Evidence continues to mount about how bad Denuvo is for PC gaming performance" (https://arstechnica.com/gaming/2018/12/evidence-continues-to-mount-about-how-bad-denuvo-is-for-pc-gaming-performance/). *Ars Technica*. Archived (https://web.archive.org/web/20190410171912/https://arstechnica.com/gaming/2018/12/evidence-continues-to-mount-about-how-bad-denuvo-is-for-pc-gaming-performance/) from the original on 10 April 2019. Retrieved 26 June 2019.
155. "Denuvo DRM Proven To Hurt Performance Of Games It's Attached To" (https://www.thegamer.com/denuvo-drm-hurts-game-performance/). *TheGamer*. 29 March 2019. Archived (https://web.archive.org/web/20190626111531/https://www.thegamer.com/denuvo-drm-hurts-game-performance/) from the original on 26 June 2019. Retrieved 26 June 2019.
156. Thoman, Peter "Durante" (9 March 2018). "Tested: Denuvo DRM has no performance impact on Final Fantasy 15" (https://www.pcgamer.com/denuvo-drm-performance-final-fantasy-15/). *PC Gamer*. Archived (https://web.archive.org/web/20190727200418/https://www.pcgamer.com/denuvo-drm-performance-final-fantasy-15/) from the original on 27 July 2019. Retrieved 26 June 2019.
157. Sicker, Douglas; Ohm, Paul; Gunaji, Shannon (2007). "The Analog Hole and the Price of Music: An Empirical Study". *Journal on Telecommunications and High Technology Law*. **5**. SSRN 969998 (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=969998).
158. "MPAA shows how to videorecord a TV set" (https://vimeo.com/4520463). Archived (https://web.archive.org/web/20111217082309/http://vimeo.com/4520463) from the original on 17 December 2011. Retrieved 4 September 2009.
159. Crosby, Scott; Goldberg, Ian; Johnson, Robert; Song, Dawn (2002). "A Cryptanalysis of the High-Bandwidth Digital Content Protection System". In Sander, Tomas (ed.). *Security and Privacy in Digital Rights Management*. Lecture Notes in Computer Science. Vol. 2320. Springer. pp. 192–200. doi:10.1007/3-540-47870-1_12 (https://doi.org/10.1007%2F3-540-47870-1_12). ISBN 978-3-540-47870-6.
160. DRM Is The Right To Make Up Your Own Copyright Laws (https://www.techdirt.com/articles/20140206/10323526118/drm-is-right-to-make-up-your-own-copyright-laws.shtml) Archived (https://web.archive.org/web/20180819083940/https://www.techdirt.com/articles/20140206/10323526118/drm-is-right-to-make-up-your-own-copyright-laws.shtml) 19 August 2018 at the Wayback Machine – Mike Masnick, Techdirt, 6 February 2014
161. "What Content Will Be Crippled When Output in Vista?" (https://web.archive.org/web/20101006042342/http://msmvps.com/blogs/chrisl/archive/2007/01/25/519180.aspx). Archived from the original (http://msmvps.com/blogs/chrisl/archive/2007/01/25/519180.aspx) on 6 October 2010. Retrieved 8 September 2010.
162. "Why is Netflix ignoring Linux?" (http://wesleytech.com/why-is-netflix-ignoring-linux/2072/). 29 May 2010. Archived (https://web.archive.org/web/20100611123221/http://wesleytech.com/why-is-netflix-ignoring-linux/2072/) from the original on 11 June 2010. Retrieved 8 September 2010.

163. Why the Kindle's DRM is anti-elderly: AARP should fight against it (http://www.teleread.org/2009/02/26/why-the-kindles-drm-is-anti-elderly-and-perhaps-even-a-target-for-an-aarp-campaign-against-it/) Archived (https://web.archive.org/web/20090503064350/http://www.teleread.org/2009/02/26/why-the-kindles-drm-is-anti-elderly-and-perhaps-even-a-target-for-an-aarp-campaign-against-it/) 3 May 2009 at the Wayback Machine, 26 February 2009
164. Cheng, Jacqui (22 April 2008). "DRM sucks redux: Microsoft to nuke MSN Music DRM keys" (https://arstechnica.com/news.ars/post/20080422-drm-sucks-redux-microsoft-to-nuke-msn-music-drm-keys.html). Ars Technica. Archived (https://web.archive.org/web/20120529102807/http://arstechnica.com/information-technology/2008/04/drm-sucks-redux-microsoft-to-nuke-msn-music-drm-keys/) from the original on 29 May 2012. Retrieved 22 April 2008.
165. Anderson, Nate (24 July 2008). "DRM still sucks: Yahoo Music going dark, taking keys with it" (https://arstechnica.com/uncategorized/2008/07/drm-still-sucks-yahoo-music-going-dark-taking-keys-with-it/). Ars Technica. Archived (https://web.archive.org/web/20121118025222/http://arstechnica.com/uncategorized/2008/07/drm-still-sucks-yahoo-music-going-dark-taking-keys-with-it/) from the original on 18 November 2012. Retrieved 18 May 2012.
166. "Adobe Content 3 Server Discontinued" (http://kb2.adobe.com/cps/409/kb409073.html). Archived (https://web.archive.org/web/20111007150727/http://kb2.adobe.com/cps/409/kb409073.html) from the original on 7 October 2011. Retrieved 22 February 2011.
167. "VOD Service Acetrax Shutting Down, Forcing Customers Through DRM Hoops To Retain Their Purchased Movies" (https://www.techdirt.com/articles/20130522/10290323174/vod-service-acetrax-shuts-down-germany-customers-movies-are-lost-due-to-drm.shtml/). 23 May 2013. Archived (https://web.archive.org/web/20130609212643/http://www.techdirt.com/articles/20130522/10290323174/vod-service-acetrax-shuts-down-germany-customers-movies-are-lost-due-to-drm.shtml) from the original on 9 June 2013. Retrieved 23 May 2013.
168. "Australia Recording industry Association Survey, 2003" (http://www.aria.com.au/pages/CurrentIssueInternetMusicFileSharingCDBurning.htm). Archived (https://web.archive.org/web/20110217014855/http://aria.com.au/pages/CurrentIssueInternetMusicFileSharingCDBurning.htm) from the original on 17 February 2011. Retrieved 5 February 2011.
169. "The Customer Is Always Wrong" (https://www.eff.org/pages/customer-always-wrong-users-guide-drm-online-music). Electronic Frontier Foundation. 25 September 2007. Archived (https://web.archive.org/web/20110205171449/http://www.eff.org/pages/customer-always-wrong-users-guide-drm-online-music) from the original on 5 February 2011. Retrieved 6 February 2011.
170. Mondok, Matt (13 March 2007). "Microsoft executive: Pirating software? Choose Microsoft!" (https://arstechnica.com/microsoft/news/2007/03/microsoft-executive-pirating-software-choose-microsoft.ars). *Ars Technica*. Archived (https://web.archive.org/web/20110223072518/http://arstechnica.com/microsoft/news/2007/03/microsoft-executive-pirating-software-choose-microsoft.ars) from the original on 23 February 2011. Retrieved 6 February 2011.
171. Conner, Kathleen; Rummelt, Richard (1991). "Software Piracy: An Analysis of Protection Strategies". *Management Science*. **37** (2): 125–139. doi:10.1287/mnsc.37.2.125 (https://doi.org/10.1287%2Fmnsc.37.2.125). JSTOR 2632386 (https://www.jstor.org/stable/2632386).
172. Oestreicher-Singer, Gal and Arun Sundararajan (2004). "Are Digital Rights Valuable? Theory and Evidence from the eBook Industry" (http://aisel.aisnet.org/icis2004/42/). *Proceedings of the International Conference on Information Systems*. Archived (https://web.archive.org/web/20110724230919/http://aisel.aisnet.org/icis2004/42/) from the original on 24 July 2011. Retrieved 26 February 2011.
173. Dinah A. Vernik; Devavrat Purohit; Preyas S. Desai (2011). "Music Downloads and the Flip Side of Digital Rights Management" (https://web.archive.org/web/20190807021340/http://static.arstechnica.net/2011/10/11/mksc.1110.0668-1.pdf) (PDF). Archived from the original (http://static.arstechnica.net/2011/10/11/mksc.1110.0668-1.pdf) (PDF) on 7 August 2019. Retrieved 12 July 2012.

174. Committee on Intellectual Property Rights in the Emerging Information Infrastructure, National Research Council. (2000) "The Digital Dilemma: Intellectual Property in the Information Age." 3 May 2011. http://www.nap.edu/catalog.php?record_id=9601#toc Archived (https://web.archive.org/web/20120120212259/http://www.nap.edu/catalog.php?record_id=9601#toc) 20 January 2012 at the Wayback Machine
175. "Surprise, surprise – almost every piracy website features cyber scams" (http://betanews.com/2014/04/30/surprise-surprise-almost-every-piracy-website-features-cyber-scams/). *BetaNews*. 30 April 2014. Archived (https://web.archive.org/web/20141013142627/http://betanews.com/2014/04/30/surprise-surprise-almost-every-piracy-website-features-cyber-scams/) from the original on 13 October 2014. Retrieved 8 October 2014.
176. Baker, Dean. (2003). "The Artistic Freedom Voucher: An Internet Age Alternative to Copyrights." Pg. 2–8. Web. 3 May. 2011. http://www.cepr.net/documents/publications/ip_2003_11.pdf Archived (https://web.archive.org/web/20110516024418/http://www.cepr.net/documents/publications/ip_2003_11.pdf) 16 May 2011 at the Wayback Machine

Further reading

- Lawrence Lessig's *Free Culture*, published by Basic Books in 2004, is available for free download in PDF format (http://www.free-culture.cc/freeculture.pdf) Archived (https://web.archive.org/web/20090916145748/http://www.free-culture.cc/freeculture.pdf) 16 September 2009 at the Wayback Machine. The book is a legal and social history of copyright. Lessig is well known, in part, for arguing landmark cases on copyright law. A professor of law at Stanford University, Lessig writes for an educated lay audience, including for non-lawyers. He is, for the most part, an opponent of DRM technologies.
- Rosenblatt, B. et al., *Digital Rights Management: Business and Technology*, published by M&T Books (John Wiley & Sons) in 2001. An overview of DRM technology, business implications for content publishers, and relationship to U.S. copyright law.
- Consumer's Guide to DRM (https://web.archive.org/web/20101208173800/http://www.microsoft.com/windows/windowsmedia/forpros/drm/faq.aspx), published in 10 languages (Czech, German, Greek, English, Spanish, French, Hungarian, Italian, Polish, Swedish), produced by the INDICARE research and dialogue project (https://web.archive.org/web/20101208173800/http://www.microsoft.com/windows/windowsmedia/forpros/drm/faq.aspx)
- Eberhard Becker, Willms Buhse, Dirk Günnewig, Niels Rump: *Digital Rights Management – Technological, Economic, Legal and Political Aspects*. An 800-page compendium from 60 different authors on DRM.
- Arun Sundararajan's uses the following digital rights conjecture, that "digital rights increases the incidence of digital piracy, and that managing digital rights therefore involves restricting the rights of usage that contribute to customer value" to show that creative pricing can be an effective substitute for excessively stringent DRM.
- Fetscherin, M., *Implications of Digital Rights Management on the Demand for Digital Content*, provides a view on DRM from a consumers perspective. "Buch- und online Publikationen" (https://web.archive.org/web/20101205230016/http://www.dissertation.de/index.php3?active_document=buch.php3&buch=4731). dissertation.de. 5 February 1998. Archived from the original (http://www.dissertation.de/index.php3?active_document=buch.php3&buch=4731) on 5 December 2010. Retrieved 31 August 2010.
- *The Pig and the Box*, a book with colorful illustrations and having a coloring book version, by 'MCM'. It describes DRM in terms suited to kids, written in reaction to a Canadian entertainment industry copyright education initiative, aimed at children.
- *Present State and Emerging Scenarios of Digital Rights Management Systems* – A paper by Marc Fetscherin which provides an overview of the various components of DRM, pro and

cons and future outlook of how, where, when such systems might be used.

- *DRM is Like Paying for Ice (https://web.archive.org/web/20101208173800/http://www.microsoft.com/windows/windowsmedia/forpros/drm/faq.aspx)* – Richard Menta article on MP3 Newswire discusses how DRM is implemented in ways to control consumers, but is undermining perceived product value in the process.
- *A Semantic Web Approach to Digital Rights Management (http://rhizomik.net/~roberto/thesis)* – PhD Thesis by Roberto García that tries to address DRM issues using Semantic Web technologies and methodologies.
- Patricia Akester, "Technological Accommodation of Conflicts between Freedom of Expression and DRM: The First Empirical Assessment" available at Technological Accommodation of Conflicts between Freedom of Expression and DRM: The First Empirical Assessment (https://ssrn.com/abstract=1469412) Archived (https://web.archive.org/web/20220216045217/https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1469412) 16 February 2022 at the Wayback Machine (unveiling, through empirical lines of enquiry, (1) whether certain acts which are permitted by law are being adversely affected by the use of DRM and (2) whether technology can accommodate conflicts between freedom of expression and DRM).

External links

- BBC News Technology (http://news.bbc.co.uk/2/hi/technology/6337781.stm) Q&A: What is DRM?
- Copyright vs Community in the Age of Computer Networks (https://csclub.uwaterloo.ca/resources/tech-talks/copyright-vs-community-in-the-age-of-computer-networks/) by Richard Stallman
- Windows Media DRM FAQ (https://web.archive.org/web/20101208173800/http://www.microsoft.com/windows/windowsmedia/forpros/drm/faq.aspx) at the Wayback Machine (archived 8 December 2010) from Microsoft
- Microsoft Research DRM talk (https://web.archive.org/web/20080308232149/http://ec.europa.eu/enterprise/ict/policy/doc/drm.pdf), by Cory Doctorow
- iTunes, DRM and competition law (https://web.archive.org/web/20090302030608/http://www.reckon.co.uk/open/iTunes) by Reckon LLP
- Digital Rights Management (https://web.archive.org/web/20080308232149/http://ec.europa.eu/enterprise/ict/policy/doc/drm.pdf) at the Wayback Machine (archived 8 March 2008) from CEN/ISSS (European Committee for Standardization / Information Society Standardization System). Contains a range of possible definitions for DRM from various stakeholders. 30 September 2003
- PC Game Piracy Examined (https://tweakguides.pcgamingwiki.com/Piracy_1.html) Article investigating the effects of DRM and piracy on the video game industry
- DRM.info (https://drm.info/) Information about DRM by Chaos Computer Club, Defective by design, Electronic Frontier Foundation, Free Software Foundation Europe, and other organisations.

License proliferation

License proliferation is the phenomenon of an abundance of already existing and the continued creation of new software licenses for software and software packages in the FOSS ecosystem. License proliferation affects the whole FOSS ecosystem negatively by the burden of increasingly complex license selection, license interaction, and license compatibility considerations.[1]

Impact

Often when a software developer would like to merge portions of different software programs they are unable to do so because the licenses are incompatible. When software under two different licenses can be merged into a larger software work, the licenses are said to be compatible. As the number of licenses increases, the probability that a free and open-source software (FOSS) developer will want to merge software that are available under incompatible licenses increases. There is also a greater cost to companies that wish to evaluate every FOSS license for software packages that they use.[1] Strictly speaking, no one is in favor of license proliferation. Rather, the issue stems from the tendency for organizations to write new licenses in order to address real or perceived needs for their software releases.

License compatibility

License proliferation is especially a problem when licenses have only limited or complicated license compatibility relationships with other licenses. Therefore, some consider compatibility with the widely used GNU General Public License (GPL) an important characteristic, for instance David A. Wheeler[2][3] as also the Free Software Foundation (FSF), who maintains a list of the licenses that are compatible with the GPL.[4] On the other hand, some recommend Permissive licenses, instead of copyleft licenses,[5] due to the better compatibility with more licenses.[6][7] The Apache Foundation for instance criticizes the fact that while the Apache License is compatible with the copyleft GPLv3, the GPLv3 is not compatible with the permissive Apache license — Apache software can be included in GPLv3 software but not vice versa.[8] As another relevant example, the GPLv2 is by itself not compatible with the GPLv3.[9] The 2007 released GPLv3 was criticized by several authors for adding another incompatible license in the FOSS ecosystem.[10][11][12][13][14][15][16]

Vanity licenses

A vanity license is a license that is written by a company or person for no other reason than to write their own license ("NIH syndrome").[17] If a new license is created that has no obvious improvement or difference over another more common FOSS license it can often be criticized as a vanity license. As of 2008, many people create a custom new license for their newly released program, without knowing the requirements for a FOSS license and without realizing that using a nonstandard license can make that program almost useless to others.[18]

Solution approaches

GitHub's stance

In July 2013, GitHub started a license selection wizard called *choosealicense*.[19] GitHub's *choosealicense* frontpage offers as a quick selection only three licenses: the MIT License, the Apache License and the GNU General Public License. Some additional licenses are offered on subpages and via links.[20] Following in 2015, approx. 77% of all licensed projects on GitHub were licensed under at least one of these three licenses.[21]

Google's stance

From 2006 Google Code only accepted projects licensed under the following seven licenses:[22]

- Apache License 2.0
- New BSD License
- MIT License
- GNU General Public License 2.0
- GNU Lesser General Public License 2.1
- Mozilla Public License 1.1
- Artistic License/GPL dual-licensed (often used by the Perl community)

One year later, around 2008, the GNU General Public License 3.0 was added and strongly recommended together with the permissive Apache license,[23] notably excluded was the AGPLv3 to reduce license proliferation.[24]

In 2010, Google removed these restrictions, and announced that it would allow projects to use any OSI-approved license (see OSI's stance below),[25] but with the limitation that public domain projects are only allowed as single case decision.

OSI's stance

Open Source Initiative (OSI) maintains a list of approved licenses.[26] Early in its history, the OSI contributed to license proliferation by approving vanity and non-reusable licenses. In 2004 an OSI License Proliferation Project was started[27] has prepared a License Proliferation Report in 2007.[28] The report defined classes of licenses:

- Licenses that are popular and widely used or with strong communities
- International licenses
- Special purpose licenses
- Other/Miscellaneous licenses
- Licenses that are redundant with more popular licenses
- Non-reusable licenses
- Superseded licenses
- Licenses that have been voluntarily retired
- Uncategorized Licenses

The group of "popular" licenses include nine licenses: Apache License 2.0, New BSD license, GPLv2, LGPLv2, MIT license, Mozilla Public License 1.1, Common Development and Distribution License, Common Public License, Eclipse Public License.

FSF's stance

Richard Stallman, former president of Free Software Foundation, and Bradley M. Kuhn, former Executive Director, have argued against license proliferation since 2000, when they instituted the FSF *license list*, which urges developers to license their software under GPL-compatible free software license(s), though multiple GPL-incompatible free software licenses are listed with a comment stating that there is no problem using and/or working on a piece of software already under the licenses in question while also urging readers of the list not to use those licenses on software they write.[29]

Ciarán O'Riordan of FSF Europe argues that the main thing that the FSF can do to prevent license proliferation is to reduce the reasons for making new licenses in the first place, in an editorial entitled *How GPLv3 tackles license proliferation*.[30] Generally the FSF Europe consistently recommends the use of the GNU GPL as much as possible, and when that is not possible, to use GPL-compatible licenses.

Others

In 2005 Intel has voluntarily retracted their Intel Open Source License from the OSI list of open source licenses and has also ceased to use or recommend this license to reduce license proliferation.[31]

The 451group created in June 2009 a proliferation report called *The Myth of Open Source License Proliferation*.[32] A 2009 paper from the University of Washington School of Law titled *Open Source License Proliferation: Helpful Diversity or Hopeless Confusion?* called for three things as a solution: *"A Wizzier Wizzard"* (for license selection), *"Best Practices and Legacy Licenses"*, *"More Legal Services For Hackers"*.[33] The OpenSource Software Collaboration Counseling (OSSCC) recommends, based on the originally nine recommended OSI licenses, five licenses: the Apache License 2.0, New BSD License, CDDL, MIT license, and to some degree the MPL, as they support collaboration, grant patent use and offer patent protection. Notably missing is the GPL as *"this license cannot be used inside other works under a different license."*[34]

See also

- License compatibility
- Rights Expression Language

References

1. "OSI and License Proliferation (https://fossbazaar.org/content/osi-and-license-proliferation/)" on FOSSBazaar by Martin Michlmayr on August 21st, 2008. "Too many different licenses makes it difficult for licensors to choose: it's difficult to choose a good license for a project because there are so many. Some licenses do not play well together: some open source licenses do not inter-operate well with other open source licenses, making it hard to incorporate code from other projects. Too many licenses makes it difficult to understand what you are agreeing to in a multi-license distribution: since a FOSS application typically

contains code with different licenses and people use many applications which each contain one or several licenses, it's difficult to see what your obligations are."

2. "The Free-Libre / Open Source Software (FLOSS) License Slide (http://www.dwheeler.com/essays/floss-license-slide.html)" by David A. Wheeler on September 27, 2007.
3. Wheeler, David A. (February 16, 2014). "Make Your Open Source Software GPL-Compatible. Or Else" (http://www.dwheeler.com/essays/gpl-compatible.html). Archived (https://web.archive.org/web/20231113231043/https://www.dwheeler.com/essays/gpl-compatible.html) from the original on November 13, 2023.
4. "Various Licenses and Comments about Them (https://www.gnu.org/philosophy/license-list.html)", GNU. Archived (https://web.archive.org/web/20000815065020/https://www.gnu.org/philosophy/license-list.html) 2000-08-15 at the Wayback Machine.
5. Laurent, Philippe (September 24, 2008). "The GPLv3 and compatibility issues" (https://web.archive.org/web/20160304082732/http://www.eolevent.eu/sites/default/files/EOLE%202008%20%E2%80%94%20Philippe%20Laurent%20%E2%80%94%20The%20GPLv3%20and%20Compatibility%20Issues.pdf) (PDF). *European Open source Lawyers Event 2008*. University of Namur – Belgium. p. 7. Archived from the original (http://www.eolevent.eu/sites/default/files/EOLE%202008%20%E2%80%94%20Philippe%20Laurent%20%E2%80%94%20The%20GPLv3%20and%20Compatibility%20Issues.pdf) (PDF) on March 4, 2016. Retrieved May 30, 2015. "*Copyleft is the main source of compatibility problems*"
6. Hanwell, Marcus D. (January 28, 2014). "Should I use a permissive license? Copyleft? Or something in the middle?" (http://opensource.com/business/14/1/what-license-should-i-use-open-source-project). opensource.com. Retrieved May 30, 2015. "*Permissive licensing simplifies things One reason the business world, and more and more developers [...], favor permissive licenses is in the simplicity of reuse. The license usually only pertains to the source code that is licensed and makes no attempt to infer any conditions upon any other component, and because of this there is no need to define what constitutes a derived work. I have also never seen a license compatibility chart for permissive licenses; it seems that they are all compatible.*"
7. "Licence Compatibility and Interoperability" (https://web.archive.org/web/20150617130550/https://joinup.ec.europa.eu/software/page/licence_compatibility_and_interoperability). *Open-Source Software - Develop, share, and reuse open source software for public administrations*. joinup.ec.europa.eu. Archived from the original (https://joinup.ec.europa.eu/software/page/licence_compatibility_and_interoperability) on June 17, 2015. Retrieved May 30, 2015. "*The licences for distributing free or open source software (FOSS) are divided in two families: permissive and copyleft. Permissive licences (BSD, MIT, X11, Apache, Zope) are generally compatible and interoperable with most other licences, tolerating to merge, combine or improve the covered code and to re-distribute it under many licences (including non-free or "proprietary").*"
8. Apache foundation (May 30, 2015). "GPL compatibility" (https://www.apache.org/licenses/GPL-compatibility.html). Retrieved May 30, 2015. "*Apache 2 software can therefore be included in GPLv3 projects, because the GPLv3 license accepts our software into GPLv3 works. However, GPLv3 software cannot be included in Apache projects. The licenses are incompatible in one direction only, and it is a result of ASF's licensing philosophy and the GPLv3 authors' interpretation of copyright law.*"
9. "Frequently Asked Questions about the GNU Licenses – Is GPLv3 compatible with GPLv2?" (https://www.gnu.org/licenses/gpl-faq.html#v2v3Compatibility). gnu.org. Retrieved June 3, 2014. "*No. Some of the requirements in GPLv3, such as the requirement to provide Installation Information, do not exist in GPLv2. As a result, the licenses are not compatible: if you tried to combine code released under both these licenses, you would violate section 6 of GPLv2. However, if code is released under GPL "version 2 or later," that is compatible with GPLv3 because GPLv3 is one of the options it permits.*"

10. Landley, Rob. "CELF 2013 Toybox talk" (http://landley.net/talks/celf-2013.txt). landley.net. Retrieved August 21, 2013. "*GPLv3 broke "the" GPL into incompatible forks that can't share code.*"
11. Asay, Clark D. "Michigan Telecommunications and Technology Law Review Volume 14 - Issue 22008 The General Public License Version 3.0: Making or Breaking the Foss Movement" (http://repository.law.umich.edu/cgi/viewcontent.cgi?article=1074&context=mttlr). law.umich.edu. "*In the end, GPLv3 constitutes license proliferation.*"
12. Nikolai Bezroukov (2000). "Comparative merits of GPL, BSD and Artistic licences (Critique of Viral Nature of GPL v.2 - or In Defense of Dual Licensing Idea)" (https://web.archive.org/web/20011222205401/http://icfcst.kiev.ua/panorama/OSS/bsd_vs_gpl.shtml). Archived from the original (http://icfcst.kiev.ua/panorama/OSS/bsd_vs_gpl.shtml) on December 22, 2001. "*Viral property stimulates proliferation of licenses and contributes to the "GPL-enforced nightmare" -- a situation when many other licenses are logically incompatible with the GPL and make life unnecessary difficult for developers working in the Linux environment (KDE is a good example here, Python is a less known example). I think that this petty efforts to interpret GPL as a "holy text" are non-productive discussion that does not bring us anywhere. And they directly contributed to the proliferation of different "free software" licenses.*"
13. Byfield, Bruce (November 22, 2011). "7 Reasons Why Free Software Is Losing Influence: Page 2" (http://www.datamation.com/open-source/7-reasons-why-free-software-is-losing-influence_2.html). Datamation.com. Retrieved August 23, 2013. "*At the time, the decision seemed sensible in the face of a deadlock. But now, GPLv2 is used for 42.5% of free software, and GPLv3 for less than 6.5%, according to Black Duck Software.*"
14. James E.J. Bottomley; Mauro Carvalho Chehab; Thomas Gleixner; Christoph Hellwig; Dave Jones; Greg Kroah-Hartman; Tony Luck; Andrew Morton; Trond Myklebust; David Woodhouse (September 15, 2006). "Kernel developers' position on GPLv3 - The Dangers and Problems with GPLv3" (https://lwn.net/Articles/200422/). LWN.net. Retrieved March 11, 2015. "[...]since the FSF is proposing to shift all of its projects to GPLv3 and apply pressure to every other GPL licensed project to move, we foresee the release of GPLv3 portends the Balkanisation of the entire Open Source Universe upon which we rely."
15. Ronacher, Armin (July 23, 2013). "Licensing in a Post Copyright World" (http://lucumr.pocoo.org/2013/7/23/licensing/). lucumr.pocoo.org. Retrieved November 18, 2015. "*The License Compatibility Clusterfuck - When the GPL is involved the complexities of licensing becomes a non fun version of a riddle. So many things to consider and so many interactions to consider. And that GPL incompatibilities are still an issue that actively effects people is something many appear to forget. For instance one would think that the incompatibility of the GPLv2 with the Apache Software License 2.0 should be a thing of the past now that everything upgrades to GPLv3, but it turns out that enough people are either stuck with GPLv2 only or do not agree with the GPLv3 that some Apache Software licensed projects are required to migrate. For instance Twitter's Bootstrap is currently migrating from ASL2.0 to MIT precisely because some people still need GPLv2 compatibility. Among those projects that were affected were Drupal, WordPress, Joomla, the MoinMoin Wiki and others. And even that case shows that people don't care that much about licenses any more as Joomla 3 just bundled bootstrap even though they were not licenses in a compatible way (GPLv2 vs ASL 2.0). The other traditional case of things not being GPL compatible is the OpenSSL project which has a license that does not go well with the GPL. That license is also still incompatible with the GPLv3. The whole ordeal is particularly interesting as some not so nice parties have started doing license trolling through GPL licenses.*"
16. Are you sure you want to use the GPL? (http://lucumr.pocoo.org/2009/2/12/are-you-sure-you-want-to-use-gpl/) by Armin Ronacher (2009)
17. Sharing medical software: FOSS licensing in medicine (http://www.freesoftwaremagazine.com/articles/sharing_medical_software_foss_licensing_in_medicine) on freesoftwaremagazine.com by Fred Trotter (2007-06-14)

18. "David A. Wheeler's Blog" (https://dwheeler.com/blog/2008/08/20/). *dwheeler.com*.
19. GitHub finally takes open source licenses seriously (http://www.infoworld.com/article/2611422/open-source-software/github-finally-takes-open-source-licenses-seriously.html) on Infoworld by Simon Phipps on July 2013
20. Choosing an open source license doesn't need to be scary - Which of the following best describes your situation? (https://choosealicense.com/) on choosealicense.com (accessed 2015-11-29)
21. Open source license usage on GitHub.com (https://github.com/blog/1964-license-usage-on-github-com) on March 9, 2015 by Ben Balter on github.com *"MIT 44.69%, [...]GPLv2 12.96%, Apache 11.19%, GPLv3 8.88%"*
22. Ed Burnette (November 2, 2006). "Google says no to license proliferation" (https://www.zdnet.com/article/google-says-no-to-license-proliferation/). *ZDNet*. Archived (https://web.archive.org/web/20070224072515/http://blogs.zdnet.com/Burnette/?p=192) from the original on February 24, 2007. Retrieved September 11, 2010.
23. Greg Stein (May 28, 2009). "Standing Against License Proliferation" (https://web.archive.org/web/20080601023839/http://google-opensource.blogspot.com/2008/05/standing-against-license-proliferation.html). Archived from the original (http://google-opensource.blogspot.com/2008/05/standing-against-license-proliferation.html) on June 1, 2008. Retrieved September 11, 2010.
24. License Proliferation - Less is More, One is Best (https://fossbazaar.org/content/license-proliferation-less-more-one-best/) on January 27th, 2009 by Ernest M. Park *"Chris DiBona from Google suffered the slings and arrows of the OSS community when he rejected the AGPLv3 license for Google Code repository, citing license proliferation as one of the reasons."*
25. Chris DiBona (September 10, 2010). "License Evolution and Hosting Projects on Code.Google.Com" (http://googlecode.blogspot.com/2010/09/license-evolution-and-hosting-projects.html). Retrieved September 11, 2010.
26. OSI Approved Licenses (http://opensource.org/licenses/alphabetical) on opensource.org
27. License Proliferation Project (http://opensource.org/proliferation) on opensource.com (2004)
28. License Proliferation Report (http://opensource.org/proliferation-report) Archived (https://web.archive.org/web/20121212025006/http://opensource.org/proliferation-report) 2012-12-12 at the Wayback Machine on opensource.com (2007)
29. The earliest archived version of the license list reflects this position. Bradley M. Kuhn (August 15, 2000). "Various Licenses and Comments about Them" (https://web.archive.org/web/20000815065020/https://www.gnu.org/philosophy/license-list.html). Free Software Foundation. pp. 37–39. Archived from the original (https://www.gnu.org/philosophy/license-list.html) on August 15, 2000. Retrieved November 29, 2015.
30. How GPLv3 tackles license proliferation (https://archive.today/20070215145800/http://www.linuxdevices.com/articles/AT7188273245.html) on linuxdevices.com
31. Marson, Ingrid (March 31, 2005). "Intel to stop using open-source license" (http://news.cnet.com/Intel-to-stop-using-open-source-license/2100-7344_3-5648518.html). *cnet.com*. CNet. Retrieved October 6, 2014.
32. The Myth of Open Source License Proliferation (https://web.archive.org/web/20111018133015/http://the451group.com/caos/caos_detail.php?icid=843) on the451group.com
33. Open Source License Proliferation: Helpful Diversity or Hopeless Confusion? (http://www.law.washington.edu/Directory/docs/Gomulkiewicz/Open%20Source%20License%20Proliferation.pdf) on law.washington.edu by Robert W. Gomulkiewicz on 2009
34. License compatibility (http://www.osscc.net/en/licenses.html#compatibility) on osscc.net

External links

- Open source license proliferation, a broader view (http://www.ipinfoblog.com/archives/licensing-law-issues-open-source-license-proliferation-a-broader-view.html) by Raymond Nimmer
- Larry Rosen argues that different licenses can be a good thing (https://web.archive.org/web/20070927225757/http://old.linux-foundation.org/newsroom/articles/License_Proliferation.pdf) Larry Rosen
- Licensing howto (http://www.catb.org/~esr/Licensing-HOWTO.html) by Eric S. Raymond
- License proliferation for Medical Software (https://web.archive.org/web/20110927140032/http://www.freesoftwaremagazine.com/columns/sharing_medical_software_foss_licensing_in_medicine) by Fred Trotter (http://www.fredtrotter.com) Advocates that for Health Software, only the Google seven should be used.
- How to choose a license for your own work (https://www.gnu.org/licenses/license-recommendations) Free Software Foundation

Debian–Mozilla trademark dispute

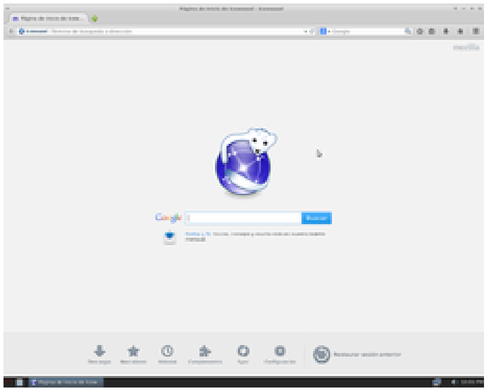
(Redirected from Mozilla software rebranding)

In 2006, a branding issue developed when Mike Connor, representing the Mozilla Corporation, requested that the Debian Project comply with Mozilla standards for use of the Thunderbird trademark when redistributing the Thunderbird software.[1][2] At issue were modifications not approved by the Mozilla Foundation, when the name for the software remained the same.

The Debian Project subsequently rebranded the Mozilla Firefox program,[3] and other software released by Mozilla, so that Debian could distribute modified software without being bound by the trademark requirements that the Mozilla Foundation had invoked. The new names established by Debian were *Iceweasel* for Mozilla Firefox, *Icedove* for Mozilla Thunderbird, and *Iceape* for SeaMonkey. These changes were implemented in the subsequent version of Debian (Etch). In July 2007, *Iceowl*, a rebranded version of Mozilla Sunbird, was added to the unstable branch of Debian.[4]

In 2016, a number of Mozilla employees and Debian maintainers argued that the branding was no longer needed,[5][6] and on 10 March 2016, Debian's unstable branch switched back to the Mozilla branding, with the stable branch planning to switch after Iceweasel's end of life.[7]

Iceweasel



Iceweasel 30 running on Debian "Jessie"

Developer(s)	Debian Project
Operating system	Linux, GNU and variants
Type	Web browser
License	MPL, GNU GPL, GNU LGPL
Website	Debian package page (https://packages.debian.org/jessie/iceweasel)

The decade-long branding issues between the Debian Project and Mozilla Corporation ended in 2017 when all Mozilla applications in Debian were reverted to their original names.[8]

Applications

Debian's Iceweasel, Icedove, Iceowl, and Iceape were based on Mozilla's Firefox, Thunderbird, Sunbird, and SeaMonkey, respectively. The rebranded products still used some Internet-based services from Mozilla, including the Mozilla plugin finder service, and Mozilla add-ons and their update notifications. There was also no change to how non-free components, such as Flash, were found or used.

Iceape

Iceape was a free software Internet suite based on SeaMonkey. It was developed by the Debian Mozilla Team in unison with the SeaMonkey Council's work on their release, but in accordance with Debian's policy of only using free content -- copyrighted artwork and proprietary plug-ins were omitted.

Iceape logo

A temporary lack of community development support resulted in Iceape being absent from the repository for Debian 5 ("Lenny"), though the `iceape-dev` and `iceape-dev-bin` libraries remained available. Support later resumed with Debian 6 ("Squeeze"), and by early 2013, the package was available for both Squeeze and June's release of Debian 7 ("Wheezy").

As of December 2013,[9] the Iceape package was no longer being maintained by the Debian project and users were encouraged to migrate to other packages for security patches.

Icedove

Icedove was the Debian project's rebranded version of the Mozilla Thunderbird e-mail client. In February 2017, Thunderbird packages were reintroduced into the Debian repositories,[10] and on April 20, 2017, the Icedove package in Debian Stable was de-branded back to Thunderbird.[11]

Icedove logo

Iceowl

Iceowl was a calendar application distributed by the Debian project. It was based on Mozilla Sunbird, but was made entirely of free software. The Lightning calendar extension for Mozilla Thunderbird was rebranded as Iceowl Extension.

Iceweasel logo

History

History and origin of name

Mozilla Foundation owns the trademark "Firefox"[12] and claims the right to deny the use of the name and other trademarks to unofficial builds.[13] The Debian Free Software Guidelines are used by the Debian project to determine whether a license is a free license, which in turn is used to determine whether something can be included in Debian. As the logo did not meet these requirements, it could not be used by software which was to be included in Debian. This effect of the Mozilla trademark policy led to a long debate within the Debian Project in 2004 and 2005.

During this debate, the name "Iceweasel" was coined to refer to rebranded versions of Firefox. The first known use of the name in this context is by Nathanael Nerode,[14] in reply to Eric Dorland's suggestion of "Icerabbit".[15] It was intended as a parody of "Firefox".[16] "Iceweasel" was subsequently used as the example name for a rebranded Firefox in the Mozilla Trademark Policy,[17] and became the most commonly used name for a hypothetical rebranded version of Firefox. By January 1, 2005, such strategic rebranding had come to be referred to as the "Iceweasel route".[18]

The term "ice weasel" appeared earlier in a line which Matt Groening fictionally attributed to Friedrich Nietzsche: "Love is a snowmobile racing across the tundra and then suddenly it flips over, pinning you underneath. At night, the ice weasels come."[19]

Debian was initially given permission to use the trademarks, and adopted the Firefox name.[20] However, because the artwork in Firefox had a proprietary copyright license which was not compatible with the Debian Free Software Guidelines, the substituted logo had to remain.[21]

Trademark agreement revocation

In February 2006, Mike Connor, representing the Mozilla Corporation, wrote to the Debian bug tracker and informed the project that Mozilla did not consider the way in which Debian was using the Firefox name to be acceptable.[1] Connor confirmed that the Mozilla Corporation was revoking the previous agreement which allowed Debian to use the Firefox name. Further messages from Mike Connor clarified Mozilla's new trademark policies: usage of the Firefox name is not allowed unless the rest of the branding is used and all of the browser's changes are approved by Mozilla Corporation.

As Debian releases are frozen on a long-term basis, software in the frozen stable releases needs to be patched for any newly discovered security issue. Under the revised guidelines, in order to use the Firefox name, approval from the Mozilla Corporation would have been required for all security patches, but the Debian project felt it could not put its security in the hands of an external corporation in that manner.[22]

The "Iceweasel" name was revived in the Debian community as a possible name to give the rebranded version of Firefox. The Iceweasel found in Debian is not GNU IceWeasel (now GNU IceCat), but rather a rebranded Firefox created by the Debian project. The Debian maintainer has stated that he will "get in touch with [the GNU IceCat/IceWeasel team] to see what we can do together".[23] Similarly, Debian renamed Mozilla Thunderbird and SeaMonkey to Icedove and Iceape, respectively.

Rebranding

According to the *Debian Package Tracking System*,[24] Iceweasel, Icedove, and Iceape were first accepted into the Debian project's *unstable* repository on November 20, 2006, on October 14, 2006, and on December 1, 2006, respectively. Icedove migrated to Etch and Thunderbird was removed on November 11, 2006. Iceape migrated to Etch on January 11, 2007 (the old Mozilla suite having previously been removed on October 6, 2006). Iceweasel migrated (and Firefox was removed) on January 18, 2007. Debian's first stable release to include Iceweasel, Icedove, and Iceape was Debian 4.0 (Etch), released April 8, 2007. Soon after the renaming Debian also replaced Mozilla's unbranded logos with new logos designed to fit with the new names, drawn by Ricardo Fernandez Fuentes.[25][26]

User agent

Some web sites do not recognize the browsers' user agent strings and refuse to work properly.[27] As a workaround, Iceweasel 3.5.5 adds a "like Firefox x.x.x" string to the user agent.[28]

Firefox issue resolution

As described by Sylvestre Ledru in a bug in the Debian bug tracking system,[29] Mozilla and Debian agreed on renaming Iceweasel to Firefox. Mozilla recognizes that Debian potential changes are not impacting the quality of the release.

Debian users who performed a `dist-upgrade` in June 2016 received the following system message. According to Chris Hoffman of *PC World*, "After a decade, Debian and Mozilla are burying the hatchet. Iceweasel is about to re-assume its proper name".[30]

```
iceweasel (45.0esr-1) unstable; urgency=medium

  * The iceweasel package was replaced with the firefox-esr package.

  * Preferences under /etc/iceweasel/prefs will need to be copied manually
    to /etc/firefox-esr.
  * Other customizations under /etc/iceweasel will need additional manual
    steps, through CCK2 or addons.
```

The developers of Parabola GNU/Linux-libre picked up the Iceweasel project[31][32] and it continues to be maintained.[33]

Licensing

The rebranded programs are available under Mozilla's standard MPL/GPL/LGPL tri-license. Like Mozilla, the default icons are under the same tri-license, but unlike Mozilla, there are no trademark restrictions.

See also

Free and open-source software portal

- Firefox version history

References

1. "Debian bug report on use of Mozilla Firefox trademark without permission" (http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=354622).
2. "mozilla thunderbird trademark restrictions / still dfsg free ?" (http://lists.debian.org/debian-legal/2004/12/msg00328.html).
3. "Re: Will IceWeasel be based on a fork or on vanilla FireFox?" (http://lists.debian.org/debian-devel/2006/10/msg00665.html).
4. "Debian Package Tracking System - iceowl" (http://packages.qa.debian.org/i/iceowl.html). *debian.org*. Retrieved 18 February 2016.
5. "#815006 - Renaming Iceweasel to Firefox - Debian Bug report logs" (https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=815006). *debian.org*. Retrieved 18 February 2016.

6. Hoffman, Chris (24 February 2016). " 'Iceweasel' will be renamed 'Firefox' as relations between Debian and Mozilla thaw" (http://www.pcworld.com/article/3036509/linux/iceweasel-will-be-renamed-firefox-as-relations-between-debian-and-mozilla-thaw.html). PC World. Retrieved 28 February 2016.
7. "RIP Iceweasel, 13 Nov 2006 – 10 Mar 2016" (https://glandium.org/blog/?p=3622). 10 March 2016. Retrieved 11 March 2016.
8. "The Great Debian Iceweasel/Icedove Saga Comes to an End | FOSS Force" (http://fossforce.com/2017/02/great-debian-iceweasel-icedove-saga/). *FOSS Force*. 27 February 2017.
9. "DSA-2819-1 iceape - end-of-life announcement for iceape" (http://www.debian.org/security/2013/dsa-2819).
10. "De-Branding of Icedove, reintroducing Thunderbird packages into Debian" (https://lists.debian.org/debian-devel-announce/2017/02/msg00004.html).
11. "Debian Security Advisory DSA-3832-1 icedove" (https://www.debian.org/security/2017/dsa-3832).
12. U.S. Trademark 78,344,043 (http://tarr.uspto.gov/servlet/tarr?regser=serial&entry=78344043); see History of Mozilla Firefox#Naming
13. "Mozilla Trademark Policy" (https://www.mozilla.org/foundation/trademarks/policy.html).
14. "Nathanael Nerode 27 Feb 2004 email to debian-legal" (http://lists.debian.org/debian-legal/2004/02/msg00279.html).
15. "Eric Dorland 27 Feb 2004 email to debian-devel" (http://lists.debian.org/debian-devel/2004/02/msg01876.html).
16. "Gnuzilla Homepage" (https://www.gnu.org/software/gnuzilla/). Gnu.org. 17 October 2010. Retrieved 9 November 2010.
17. "Mozilla Trademark Policy (Version 0.6 (draft))" (https://archive.today/20120723162603/http://viewvc.svn.mozilla.org/vc/projects/mozilla.org/trunk/foundation/trademarks/policy.html?revision=39674&view=markup). Mozilla Foundation. 28 September 2004. Archived from the original (http://viewvc.svn.mozilla.org/vc/projects/mozilla.org/trunk/foundation/trademarks/policy.html?revision=39674&view=markup) on 23 July 2012. Retrieved 12 June 2011.
18. "Joel Aelwyn 01 Jan 2005 email to debian-legal" (http://lists.debian.org/debian-legal/2005/01/msg00010.html).
19. Groening, Matt (1986). *Love Is Hell*. Pantheon Books. ISBN 0-394-74454-3.
20. "Gervase Markham 14 June 2005 email to debian-devel" (http://lists.debian.org/debian-devel/2005/06/msg01182.html).
21. "Gervase Markham 19 June 2005 email to debian-devel" (http://lists.debian.org/debian-legal/2005/01/msg00757.html).
22. "Debian Bug Report #354662 Message #256 (Approval of Security Backports)" (http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=354622#256).
23. "Facts about Debian and Mozilla Firefox" (http://web.glandium.org/blog/?p=97).
24. Debian Package Tracking System (http://packages.qa.debian.org), retrieved 19 July 2013
25. "copyright" file from Iceweasel package in Debian, version 2.0.0.2+dfsg-3
26. Dorland, Eric; Hommey, Mike (13 November 2006). "iceweasel (2.0+dfsg-1) unstable" (http://packages.debian.org/changelogs/pool/main/i/iceweasel/current/changelog#version2.0_dfsg-1). *Debian Changelog iceweasel*. Retrieved 22 November 2011.
27. "Microsoft breaks HotMail for Linux users?" (https://wayback.archive-it.org/all/20090303010012/http://linux-watch.com/news/NS6023147333.html). Linux-watch.com. 7 November 2008. Archived from the original (http://www.linux-watch.com/news/NS6023147333.html) on 3 March 2009. Retrieved 30 January 2011.
28. "#399633 - iceweasel: Please mention Firefox in the user-agent string. - Debian Bug report logs" (http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=399633#102). Bugs.debian.org. Retrieved 9 November 2010.

29. Ledru, Sylvestre. "Renaming Iceweasel to Firefox" (https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=815006). Debian BTS. Retrieved 19 August 2023.
30. Hoffman, Chris. " 'Iceweasel' will be renamed 'Firefox' as relations between Debian and Mozilla thaw" (http://www.pcworld.com/article/3036509/linux/iceweasel-will-be-renamed-firefox-as-relations-between-debian-and-mozilla-thaw.html). PC World. Retrieved 12 June 2016.
31. "IceWeasel History" (https://wiki.parabola.nu/IceWeasel_History). *Parabola Wiki*. Parabola GNU/Linux-libre Hackers. 25 June 2020. Retrieved 8 February 2021. "The Gnuzilla, Connochaetos, and Parabola web browsers however, were and still are, re-configured significantly more than the Debian IceWeasel was, in order to meet the GNU guidelines. That is the primary reason why Gnuzilla, Parabola, and Connochaetos still refer to their web browsers as something other than: "Firefox"."
32. "Iceweasel" (https://wiki.parabola.nu/Iceweasel). *Parabola Wiki*. Parabola GNU/Linux-libre Hackers. 8 April 2020. Retrieved 8 February 2021. "Iceweasel differs technically from Firefox, mainly in that it removes support for DRM and other features that the FSDG considers to be non-free or anti-features, such as DRM and recommendations of non-free add-ons."
33. "Iceweasel development in Parabola's abslibre.git repository" (https://git.parabola.nu/abslibre.git/log/libre/iceweasel).

Proprietary firmware

Proprietary firmware is any firmware that has had its use, private modification, copying, or republishing restricted by the producer. Proprietors may enforce restrictions by technical means, such as by restricting source code access, firmware replacement restrictions (by denying complete tooling that may be necessary in order to recompile and replace the firmware), or by legal means, such as through copyright and patents. Alternatives to proprietary firmware may be free (libre) or open-source.

Distribution

Proprietary firmware (and especially the microcode) is much more difficult to avoid than proprietary software or even proprietary device drivers, because the firmware is usually very specific to the manufacturer of each device (often being unique for each model), and the programming documentation and complete specifications that would be necessary to create a replacement are often withheld by the hardware manufacturer.[1]

Many open-source operating systems reluctantly choose to include proprietary firmware files in their distributions simply to make their device drivers work,[2] because manufacturers try to save money by removing flash memory or EEPROM from their devices, requiring the operating system to upload the firmware each time the device is used.[3] However, in order to do so, the operating system still has to have distribution rights for this proprietary microcode.[3]

Security concerns

Proprietary firmware poses a significant security risk to the user because of the direct memory access (DMA) architecture of modern computers and the potential for DMA attacks. Theo de Raadt of OpenBSD suggests that wireless firmware are kept proprietary because of poor design quality and firmware defects.[4][5] Mark Shuttleworth of Ubuntu suggests that "it's reasonable to assume that all firmware is a cesspool of insecurity courtesy of incompetence of the worst degree from manufacturers, and competence of the highest degree from a very wide range of such agencies".[6]

The security and reliability risks posed by proprietary microcode may be lower than those posed by proprietary device drivers, because the microcode in this context isn't linked against the operating system, and doesn't run on the host's main processor.[2]

Alternatives

Custom firmware may still be available for certain products, which is often free and open-source software, and is especially popular in certain segments of hardware like gaming consoles, wireless routers and Android phones, which are capable of running complete general-purpose operating systems like

Linux, FreeBSD or NetBSD, which are often the systems used by the manufacturer in their original proprietary firmware.

Another potential solution is going with open-source hardware, which goes a step further by also providing schematics for replicating the hardware itself.

Examples

- Breathalysers[7]
- Most BIOS found in IBM-compatible Personal Computers[6]
- Most UEFI found in modern x86 Computers
- ARCS, used in computers from Silicon Graphics
- Run-Time Abstraction Services (RTAS), used in computers from IBM
- The iPod's control menus
- Cisco IOS
- Microcode in wireless network interface controllers, video cards, x86 processors, etc.

See also

- ROM image
- Open-source hardware
- Coreboot
- Open Firmware

References

1. Jeremy Andrews (2005-03-08). "Feature: OpenBSD's "Out of the Box" Wireless Support" (https://web.archive.org/web/20050309035103/http://kerneltrap.org/node/4818). *KernelTrap*. Archived from the original (http://kerneltrap.org/node/4818) on 2005-03-09.
2. Jeremy Andrews (2006-05-02). "Interview: Theo de Raadt" (https://web.archive.org/web/20060603230017/http://kerneltrap.org/node/6550). *KernelTrap*. Archived from the original (http://kerneltrap.org/node/6550) on 2006-06-03.
3. Jeremy Andrews (2004-11-02). "Feature: OpenBSD Works To Open Wireless Chipsets" (https://web.archive.org/web/20060620051155/http://kerneltrap.org/node/4118). *KernelTrap*. Archived from the original (http://kerneltrap.org/node/4118) on 2006-06-20.
4. Theo de Raadt (2016-12-03). "Page 13: The hardware: 802.11 wireless networking (more detail)" (http://www.openbsd.org/papers/opencon06-docs/mgp00013.html). *Open Documentation for Hardware* (http://www.openbsd.org/papers/opencon06-docs/). OpenCON 2006, 2–3 December 2006 (https://web.archive.org/web/20070109032258/http://www.opencon.org/program.php). Courtyard Venice Airport, Venice/Tessera, Italy.
5. Constantine A. Murenin (2006-12-10). "Почему так важно иметь документацию по программированию железа" (https://www.linux.org.ru/news/hardware/1690470). *Linux.org.ru* (in Russian).
6. Mark Shuttleworth (2014-03-17). "ACPI, firmware and your security" (https://www.markshuttleworth.com/archives/1332).
7. "Drunk drivers granted access to breathalyser source code" (https://web.archive.org/web/20080930155902/www.vnunet.com/vnunet/news/2145410/judge-gets-bottom-possible). 2005-11-03. Archived from the original (http://www.vnunet.com/vnunet/news/2145410/judge-gets-bottom-possible) on 2008-09-30.

Proprietary software

Proprietary software is software that grants its creator, publisher, or other rightsholder or rightsholder partner a legal monopoly by modern copyright and intellectual property law to exclude the recipient from freely sharing the software or modifying it, and—in some cases, as is the case with some patent-encumbered and EULA-bound software—from making use of the software on their own, thereby restricting their freedoms.[1]

Proprietary software is a subset of **non-free software**, a term defined in contrast to free and open-source software; non-commercial licenses such as CC BY-NC are not deemed proprietary, but are non-free. Proprietary software may either be **closed-source software** or source-available software.[1][2]

Types

	Free/Open Licenses			**Non-free Licenses**		
	Public domain & equivalents	**Permissive license**	**Copyleft (protective license)**	**Noncommercial license**	**Proprietary license**	**Trade secret**
Software	PD, CC0	BSD, MIT, Apache	GPL, AGPL	JRL, AFPL	proprietary software, no public license	private, internal software
Other creative works	PD, CC0	CC BY	CC BY-SA	CC BY-NC	Copyright, no public license	unpublished

Origin

Until the late 1960s, computers—especially large and expensive mainframe computers, machines in specially air-conditioned computer rooms—were usually leased to customers rather than sold.[3][4] Service and all software available were usually supplied by manufacturers without separate charge until 1969. Computer vendors usually provided the source code for installed software to customers. Customers who developed software often made it available to the public without charge.[5] Closed source means computer programs whose source code is not published except to licensees. It is available to be modified only by the organization that developed it and those licensed to use the software.

In 1969, IBM, which had antitrust lawsuits pending against it, led an industry change by starting to charge separately for mainframe software[6][7] and services, by unbundling hardware and software.[8]

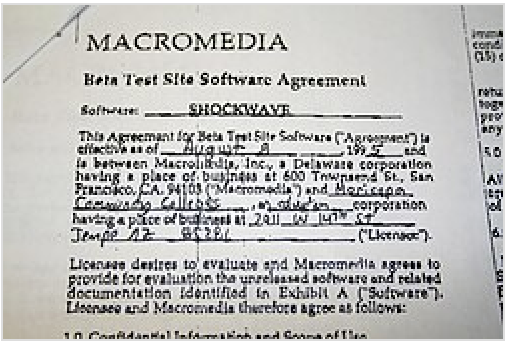
Bill Gates' "Open Letter to Hobbyists" in 1976 decried computer hobbyists' rampant copyright infringement of software, particularly Microsoft's Altair BASIC interpreter, and asserted that their unauthorized use hindered his ability to produce quality software. But the legal status of software copyright, especially for object code, was not clear until the 1983 appeals court ruling in Apple Computer, Inc. v. Franklin Computer Corp.[9][10][11]

According to Brewster Kahle the legal characteristic of software changed also due to the U.S. Copyright Act of 1976.[12]

Starting in February 1983 IBM adopted an "object-code-only" model for a growing list of their software and stopped shipping much of the source code,[13][14] even to licensees.

In 1983, binary software became copyrightable in the United States as well by the Apple vs. Franklin law decision,[15] before which only source code was copyrightable.[16] Additionally, the growing availability of millions of computers based on the same microprocessor architecture created for the first time an unfragmented and big enough market for binary distributed software.[16]

Licenses


MACROMEDIA

Beta Test Site Software Agreement

Software: SHOCKWAVE

This Agreement for Beta Test Site Software ("Agreement") is effective as of August 8, 1995 and is between Macromedia, Inc., a Delaware corporation having a place of business at 600 Townsend St., San Francisco, CA 94103 ("Macromedia") and Maricopa Community Colleges, an Arizona corporation having a place of business at 2411 W 14th St Tempe AZ 85281 ("Licensee").

Licensee desires to evaluate and Macromedia agrees to provide for evaluation the unreleased software and related documentation identified in Exhibit A ("Software"). Licensee and Macromedia therefore agree as follows:

A brief, written-out beta test software license issued by Macromedia in 1995

The tendency to license proprietary software, rather than sell it, dates from the time period before the existence, then the scope of software copyright protection was clear. These licenses have continued in use after software copyright was recognized in the courts, and are considered to grant the company extra protection compared to copyright law.[17] According to United States federal law, a company can restrict the parties to which it sells but it cannot prevent a buyer from reselling the product. Software licensing agreements usually prohibit resale, enabling the company to maximize revenue.[18]

Traditionally, software was distributed in the form of binary object code that could not be understood or modified by the user,[19] but could be downloaded and run. The user bought a perpetual license to use a particular version of the software.[20] Software as service (SaaS) vendors—who have the majority market share in application software as of 2023[21]—rarely offer perpetual licenses.[22] SaaS licenses are usually temporary and charged on a pay-per-usage or subscription basis,[23] although other revenue models such as freemium are also used.[24] For customers, the advantages of temporary licenses include reduced upfront cost, increased flexibility, and lower overall cost compared to a perpetual license.[20] In some cases, the steep one-time cost demanded by sellers of traditional software were out of the reach of smaller businesses, but pay-per-use SaaS models makes the software affordable.[25]

Mixed-source software

Software distributions considered as proprietary may in fact incorporate a "mixed source" model including both free and non-free software in the same distribution.[26] Most if not all so-called proprietary UNIX distributions are mixed source software, bundling open-source components like BIND, Sendmail, X Window System, DHCP, and others along with a purely proprietary kernel and system utilities.[27][28]

Multi-licensing

Some free software packages are also simultaneously available under proprietary terms. Examples include MySQL, Sendmail and ssh. The original copyright holders for a work of free software, even copyleft free software, can use dual-licensing to allow themselves or others to redistribute proprietary versions. Non-copyleft free software (i.e. software distributed under a permissive free software license or released to the public domain) allows anyone to make proprietary redistributions.[29][30] Free software that depends on proprietary software is considered "trapped" by the Free Software Foundation. This includes software written only for Microsoft Windows,[31] or software that could only run on Java, before it became free software.[32]

Legal basis

Most of the software is covered by copyright which, along with contract law, patents, and trade secrets, provides legal basis for its owner to establish exclusive rights.[33]

A software vendor delineates the specific terms of use in an end-user license agreement (EULA). The user may agree to this contract in writing, interactively on screen (clickwrap), or by opening the box containing the software (shrink wrap licensing). License agreements are usually not negotiable.[34] Software patents grant exclusive rights to algorithms, software features, or other patentable subject matter, with coverage varying by jurisdiction. Vendors sometimes grant patent rights to the user in the license agreement.[35] The source code for a piece of proprietary software is routinely handled as a trade secret.[36] Software can be made available with fewer restrictions on licensing or source-code access; software that satisfies certain conditions of freedom and openness is known as "free" or "open-source."[37]

Limitations

Since license agreements do not override applicable copyright law or contract law, provisions in conflict with applicable law are not enforceable.[38] Some software is specifically licensed and not sold, in order to avoid limitations of copyright such as the first-sale doctrine.[39]

Exclusive rights

The owner of proprietary software exercises certain exclusive rights over the software. The owner can restrict the use, inspection of source code, modification of source code, and redistribution.

Use of the software

Vendors typically limit the number of computers on which software can be used, and prohibit the user from installing the software on extra computers. Restricted use is sometimes enforced through a technical measure, such as product activation, a product key or serial number, a hardware key, or copy protection.

Vendors may also distribute versions that remove particular features, or versions which allow only certain fields of endeavor, such as non-commercial, educational, or non-profit use.

Use restrictions vary by license:

- Windows Vista Starter is restricted to running a maximum of three concurrent applications.
- The retail edition of Microsoft Office Home and Student 2007 is limited to non-commercial use on up to three devices in one household.
- Windows XP can be installed on one computer, and limits the number of network file sharing connections to 10.[40] The Home Edition disables features present in Windows XP Professional.
- Traditionally, Adobe licenses are limited to one user, but allow the user to install a second copy on a home computer or laptop.[41] This is no longer true with the switching to Creative Cloud.
- iWork '09, Apple's productivity suite, is available in a five-user family pack, for use on up to five computers in a household.[42]

Inspection and modification of source code

Vendors typically distribute proprietary software in compiled form, usually the machine language understood by the computer's central processing unit. They typically retain the source code, or human-readable version of the software, often written in a higher level programming language.[43] This scheme is often referred to as closed source.[44]

While most proprietary software is distributed without the source code, some vendors distribute the source code or otherwise make it available to customers. For example, users who have purchased a license for the Internet forum software vBulletin can modify the source for their own site but cannot redistribute it. This is true for many web applications, which must be in source code form when being run by a web server. The source code is covered by a non-disclosure agreement or a license that allows, for example, study and modification, but not redistribution.[45] The text-based email client Pine and certain implementations of Secure Shell are distributed with proprietary licenses that make the source code available.Some licenses for proprietary software allow distributing changes to the source code, but only to others licensed for the product, and some[46] of those modifications are eventually picked up by the vendor.

Some governments fear that proprietary software may include defects or malicious features which would compromise sensitive information. In 2003 Microsoft established a Government Security Program (GSP) to allow governments to view source code and Microsoft security documentation, of which the Chinese government was an early participant.[47][48] The program is part of Microsoft's broader Shared Source Initiative which provides source code access for some products. The Reference Source License (Ms-RSL) and Limited Public License (Ms-LPL) are proprietary software licenses where the source code is made available.

Governments have also been accused of adding such malware to software themselves. According to documents released by Edward Snowden, the NSA has used covert partnerships with software companies to make commercial encryption software exploitable to eavesdropping, or to insert backdoors.[49][50]

Software vendors sometimes use obfuscated code to impede users who would reverse engineer the software.[51] This is particularly common with certain programming languages. For example, the bytecode for programs written in Java can be easily decompiled to somewhat usable code, and the source code for programs written in scripting languages such as PHP or JavaScript is available at run time.[52]

Redistribution

Proprietary software vendors can prohibit the users from sharing the software with others. Another unique license is required for another party to use the software.

In the case of proprietary software with source code available, the vendor may also prohibit customers from distributing their modifications to the source code.

Shareware is closed-source software whose owner encourages redistribution at no cost, but which the user sometimes must pay to use after a trial period. The fee usually allows use by a single user or computer. In some cases, software features are restricted during or after the trial period, a practice sometimes called crippleware.

Interoperability with software and hardware

Proprietary file formats and protocols

Proprietary software often stores some of its data in file formats that are incompatible with other software, and may also communicate using protocols which are incompatible. Such formats and protocols may be restricted as trade secrets or subject to patents.

Proprietary APIs

A proprietary application programming interface (API) is a software library interface "specific to one device or, more likely to a number of devices within a particular manufacturer's product range."[53] The motivation for using a proprietary API can be vendor lock-in or because standard APIs do not support the device's functionality.[53]

The European Commission, in its March 24, 2004, decision on Microsoft's business practices,[54] quotes, in paragraph 463, Microsoft general manager for C++ development Aaron Contorer as stating in a February 21, 1997, internal Microsoft memo drafted for Bill Gates:

> The Windows API is so broad, so deep, and so functional that most ISVs would be crazy not to use it. And it is so deeply embedded in the source code of many Windows apps that there is a huge switching cost to using a different operating system instead.

Early versions of the iPhone SDK were covered by a non-disclosure agreement. The agreement forbade independent developers from discussing the content of the interfaces. Apple discontinued the NDA in October 2008.[55]

Vendor lock-in

Any dependency on the future versions and upgrades for a proprietary software package can create *vendor lock-in*, entrenching a monopoly position.[56]

Software limited to certain hardware configurations

Proprietary software may also have licensing terms that limit the usage of that software to a specific set of hardware. Apple has such a licensing model for macOS, an operating system which is limited to Apple hardware, both by licensing and various design decisions. This licensing model has been affirmed by the United States Court of Appeals for the Ninth Circuit.[57]

Abandonment by proprietors

Proprietary software which is no longer marketed, supported or sold by its owner is called abandonware, the digital form of orphaned works. If the proprietor of a software package should cease to exist, or decide to cease or limit production or support for a proprietary software package, recipients and users of the package may have no recourse if problems are found with the software. Proprietors can fail to improve and support software because of business problems.[58] Support for older or existing versions of a software package may be ended to force users to upgrade and pay for newer versions[59](planned obsolescence). Sometimes another vendor or a software's community themselves can provide support for the software, or the users can migrate to either competing systems with longer support life cycles or to FOSS-based systems.[60]

Some proprietary software is released by their owner at end-of-life as open-source or source available software, often to prevent the software from becoming unsupported and unavailable abandonware.[61][62][63] 3D Realms and id Software are famous for the practice of releasing closed source software into the open source. Some of those kinds are free-of-charge downloads (freeware), some are still commercially sold (e.g. Arx Fatalis). More examples of formerly closed-source software in the List of commercial software with available source code and List of commercial video games with available source code.

Pricing and economics

Proprietary software is not synonymous with commercial software,[64][65] although the two terms are sometimes used synonymously in articles about free software.[66][67] Proprietary software can be distributed at no cost or for a fee, and free software can be distributed at no cost or for a fee.[68] The difference is that whether proprietary software can be distributed, and what the fee would be, is at the proprietor's discretion. With free software, anyone who has a copy can decide whether, and how much, to charge for a copy or related services.[69]

Proprietary software that comes for no cost is called freeware.

Proponents of commercial proprietary software argue that requiring users to pay for software as a product increases funding or time available for the research and development of software. For example, Microsoft says that per-copy fees maximize the profitability of software development.[70]

Proprietary software generally creates greater commercial activity over free software, especially in regard to market revenues.[71] Proprietary software is often sold with a license that gives the end user right to use the software.

Technical support for proprietary software can often be provided only by employees of the company that created the program and such service is included with the software. However, a dedicated technical support system increases the cost of software maintenance, which has an impact on its price.[72]

See also

- Business software
- Commercial off-the-shelf
- Comparison of open-source and closed-source software
- Proprietary hardware
- Retail software
- Enshittification

References

1. Saraswati Experts. "2.5.3" (https://books.google.com/books?id=OtlBDAAAQBAJ&pg=SA1-PA31). *COMPUTER SCIENCE WITH C++*. Saraswati House Pvt Ltd. p. 1.27. ISBN 978-93-5199-877-8. Retrieved 29 June 2017.
2. Brendan Scott (March 2003). "Why Free Software's Long Run TCO must be lower" (https://books.google.com/books?id=C55YaWmDBiAC&pg=PA51). *AUUGN*. **24** (1). AUUG, Inc. 1. Definitions. Retrieved 29 June 2017.
3. Ceruzzi, Paul E. (2003). *A History of Modern Computing* (https://archive.org/details/historyofmodernc00ceru_0/page/128). Cambridge, MA: MIT Press. p. 128 (https://archive.org/details/historyofmodernc00ceru_0/page/128). ISBN 0-262-53203-4. "Although IBM agreed to sell its machines as part of a Consent Decree effective January 1956, leasing continued to be its preferred way of doing business."
4. "The History of Equipment Leasing" (https://web.archive.org/web/20080411024345/http://www.leasegenie.com/History_of_Leasing.html), *Lease Genie*, archived from the original (http://www.leasegenie.com/History_of_Leasing.html) on April 11, 2008, retrieved November 12, 2010, "In the 1960s, IBM and Xerox recognized that substantial sums could be made from the financing of their equipment. The leasing of computer and office equipment that occurred then was a significant contribution to leasings [*sic*] growth, since many companies were exposed to equipment leasing for the first time when they leased such equipment."
5. "Overview of the GNU System" (https://www.gnu.org/gnu/gnu-history.en.html). *GNU Operating System*. Free Software Foundation. 2016-06-16. Retrieved 2017-05-01.
6. Pugh, Emerson W. (2002). "Origins of Software Bundling". *IEEE Annals of the History of Computing*. **24** (1): 57–58. doi:10.1109/85.988580 (https://doi.org/10.1109%2F85.988580).
7. Hamilton, Thomas W. (1969). *IBM's Unbundling Decision: Consequences for Users and the Industry*. Programming Sciences Corporation.
8. "Chronological History of IBM: 1960s" (https://web.archive.org/web/20160703062451/http://www-03.ibm.com/ibm/history/history/decade_1960.html). IBM. n.d. Archived from the original (http://www-03.ibm.com/ibm/history/history/decade_1960.html) on July 3, 2016. Retrieved May 28, 2016. "Rather than offer hardware, services and software exclusively in packages, marketers **'unbundled'** the components and offered them for sale individually. Unbundling gave birth to the multibillion-dollar software and services industries, of which IBM is today a world leader."
9. Gates, Bill (February 3, 1976). "An Open Letter to Hobbyists" (https://upload.wikimedia.org/wikipedia/commons/1/14/Bill_Gates_Letter_to_Hobbyists.jpg). Retrieved May 28, 2016.

10. Swann, Matthew (18 November 2004). *Executable Code is Not the Proper Subject of Copyright Law* (Technical report). Cal Poly State University. CPSLO-CSC-04-02.
11. Pamela Samuelson (Sep 1984), "CONTU Revisited: The Case against Copyright Protection for Computer Programs in Machine-Readable Form" (https://web.archive.org/web/20170804014725/http://scholarship.law.berkeley.edu:80/facpubs/333/), *Duke Law Journal*, **1984** (4): 663–769, doi:10.2307/1372418 (https://doi.org/10.2307%2F1372418), JSTOR 1372418 (https://www.jstor.org/stable/1372418), archived from the original (https://scholarship.law.berkeley.edu/facpubs/333) on Aug 4, 2017
12. Robert X. Cringely. *Cringely's interview with Brewster Kahle* (https://web.archive.org/web/20190118221148/https://www.youtube.com/watch?v=-1jUr0JrYEk&gl=US&hl=en). *YouTube*. 46 minutes in. Archived from the original (https://www.youtube.com/watch?v=-1jUr0JrYEk) on 2019-01-18.
13. Cantrill, Bryan (2014-09-17). *Corporate Open Source Anti-patterns* (https://ghostarchive.org/varchive/youtube/20211027/Pm8P4oCIY3g). *YouTube*. Event occurs at 3:15. Archived from the original (https://www.youtube.com/watch?v=Pm8P4oCIY3g&t=3m15s) on 2021-10-27. Retrieved 2015-12-26.
14. Gallant, John (1985-03-18). "IBM policy draws fire - Users say source code rules hamper change" (https://books.google.com/books?id=4Wgmey4obagC&q=1983object-only+model+IBM&pg=PA8). Computerworld. Retrieved 2015-12-27. "While IBM's policy of withholding source code for selected software products has already marked its second anniversary, users are only now beginning to cope with the impact of that decision. But whether or not the advent of object-code-only products has affected their day-to-day DP operations, some users remain angry about IBM's decision. Announced in February 1983, IBM's object-code-only policy has been applied to a growing list of Big Blue system software products"
15. Hassett, Rob (Dec 18, 2012). "Impact of Apple vs. Franklin Decision" (http://www.internetlegal.com/impact-of-apple-vs-franklin-decision/). *InternetLegal.com*. Archived (https://web.archive.org/web/20230908092517/https://internetlegal.com/impact-of-apple-vs-franklin-decision/) from the original on Sep 8, 2023.
16. Landley, Rob (2009-05-23). "May 23, 2009" (https://landley.net/notes-2009.html#23-05-2009). landley.net. Retrieved 2024-06-22. "So if open source used to be the norm back in the 1960's and 70's, how did this _change_? Where did proprietary software come from, and when, and how? How did Richard Stallman's little utopia at the MIT AI lab crumble and force him out into the wilderness to try to rebuild it? Two things changed in the early 80's: the exponentially growing installed base of microcomputer hardware reached critical mass around 1980, and a legal decision altered copyright law to cover binaries in 1983. Increasing volume: The microprocessor creates millions of identical computers"
17. Terasaki 2013, p. 469.
18. Terasaki 2013, pp. 469–470.
19. Boyle 2003, p. 45.
20. Clohessy *et al.* 2020, pp. 40–41.
21. Watt 2023, p. 4.
22. Dempsey & Kelliher 2018, p. 48.
23. Dempsey & Kelliher 2018, pp. 48, 57.
24. Dempsey & Kelliher 2018, pp. 61–63.
25. Dempsey & Kelliher 2018, p. 2.
26. Engelfriet, Arnoud (August–September 2006). "The best of both worlds" (https://web.archive.org/web/20130914013626/http://www.iam-magazine.com/issues/Article.ashx?g=64d0a423-1249-4de3-929c-91b57c15f702). *Intellectual Asset Management (IAM)* (19). Gavin Stewart. Archived from the original (http://www.iam-magazine.com/issues/article.ashx?g=64d0a423-1249-4de3-929c-91b57c15f702) on 2013-09-14. Retrieved 2008-05-19.

27. Loftus, Jack (2007-02-19). "Managing mixed source software stacks" (https://web.archive.org/web/20100603162347/http://searchenterpriselinux.techtarget.com/news/article/0,289142,sid39_gci1244277,00.html). LinuxWorld. Archived from the original (http://searchenterpriselinux.techtarget.com/news/article/0,289142,sid39_gci1244277,00.html) on 2010-06-03.
28. Tan, Aaron (2006-12-28). "Novell: We're a 'mixed-source' company" (http://www.zdnetasia.com/news/software/0,39044164,61977995,00.htm). CNET Networks, Inc.
29. Rosenberg, Donald (2000). *Open Source: The Unauthorized White Papers* (https://archive.org/details/opensourceunauth00rose/page/109). Foster City: IDG. p. 109 (https://archive.org/details/opensourceunauth00rose/page/109). ISBN 0-7645-4660-0.
30. "Categories of Free and Non-Free Software" (https://www.gnu.org/philosophy/categories.html#PublicDomainSoftware). GNU Project.
31. Free Software Foundation (2009-05-05). "Frequently Asked Questions about the GNU Licenses" (https://www.gnu.org/licenses/gpl-faq.html#WindowsRuntimeAndGPL). Retrieved 2017-05-01.
32. Richard Stallman (2004-04-12). "Free But Shackled - The Java Trap" (https://www.gnu.org/philosophy/java-trap.html). Retrieved 2017-05-01.
33. Liberman, Michael (1995). "Overreaching Provisions in Software License Agreements" (http://jolt.richmond.edu/v1i1/liberman.html). *Richmond Journal of Law and Technology*. **1**: 4. Retrieved November 29, 2011.
34. Limitations and Exceptions to Copyright and Neighbouring Rights in the Digital Environment: An International Library Perspective (2004) (http://www.ifla.org/publications/limitations-and-exceptions-to-copyright-and-neighbouring-rights-in-the-digital-environm). IFLA (2013-01-22). Retrieved on 2013-06-16.
35. Daniel A. Tysver (2008-11-23). "Why Protect Software Through Patents" (http://www.bitlaw.com/software-patent/why-patent.html). Bitlaw. Retrieved 2009-06-03. "In connection with the software, an issued patent may prevent others from utilizing a certain algorithm (such as the GIF image compression algorithm) without permission, or may prevent others from creating software programs that perform a function in a certain way. In connection with computer software, copyright law can be used to prevent the total duplication of a software program, as well as the copying of a portion of software code."
36. Donovan, S. (1994). "Patent, copyright and trade secret protection for software". *IEEE Potentials*. **13** (3): 20. doi:10.1109/45.310923 (https://doi.org/10.1109%2F45.310923). S2CID 19873766 (https://api.semanticscholar.org/CorpusID:19873766). "Essentially there are only three ways to protect computer software under the law: patent it, register a copyright for it, or keep it as a trade secret."
37. Eben Moglen (2005-02-12). "Why the FSF gets copyright assignments from contributors" (http://www.gnu.org/licenses/why-assign.html). Retrieved 2017-05-01. "Under US copyright law, which is the law under which most free software programs have historically been first published, [...] only the copyright holder or someone having assignment of the copyright can enforce the license."
38. White, Aoife (2012-07-03). "Oracle Can't Stop Software License Resales, EU Court Says" (https://www.bloomberg.com/news/2012-07-03/oracle-can-t-stop-software-license-resales-eu-court-says-1-.html). *Bloomberg*.
39. Microsoft Corporation (2005-04-01). "End-User License Agreement for Microsoft Software: Microsoft Windows XP Professional Edition Service Pack 2" (http://www.microsoft.com/about/legal/useterms/) (PDF). *Microsoft*. p. Page 3. Retrieved 2009-04-29.

40. Microsoft Corporation (2005-04-01). "End-User License Agreement for Microsoft Software: Microsoft Windows XP Professional Edition Service Pack 2" (http://www.microsoft.com/about/legal/useterms/) (PDF). *Microsoft*. p. Page 1. Retrieved 2009-04-29. "You may install, use, access, display and run one copy of the Software on a single computer, such as a workstation, terminal or another device ("Workstation Computer"). The Software may not be used by more than two (2) processors at any one time on any single Workstation Computer. ... You may permit a maximum of ten (10) computers or other electronic devices (each a 'Device') to connect to the Workstation Computer to utilize one or more of the following services of the Software: File Services, Print Services, Internet Information Services, Internet Connection Sharing and telephony services."
41. Adobe Systems, *Adobe Software License Agreement* (https://www.adobe.com/products/eulas/pdfs/gen_wwcombined_20091001_1604.pdf) (PDF), retrieved 2010-06-09
42. Parker, Jason (January 27, 2009). "Apple iWork '09 review: Apple iWork '09" (https://www.cnet.com/reviews/apple-iwork-09-review/). *CNET*. Retrieved May 2, 2022.
43. Heffan, Ira V. (1997). "Copyleft: Licensing Collaborative Works in the Digital Age" (https://web.archive.org/web/20130514171754/http://www.open-bar.org/docs/copyleft.pdf) (PDF). *Stanford Law Review*. **49** (6): 1490. doi:10.2307/1229351 (https://doi.org/10.2307%2F1229351). JSTOR 1229351 (https://www.jstor.org/stable/1229351). Archived from the original (http://www.open-bar.org/docs/copyleft.pdf) (PDF) on 2013-05-14. Retrieved 2009-07-27. "Under the proprietary software model, most software developers withhold their source code from users."
44. David A. Wheeler (2009-02-03). "Free-Libre / Open Source Software (FLOSS) is Commercial Software" (http://www.dwheeler.com/essays/commercial-floss.html). Retrieved 2009-06-03.
45. "Distribution of IBM Licensed Programs and Licensed Program Materials and Modified Agreement for IBM Licensed Programs" (http://www.landley.net/history/mirror/ibm/oco.html). *Announcement Letters*. IBM. February 8, 1983. 283-016.
46. Greg Mushial (July 20, 1983), "Module 24: SLAC Enhancements to and Beautifications of the IBM H-Level Assembler for Version 2.8" (https://www.gsf-soft.com/Documents/SLAC-MODS.html), *SLAC VM NOTEBOOK*, Stanford Linear Accelerator Center
47. Shankland, Stephen (January 30, 2003). "Governments to see Windows code" (https://www.cnet.com/tech/tech-industry/governments-to-see-windows-code/). CNET. Retrieved May 2, 2022.
48. Gao, Ken (February 28, 2003). "China to view Windows code" (https://www.cnet.com/tech/tech-industry/china-to-view-windows-code/). CNET. Retrieved May 2, 2022.
49. James Ball, Julian Borger and Glenn Greenwald (2013-09-06). "US and UK spy agencies defeat privacy and security on the internet" (https://www.theguardian.com/world/2013/sep/05/nsa-gchq-encryption-codes-security). *The Guardian*.
50. Bruce Schneier (2013-09-06). "How to remain secure against NSA surveillance" (https://www.theguardian.com/world/2013/sep/05/nsa-how-to-remain-secure-surveillance). *The Guardian*.
51. Jacob, Matthias; Boneh, Dan; Felten, Edward (30 October 2003). "Attacking an Obfuscated Cipher by Injecting Faults" (https://archive.org/details/springer_10.1007-b11725/page/n26/mode/1up). In Feigenbaum, Joan (ed.). *Digital Rights Management: ACM CCS-9 Workshop, DRM 2002, Washington, DC, USA, November 18, 2002, Revised Papers* (https://archive.org/details/springer_10.1007-b11725). Second International Workshop on Digital Rights Management. Lecture Notes in Computer Science. Vol. 2696. Springer Berlin Heidelberg. p. 17. ISBN 978-3-540-44993-5. Retrieved 12 January 2024 – via Internet Archive.

52. Tony Patton (2008-11-21). "Protect your JavaScript with obfuscation" (https://archive.today/20140315082054/http://www.techrepublic.com/blog/software-engineer/protect-your-javascript-with-obfuscation/). *TechRepublic*. Archived from the original (https://www.techrepublic.com/article/protect-your-javascript-with-obfuscation/) on March 15, 2014. Retrieved May 2, 2022. "While the Web promotes the sharing of such code, there are times when you or a client may not want to share their JavaScript code. This may be due to the sensitive nature of data within the code, proprietary calculations, or any other scenario."
53. Orenstein, David (January 10, 2000). "Application Programming Interface" (https://www.computerworld.com/article/2593623/application-programming-interface.html). *Computerworld*. Retrieved May 2, 2022.
54. "Commission Decision of 24.03.2004 relating to a proceeding under Article 82 of the EC Treaty (Case COMP/C-3/37.792 Microsoft)" (https://web.archive.org/web/20081028213407/http://ec.europa.eu/comm/competition/antitrust/cases/decisions/37792/en.pdf) (PDF). European Commission. March 24, 2004. Archived from the original (http://ec.europa.eu/comm/competition/antitrust/cases/decisions/37792/en.pdf) (PDF) on October 28, 2008. Retrieved June 17, 2009.
55. Wilson, Ben (2008-10-01). "Apple Drops NDA for Released iPhone Software" (https://web.archive.org/web/20130308181607/http://reviews.cnet.com/8301-19512_7-10115774-233.html). *CNET*. Archived from the original (https://www.cnet.com/tech/mobile/apple-drops-nda-for-released-iphone-software/) on 2013-03-08. Retrieved 2022-05-02.
56. The Linux Information Project (2006-04-29). "Vendor Lock-in Definition" (http://www.linfo.org/vendor_lockin.html). Retrieved 2009-06-11. "Vendor lock-in, or just lock-in, is the situation in which customers are dependent on a single manufacturer or supplier for some product [...] This dependency is typically a result of standards that are controlled by the vendor [...] It can grant the vendor some extent of monopoly power [...] The best way for an organization to avoid becoming a victim of vendor lock-in is to use products that conform to free, industry-wide standards. Free standards are those that can be used by anyone and are not controlled by a single company. In the case of computers, this can usually be accomplished by using free software rather than proprietary software (i.e., commercial software)."
57. Don Reisinger (2011-09-29). "Apple wins key battle against Psystar over Mac clones" (https://www.cnet.com/home/smart-home/apple-wins-key-battle-against-psystar-over-mac-clones/). Retrieved 2022-05-02.
58. "What happens when a proprietary software company dies?" (https://www.linux.com/news/what-happens-when-proprietary-software-company-dies/). Linux. October 24, 2003. Retrieved May 2, 2022.
59. Livingston, Brian (December 15, 2006). "Microsoft Turns Up The Heat On Windows 2000 Users" (https://web.archive.org/web/20220503022710/https://www.crn.com/news/channel-programs/196700124/microsoft-turns-up-the-heat-on-windows-2000-users.htm). *CRN*. Archived from the original (https://www.crn.com/news/channel-programs/196700124/microsoft-turns-up-the-heat-on-windows-2000-users.htm) on May 3, 2022. Retrieved May 2, 2022.
60. Cassia, Fernando (March 28, 2007). "Open Source, the only weapon against 'planned obsolescence' " (https://web.archive.org/web/20121122142811/http://www.theinquirer.net/inquirer/news/1001739/open-source-weapon-planned-obsolescence). *The Inquirer*. Archived from the original (http://www.theinquirer.net/inquirer/news/1001739/open-source-weapon-planned-obsolescence) on November 22, 2012. Retrieved August 2, 2012.

61. Bell, John (October 1, 2009). "Opening the Source of Art" (https://web.archive.org/web/20140330084636/http://timreview.ca/article/294). Technology Innovation Management Review. Archived from the original (http://www.johnpbell.com/opening-the-source-of-art/) on March 30, 2014. Retrieved May 2, 2022. "that no further patches to the title would be forthcoming. The community was predictably upset. Instead of giving up on the game, users decided that if Activision wasn't going to fix the bugs, they would. They wanted to save the game by getting Activision to open the source so it could be kept alive beyond the point where Activision lost interest. With some help from members of the development team that were active on fan forums, they were eventually able to convince Activision to release Call to Power II's source code in October of 2003."
62. Wen, Howard (June 10, 2004). "Keeping the Myths Alive" (https://web.archive.org/web/20130406161344/http://www.linuxdevcenter.com/pub/a/linux/2004/06/10/mythdevelopers.html). Linux Dev Center. Archived from the original (http://linuxdevcenter.com/pub/a/linux/2004/06/10/mythdevelopers.html) on April 6, 2013. Retrieved December 22, 2012. "fans of the Myth trilogy have taken this idea a step further: they have official access to the source code for the Myth games. Organized under the name MythDevelopers, this all-volunteer group of programmers, artists, and other talented people devote their time to improving and supporting further development of the Myth game series."
63. Largent, Andy (October 8, 2003). "Homeworld Source Code Released" (https://web.archive.org/web/20131012012745/http://www.insidemacgames.com/news/story.php?ArticleID=8516). Inside Mac Games. Archived from the original (http://www.insidemacgames.com/news/story.php?ArticleID=8516) on October 12, 2013. Retrieved November 24, 2012. "With the release of Homeworld 2 for the PC, Relic Entertainment has decided to give back to their impressive fan community by releasing the source code to the original Homeworld."
64. Rosen, Lawrence (2004). *Open Source Licensing* (https://archive.org/details/opensourcelicens00rose_0). Upper Saddle River: Prentice Hall. pp. 52 (https://archive.org/details/opensourcelicens00rose_0/page/52), 255, 259. ISBN 978-0-13-148787-1.
65. Havoc Pennington (2008-03-02). "Debian Tutorial" (https://web.archive.org/web/20180129072039/https://www.debian.org/doc/manuals/debian-tutorial/). Archived from the original (http://www.debian.org/doc/manuals/debian-tutorial/) on 2018-01-29. Retrieved 2009-06-04. "It is important to distinguish commercial software from proprietary software. Proprietary software is non-free software, while commercial software is software sold for money."
66. Russell McOrmond (2000-01-04). "What is "Commercial Software"?" (https://web.archive.org/web/20121004013215/http://www.linuxtoday.com/developer/2000010400505NWSM). Archived from the original (http://www.linuxtoday.com/developer/2000010400505NWSM) on 2012-10-04. Retrieved 2009-05-02.
67. Michael K. Johnson (1996-09-01). "Licenses and Copyright" (http://www.linuxjournal.com/article/1297). Retrieved 2009-06-16. "If you program for Linux, you do need to understand licensing, no matter if you are writing free software or commercial software."
68. Eric S. Raymond (2003-12-29). "*Proprietary*, Jargon File" (http://www.catb.org/~esr/jargon/html/P/proprietary.html). Retrieved 2009-06-12. "Proprietary software should be distinguished from commercial software. It is possible for the software to be commercial [...] without being proprietary. The reverse is also possible, for example in binary-only freeware."
69. "Selling Free Software" (https://www.gnu.org/philosophy/selling.html). GNU Project.
70. "The Commercial Software Model" (https://web.archive.org/web/20070305010226/http://www.microsoft.com/presspass/exec/craig/05-03sharedsource.mspx). Microsoft. May 2001. Archived from the original (http://www.microsoft.com/presspass/exec/craig/05-03sharedsource.mspx) on 2007-03-05.
71. *Open Source Versus Commercial Software: Why Proprietary Software is Here to Stay* (https://www.informit.com/articles/article.aspx?p=420290). Sams Publishing. October 2005. Retrieved 2022-05-02.
72. "Proprietary software | Definition, History, & Facts | Britannica" (https://www.britannica.com/technology/proprietary-software). *www.britannica.com*. Retrieved 2025-04-24.

External links

- Media related to Proprietary software at Wikimedia Commons

Wikipedia
The Free Encyclopedia

Software patent

A **software patent** is a patent on a piece of software, such as a computer program, library, user interface, or algorithm. The validity of these patents can be difficult to evaluate, as software is often at once a product of engineering, something typically eligible for patents, and an abstract concept, which is typically not. This gray area, along with the difficulty of patent evaluation for intangible, technical works such as libraries and algorithms, makes software patents a frequent subject of controversy and litigation.

Different jurisdictions have radically different policies concerning software patents, including a blanket ban, no restrictions, or attempts to distinguish between purely mathematical constructs and "embodiments" of these constructs. For example, an algorithm itself may be judged unpatentable, but its use in software judged patentable.

Background

A patent is a set of exclusionary rights granted by a state to a patent holder for a limited period of time, usually 20 years. These rights are granted to patent applicants in exchange for their disclosure of the inventions. Once a patent is granted in a given country, no person may make, use, sell or import/export the claimed invention in that country without the permission of the patent holder. Permission, where granted, is typically in the form of a license which conditions are set by the patent owner: it may be free or in return for a royalty payment or lump sum fee.

Patents are territorial in nature. To obtain a patent, inventors must file patent applications in each and every country in which they want a patent. For example, separate applications must be filed in Japan, China, the United States and India if the applicant wishes to obtain patents in those countries. However, some regional offices exist, such as the European Patent Office (EPO), which act as supranational bodies with the power to grant patents which can then be brought into effect in the member states, and an international procedure also exists for filing a single international application under the Patent Cooperation Treaty (PCT), which can then give rise to patent protection in most countries.

These different countries and regional offices have different standards for granting patents. This is particularly true of software or computer-implemented inventions, especially where the software is implementing a business method.

Early example of a software patent

On 21 May 1962, a British patent application entitled "*A Computer Arranged for the Automatic Solution of Linear Programming Problems*" was filed.[1] The invention was concerned with efficient memory management for the simplex algorithm, and could be implemented by purely software means. The patent struggled to establish that it represented a 'vendible product'. "The focus of attention shifted to look at the relationship between the [unpatentable] computer program and the [potentially patentable] programmed computer".[2] The patent was granted on August 17, 1966, and seems to be one of the first software patents, establishing the principle that the computer program itself was unpatentable and therefore covered by copyright law, while the computer program embedded in hardware was potentially patentable.[3]

Jurisdictions

Most countries place some limits on the patenting of inventions involving software, but there is no one legal definition of a software patent. For example, U.S. patent law excludes "abstract ideas", and this has been used to refuse some patents involving software. In Europe, "computer programs as such" are excluded from patentability, thus European Patent Office policy is consequently that a program for a computer is not patentable if it does not have the potential to cause a "technical effect" which is by now understood as a material effect (a "transformation of nature").[4] Substantive law regarding the patentability of software and computer-implemented inventions, and case law interpreting the legal provisions, are different under different jurisdictions.

Software patents under multilateral treaties:

- Software patents under TRIPs Agreement
- Software patents under the European Patent Convention
- Computer programs and the Patent Cooperation Treaty

Software patents under national laws:

- Software patents under United States patent law
- Software patents under United Kingdom patent law

Australia

In Australia, there is no particular exclusion for patents relating to software. The subject matter of an invention is patentable in Australia, if it is a *manner of manufacture* within the meaning of section 6 of the Statute of Monopolies.[5] The High Court of Australia has refrained from ruling on the precise definition of manner of manufacture stating that any such attempt is bound to fail for the policy reason of encouraging national development in fields that may be unpredictable.[6] In assessing whether an invention is a manner of manufacture, the High Court has relied on the inquiry of whether the subject of the claims defining the invention has as its end result an *artificially created state of affairs*.[6]

In a decision of the Federal Court of Australia, on the patentability of an improved method of representing curved images in computer graphics displays, it was held that the application of selected mathematical methods to computers may involve steps which are foreign to the normal use of computers and hence amount to a manner of manufacture.[7] In another unanimous decision by the Full Federal Court of Australia, an invention for methods of storing and retrieving Chinese characters to perform word processing was held to be an artificially created state of affairs and consequently within the concept of a manner of manufacture.[8]

Nevertheless, in a recent decision on the patentability of a computer implemented method of generating an index based on selection and weighing of data based on certain criterion, the Full Federal Court of Australia reaffirmed that mere methods, schemes and plans are not manners of manufacture.[9] The Full

Court went on to hold that the use of a computer to implement a scheme did not contribute to the invention or the artificial effect of the invention.[9] The subject matter of the invention was held to be an abstract idea and not a manner of manufacture within the meaning of the term in the Patents Act. The same Full Federal Court in another decision regarding the patentability of an invention regarding a method and system for assessing an individual's competency in relation to certain criterion, reiterated that a business method or mere scheme were per se are not patentable.[10]

In principle, computer software is still a valid patentable subject matter in Australia. But, in circumstances where patents have been sought over software to merely implement abstract ideas or business methods, the courts and the Commissioner of Patents have resisted granting patent protection to such applications both as a matter of statutory interpretation and policy.

Canada

In Canada, courts have held that the use of a computer alone neither lends, nor reduces patentability of an invention. However, it is the position of the Canadian Patent Office that where a computer is an "essential element" of a patent's claims, the claimed invention is generally patentable subject matter.[11]

China

In China, the starting time of software patent is relatively late. Before 2006, software patents were basically not granted, and software and hardware had to be combined when applying for a patent. With the development of network technology and software technology, China's patent examination system has been constantly updated. Recently, the design idea of the software itself has been allowed to apply for patent separately, instead of requiring to be combined with hardware. However, software patent writing requirements are relatively high.

Software patents can be written as either a product or a method, depending on the standards of review. However, no matter what form it is written in, it is difficult to highlight the creativity of the scheme, which requires specific case analysis.

Software that can be patented mainly includes (but is not limited to):

(1) Industrial control software, such as controlling the movement of mechanical equipment;

(2) Software to improve the internal performance of the computer, such as a software can improve the virtual memory of the computer;

(3) External technical data processing software, such as digital camera image processing software.

It is fair to say that a considerable proportion of software belongs to category (3).

The patent protection measures can be seen in the patent law and the regulations on the protection of computer software.

Europe

Within European Union member states, the EPO and other national patent offices have issued many patents for inventions involving software since the European Patent Convention (EPC) came into force in the late 1970s. Article 52 (https://new.epo.org/en/legal/epc/2020/a52.html) EPC excludes "programs for computers" from patentability (Art. 52(2)) to the extent that a patent application relates to a computer program "as such" (Art. 52(3)). This has been interpreted to mean that any invention that makes a non-obvious "technical contribution" or solves a "technical problem" in a non-obvious way is patentable even if that technical problem is solved by running a computer program.[12] When the EPO examines a patent application with questionable subject matter eligibility, their approach is to simply disregard any ineligible portions or aspects and evaluate the rest.[13] This is notably different from the U.S. approach (see below).

Computer-implemented inventions that *only* solve a business problem using a computer, rather than a technical problem, are considered unpatentable as lacking an inventive step (see T 258/03). Nevertheless, the fact that an invention is useful in business does not mean it is not patentable if it also solves a technical problem.

A summary of the developments concerning patentability of computer programs under the European Patent Convention is given in (see G 3/08) as a response of the Enlarged Board of Appeal to questions filed by the President of the European Patent Office according to Article 112(1)(b) (https://new.epo.org/en/legal/epc/2020/a112.html) EPC.

Concerns have been raised by free software campaigners, such as the Free Software Foundation, that the Unified Patent Court will be much more open to patents generally and software patents in particular.[14]

Germany

In April 2013, the German Parliament adopted a joint motion "against the growing trend of patent offices to grant patents on software programs".[15]

United Kingdom

United Kingdom patent law is interpreted to have the same effect as the European Patent Convention such that "programs for computers" are excluded from patentability to the extent that a patent application relates to a computer program "as such". Current case law in the UK states that an (alleged) invention will only be regarded as an invention if it provides a contribution that is not excluded and that is also technical. A computer program implementing a business process is therefore not an invention, but a computer program implementing an industrial process may well be.

India

In India, a clause to include software patents was quashed by the Indian Parliament in April 2005.[16] However, following publication of the new guidelines on the examination of computer-related inventions on 19 February 2016, the Office of the Controller General of Patents, Designs and Trade marks accepts applications for software patents, as long as the software is claimed in conjunction with a novel hardware.[17] On 30 June 2017, revised guidelines on the examination of computer related inventions were published. This 2017 guidelines provides clarity on patentability of software invention in India, i.e., the claimed computer-related invention needs to be ascertained whether it is of a technical nature involving technical advancement as compared to the existing knowledge or having economic significance or both, and is not subject to exclusion under Section 3 of the Patents Act.[18] In 2019, the Court observed,

> In today's digital world, when most inventions are based on computer programs, it would be retrograde to argue that all such inventions would not be patentable. Innovation in the field of artificial intelligence, blockchain technologies and other digital products would be based on computer programs, however the same would not become nonpatentable inventions – simply for that reason. It is rare to see a product which is not based on a computer program. Whether they are cars and other automobiles, microwave ovens, washing machines, refrigerators, they all have some sort of computer programs in-built in them. Thus, the effect that such programs produce including in digital and electronic products is crucial in determining the test of patentability.

Patent applications in these fields would have to be examined to see if they result in a "technical contribution", it added. Further elaborating on the usage of the term 'per se' in Section 3(k), the Court said,

> The words 'per se' were incorporated so as to ensure that genuine inventions which are developed, based on computer programs are not refused patents.[19]

With respect to the term per se, the joint parliamentary committee had expressed the following view:

> In the new proposed clause (k) the words: "per se" have been inserted. This change has been proposed because sometimes the computer programme may include certain other things, ancillary thereto or developed thereon. The intention here is not to reject them for grant of patent if they are inventions. However, the computer programs as such are not intended to be granted patent. This amendment has been proposed to clarify the purpose.[20]

Japan

Software-related inventions are patentable. To qualify as an invention, however, there must be "a creation of technical ideas utilizing a law of nature"[21] although this requirement is typically met by "concretely realising the information processing performed by the software by using hardware resources".[22] Software-related inventions may be considered obvious if they involve the application of an operation known in other fields, the addition of a commonly known means or replacement by equivalent, the implementation in software of functions which were previously performed by hardware, or the systematisation of known human transactions.[23]

In 1999, the allowance rate for business method patents at the Japan Patent Office (JPO) reached an all-time high of roughly 35 percent. Subsequently, the JPO experienced a surge in business method patent filings. This surge was met with a dramatic decrease in the average grant rate of business method patents during the following six years; it lingered around 8 percent between 2003 and 2006 (8 percent is extremely low in comparison to the average of 50 percent across all technical fields). A report from 2012 found that the average grant rate since 2006 for business method patents has risen to the current rate of roughly 25 percent.[24]

New Zealand

In New Zealand computer programs are excluded from patentability under the Patents Act 2013,[25] but guidelines permitting embedded software were added since the initial Patents Bill.[26] From 2013 computer programs 'as such' are excluded from patentability. The as such wording rules out only those software based patents where novelty lies solely in the software. Similar to Europe.[27]

Philippines

In the Philippines, "schemes, rules and methods of performing mental acts, playing games or doing business, and programs for computers" are non-patentable inventions under Sec. 22.2 of Republic Act No. 8293, otherwise known as the "Intellectual Property Code of the Philippines".

Russian Federation

In the Russian Federation according to article #1350 of the Civil Code of the Russian Federation the following do not qualify as inventions:

1. discoveries;
2. scientific theories and mathematical methods;
3. solutions concerning only the appearance of products and aimed at meeting the aesthetic needs;
4. the rules and methods of games, intellectual or economic activities;
5. computer programs;
6. solutions consisting only in the presentation of information.

However, the article provides for that the patentability of these objects is excluded only in the case when the application for the grant of a patent for an invention concerns these objects *as such*.

South Africa

In South Africa, "a program for a computer" is excluded from recognition as an invention by section 25(2) of the Patents Act.[28] However, this restriction applies "only to the extent to which a patent or an application for a patent relates to that thing as such"[29] and should not prevent, for example, a product, process, or method which may be implemented on a computer from being an invention, provided that the requirements of novelty and inventiveness are met.

South Korea

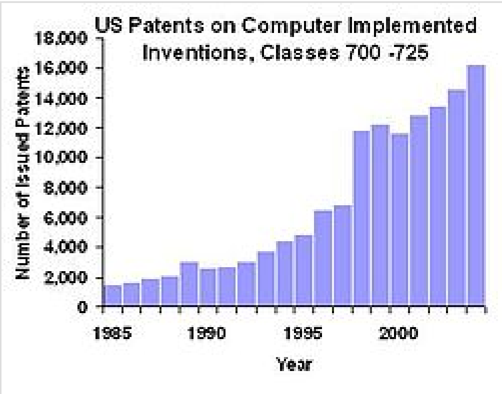
In South Korea, software is considered patentable and many patents directed towards "computer programs" have been issued.[30] In 2006, Microsoft's sales of its "Office" suite were jeopardized due to a possible patent infringement.[31] A ruling by the Supreme Court of Korea found that patents directed towards automatic language translation within software programs were valid and possibly violated by its software.[31]

Thailand

As like as 52(2) of the European Patent Convention (EPC), section 9 of the Thai Patent Act 1999 states that Thai patent law does not include software (or computer program) from patentability because the computer software is not considered as an "invention", in which it is not the idea of the product itself. Hence, the software is considered as the manual or instruction that was controlled by users to perform the tasks.[32]

A software patents law in Thailand has been controversial debates among the economists and national developers' overtime since there were two significant developments in the international patent law; (1) the European Union's attempt to harmonize national patent laws by the Proposal for a Directive of the European Parliament and Council on the patentability of computer-implemented inventions,[33] and (2) the US court decision to expand patent protection to business methods.[33] The opinions are divided into two sides. Dr. Tangkitvanich, the IT specialist of Thailand Development Research Institute (TDRI), raised his concern that Thailand is not in a good stage for a software patent as there were several flaws in patent rights. For example, the business method prevention has high tendency to hinder the growth in innovations especially for the infant software companies.[34] Moreover, the software patent may cause monopoly and innovation problems. "Monopoly will thwart innovations of new software products, particularly open-source software", said by a group of Thai Economists. However, Dr. Hirapruk who is the Director of Software Park Thailand, on the other hand, provides his support on allowing the computer programs to be patentable: "Thailand had to provide a patent-right protection for computer software to ensure foreign high-tech investors that software producers' creativity would be secured from violations in Thailand". As a result, Mr. Sribhibhadh, president of the Association of Thai Software Industry, emphasized that there will need to be a clear overview of the impact on the local industry if Thailand really had to fully implement the patent right protections.

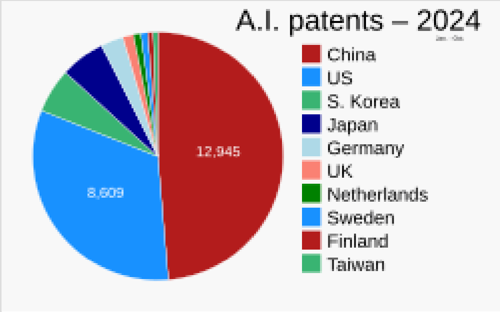
United States



Growth of software patents in US

The first software patent was issued June 19, 1968 to Martin Goetz for a data sorting algorithm.[36] The United States Patent and Trademark Office has granted patents that may be referred to as software patents since at least the early 1970s.[37] In Gottschalk v. Benson (1972), the United States Supreme Court ruled that a patent for a process should not be allowed if it would "wholly pre-empt the mathematical formula and in practical effect would be a patent on the algorithm itself", adding that "it is said that the decision precludes a patent for any program servicing a computer. We do not so hold."[38] In 1981, the Supreme Court stated that "a claim drawn to subject matter otherwise statutory does not become nonstatutory simply because it uses a mathematical formula, computer program, or digital computer" and a claim is patentable if it contains "a mathematical formula [and] implements or applies the formula in a structure or process which, when considered as a whole, is performing a function which the patent laws were designed to protect".[39] When a patent application is examined by the USPTO, the initial threshold question (for each claim) is whether the subject matter is eligible, so this is evaluated separately and prior to the other patentability criteria (novelty, nonobviousness).[40] This is notably different than the European approach (see above).

Due to different treatment of federal patent rights in different parts of the country, in 1982 the U.S. Congress created a new court (the Federal Circuit) to hear patent cases. Following several landmark decisions by this court, by the early 1990s the patentability of software was well established, and in 1996 the USPTO issued Final Computer Related Examination Guidelines stating that "A *practical application* of a computer-related invention is statutory subject matter. This requirement can be discerned from the variously phrased prohibitions against the patenting of abstract ideas, laws of nature or natural phenomena" (emphasis added).[41]



In 2024, AI patents in China and the US numbered more than three-fourths of AI patents worldwide.[35] Though China had more AI patents, the US had 35% more patents per AI patent-applicant company than China.[35]

The emergence of the Internet and e-commerce led to many patents being applied for and being granted for business methods implemented in software and the question of whether business methods are statutory subject matter is a separate issue from the question of whether software is. Critics of the Federal Circuit believe that the non-obviousness standard is partly responsible for the large increase in patents for software and business methods.[42] There have been several successful enforcement trials in the United States, some of which are listed in the list of software patents article.

An issue with software patent intellectual property rights is typically revolved around deciding whether the company or inventor owns it.

As a matter of law, in the United States, the employee generally owns the IP right unless the employee's inventing skills or task to create the invention is the main specific hiring reason or a specific clause in the employment agreement assigning invention rights.[43][44]

A work for hire created after 1978 has copyright protection for 120 years from its creation date or 90 years from its publication date whichever comes first.[45] Patent protection for software lasts 20 years.[46]

Indonesia

In Indonesia, software cannot be protected by patents, until the implementation of the Law No. 13 Year 2016, Patent Law in Indonesia.[47] To begin evaluation, it is necessary to distinguish whether or not the application is considered an invention. Under Law No. 14 Year 2001, Article 1 of Patent Law in Indonesia,[48] application is considered as an invention if the activity is created to solve a particular conflict or problem in the technology sector. Furthermore, it can be executed in the medium of a new process or product or a developmental enhancement in a product or process. According to Law No. 14 Year 2001, Article 7 of Patent Law in Indonesia.,[48] an application can not be patented as an invention if the product or process contradicts or challenges the current regulations and rules, public order or ethics, and religious morality. In addition, if the application is treated as a method or theory in the scientific or mathematics, argued to be any type of living creatures, with the exception of micro-organisms, or is considered as an essential biological measure to produce plants or animals, the application is not a patentable invention.[48]

As software contains algorithms, it is deemed to be part of the field of mathematics; hence, software cannot be protected by patents in Indonesia. However, one way for the Indonesian Intellectual Property office to grant software patents in Indonesia is if the application has been patented in other nations, which have ratified the Patent Corporation Treaty (PCT). Therefore, in accordance to the regulations under the Patent Cooperation Treaty, a software will have a regional protection among the participating entities of World Intellectual Property Organization (WIPO).[49]

An important update was enacted on 26 August 2016, the Law No. 13 Year 2016, Patent Law in Indonesia.[47] This update is geared to encourage innovation and growth by augmenting the number of patents within the public and private sector in Indonesia. This update proposes an extension of protection for simple patent, which grants application for patents for new improvements or inventions to existing processes. Intangible inventions can also be patented; under the former law, simple patent is restricted for tangible inventions, which has a positive implication for software patents in Indonesia. Furthermore, these changes provide more protection to the pharmaceutical industry and encourage public access to medical knowledge. This can boost new software ideas and processes within the healthcare and pharmaceutical sector. This update provides a stronger protection of traditional knowledge. In addition, a significant update is the usability of electronic filling and electronic media. Under this new law, application can be made electronically.

Purpose of patents

For the U.S., the purpose of patents is laid down in the constitutional clause that gives Congress the power to "promote the progress of science and useful arts, by securing for limited times to authors and inventors the exclusive right to their respective writings and discoveries;" (Article I, Section 8, Clause 8).[50] For Europe, there is no similar definition. Commonly four patent justification theories are recognised, as laid down for instance by Machlup in 1958,[51] which include justice to the inventor and benefit for society by rewarding inventors. Disclosure is required in return for the exclusive right, and disclosure may promote further development. However, the value of disclosure should not be overestimated: some inventions could not be kept secret otherwise, and patents also prohibit independent reinventions to be exploited.

There is debate as to whether or not these aims are achieved with software patents.

Proposals

In seeking to find a balance, different countries have different policies as to where the boundary between patentable and non-patentable software should lie. In Europe, a number of different proposals for setting a boundary line were put forward during the debate concerning the proposed Directive on the patentability of computer-implemented inventions, none of which were found acceptable by the various parties to the debate. Two particular suggestions for a hurdle that software must pass to be patentable include:

- A computer program that utilises "controllable forces of nature to achieve predictable results".[52]
- A computer program which provides a "technical effect".[53]

In the US, Ben Klemens, a Guest Scholar at the Brookings Institution, proposed that patents should be granted only to inventions that include a physical component that is by itself nonobvious.[54] This is based on Justice William Rehnquist's ruling in the U.S. Supreme Court case of Diamond v. Diehr that stated that "... insignificant postsolution activity will not transform an unpatentable principle into a patentable process."[55] By this rule, one would consider software loaded onto a stock PC to be an abstract algorithm with obvious postsolution activity, while a new circuit design implementing the logic would likely be a nonobvious physical device. Upholding an "insignificant postsolution activity" rule as per Justice Rehnquist's ruling would also eliminate most business method patents.

Obviousness

A common objection to software patents is that they relate to trivial inventions.[56] A patent on an invention that many people would easily develop independently of one another should not, it is argued, be granted since this impedes development.

Different countries have different ways of dealing with the question of inventive step and non-obviousness in relation to software patents. Europe uses an 'Inventive step test'; see the Inventive step requirement in Europe and, for instance, T 258/03.

Criticism

Compatibility

There are a number of high-profile examples where the patenting of a data exchange standards forced another programming group to introduce an alternative format. For instance, the Portable Network Graphics (PNG) format was largely introduced to avoid the Graphics Interchange Format (GIF) patent problems, and Ogg Vorbis to avoid MP3. If it is discovered that these new suggested formats are themselves covered by existing patents, the final result may be a large number of incompatible formats. Creating such formats and supporting them costs money and creates inconvenience to users.

Computer-implemented invention (CII)

Under the European Patent Convention (EPC), and in particular its Article 52,[57] "programs for computers" are not regarded as inventions for the purpose of granting European patents,[58] but this exclusion from patentability only applies to the extent to which a European patent application or European patent relates to a computer program as such.[59] As a result of this partial exclusion, and despite the fact that the EPO subjects patent applications in this field to a much stricter scrutiny[60] when compared to their American counterpart, that does not mean that all inventions including some software are *de jure* not patentable.

Overlap with copyright

Patent and copyright protection constitute two different means of legal protection which may cover the same subject matter, such as computer programs, since each of these two means of protection serves its own purpose.[61] Software is protected as works of literature under the Berne Convention. This allows the creator to prevent another entity from copying the program and there is generally no need to register code in order for it to be copyrighted.

Patents, on the other hand, give their owners the right to prevent others from using the technology defined by the patent claims, even if the technology was independently developed and there was no copying of a software or software code involved. In fact, one of the most recent EPO decisions[62] clarifies the distinction, stating that software is patentable, because it is basically only a technical method executed on a computer, which is to be distinguished from the program itself for executing the method, the program being merely an expression of the method, and thus being copyrighted.

Patents cover the underlying methodologies embodied in a given piece of software, or the function that the software is intended to serve, independent of the particular language or code that the software is written in. Copyright prevents the direct copying of some or all of a particular version of a given piece of software, but does not prevent other authors from writing their own embodiments of the underlying methodologies. Assuming a dataset meets certain criteria, copyright can also be used to prevent a given set of data from being copied while still allowing the author to keep the contents of said set of data a trade secret.[63]

Whether and how the numerus clausus principle shall apply to the legal hybrid software[64] to provide a judicious balance between property rights of the title holders and freedom rights of computing professionals[65] and society as a whole,[66] is in dispute.[67][68][69]

Debate

There is a debate over the extent to which software patents should be granted, if at all. Important issues concerning software patents include:

- Whether software patents should be allowed, and if so, where the boundary between patentable and non-patentable software should lie;[70]
- Whether the inventive step and non-obviousness requirement is applied too loosely to software;[71] and
- Whether patents covering software discourage, rather than encourage, innovation;[72]
- Whether software based on mathematical methods may be allowed if the mathematics or algorithm in question is complicated enough and may not be implemented with pencil and paper.[73]

Open source software

There is strong dislike in the free software community towards software patents. Much of this has been caused by free software or open source projects terminating[74] when the owners of patents covering aspects of a project demanded license fees that the project could not pay, or was not willing to pay, or offered licenses with terms that the project was unwilling to accept, or could not accept, because it conflicted with the free software license in use.[75]

Several patent holders have offered royalty-free patent licenses for a very small portion of their patent portfolios. Such actions have provoked only minor reaction from the free and open source software communities for reasons such as fear of the patent holder changing their mind or the license terms being so narrow as to have little use.[76] Companies that have done this include Apple,[77] IBM,[78] Microsoft,[79] Nokia,[80] Novell,[81] Red Hat,[82] and Sun (now Oracle).[83]

In 2005, Sun Microsystems announced that they were making a portfolio of 1,600 patents available through a patent license called Common Development and Distribution License.[84]

In 2006, Microsoft's pledge not to sue Novell Linux customers, openSUSE contributors, and free/open source software developers over patents[85] and the associated collaboration agreement with Novell[86] was met with disdain from the Software Freedom Law Center[87] while commentators from the Free Software Foundation stated that the agreement would not comply with GPLv3. Meanwhile, Microsoft has reached similar agreements with Dell and Samsung,[88] due to alleged patent infringements of the Linux operating system. Microsoft has also derived revenue from Android by making such agreements-not-to-sue with Android vendors.[89]

Unisys case

In the late 1990s, Unisys claimed to have granted royalty free licenses to hundreds of not-for-profit organizations that used the patented LZW compression method and, by extension, the GIF image format. However, this did not include most software developers and Unisys were "barraged" by negative and "sometimes obscene" emails from software developers.[90]

Licensing

Patenting software is widespread in the US. As of 2015, approximately 500,000 patents had issued in the 23 classes of patents covering "computer implemented inventions" (see table).

Many software companies cross license their patents to each other. These agreements allow each party to practice the other party's patented inventions without the threat of being sued for patent infringement. Microsoft, for example, has agreements with IBM, Sun (now Oracle), SAP, Hewlett-Packard, Siemens, Cisco, Autodesk,[92] and recently Novell. Microsoft cross-licensed its patents with Sun, despite being direct competitors, and with Autodesk even though Autodesk has far fewer patents than Microsoft.

The ability to negotiate cross licensing agreements is a major reason that many software companies, including those providing open source software, file patents. As of June 2006, for example, Red Hat had developed a portfolio of 10 issued US patents, 1 issued European patent, 163 pending US patent applications, and 33 pending international PCT (Patent Cooperation Treaty) patent applications. Red Hat uses this portfolio to cross license with proprietary software companies so that they can preserve their freedom to operate.[82]

Other patent holders are in the business of inventing new "computer implemented inventions" and then commercializing the inventions by licensing the patents to other companies that manufacture the inventions. Walker Digital, for example, has generated a large patent portfolio from its research efforts, including the basic patent on the Priceline.com reverse auction technology. US universities also fall into this class of patent owners. They

Total US software patent counts by class of invention as of 2015[91]

US class	Description	Total patents issued
700	Data Processing: Generic Control Systems or Specific Applications	26042
701	Data Processing: Vehicles, Navigation, and Relative Location	38566
702	Data Processing: Measuring, Calibrating, or Testing	27130
703	Data Processing: Structural Design, Modeling, Simulation, and Emulation	10126
704	Data Processing: Speech Signal Processing, Linguistics, Language Translation, and Audio Compression/Decompression	17944
705	Data Processing: Financial, Business Practice, Management, or Cost/Price Determination	38284
706	Data Processing: Artificial Intelligence	9161
707	Data Processing: Database and File Management or Data Structures	47593
708	Electrical Computers: Arithmetic Processing and Calculating	9993
709	Electrical Computers and Digital Processing Systems: Multicomputer Data Transferring	56001
710	Electrical Computers and Digital Data Processing Systems: Input/Output	23991
711	Electrical Computers and Digital Processing Systems: Memory	34025
712	Electrical Computers and Digital Processing Systems: Processing Architectures and Instruction Processing (e.g., Processors)	10461
713	Electrical Computers and Digital Processing Systems: Support	30695
714	Error Detection/Correction and Fault Detection/Recovery	38532
715	Data Processing: Presentation Processing of Document, Operator Interface Processing, and Screen Saver Display Processing	25413
716	Computer-Aided Design and Analysis of Circuits and Semiconductor Masks	13809
717	Data Processing: Software Development, Installation, and Management	17336
718	Electrical Computers and Digital Processing Systems: Virtual Machine Task or Process Management or Task Management/Control	7615
719	Electrical Computers and Digital Processing Systems: Interprogram Communication or Interprocess Communication (Ipc)	5456
720	Dynamic Optical Information Storage or Retrieval	3877
725	Interactive Video Distribution Systems	12076
726	Information Security	21144
	Total	**525270**

collectively generate about $1.4 billion per year through licensing the inventions they develop to both established and start up companies in all fields of technology, including software.[93]

Still other patent holders focus on obtaining patents from original inventors and licensing them to companies that have introduced commercial products into the marketplace after the patents were filed. Some of these patent holders, such as Intellectual Ventures, are privately held companies financed by large corporations such as Apple, Microsoft, Intel, Google, etc. Others, such as Acacia Technologies, are publicly traded companies with institutional investors being the primary shareholders.[94]

The practice of acquiring patents merely to license them is controversial in the software industry. Companies that have this business model are pejoratively referred to as patent trolls. It is an integral part of the business model that patent licensing companies sue infringers that do not take a license. Furthermore, they may take advantage of the fact that many companies will pay a modest license fee (e.g. $100,000 to $1,000,000) for rights to a patent of questionable validity, rather than pay the high legal fees ($2,000,000 or more) to demonstrate in court that the patent is invalid.

See also

- Open Invention Network (OIN)
- Patent Commons Project
- Piano roll blues
- Social networking patents

References

1. GB 1039141 (https://worldwide.espacenet.com/textdoc?DB=EPODOC&IDX=GB1039141), "A computer arranged for the automatic solution of linear programming problems by British Petroleum, Patrick Vincent Slee and Pauline Margaret Joyce Harris"
2. Bellido, Jose (2017). *Landmark Cases in Intellectual Property Law*. Hart Publishing.
3. Beresford, K (2000). *Patenting Software under the European Patent Convention*. London: Sweet & Maxwell. p. 4.
   Kretschmer, Martin (4 July 2003). "Software as Text and Machine: The Legal Capture of Digital Innovation" (http://www2.warwick.ac.uk/fac/soc/law/elj/jilt/2003_1/kretschmer/). *Journal of Information Law & Technology*. Centre for Intellectual Property Policy and Management, Bournemouth University.
4. Dhenne, M. (2016). *Technique et droit des brevets. L'invention en droit des brevets*. France: LexisNexis. pp. 82 and following. ISBN 9782711024100.
5. *Patents Act 1990* (Cth) s 18 (https://www.austlii.edu.au/au/legis/cth/consol_act//pa1990109/s18.html) Patentable inventions
6. *National Research Development Corp v Commissioner of Patents* [1959] HCA 67 (https://www.austlii.edu.au/au/cases/cth/HCA/1959/67.html), (1959) 102 CLR 252, High Court (Australia).
7. *International Business Machines Corp v Commissioner of Patents* [1991] FCA 811 (https://www.austlii.edu.au/au/cases/cth/FCA/1991/811.html), 105 ALR 388, Federal Court (Australia).
8. *CCOM Pty Ltd v Jie-jing Pty Ltd* [1994] FCA 396 (https://www.austlii.edu.au/au/cases/cth/FCA/1994/396.html), (1994) 51 FCR 260, Federal Court (Full Court) (Australia).
9. *Research Affiliates LLC v Commissioner of Patents* [2014] FCAFC 150 (https://www.austlii.edu.au/au/cases/cth/FCAFC/2014/150.html) (10 November 2014), Federal Court (Full Court) (Australia).

10. *Commissioner of Patents v RPL Central Pty Ltd* [2015] FCAFC 177 (https://www.austlii.edu.au/au/cases/cth/FCAFC/2015/177.html) (11 December 2015), Federal Court (Full Court) (Australia).
11. *Examination Practice Respecting Computer-Implemented Inventions – PN 2013-03* (http://www.cipo.ic.gc.ca/eic/site/cipointernet-internetopic.nsf/eng/wr03627.html), Canadian Intellectual Property Office, March 8, 2013
12. *DECISION of 24 February 2006, Case Number: T 0469/03 – 3.5.01* (https://www.epo.org/law-practice/case-law-appeals/recent/t030469eu1.html), European Patent Office, Boards of Appeal, 24 February 2006, pp. Reasons 5.1 to 5.3
13. Guidelines for Examination in the EPO, section G-VII, 5.4 (https://www.epo.org/law-practice/legal-texts/html/guidelines/e/g_vii_5_4.htm)
14. "End Software Patents" (https://endsoftpatents.org/pages/unitary-patent/).
"Europe's "unitary patent" could mean unlimited software patents" (https://www.gnu.org/philosophy/europes-unitary-patent.en.html).
"EU unitary patent and Unified Patent Court" (https://en.swpat.org/wiki/EU_unitary_patent_and_Unified_Patent_Court).
15. Ermert, Monika (22 April 2013). "German Parliament Sends Message: Stop Granting Software Patents" (http://www.ip-watch.org/2013/04/22/german-parliament-sends-message-stop-granting-software-patents). *Intellectual Property Watch*. Retrieved April 29, 2013.
16. "Software patents under Ordinance face reversal" (http://www.financialexpress.com/news/software-patents-under-ordinance-face-reversal/82155). Financialexpress.com. 2005-03-29. Retrieved 2012-10-09.
17. Guido Noto La Diega, PhD. *Software Patents and the Internet of Things in Europe, the United States and India (FULL TEXT)* (https://www.academia.edu/34244396) – via www.academia.edu.
18. "Guidelines for Examination of Computer Related Inventions (CRIs)" (http://www.ipindia.nic.in/writereaddata/Portal/Images/pdf/Revised__Guidelines_for_Examination_of_Computer-related_Inventions_CRI__.pdf) (PDF). *www.ipindia.nic.in*.
19. Aditi, Singh (29 December 2019). "A computer program which makes 'technical contribution' patentable, not hit by Sec 3(k) of Patents Act" (https://www.barandbench.com/news/a-computer-program-which-makes-technical-contribution-patentable-not-hit-by-sec-3k-of-patents-act). Bar and Bench. Retrieved 13 February 2020.
20. Parliament of India, Report of the Joint Committee on Patents (Second Amendment) Bill, 1999 (Rajya Sabha Secretariat 2001)
21. "Examination Guidelines for Patent and Utility Model in Japan (REQUIREMENTS FOR PATENTABILITY)" (https://web.archive.org/web/20110323200823/https://www.jpo.go.jp/tetuzuki_e/t_tokkyo_e/Guidelines/2_1.pdf) (PDF). jpo.go.jp. pp. 1–3. Archived from the original (http://www.jpo.go.jp/tetuzuki_e/t_tokkyo_e/Guidelines/2_1.pdf) (PDF) on 2011-03-23. Retrieved 2009-11-21.
22. *Examination Guidelines for Inventions for Specific Fields (Computer Software-Related Inventions) in Japan* (https://web.archive.org/web/20110323200910/https://www.jpo.go.jp/tetuzuki_e/t_tokkyo_e/Guidelines/7_1.pdf) (PDF), Japanese Patent Office, April 2005, p. 10 (2.2.1), archived from the original (http://www.jpo.go.jp/tetuzuki_e/t_tokkyo_e/Guidelines/7_1.pdf) (PDF) on 2011-03-23, retrieved 2009-11-21
23. *Examination Guidelines for Inventions for Specific Fields (Computer Software-Related Inventions) in Japan* (https://web.archive.org/web/20110323200910/https://www.jpo.go.jp/tetuzuki_e/t_tokkyo_e/Guidelines/7_1.pdf) (PDF), Japanese Patent Office, April 2005, pp. 16–17 ("Systematization of human transactions"), archived from the original (http://www.jpo.go.jp/tetuzuki_e/t_tokkyo_e/Guidelines/7_1.pdf) (PDF) on 2011-03-23, retrieved 2009-11-21
24. Sugimura, Kenji; Chen, Rebecca (1 October 2012), *An important market: software patenting in Japan* (http://www.worldipreview.com/article/an-important-market-software-patenting-in-japan), World Intellectual Property Review

25. "Patents Act 2013, section 11 'Computer programs' " (https://www.legislation.govt.nz/act/public/2013/0068/latest/whole.html#DLM5516600). 2020-08-07. Retrieved 2022-01-25.
26. "Minister announces way forward for software patents" (https://web.archive.org/web/20100920044258/http://www.beehive.govt.nz/release/minister+announces+way+forward+software+patents). beehive.govt.nz. 2010-07-15. Archived from the original (https://www.beehive.govt.nz/release/minister+announces+way+forward+software+patents) on 2010-09-20. Retrieved 2012-10-09.
    Varghese, Sam (16 July 2010). "New Zealand says no to software patents" (https://web.archive.org/web/20130116134330/http://www.itwire.com/it-policy-news/govenrment-tech-policy/40451-new-zealand-says-no-to-software-patents). *Government Tech Policy*. Archived from the original (http://www.itwire.com/it-policy-news/govenrment-tech-policy/40451-new-zealand-says-no-to-software-patents) on 16 January 2013. Retrieved 7 November 2012.
    "South Pacific cousins part ways over patents" (https://web.archive.org/web/20120826000324/http://www.cpaglobal.com/newlegalreview/4617/south_pacific_cousins_part_way). Cpaglobal.com. 2010-07-28. Archived from the original (http://www.cpaglobal.com/newlegalreview/4617/south_pacific_cousins_part_way) on 2012-08-26. Retrieved 2012-10-09.
    "New Zealand bans software patents" (https://web.archive.org/web/20120406172154/http://www.tgdaily.com/business-and-law-features/50667-new-zealand-bans-software-patents). TG Daily. 2010-07-15. Archived from the original (http://www.tgdaily.com/business-and-law-features/50667-new-zealand-bans-software-patents) on 2012-04-06. Retrieved 2012-10-09.
    Fairfax Media Business Group (2010-04-15). "NZICT says patents 'integral' to software protection | Computerworld New Zealand" (https://web.archive.org/web/20130115221652/http://computerworld.co.nz/news.nsf/news/nzict-says-patents-integral-to-software-protection). Computerworld.co.nz. Archived from the original (http://computerworld.co.nz/news.nsf/news/nzict-says-patents-integral-to-software-protection) on 2013-01-15. Retrieved 2012-10-09.
    {{cite web}}: |author= has generic name (help)
27. Rob O'Neill (August 28, 2013). "New Zealand bans software patents" (https://www.zdnet.com/article/new-zealand-bans-software-patents/). *ZDNet*. Retrieved September 6, 2013.
28. *Patents Act, No. 57 of 1978, as amended* (https://web.archive.org/web/20111212164224/http://www.cipc.co.za/Patents_files/Patent_Act.pdf) (PDF), 26 April 1978, archived from the original (http://www.cipc.co.za/Patents_files/Patent_Act.pdf) (PDF) on 12 December 2011, retrieved 25 September 2011
29. *Patents Act, No. 57 of 1978, as amended Section 25(3)* (https://web.archive.org/web/20111212164224/http://www.cipc.co.za/Patents_files/Patent_Act.pdf) (PDF), 26 April 1978, archived from the original (http://www.cipc.co.za/Patents_files/Patent_Act.pdf) (PDF) on 12 December 2011, retrieved 25 September 2011
30. *FAQ – Korea* (https://web.archive.org/web/20090913083903/http://www.epo.org/patents/patent-information/east-asian/helpdesk/korea/faq.html#new2), European Patent Office, archived from the original (http://www.epo.org/patents/patent-information/east-asian/helpdesk/korea/faq.html#new2) on 13 September 2009, retrieved 29 October 2008
31. Lombardi, Candace (November 27, 2006), "Microsoft lost in translation" (http://news.cnet.com/8301-10784_3-6138379-7.html), *News Blog*, CNET, retrieved October 29, 2008
32. Sirivish Toomgum and Kwanchai Rungfapaisan, THE NATION, July 24, 2000, at 7.
33. Pitiyasak, Saravuth. "Thailand Debates Software Patents." *Managing Intellectual Property*.123 (2002): 62-4. *ProQuest.* Web. 11 Apr. 2020.
34. Newsbytes, Thailand Mulls Software Patent Rules, at (last visited Jul. 25, 2000). (http://www.washingtonpost.com/wp-dyn/technology/)
35. Buntz, Brian (3 November 2024). "Quality vs. quantity: US and China chart different paths in global AI patent race in 2024 / Geographical breakdown of AI patents in 2024" (https://www.rdworldonline.com/quality-vs-quantity-us-and-china-chart-different-paths-in-global-ai-patent-race-in-2024/). R&D World. Archived (https://web.archive.org/web/20241209072113/https://www.rdworldonline.com/quality-vs-quantity-us-and-china-chart-different-paths-in-global-ai-patent-race-in-2024/) from the original on 9 December 2024.

36. Martin Goetz, "Unsung innovators: Marty Goetz, holder of first [US] software patent" (https://www.computerworld.com/article/2540020/unsung-innovators--marty-goetz--holder-of-first-software-patent.html). Computerworld. 1968.
37. U.S. patent 3,552,738 (https://patents.google.com/patent/US3552738), U.S. patent 3,553,358 (https://patents.google.com/patent/US3553358) and U.S. patent 3,553,384 (https://patents.google.com/patent/US3553384) granted 5 January 1971, and U.S. patent 3,996,564 (https://patents.google.com/patent/US3996564) granted December 7, 1976, can be easily found using the Bessen/Hunt technique. Earlier patents may exist but US patent database does not permit full text searching for earlier patents
38. Gottschalk v Benson, 409 U.S. 63 (https://supreme.justia.com/cases/federal/us/409/63/) (1972)
39. Diamond v. Diehr, 450 U.S. 175 (https://supreme.justia.com/cases/federal/us/450/175/) (1981)
40. "MPEP" (https://www.uspto.gov/web/offices/pac/mpep/s2106.html).
41. "Examination Guidelines for Computer-Related Inventions" (http://www.uspto.gov/web/offices/com/sol/og/con/files/cons093.htm). United States Patent Office. 1996-03-26. Retrieved 2014-05-20.
42. Bessen, James, and Michael J. Meurer. Patent Failure: How Judges, Bureaucrats, and Lawyers Put Innovators at Risk. Princeton, NJ: Princeton University Press, 2008
43. Employment, Small Business and Training (2010-12-03). "Home" (https://www.business.qld.gov.au/). *www.business.qld.gov.au*. Retrieved 2020-04-11. `{{cite web}}`: `|first=` has generic name (help)
44. "Who owns the rights to a patent? The employer or inventor?" (https://www.adlilaw.com/who-owns-the-rights-to-a-patent-the-employer-or-inventor/). *Adli Law Group*. 2016-04-12. Retrieved 2020-04-11.
45. Peter B. Hirtle, Copyright Term and the Public Domain in the United States, 1 January 2007 (http://www.copyright.cornell.edu/training/Hirtle_Public_Domain.htm). Archived (https://web.archive.org/web/20120704071954/http://www.copyright.cornell.edu/training/Hirtle_Public_Domain.htm) July 4, 2012, at the Wayback Machine
46. "SOFTWARE PATENT LAW: UNITED STATES AND EUROPE COMPARED" (https://scholarship.law.duke.edu/cgi/viewcontent.cgi?article=1075&context=dltr). *Duke Law*. Retrieved 19 October 2021.
47. "Law No. 13 of 2016 on Patents, Indonesia, WIPO Lex" (https://www.wipo.int/wipolex/en/legislation/details/16392). *www.wipo.int*. Retrieved 2025-05-15.
48. "Law of the Republic of Indonesia Number 14 Year 2001" (https://www.wipo.int/edocs/lexdocs/laws/en/id/id044en.pdf) (PDF). *WIPO*.
49. "Regulations under the Patent Cooperation Treaty" (https://www.wipo.int/wipolex/en/text/288644). *wipo.int*. Retrieved 2025-05-15.
50. "Intellectual Property and Software-Copyright" (https://aristeksystems.com/blog/intellectual-property-and-software-copyright/). Retrieved 2021-12-16.
51. "An Economic Review of the Patent System (1958)" (https://www.mises.org/etexts/patentsystem.pdf) (PDF). 20 July 2005. Retrieved 2012-10-09.
52. *Amendment 23 to the proposed Directive on the patentability of computer-implemented inventions* (https://web.archive.org/web/20060210231145/http://www.europarl.eu.int/meetdocs/committees/itre/20030219/488498en.pdf) (PDF), European Parliament, September 2003, archived from the original (http://www.europarl.eu.int/meetdocs/committees/itre/20030219/488498en.pdf) (PDF) on February 10, 2006
Dispositions program decision, Bundesgerichtshof (, BGH, Federal Court of Justice of Germany), 1976-06-22

53. "Reasons, point 3.4" (https://www.epo.org/law-practice/case-law-appeals/recent/t930059eu1.html), *Decision T 0059/93 () of 20 April 1994*, European Patent Office, Boards of Appeal, 20 April 1994
54. "The Gravity of the U.S. Patent Swindle" (https://www.wsj.com/articles/SB114325217495908172). *Wall Street Journal*. 25 March 2006. p. A9.
55. *Diamond v. Diehr*, 49 U.S. 1112 (https://caselaw.findlaw.com/us-supreme-court/450/175.html) (SCOTUS 1981) (""... insignificant postsolution activity will not transform an unpatentable principle into a patentable process."").
56. Stallman, Richard (June 20, 2005). "Patent absurdity" (https://www.theguardian.com/technology/2005/jun/20/comment.comment). *The Guardian*. London. Retrieved May 23, 2010.
57. Article 52 (https://new.epo.org/en/legal/epc/2020/a52.html) EPC
58. Article 52(1) (https://new.epo.org/en/legal/epc/2020/a52.html) EPC
59. Article 52(3) (https://new.epo.org/en/legal/epc/2020/a52.html) EPC
60. Laub, Christoph (2005) [Academic Year 2004/2005]. *International Software Patent Filing: The Problem of Statutory Subject Matter in view of Legal Standards at the EPO-USPTO and Economic Implications* (https://web.archive.org/web/20070621083254/http://www.miplc.de/research/master_theses/2004_2005/abstracts/abstract_laub.pdf) (PDF) (Master's). Munich Intellectual Property Law Center. Archived from the original (http://www.miplc.de/research/master_theses/2004_2005/abstracts/abstract_laub.pdf) (PDF) on June 21, 2007. Retrieved 21 March 2006.
61. "Reasons 2.4" (https://www.epo.org/law-practice/case-law-appeals/recent/t971173ex1.html), *Decision T 1173/97 (Computer program product/IBM) of 1.7.1998*, European Patent Office, Boards of Appeal, 1 July 1998
62. *Decision T 0424/03 (Clipboard formats I/MICROSOFT) of 23.2.2006* (https://www.epo.org/law-practice/case-law-appeals/recent/t030424eu1.html), European Patent Office Boards of Appeal, 23 February 2006
63. Bro, Sarah (September 6, 2018). "Copyright in Data Compilations is Thin" (http://www.mondaq.com/unitedstates/x/733688/Trade+Secrets/Copyright+in+Data+Compilations+Is+Thin). *Mondaq Business Briefing*. Retrieved January 25, 2019.
64. Burk, Dan L. (February 2001). "Copyrightable functions and patentable speech" (https://doi.org/10.1145%2F359205.359231). *Communications of the ACM*. **44** (2). Association for Computing Machinery (ACM): 69–75. doi:10.1145/359205.359231 (https://doi.org/10.1145%2F359205.359231). S2CID 10784660 (https://api.semanticscholar.org/CorpusID:10784660).
65. Kiesewetter-Köbinger, Swen (2010). "Programmers' Capital". *Computer*. **43** (2): 106–108. doi:10.1109/MC.2010.47 (https://doi.org/10.1109%2FMC.2010.47).
66. Standing Committee on the Law of Patents (January 20, 2010). "Proposal from Brazil in respect of exceptions and limitations to patent rights" (http://www.wipo.int/edocs/mdocs/patent_policy/en/scp_14/scp_14_7.pdf) (PDF). World Intellectual Property Organization. Retrieved 4 February 2010.
67. Max-Planck-Institut (20 December 2000), *Stellungnahme zum Sondierungspapier der Kommission der Europaischen Gemeinschaften: Die Patentierbarkeit Computer-implementierter Erfindungen* (http://ec.europa.eu/internal_market/indprop/docs/comp/replies/planck_en.pdf) [*Opinion on the consultation paper by the Commission of the European Communities: The patentability of computer-implemented inventions*] (PDF)
68. Hoffmann, Marcus (2008). *Mehrfachschutz geistigen Eigentums im deutschen Rechtssystem* (http://www.buchhandel.de/WebApi1/GetMmo.asp?MmoId=4804929&mmoType=PDF) [*Multiple protection of intellectual property in the German legal system*]. ISBN 978-3-8316-0806-5.
69. Peukert, Alexander (2008). *Güterzuordnung als Rechtsprinzip* [*Goods classification as a legal principle*]. Mohr Siebeck. ISBN 978-3-16-149724-7.

70. Gray, John (2004-08-19), *Software patents in Europe: debunking the myths* (http://www.out-law.com/page-4814), OUT-LAW News
71. "Public consultation on level of the inventive step required for obtaining patents" (https://web.archive.org/web/20120104231807/http://www.ipo.gov.uk/response-inventive.pdf) (PDF). UK Intellectual Property Office. Archived from the original (http://www.ipo.gov.uk/response-inventive.pdf) (PDF) on 2012-01-04. Retrieved 2007-06-05.
72. "Patent Economics: Part 4 – Incentives" (http://www.patenthawk.com/blog/2005/04/patent_economics_part_4_incent.html) (blog entry). Patenthawk. 2005-04-17. Archived (https://archive.today/20120906032234/http://www.patenthawk.com/blog/2005/04/patent_economics_part_4_incent.html) from the original on 2012-09-06. Retrieved 2012-10-09.
73. "The India Patents Act, 1970 and Software Patentability" (https://web.archive.org/web/20160417223948/http://www.managingip.com/pdfs/Events/IndiaIP2013_15.15_Akhilesh%20Gupta.pdf) (PDF). *www.managingip.com*. 7 March 2013. Archived from the original (http://www.managingip.com/pdfs/Events/IndiaIP2013_15.15_Akhilesh%20Gupta.pdf) (PDF) on 2016-04-17. Retrieved 2020-03-27.
74. "Software patents that hurt free software" (https://www.gnu.org/patent-examp/patent-examples.html). Free Software Foundation. Retrieved 2012-10-09.
75. "Effects" (https://web.archive.org/web/20070210164646/http://swpat.ffii.org/patents/effects/). Patentability and Democracy in Europe. Foundation for a Free Information Infrastructure. Archived from the original (http://swpat.ffii.org/patents/effects) on February 10, 2007.
76. "Microsoft's Open Specification Promise: No Assurance for GPL" (http://www.softwarefreedom.org/resources/2008/osp-gpl.html). Software Freedom Law Center.
77. "License - APSL" (https://opensource.apple.com/apsl/). *opensource.apple.com*.
78. Scherer, Markus; Davis, Mark (2006-02-04). "UTN #6: BOCU-1" (https://www.unicode.org/notes/tn6/#Intellectual_Property). Retrieved 2014-02-05.
79. "Silverlight Controls License" (http://www.microsoft.com/resources/sharedsource/silverlightcontrolslicense.mspx). Microsoft Corporation.
80. "symbian-dump" (https://sourceforge.net/projects/symbiandump/). *SourceForge*. 11 April 2013.
81. "Patent Policy" (https://web.archive.org/web/20130730235511/http://www.novell.com/company/policies/patent/). Novell. Archived from the original (http://www.novell.com/company/policies/patent/) on 2013-07-30. Retrieved 2012-10-09.
82. "Red Hat Patent Policy" (http://www.redhat.com/legal/patent_policy.html). Red Hat.
83. "Oracle Technology Network for Java Developers" (http://java.sun.com/javase/6/jdk-6u2-license.txt). Sun Microsystems. Retrieved 2012-10-09.
84. "Sun Grants Global Open Source Community Access to More than 1,600 Patents" (https://web.archive.org/web/20081202014850/http://www.sun.com/smi/Press/sunflash/2005-01/sunflash.20050125.2.xml) (Press release). Sun Microsystems. January 25, 2005. Archived from the original (http://www.sun.com/smi/Press/sunflash/2005-01/sunflash.20050125.2.xml) on December 2, 2008.
85. "Community Commitments – Microsoft & Novell Interoperability Collaboration" (http://www.microsoft.com/interop/msnovellcollab/community.mspx). Microsoft. November 2, 2006.
86. "Microsoft and Novell Announce Broad Collaboration on Windows and Linux Interoperability and Support" (http://www.novell.com/news/press/item.jsp?id=1196) (Press release). Novell. November 2, 2006.
87. "Microsoft's developer patent pledge 'worse than useless' says SFLC" (http://www.businessreviewonline.com/os/archives/2006/11/microsofts_deve.html). November 10, 2006.
88. Eric Lai; Sumner Lemon (2007-05-13). "Update: Microsoft wants royalties for open-source software" (http://www.computerworld.com/s/article/9019238/Update_Microsoft_wants_royalties_for_open_source_software). Computerworld. Retrieved 2012-10-09.

89. "Microsoft and LG Sign Patent Agreement Covering Android and Chrome OS Based Devices" (http://www.microsoft.com/Presspass/press/2012/jan12/01-12LGPR.mspx). Microsoft. 2012-01-12. Retrieved 2012-10-09.
90. Miller, Robin ("Roblimo") (August 31, 1999). "Unisys Not Suing (most) Webmasters for Using GIFs" (http://slashdot.org/articles/99/08/31/0143246.shtml). Slashdot.
91. "Patent Counts By Class By Year, CY 1977 - 2015" (https://www.uspto.gov/web/offices/ac/ido/oeip/taf/cbcby.htm). United States Patent and Trademark Office. Retrieved 2017-03-13.
92. "Microsoft, Autodesk in patent licensing deal" (https://web.archive.org/web/20130115222332/http://www.arnnet.com.au/article/5043/microsoft_autodesk_patent_licensing_deal/?taxid=620938001). *IDG News Service*. 2004-12-17. Archived from the original (http://www.arn.idg.com.au/index.php?taxid=620938001&id=63439861) on January 15, 2013. Retrieved 2012-10-09.
93. *FY 2004 U.S. Licensing Survey* (https://web.archive.org/web/20121015213045/http://www.autm.net/FY_2004_Licensing_Survey/8932.htm), Association of University Technology Managers, 2012-09-08, archived from the original (http://www.autm.net/FY_2004_Licensing_Survey/8932.htm) on 2012-10-15, retrieved 2012-11-07
94. *Acacia Technologies LLC: Company Information* (http://premium.hoovers.com/subscribe/co/overview.xhtml?ID=fffrfkrhrrxhjcxxkh), Hoovers

External links

- Media related to Software patents at Wikimedia Commons
- *Software and Business Methods* (https://web.archive.org/web/20061022090935/http://www.wipo.int/patent/law/en/developments/software.html) on the WIPO web site
- FFII Annotated Bibliography (https://web.archive.org/web/20050410183700/http://swpat.ffii.org/archive/mirror/impact/index.en.html)
- Softwarepatents.eu (http://www.softwarepatents.eu/) Archived (https://web.archive.org/web/20170626075717/http://softwarepatents.eu/) 2017-06-26 at the Wayback Machine, an introduction to software patents in Europe and Germany (in German)

Economic studies

- Bessen; Hunt (2004), *An Empirical Look at Software Patents* (http://www.researchoninnovation.org/swpat.pdf) (PDF) This paper includes a method of identifying software patents that has proved popular with organisations such as the Public Patent Foundation.
- Bessen, James; Hunt, Robert M. (March 16, 2004), *The Software Patent Experiment* (http://www.researchoninnovation.org/softpat.pdf) (PDF)
- Hahn, Robert; Wallsten, Scott, *A Review of Bessen and Hunt's Analysis of Software Patents* (https://web.archive.org/web/20070929094042/http://www.researchineurope.org/policy/hahn_wallsten.pdf) (PDF), American Enterprise Institute, archived from the original (http://www.researchineurope.org/policy/hahn_wallsten.pdf) (PDF) on 2007-09-29
- Bessen, James; Hunt, Robert M. (March 10, 2004), *A Reply to Hahn and Wallsten* (http://www.researchoninnovation.org/hahn.pdf) (PDF)
- Federal Trade Commission (October 2003), *To Promote Innovation: The Proper Balance of Competition and Patent Law and Policy, A Report by the Federal Trade Commission* (https://www.ftc.gov/os/2003/10/innovationrpt.pdf) (PDF)
- *FTC 2003 Report on Patents and Competition* (https://web.archive.org/web/20121204023208/http://eupat.ffii.org/papri/ftc03/), Foundation for a Free Information Infrastructure (FFII), 2004-08-16, archived from the original (http://eupat.ffii.org/papri/ftc03/) on 2012-12-04,

retrieved 2012-11-07 – commentary on the FTC "To Promote Innovation" report, 2003 (translated into English)

- Noel, Michael; Schankerman, Mark (August 2006), *CEP Discussion Paper No 740: Strategic Patenting and Software Innovation* (http://cep.lse.ac.uk/pubs/download/dp0740.pdf) (PDF), Centre for Economic Performance, London School of Economics and Political Science

Books

- Bessen, James; Meurer, Michael (2008). *Patent Failure* (http://press.princeton.edu/titles/8634.html). Princeton University Press. ISBN 9780691143217.
- Klemens, Ben (2005). *Math You Can't Use: Patents, Copyright, and Software* (http://www.brookings.edu/press/books/mathyoucantuse.htm). Brookings Institution Press.
- Warshofsky, Fred (October 3, 1994). *The Patent Wars: The Battle to Own the World's Technology* (https://archive.org/details/patentwarsbattle00wars). Wiley. ISBN 978-0-471-59902-9.
- Information Feudalism – Peter Drahos Drahos, Peter (May 1, 2003). *Information Feudalism: Who Owns the Knowledge Economy?*. The New Press. ISBN 1-56584-804-7.
- Müller, Florian (2006). *No Lobbyists As Such: The War over Software Patents in the European Union* (https://web.archive.org/web/20090326022646/http://crashrecovery.org/NoLobbyistsAsSuch.pdf) (PDF). Archived from the original (http://crashrecovery.org/NoLobbyistsAsSuch.pdf) (PDF) on 2009-03-26. Retrieved 2008-05-28. – 377-page play-by-play memoir of the story of Florian's part in the campaign over the EU software patent directive.
- Leith, Philip (2007). *Software and Patents in Europe* (http://www.cambridge.org/gb/knowledge/isbn/item6487700/?site_locale=en_GB). Cambridge University Press.

Papers and presentations

- Edwards, David, *Platonism Is the Law of the Land* (http://www.ams.org/notices/201304/rnoti-p475.pdf) (PDF)
- Hunt, R, *You can patent that?* (https://web.archive.org/web/20130112210524/http://www.phil.frb.org/research-and-data/publications/business-review/2001/q1/brq101bh.pdf) (PDF), archived from the original (http://www.phil.frb.org/research-and-data/publications/business-review/2001/q1/brq101bh.pdf) (PDF) on 2013-01-12, retrieved 2019-01-21 – Overview of the legal history and trends
- Szabo, N., *Elemental Subject Matter* (https://web.archive.org/web/20060519040201/http://szabo.best.vwh.net/elementalsubjectmatter.pdf) (PDF), archived from the original (http://szabo.best.vwh.net/elementalsubjectmatter.pdf) (PDF) on 2006-05-19, retrieved 2006-03-25 – How software came to be patentable in the United States
- Pellegrini, François (2007), "Analysis of software patentability in Europe (Chapter 11)", *Software Patents – Legal Perspectives* (http://www.labri.fr/perso/pelegrin/papers/swpat_europe_20061030.pdf) (PDF), Amicus Books – ICFAI University Press, pp. 192–209, ISBN 978-81-314-0653-3
- Furutani, Hideo (2003), *Patentability of Business Method Inventions in Japan Compared with the US and Europe* (http://www.furutani.co.jp/office/ronbun/Business_method_patents_in_Japan.pdf) (PDF presentation), Arlington, Virginia; United States Patent and Trademark Office
- Stallman, Richard M. (May 24, 2004), *The Dangers of Software Patents* (https://web.archive.org/web/20130115221652/http://www.ifso.ie/documents/rms-2004-05-24.html), archived from the original (http://www.ifso.ie/documents/rms-2004-05-24.html) on 2013-01-15, retrieved 2007-01-29

- Abecassis, Alexandre. "Patenting Software Innovations: A brief overview of the situation in some jurisdictions of interest" (https://ewh.ieee.org/reg/7/canrev/cr55/IEEECanadianReview_no55.pdf) (PDF). *IEEE Canadian Review* (55). Institute of Electrical and Electronics Engineers (IEEE): 24–27.
- Kahin, Brian (April 1990). "The Software Patent Crisis" (http://reoxy.org/swpc.htm). *Technology Review*. Massachusetts Institute of Technology.

Computer security

(Redirected from Software security)

Computer security (also **cybersecurity**, **digital security**, or **information technology (IT) security**) is a subdiscipline within the field of information security. It consists of the protection of computer software, systems and networks from threats that can lead to unauthorized information disclosure, theft or damage to hardware, software, or data, as well as from the disruption or misdirection of the services they provide.[1][2]

An example of a physical security measure: a metal lock on the back of a personal computer to prevent hardware tampering.

The significance of the field stems from the expanded reliance on computer systems, the Internet,[3] and wireless network standards. Its importance is further amplified by the growth of smart devices, including smartphones, televisions, and the various devices that constitute the Internet of things (IoT). Cybersecurity has emerged as one of the most significant new challenges facing the contemporary world, due to both the complexity of information systems and the societies they support. Security is particularly crucial for systems that govern large-scale systems with far-reaching physical effects, such as power distribution, elections, and finance.[4][5]

Although many aspects of computer security involve digital security, such as electronic passwords and encryption, physical security measures such as metal locks are still used to prevent unauthorized tampering. IT security is not a perfect subset of information security, therefore does not completely align into the security convergence schema.

Vulnerabilities and attacks

A vulnerability refers to a flaw in the structure, execution, functioning, or internal oversight of a computer or system that compromises its security. Most of the vulnerabilities that have been discovered are documented in the Common Vulnerabilities and Exposures (CVE) database.[6] An *exploitable* vulnerability is one for which at least one working attack or *exploit* exists.[7] Actors maliciously seeking vulnerabilities are known as *threats*. Vulnerabilities can be researched, reverse-engineered, hunted, or exploited using automated tools or customized scripts.[8][9]

Various people or parties are vulnerable to cyber attacks; however, different groups are likely to experience different types of attacks more than others.[10]

In April 2023, the United Kingdom Department for Science, Innovation & Technology released a report on cyber attacks over the previous 12 months.[11] They surveyed 2,263 UK businesses, 1,174 UK registered charities, and 554 education institutions. The research found that "32% of businesses and 24% of charities overall recall any breaches or attacks from the last 12 months." These figures were much higher for "medium businesses (59%), large businesses (69%), and high-income charities with £500,000 or more in annual income (56%)."[11] Yet, although medium or large businesses are more often the victims, since larger companies have generally improved their security over the last decade, small and midsize businesses (SMBs) have also become increasingly vulnerable as they often "do not have advanced tools to defend the business."[10] SMBs are most likely to be affected by malware, ransomware, phishing, man-in-the-middle attacks, and Denial-of Service (DoS) Attacks.[10]

Normal internet users are most likely to be affected by untargeted cyberattacks.[12] These are where attackers indiscriminately target as many devices, services, or users as possible. They do this using techniques that take advantage of the openness of the Internet. These strategies mostly include phishing, ransomware, water holing and scanning.[12]

To secure a computer system, it is important to understand the attacks that can be made against it, and these threats can typically be classified into one of the following categories:

Backdoor

A backdoor in a computer system, a cryptosystem, or an algorithm is any secret method of bypassing normal authentication or security controls. These weaknesses may exist for many reasons, including original design or poor configuration.[13] Due to the nature of backdoors, they are of greater concern to companies and databases as opposed to individuals.

Backdoors may be added by an authorized party to allow some legitimate access or by an attacker for malicious reasons. Criminals often use malware to install backdoors, giving them remote administrative access to a system.[14] Once they have access, cybercriminals can "modify files, steal personal information, install unwanted software, and even take control of the entire computer."[14]

Backdoors can be difficult to detect, as they often remain hidden within the source code or system firmware intimate knowledge of the operating system of the computer.

Denial-of-service attack

Denial-of-service attacks (DoS) are designed to make a machine or network resource unavailable to its intended users.[15] Attackers can deny service to individual victims, such as by deliberately entering a wrong password enough consecutive times to cause the victim's account to be locked, or they may overload the capabilities of a machine or network and block all users at once. While a network attack from a single IP address can be blocked by adding a new firewall rule, many forms of distributed denial-of-service (DDoS) attacks are possible, where the attack comes from a large number of points. In this case, defending against these attacks is much more difficult. Such attacks can originate from the zombie computers of a botnet or from a range of other possible techniques, including distributed reflective denial-of-service (DRDoS), where innocent systems are fooled into sending traffic to the victim.[15] With such attacks, the amplification factor makes the attack easier for the attacker because they have to use little bandwidth themselves. To understand why attackers may carry out these attacks, see the 'attacker motivation' section.

Physical access attacks

A direct-access attack is when an unauthorized user (an attacker) gains physical access to a computer, most likely to directly copy data from it or steal information.[16] Attackers may also compromise security by making operating system modifications, installing software worms, keyloggers, covert listening devices or using wireless microphones. Even when the system is protected by standard security measures, these may be bypassed by booting another operating system or tool from a CD-ROM or other bootable media. Disk encryption and the Trusted Platform Module standard are designed to prevent these attacks.

Direct service attackers are related in concept to direct memory attacks which allow an attacker to gain direct access to a computer's memory.[17] The attacks "take advantage of a feature of modern computers that allows certain devices, such as external hard drives, graphics cards, or network cards, to access the computer's memory directly."[17]

Eavesdropping

Eavesdropping is the act of surreptitiously listening to a private computer conversation (communication), usually between hosts on a network. It typically occurs when a user connects to a network where traffic is not secured or encrypted and sends sensitive business data to a colleague, which, when listened to by an attacker, could be exploited.[18] Data transmitted across an *open network* allows an attacker to exploit a vulnerability and intercept it via various methods.

Unlike malware, direct-access attacks, or other forms of cyber attacks, eavesdropping attacks are unlikely to negatively affect the performance of networks or devices, making them difficult to notice.[18] In fact, "the attacker does not need to have any ongoing connection to the software at all. The attacker can insert the software onto a compromised device, perhaps by direct insertion or perhaps by a virus or other malware, and then come back some time later to retrieve any data that is found or trigger the software to send the data at some determined time."[19]

Using a virtual private network (VPN), which encrypts data between two points, is one of the most common forms of protection against eavesdropping. Using the best form of encryption possible for wireless networks is best practice, as well as using HTTPS instead of an unencrypted HTTP.[20]

Programs such as Carnivore and NarusInSight have been used by the Federal Bureau of Investigation (FBI) and NSA to eavesdrop on the systems of internet service providers. Even machines that operate as a closed system (i.e., with no contact with the outside world) can be eavesdropped upon by monitoring the faint electromagnetic transmissions generated by the hardware. TEMPEST is a specification by the NSA referring to these attacks.

Malware

Malicious software (malware) is any software code or computer program "intentionally written to harm a computer system or its users."[21] Once present on a computer, it can leak sensitive details such as personal information, business information and passwords, can give control of the system to the attacker, and can corrupt or delete data permanently.[22][23]

Types of malware

- **Viruses** are a specific type of malware, and are normally a malicious code that hijacks software with the intention to "do damage and spread copies of itself." Copies are made with the aim to spread to other programs on a computer.[21]
- **Worms** are similar to viruses, however viruses can only function when a user runs (opens) a compromised program. Worms are self-replicating malware that spread between programs, apps and devices *without* the need for human interaction.[21]
- **Trojan horses** are programs that pretend to be helpful or hide themselves within desired or legitimate software to "trick users into installing them." Once installed, a RAT (Remote Access Trojan) can create a secret backdoor on the affected device to cause damage.[21]
- **Spyware** is a type of malware that secretly gathers information from an infected computer and transmits the sensitive information back to the attacker. One of the most common forms of spyware are keyloggers, which record all of a user's keyboard inputs/keystrokes, to "allow hackers to harvest usernames, passwords, bank account and credit card numbers."[21]
- **Scareware**, as the name suggests, is a form of malware which uses social engineering (manipulation) to scare, shock, trigger anxiety, or suggest the perception of a threat in order to manipulate users into buying or installing unwanted software. These attacks often begin with a "sudden pop-up with an urgent message, usually warning the user that they've broken the law or their device has a virus."[21]
- **Ransomware** is when malware installs itself onto a victim's machine, encrypts their files, and then turns around and demands a ransom (usually in Bitcoin) to return that data to the user.

Man-in-the-middle attacks

Man-in-the-middle attacks (MITM) involve a malicious attacker trying to intercept, surveil or modify communications between two parties by spoofing one or both party's identities and injecting themselves in-between.[24] Types of MITM attacks include:

- IP address spoofing is where the attacker hijacks routing protocols to reroute the targets traffic to a vulnerable network node for traffic interception or injection.
- Message spoofing (via email, SMS or OTT messaging) is where the attacker spoofs the identity or carrier service while the target is using messaging protocols like email, SMS or OTT (IP-based) messaging apps. The attacker can then monitor conversations, launch social attacks or trigger zero-day-vulnerabilities to allow for further attacks.
- WiFi SSID spoofing is where the attacker simulates a WIFI base station SSID to capture and modify internet traffic and transactions. The attacker can also use local network addressing and reduced network defenses to penetrate the target's firewall by breaching known vulnerabilities. Sometimes known as a Pineapple attack thanks to a popular device. See also Malicious association.
- DNS spoofing is where attackers hijack domain name assignments to redirect traffic to systems under the attackers control, in order to surveil traffic or launch other attacks.
- SSL hijacking, typically coupled with another media-level MITM attack, is where the attacker spoofs the SSL authentication and encryption protocol by way of Certificate Authority injection in order to decrypt, surveil and modify traffic. See also TLS interception[24]

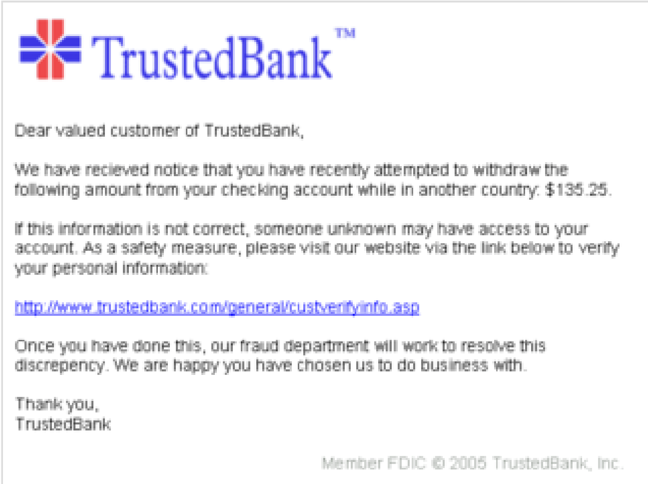
Multi-vector, polymorphic attacks

Surfacing in 2017, a new class of multi-vector,[25] polymorphic[26] cyber threats combine several types of attacks and change form to avoid cybersecurity controls as they spread.

Multi-vector polymorphic attacks, as the name describes, are both multi-vectored and polymorphic.[27] Firstly, they are a singular attack that involves multiple methods of attack. In this sense, they are "multi-vectored (i.e. the attack can use multiple means of propagation such as via the Web, email and applications." However, they are also multi-staged, meaning that "they can infiltrate networks and move laterally inside the network."[27] The attacks can be polymorphic, meaning that the cyberattacks used such as viruses, worms or trojans "constantly change ("morph") making it nearly impossible to detect them using signature-based defences."[27]

Phishing

Phishing is the attempt of acquiring sensitive information such as usernames, passwords, and credit card details directly from users by deceiving the users.[28] Phishing is typically carried out by email spoofing, instant messaging, text message, or on a phone call. They often direct users to enter details at a fake website whose look and feel are almost identical to the legitimate one.[29] The fake website often asks for personal information, such as login details and passwords. This information can then be used to gain access to the individual's real account on the real website.



An example of a phishing email, disguised as an official email from a (fictional) bank. The sender is attempting to trick the recipient into revealing confidential information by confirming it at the phisher's website. Note the misspelling of the words *received* and *discrepancy* as recieved and discrepency, respectively. Although the URL of the bank's webpage appears to be legitimate, the hyperlink points at the phisher's webpage.

Preying on a victim's trust, phishing can be classified as a form of social engineering. Attackers can use creative ways to gain access to real accounts. A common scam is for attackers to send fake electronic invoices[30] to individuals showing that they recently purchased music, apps, or others, and instructing them to click on a link if the purchases were not authorized. A more strategic type of phishing is spear-phishing which leverages personal or organization-specific details to make the attacker appear like a trusted source. Spear-phishing attacks target specific individuals, rather than the broad net cast by phishing attempts.[31]

Privilege escalation

Privilege escalation describes a situation where an attacker with some level of restricted access is able to, without authorization, elevate their privileges or access level.[32] For example, a standard computer user may be able to exploit a vulnerability in the system to gain access to restricted data; or even become *root* and have full unrestricted access to a system. The severity of attacks can range from attacks simply sending an unsolicited email to a ransomware attack on large amounts of data. Privilege escalation usually starts with social engineering techniques, often phishing.[32]

Privilege escalation can be separated into two strategies, horizontal and vertical privilege escalation:

- Horizontal escalation (or account takeover) is where an attacker gains access to a normal user account that has relatively low-level privileges. This may be through stealing the user's username and password. Once they have access, they have gained a *foothold*, and using this foothold the attacker then may move around the network of users at this same lower level, gaining access to information of this similar privilege.[32]

- Vertical escalation however targets people higher up in a company and often with more administrative power, such as an employee in IT with a higher privilege. Using this privileged account will then enable the attacker to invade other accounts.[32]

Side-channel attack

Any computational system affects its environment in some form. This effect it has on its environment can range from electromagnetic radiation, to residual effect on RAM cells which as a consequence make a Cold boot attack possible, to hardware implementation faults that allow for access or guessing of other values that normally should be inaccessible. In Side-channel attack scenarios, the attacker would gather such information about a system or network to guess its internal state and as a result access the information which is assumed by the victim to be secure. The target information in a side channel can be challenging to detect due to its low amplitude when combined with other signals [33]

Social engineering

Social engineering, in the context of computer security, aims to convince a user to disclose secrets such as passwords, card numbers, etc. or grant physical access by, for example, impersonating a senior executive, bank, a contractor, or a customer.[34] This generally involves exploiting people's trust, and relying on their cognitive biases. A common scam involves emails sent to accounting and finance department personnel, impersonating their CEO and urgently requesting some action. One of the main techniques of social engineering are phishing attacks.

In early 2016, the FBI reported that such business email compromise (BEC) scams had cost US businesses more than $2 billion in about two years.[35]

In May 2016, the Milwaukee Bucks NBA team was the victim of this type of cyber scam with a perpetrator impersonating the team's president Peter Feigin, resulting in the handover of all the team's employees' 2015 W-2 tax forms.[36]

Spoofing

Spoofing is an act of pretending to be a valid entity through the falsification of data (such as an IP address or username), in order to gain access to information or resources that one is otherwise unauthorized to obtain. Spoofing is closely related to phishing.[37][38] There are several types of spoofing, including:

- Email spoofing, is where an attacker forges the sending (*From*, or source) address of an email.
- IP address spoofing, where an attacker alters the source IP address in a network packet to hide their identity or impersonate another computing system.
- MAC spoofing, where an attacker modifies the Media Access Control (MAC) address of their network interface controller to obscure their identity, or to pose as another.
- Biometric spoofing, where an attacker produces a fake biometric sample to pose as another user.[39]
- Address Resolution Protocol (ARP) spoofing, where an attacker sends spoofed address resolution protocol onto a local area network to associate their Media Access Control

address with a different host's IP address. This causes data to be sent to the attacker rather than the intended host.

In 2018, the cybersecurity firm Trellix published research on the life-threatening risk of spoofing in the healthcare industry.[40]

Tampering

Tampering describes a malicious modification or alteration of data. It is an intentional but unauthorized act resulting in the modification of a system, components of systems, its intended behavior, or data. So-called Evil Maid attacks and security services planting of surveillance capability into routers are examples.[41]

HTML smuggling

HTML smuggling allows an attacker to *smuggle* a malicious code inside a particular HTML or web page.[42] HTML files can carry payloads concealed as benign, inert data in order to defeat content filters. These payloads can be reconstructed on the other side of the filter.[43]

When a target user opens the HTML, the malicious code is activated; the web browser then *decodes* the script, which then unleashes the malware onto the target's device.[42]

Information security practices

Employee behavior can have a big impact on information security in organizations. Cultural concepts can help different segments of the organization work effectively or work against effectiveness toward information security within an organization. Information security culture is the "...totality of patterns of behavior in an organization that contributes to the protection of information of all kinds."[44]

Andersson and Reimers (2014) found that employees often do not see themselves as part of their organization's information security effort and often take actions that impede organizational changes.[45] Indeed, the Verizon Data Breach Investigations Report 2020, which examined 3,950 security breaches, discovered 30% of cybersecurity incidents involved internal actors within a company.[46] Research shows information security culture needs to be improved continuously. In "Information Security Culture from Analysis to Change", authors commented, "It's a never-ending process, a cycle of evaluation and change or maintenance." To manage the information security culture, five steps should be taken: pre-evaluation, strategic planning, operative planning, implementation, and post-evaluation.[47]

- Pre-evaluation: To identify the awareness of information security within employees and to analyze the current security policies.
- Strategic planning: To come up with a better awareness program, clear targets need to be set. Assembling a team of skilled professionals is helpful to achieve it.
- Operative planning: A good security culture can be established based on internal communication, management buy-in, security awareness and a training program.[47]
- Implementation: Four stages should be used to implement the information security culture. They are:

    1. Commitment of the management

2. Communication with organizational members
3. Courses for all organizational members
4. Commitment of the employees[47]

- Post-evaluation: To assess the success of the planning and implementation, and to identify unresolved areas of concern.

Computer protection (countermeasures)

In computer security, a countermeasure is an action, device, procedure or technique that reduces a threat, a vulnerability, or an attack by eliminating or preventing it, by minimizing the harm it can cause, or by discovering and reporting it so that corrective action can be taken.[48][49][50]

Some common countermeasures are listed in the following sections:

Security by design

Security by design, or alternately secure by design, means that the software has been designed from the ground up to be secure. In this case, security is considered a main feature.

The UK government's National Cyber Security Centre separates secure cyber design principles into five sections:[51]

1. Before a secure system is created or updated, companies should ensure they understand the fundamentals and the context around the system they are trying to create and identify any weaknesses in the system.
2. Companies should design and centre their security around techniques and defences which make attacking their data or systems inherently more challenging for attackers.
3. Companies should ensure that their core services that rely on technology are protected so that the systems are essentially never down.
4. Although systems can be created which are safe against a multitude of attacks, that does not mean that attacks will not be attempted. Despite one's security, all companies' systems should aim to be able to detect and spot attacks as soon as they occur to ensure the most effective response to them.
5. Companies should create secure systems designed so that any attack that is successful has minimal severity.

These design principles of security by design can include some of the following techniques:

- The principle of least privilege, where each part of the system has only the privileges that are needed for its function. That way, even if an attacker gains access to that part, they only have limited access to the whole system.
- Automated theorem proving to prove the correctness of crucial software subsystems.
- Code reviews and unit testing, approaches to make modules more secure where formal correctness proofs are not possible.
- Defense in depth, where the design is such that more than one subsystem needs to be violated to compromise the integrity of the system and the information it holds.
- Default secure settings, and design to *fail secure* rather than *fail insecure* (see fail-safe for the equivalent in safety engineering). Ideally, a secure system should require a deliberate,

conscious, knowledgeable and free decision on the part of legitimate authorities in order to make it insecure.

- Audit trails track system activity so that when a security breach occurs, the mechanism and extent of the breach can be determined. Storing audit trails remotely, where they can only be appended to, can keep intruders from covering their tracks.
- Full disclosure of all vulnerabilities, to ensure that the *window of vulnerability* is kept as short as possible when bugs are discovered.

Security architecture

Security architecture can be defined as the "practice of designing computer systems to achieve security goals."[52] These goals have overlap with the principles of "security by design" explored above, including to "make initial compromise of the system difficult," and to "limit the impact of any compromise."[52] In practice, the role of a security architect would be to ensure the structure of a system reinforces the security of the system, and that new changes are safe and meet the security requirements of the organization.[53][54]

Similarly, Techopedia defines security architecture as "a unified security design that addresses the necessities and potential risks involved in a certain scenario or environment. It also specifies when and where to apply security controls. The design process is generally reproducible." The key attributes of security architecture are:[55]

- the relationship of different components and how they depend on each other.
- determination of controls based on risk assessment, good practices, finances, and legal matters.
- the standardization of controls.

Practicing security architecture provides the right foundation to systematically address business, IT and security concerns in an organization.

Security measures

A state of computer security is the conceptual ideal, attained by the use of three processes: threat prevention, detection, and response. These processes are based on various policies and system components, which include the following:

- Limiting the access of individuals using user account access controls and using cryptography can protect systems files and data, respectively.
- Firewalls are by far the most common prevention systems from a network security perspective as they can (if properly configured) shield access to internal network services and block certain kinds of attacks through packet filtering. Firewalls can be both hardware and software-based. Firewalls monitor and control incoming and outgoing traffic of a computer network and establish a barrier between a trusted network and an untrusted network.[56]
- Intrusion Detection System (IDS) products are designed to detect network attacks in-progress and assist in post-attack forensics, while audit trails and logs serve a similar function for individual systems.
- *Response* is necessarily defined by the assessed security requirements of an individual system and may cover the range from simple upgrade of protections to notification of legal authorities, counter-attacks, and the like. In some special cases, the complete destruction of

the compromised system is favored, as it may happen that not all the compromised resources are detected.
- Cyber security awareness training to cope with cyber threats and attacks.[57]
- Forward web proxy solutions can prevent the client to visit malicious web pages and inspect the content before downloading to the client machines.

Today, computer security consists mainly of preventive measures, like firewalls or an exit procedure. A firewall can be defined as a way of filtering network data between a host or a network and another network, such as the Internet. They can be implemented as software running on the machine, hooking into the network stack (or, in the case of most UNIX-based operating systems such as Linux, built into the operating system kernel) to provide real-time filtering and blocking.[56] Another implementation is a so-called *physical firewall*, which consists of a separate machine filtering network traffic. Firewalls are common amongst machines that are permanently connected to the Internet.

Some organizations are turning to big data platforms, such as Apache Hadoop, to extend data accessibility and machine learning to detect advanced persistent threats.[58]

In order to ensure adequate security, the confidentiality, integrity and availability of a network, better known as the CIA triad, must be protected and is considered the foundation to information security.[59] To achieve those objectives, administrative, physical and technical security measures should be employed. The amount of security afforded to an asset can only be determined when its value is known.[60]

Vulnerability management

Vulnerability management is the cycle of identifying, fixing or mitigating vulnerabilities,[61] especially in software and firmware. Vulnerability management is integral to computer security and network security.

Vulnerabilities can be discovered with a vulnerability scanner, which analyzes a computer system in search of known vulnerabilities,[62] such as open ports, insecure software configuration, and susceptibility to malware. In order for these tools to be effective, they must be kept up to date with every new update the vendor release. Typically, these updates will scan for the new vulnerabilities that were introduced recently.

Beyond vulnerability scanning, many organizations contract outside security auditors to run regular penetration tests against their systems to identify vulnerabilities. In some sectors, this is a contractual requirement.[63]

Reducing vulnerabilities

The act of assessing and reducing vulnerabilities to cyber attacks is commonly referred to as information technology security assessments. They aim to assess systems for risk and to predict and test for their vulnerabilities. While formal verification of the correctness of computer systems is possible,[64][65] it is not yet common. Operating systems formally verified include seL4,[66] and SYSGO's PikeOS[67][68] – but these make up a very small percentage of the market.

It is possible to reduce an attacker's chances by keeping systems up to date with security patches and updates and by hiring people with expertise in security. Large companies with significant threats can hire Security Operations Centre (SOC) Analysts. These are specialists in cyber defences, with their role ranging from "conducting threat analysis to investigating reports of any new issues and preparing and testing disaster recovery plans."[69]

Whilst no measures can completely guarantee the prevention of an attack, these measures can help mitigate the damage of possible attacks. The effects of data loss/damage can be also reduced by careful backing up and insurance.

Outside of formal assessments, there are various methods of reducing vulnerabilities. Two factor authentication is a method for mitigating unauthorized access to a system or sensitive information.[70] It requires *something you know:* a password or PIN, and *something you have*: a card, dongle, cellphone, or another piece of hardware. This increases security as an unauthorized person needs both of these to gain access.

Protecting against social engineering and direct computer access (physical) attacks can only happen by non-computer means, which can be difficult to enforce, relative to the sensitivity of the information. Training is often involved to help mitigate this risk by improving people's knowledge of how to protect themselves and by increasing people's awareness of threats.[71] However, even in highly disciplined environments (e.g. military organizations), social engineering attacks can still be difficult to foresee and prevent.

Inoculation, derived from inoculation theory, seeks to prevent social engineering and other fraudulent tricks and traps by instilling a resistance to persuasion attempts through exposure to similar or related attempts.[72]

Hardware protection mechanisms

Hardware-based or assisted computer security also offers an alternative to software-only computer security. Using devices and methods such as dongles, trusted platform modules, intrusion-aware cases, drive locks, disabling USB ports, and mobile-enabled access may be considered more secure due to the physical access (or sophisticated backdoor access) required in order to be compromised. Each of these is covered in more detail below.

- USB dongles are typically used in software licensing schemes to unlock software capabilities,[73] but they can also be seen as a way to prevent unauthorized access to a computer or other device's software. The dongle, or key, essentially creates a secure encrypted tunnel between the software application and the key. The principle is that an encryption scheme on the dongle, such as Advanced Encryption Standard (AES) provides a stronger measure of security since it is harder to hack and replicate the dongle than to simply copy the native software to another machine and use it. Another security application for dongles is to use them for accessing web-based content such as cloud software or Virtual Private Networks (VPNs).[74] In addition, a USB dongle can be configured to lock or unlock a computer.[75]
- Trusted platform modules (TPMs) secure devices by integrating cryptographic capabilities onto access devices, through the use of microprocessors, or so-called computers-on-a-chip. TPMs used in conjunction with server-side software offer a way to detect and authenticate hardware devices, preventing unauthorized network and data access.[76]
- Computer case intrusion detection refers to a device, typically a push-button switch, which detects when a computer case is opened. The firmware or BIOS is programmed to show an alert to the operator when the computer is booted up the next time.

- Drive locks are essentially software tools to encrypt hard drives, making them inaccessible to thieves.[77] Tools exist specifically for encrypting external drives as well.[78]
- Disabling USB ports is a security option for preventing unauthorized and malicious access to an otherwise secure computer. Infected USB dongles connected to a network from a computer inside the firewall are considered by the magazine Network World as the most common hardware threat facing computer networks.
- Disconnecting or disabling peripheral devices (like camera, GPS, removable storage, etc.), that are not in use.[79]
- Mobile-enabled access devices are growing in popularity due to the ubiquitous nature of cell phones.[80] Built-in capabilities such as Bluetooth, the newer Bluetooth low energy (LE), near-field communication (NFC) on non-iOS devices and biometric validation such as thumbprint readers, as well as QR code reader software designed for mobile devices, offer new, secure ways for mobile phones to connect to access control systems. These control systems provide computer security and can also be used for controlling access to secure buildings.[81]
- IOMMUs allow for hardware-based sandboxing of components in mobile and desktop computers by utilizing direct memory access protections.[82][83]
- Physical Unclonable Functions (PUFs) can be used as a digital fingerprint or a unique identifier to integrated circuits and hardware, providing users the ability to secure the hardware supply chains going into their systems.[84][85]

Secure operating systems

One use of the term *computer security* refers to technology that is used to implement secure operating systems. Using secure operating systems is a good way of ensuring computer security. These are systems that have achieved certification from an external security-auditing organization, the most popular evaluations are Common Criteria (CC).[86]

Secure coding

In software engineering, secure coding aims to guard against the accidental introduction of security vulnerabilities. It is also possible to create software designed from the ground up to be secure. Such systems are *secure by design*. Beyond this, formal verification aims to prove the correctness of the algorithms underlying a system;[87] important for cryptographic protocols for example.

Capabilities and access control lists

Within computer systems, two of the main security models capable of enforcing privilege separation are access control lists (ACLs) and role-based access control (RBAC).

An access-control list (ACL), with respect to a computer file system, is a list of permissions associated with an object. An ACL specifies which users or system processes are granted access to objects, as well as what operations are allowed on given objects.

Role-based access control is an approach to restricting system access to authorized users,[88][89][90] used by the majority of enterprises with more than 500 employees,[91] and can implement mandatory access control (MAC) or discretionary access control (DAC).

A further approach, capability-based security has been mostly restricted to research operating systems. Capabilities can, however, also be implemented at the language level, leading to a style of programming that is essentially a refinement of standard object-oriented design. An open-source project in the area is the E language.

User security training

The end-user is widely recognized as the weakest link in the security chain[92] and it is estimated that more than 90% of security incidents and breaches involve some kind of human error.[93][94] Among the most commonly recorded forms of errors and misjudgment are poor password management, sending emails containing sensitive data and attachments to the wrong recipient, the inability to recognize misleading URLs and to identify fake websites and dangerous email attachments. A common mistake that users make is saving their user id/password in their browsers to make it easier to log in to banking sites. This is a gift to attackers who have obtained access to a machine by some means. The risk may be mitigated by the use of two-factor authentication.[95]

As the human component of cyber risk is particularly relevant in determining the global cyber risk[96] an organization is facing, security awareness training, at all levels, not only provides formal compliance with regulatory and industry mandates but is considered essential[97] in reducing cyber risk and protecting individuals and companies from the great majority of cyber threats.

The focus on the end-user represents a profound cultural change for many security practitioners, who have traditionally approached cybersecurity exclusively from a technical perspective, and moves along the lines suggested by major security centers[98] to develop a culture of cyber awareness within the organization, recognizing that a security-aware user provides an important line of defense against cyber attacks.

Digital hygiene

Related to end-user training, **digital hygiene** or **cyber hygiene** is a fundamental principle relating to information security and, as the analogy with personal hygiene shows, is the equivalent of establishing simple routine measures to minimize the risks from cyber threats. The assumption is that good cyber hygiene practices can give networked users another layer of protection, reducing the risk that one vulnerable node will be used to either mount attacks or compromise another node or network, especially from common cyberattacks.[99] Cyber hygiene should also not be mistaken for proactive cyber defence, a military term.[100]

The most common acts of digital hygiene can include updating malware protection, cloud back-ups, passwords, and ensuring restricted admin rights and network firewalls.[101] As opposed to a purely technology-based defense against threats, cyber hygiene mostly regards routine measures that are technically simple to implement and mostly dependent on discipline[102] or education.[103] It can be thought of as an abstract list of tips or measures that have been demonstrated as having a positive effect on personal or collective digital security. As such, these measures can be performed by laypeople, not just security experts.

Cyber hygiene relates to personal hygiene as computer viruses relate to biological viruses (or pathogens). However, while the term *computer virus* was coined almost simultaneously with the creation of the first working computer viruses,[104] the term *cyber hygiene* is a much later invention, perhaps as late as

2000[105] by Internet pioneer Vint Cerf. It has since been adopted by the Congress[106] and Senate of the United States,[107] the FBI,[108] EU institutions[99] and heads of state.[100]

Difficulty of responding to breaches

Responding to attempted security breaches is often very difficult for a variety of reasons, including:

- Identifying attackers is difficult, as they may operate through proxies, temporary anonymous dial-up accounts, wireless connections, and other anonymizing procedures which make back-tracing difficult – and are often located in another jurisdiction. If they successfully breach security, they have also often gained enough administrative access to enable them to delete logs to cover their tracks.
- The sheer number of attempted attacks, often by automated vulnerability scanners and computer worms, is so large that organizations cannot spend time pursuing each.
- Law enforcement officers often lack the skills, interest or budget to pursue attackers. Furthermore, identifying attackers across a network may necessitate collecting logs from multiple locations within the network and across various countries, a process that can be both difficult and time-consuming.

Where an attack succeeds and a breach occurs, many jurisdictions now have in place mandatory security breach notification laws.

Types of security and privacy

- Access control
- Anti-keyloggers
- Anti-malware
- Anti-spyware
- Anti-subversion software
- Anti-tamper software
- Anti-theft
- Antivirus software
- Cryptographic software
- Computer-aided dispatch (CAD)
- Data loss prevention software
- Firewall
- Intrusion detection system (IDS)
- Intrusion prevention system (IPS)
- Log management software
- Parental control
- Records management
- Sandbox
- Security information management
- Security information and event management (SIEM)
- Software and operating system updating
- Vulnerability Management

Systems at risk

The growth in the number of computer systems and the increasing reliance upon them by individuals, businesses, industries, and governments means that there are an increasing number of systems at risk.

Financial systems

The computer systems of financial regulators and financial institutions like the U.S. Securities and Exchange Commission, SWIFT, investment banks, and commercial banks are prominent hacking targets for cybercriminals interested in manipulating markets and making illicit gains.[109] Websites and apps that accept or store credit card numbers, brokerage accounts, and bank account information are also prominent hacking targets, because of the potential for immediate financial gain from transferring money, making purchases, or selling the information on the black market.[110] In-store payment systems and ATMs have also been tampered with in order to gather customer account data and PINs.

The UCLA Internet Report: Surveying the Digital Future (2000) found that the privacy of personal data created barriers to online sales and that more than nine out of 10 internet users were somewhat or very concerned about credit card security.[111]

The most common web technologies for improving security between browsers and websites are named SSL (Secure Sockets Layer), and its successor TLS (Transport Layer Security), identity management and authentication services, and domain name services allow companies and consumers to engage in secure communications and commerce. Several versions of SSL and TLS are commonly used today in applications such as web browsing, e-mail, internet faxing, instant messaging, and VoIP (voice-over-IP). There are various interoperable implementations of these technologies, including at least one implementation that is open source. Open source allows anyone to view the application's source code, and look for and report vulnerabilities.

The credit card companies Visa and MasterCard cooperated to develop the secure EMV chip which is embedded in credit cards. Further developments include the Chip Authentication Program where banks give customers hand-held card readers to perform online secure transactions. Other developments in this arena include the development of technology such as Instant Issuance which has enabled shopping mall kiosks acting on behalf of banks to issue on-the-spot credit cards to interested customers.

Utilities and industrial equipment

Computers control functions at many utilities, including coordination of telecommunications, the power grid, nuclear power plants, and valve opening and closing in water and gas networks. The Internet is a potential attack vector for such machines if connected, but the Stuxnet worm demonstrated that even equipment controlled by computers not connected to the Internet can be vulnerable. In 2014, the Computer Emergency Readiness Team, a division of the Department of Homeland Security, investigated 79 hacking incidents at energy companies.[112]

Aviation

The aviation industry is very reliant on a series of complex systems which could be attacked.[113] A simple power outage at one airport can cause repercussions worldwide,[114] much of the system relies on radio transmissions which could be disrupted,[115] and controlling aircraft over oceans is especially dangerous because radar surveillance only extends 175 to 225 miles offshore.[116] There is also potential for attack from within an aircraft.[117]

Implementing fixes in aerospace systems poses a unique challenge because efficient air transportation is heavily affected by weight and volume. Improving security by adding physical devices to airplanes could increase their unloaded weight, and could potentially reduce cargo or passenger capacity.[118]

In Europe, with the (Pan-European Network Service)[119] and NewPENS,[120] and in the US with the NextGen program,[121] air navigation service providers are moving to create their own dedicated networks.

Many modern passports are now biometric passports, containing an embedded microchip that stores a digitized photograph and personal information such as name, gender, and date of birth. In addition, more countries are introducing facial recognition technology to reduce identity-related fraud. The introduction of the ePassport has assisted border officials in verifying the identity of the passport holder, thus allowing for quick passenger processing.[122] Plans are under way in the US, the UK, and Australia to introduce SmartGate kiosks with both retina and fingerprint recognition technology.[123] The airline industry is moving from the use of traditional paper tickets towards the use of electronic tickets (e-tickets). These have been made possible by advances in online credit card transactions in partnership with the airlines. Long-distance bus companies are also switching over to e-ticketing transactions today.

The consequences of a successful attack range from loss of confidentiality to loss of system integrity, air traffic control outages, loss of aircraft, and even loss of life.

Consumer devices

Desktop computers and laptops are commonly targeted to gather passwords or financial account information or to construct a botnet to attack another target. Smartphones, tablet computers, smart watches, and other mobile devices such as quantified self devices like activity trackers have sensors such as cameras, microphones, GPS receivers, compasses, and accelerometers which could be exploited, and may collect personal information, including sensitive health information. WiFi, Bluetooth, and cell phone networks on any of these devices could be used as attack vectors, and sensors might be remotely activated after a successful breach.[124]

The increasing number of home automation devices such as the Nest thermostat are also potential targets.[124]

Healthcare

Today many healthcare providers and health insurance companies use the internet to provide enhanced products and services. Examples are the use of tele-health to potentially offer better quality and access to healthcare, or fitness trackers to lower insurance premiums. Patient records are increasingly being placed on secure in-house networks, alleviating the need for extra storage space.[125]

Large corporations

Large corporations are common targets. In many cases attacks are aimed at financial gain through identity theft and involve data breaches. Examples include the loss of millions of clients' credit card and financial details by Home Depot,[126] Staples,[127] Target Corporation,[128] and Equifax.[129]

Medical records have been targeted in general identify theft, health insurance fraud, and impersonating patients to obtain prescription drugs for recreational purposes or resale.[130] Although cyber threats continue to increase, 62% of all organizations did not increase security training for their business in 2015.[131]

Not all attacks are financially motivated, however: security firm HBGary Federal had a serious series of attacks in 2011 from hacktivist group Anonymous in retaliation for the firm's CEO claiming to have infiltrated their group,[132][133] and Sony Pictures was hacked in 2014 with the apparent dual motive of embarrassing the company through data leaks and crippling the company by wiping workstations and servers.[134][135]

Automobiles

Vehicles are increasingly computerized, with engine timing, cruise control, anti-lock brakes, seat belt tensioners, door locks, airbags and advanced driver-assistance systems on many models. Additionally, connected cars may use WiFi and Bluetooth to communicate with onboard consumer devices and the cell phone network.[136] Self-driving cars are expected to be even more complex. All of these systems carry some security risks, and such issues have gained wide attention.[137][138][139]

Simple examples of risk include a malicious compact disc being used as an attack vector,[140] and the car's onboard microphones being used for eavesdropping. However, if access is gained to a car's internal controller area network, the danger is much greater[136] – and in a widely publicized 2015 test, hackers remotely carjacked a vehicle from 10 miles away and drove it into a ditch.[141][142]

Manufacturers are reacting in numerous ways, with Tesla in 2016 pushing out some security fixes *over the air* into its cars' computer systems.[143] In the area of autonomous vehicles, in September 2016 the United States Department of Transportation announced some initial safety standards, and called for states to come up with uniform policies.[144][145][146]

Additionally, e-Drivers' licenses are being developed using the same technology. For example, Mexico's licensing authority (ICV) has used a smart card platform to issue the first e-Drivers' licenses to the city of Monterrey, in the state of Nuevo León.[147]

Shipping

Shipping companies[148] have adopted RFID (Radio Frequency Identification) technology as an efficient, digitally secure, tracking device. Unlike a barcode, RFID can be read up to 20 feet away. RFID is used by FedEx[149] and UPS.[150]

Government

Government and military computer systems are commonly attacked by activists[151][152][153] and foreign powers.[154][155][156][157] Local and regional government infrastructure such as traffic light controls, police and intelligence agency communications, personnel records, as well as student records.[158]

The FBI, CIA, and Pentagon, all utilize secure controlled access technology for any of their buildings. However, the use of this form of technology is spreading into the entrepreneurial world. More and more companies are taking advantage of the development of digitally secure controlled access technology. GE's ACUVision, for example, offers a single panel platform for access control, alarm monitoring and digital recording.[159]

Internet of things and physical vulnerabilities

The Internet of things (IoT) is the network of physical objects such as devices, vehicles, and buildings that are embedded with electronics, software, sensors, and network connectivity that enables them to collect and exchange data.[160] Concerns have been raised that this is being developed without appropriate consideration of the security challenges involved.[161][162]

While the IoT creates opportunities for more direct integration of the physical world into computer-based systems,[163][164] it also provides opportunities for misuse. In particular, as the Internet of Things spreads widely, cyberattacks are likely to become an increasingly physical (rather than simply virtual) threat.[165] If a front door's lock is connected to the Internet, and can be locked/unlocked from a phone, then a criminal could enter the home at the press of a button from a stolen or hacked phone. People could stand to lose much more than their credit card numbers in a world controlled by IoT-enabled devices. Thieves have also used electronic means to circumvent non-Internet-connected hotel door locks.[166]

An attack aimed at physical infrastructure or human lives is often called a cyber-kinetic attack. As IoT devices and appliances become more widespread, the prevalence and potential damage of cyber-kinetic attacks can increase substantially.

Medical systems

Medical devices have either been successfully attacked or had potentially deadly vulnerabilities demonstrated, including both in-hospital diagnostic equipment[167] and implanted devices including pacemakers[168] and insulin pumps.[169] There are many reports of hospitals and hospital organizations getting hacked, including ransomware attacks,[170][171][172][173] Windows XP exploits,[174][175] viruses,[176][177] and data breaches of sensitive data stored on hospital servers.[178][171][179][180] On 28 December 2016 the US Food and Drug Administration released its recommendations for how medical device manufacturers should maintain the security of Internet-connected devices – but no structure for enforcement.[181][182]

Energy sector

In distributed generation systems, the risk of a cyber attack is real, according to *Daily Energy Insider*. An attack could cause a loss of power in a large area for a long period of time, and such an attack could have just as severe consequences as a natural disaster. The District of Columbia is considering creating a Distributed Energy Resources (DER) Authority within the city, with the goal being for customers to have more insight into their own energy use and giving the local electric utility, Pepco, the chance to better estimate energy demand. The D.C. proposal, however, would "allow third-party vendors to create numerous points of energy distribution, which could potentially create more opportunities for cyber attackers to threaten the electric grid."[183]

Telecommunications

Perhaps the most widely known digitally secure telecommunication device is the SIM (Subscriber Identity Module) card, a device that is embedded in most of the world's cellular devices before any service can be obtained. The SIM card is just the beginning of this digitally secure environment.

The Smart Card Web Servers draft standard (SCWS) defines the interfaces to an HTTP server in a smart card.[184] Tests are being conducted to secure OTA ("over-the-air") payment and credit card information from and to a mobile phone. Combination SIM/DVD devices are being developed through Smart Video Card technology which embeds a DVD-compliant optical disc into the card body of a regular SIM card.

Other telecommunication developments involving digital security include mobile signatures, which use the embedded SIM card to generate a legally binding electronic signature.

Cost and impact of security breaches

Serious financial damage has been caused by security breaches, but because there is no standard model for estimating the cost of an incident, the only data available is that which is made public by the organizations involved. "Several computer security consulting firms produce estimates of total worldwide losses attributable to virus and worm attacks and to hostile digital acts in general. The 2003 loss estimates by these firms range from $13 billion (worms and viruses only) to $226 billion (for all forms of covert attacks). The reliability of these estimates is often challenged; the underlying methodology is basically anecdotal."[185]

However, reasonable estimates of the financial cost of security breaches can actually help organizations make rational investment decisions. According to the classic Gordon-Loeb Model analyzing the optimal investment level in information security, one can conclude that the amount a firm spends to protect information should generally be only a small fraction of the expected loss (i.e., the expected value of the loss resulting from a cyber/information security breach).[186]

Attacker motivation

As with physical security, the motivations for breaches of computer security vary between attackers. Some are thrill-seekers or vandals, some are activists, others are criminals looking for financial gain. State-sponsored attackers are now common and well resourced but started with amateurs such as Markus Hess who hacked for the KGB, as recounted by Clifford Stoll in *The Cuckoo's Egg*.

Attackers motivations can vary for all types of attacks from pleasure to political goals.[15] For example, hacktivists may target a company or organization that carries out activities they do not agree with. This would be to create bad publicity for the company by having its website crash.

High capability hackers, often with larger backing or state sponsorship, may attack based on the demands of their financial backers. These attacks are more likely to attempt more serious attack. An example of a more serious attack was the 2015 Ukraine power grid hack, which reportedly utilised the spear-phising, destruction of files, and denial-of-service attacks to carry out the full attack.[187][188]

Additionally, recent attacker motivations can be traced back to extremist organizations seeking to gain political advantage or disrupt social agendas.[189] The growth of the internet, mobile technologies, and inexpensive computing devices have led to a rise in capabilities but also to the risk to environments that are deemed as vital to operations. All critical targeted environments are susceptible to compromise and this has led to a series of proactive studies on how to migrate the risk by taking into consideration motivations by these types of actors. Several stark differences exist between the hacker motivation and that of nation state actors seeking to attack based on an ideological preference.[190]

A key aspect of threat modeling for any system is identifying the motivations behind potential attacks and the individuals or groups likely to carry them out. The level and detail of security measures will differ based on the specific system being protected. For instance, a home personal computer, a bank, and a classified military network each face distinct threats, despite using similar underlying technologies.[191]

Computer security incident management

Computer security incident management is an organized approach to addressing and managing the aftermath of a computer security incident or compromise with the goal of preventing a breach or thwarting a cyberattack. An incident that is not identified and managed at the time of intrusion typically escalates to a more damaging event such as a data breach or system failure. The intended outcome of a computer security incident response plan is to contain the incident, limit damage and assist recovery to business as usual. Responding to compromises quickly can mitigate exploited vulnerabilities, restore services and processes and minimize losses.[192] Incident response planning allows an organization to establish a series of best practices to stop an intrusion before it causes damage. Typical incident response plans contain a set of written instructions that outline the organization's response to a cyberattack. Without a documented plan in place, an organization may not successfully detect an intrusion or compromise and stakeholders may not understand their roles, processes and procedures during an escalation, slowing the organization's response and resolution.

There are four key components of a computer security incident response plan:

1. Preparation: Preparing stakeholders on the procedures for handling computer security incidents or compromises
2. Detection and analysis: Identifying and investigating suspicious activity to confirm a security incident, prioritizing the response based on impact and coordinating notification of the incident
3. Containment, eradication and recovery: Isolating affected systems to prevent escalation and limit impact, pinpointing the genesis of the incident, removing malware, affected systems and bad actors from the environment and restoring systems and data when a threat no longer remains
4. Post incident activity: Post mortem analysis of the incident, its root cause and the organization's response with the intent of improving the incident response plan and future response efforts.[193]

Notable attacks and breaches

Some illustrative examples of different types of computer security breaches are given below.

Robert Morris and the first computer worm

In 1988, 60,000 computers were connected to the Internet, and most were mainframes, minicomputers and professional workstations. On 2 November 1988, many started to slow down, because they were running a malicious code that demanded processor time and that spread itself to other computers – the first internet computer worm.[194] The software was traced back to 23-year-old Cornell University graduate student Robert Tappan Morris who said "he wanted to count how many machines were connected to the Internet".[194]

Rome Laboratory

In 1994, over a hundred intrusions were made by unidentified crackers into the Rome Laboratory, the US Air Force's main command and research facility. Using trojan horses, hackers were able to obtain unrestricted access to Rome's networking systems and remove traces of their activities. The intruders were able to obtain classified files, such as air tasking order systems data and furthermore able to penetrate connected networks of National Aeronautics and Space Administration's Goddard Space Flight Center, Wright-Patterson Air Force Base, some Defense contractors, and other private sector organizations, by posing as a trusted Rome center user.[195]

TJX customer credit card details

In early 2007, American apparel and home goods company TJX announced that it was the victim of an unauthorized computer systems intrusion[196] and that the hackers had accessed a system that stored data on credit card, debit card, check, and merchandise return transactions.[197]

Stuxnet attack

In 2010, the computer worm known as Stuxnet reportedly ruined almost one-fifth of Iran's nuclear centrifuges.[198] It did so by disrupting industrial programmable logic controllers (PLCs) in a targeted attack. This is generally believed to have been launched by Israel and the United States to disrupt Iran's nuclear program[199][200][201][202] – although neither has publicly admitted this.

Global surveillance disclosures

In early 2013, documents provided by Edward Snowden were published by *The Washington Post* and *The Guardian*[203][204] exposing the massive scale of NSA global surveillance. There were also indications that the NSA may have inserted a backdoor in a NIST standard for encryption.[205] This standard was later withdrawn due to widespread criticism.[206] The NSA additionally were revealed to have tapped the links between Google's data centers.[207]

Target and Home Depot breaches

A Ukrainian hacker known as Rescator broke into Target Corporation computers in 2013, stealing roughly 40 million credit cards,[208] and then Home Depot computers in 2014, stealing between 53 and 56 million credit card numbers.[209] Warnings were delivered at both corporations, but ignored; physical security breaches using self checkout machines are believed to have played a large role. "The malware utilized is absolutely unsophisticated and uninteresting," says Jim Walter, director of threat intelligence operations at security technology company McAfee – meaning that the heists could have easily been stopped by existing antivirus software had administrators responded to the warnings. The size of the thefts has resulted in major attention from state and Federal United States authorities and the investigation is ongoing.

Office of Personnel Management data breach

In April 2015, the Office of Personnel Management discovered it had been hacked more than a year earlier in a data breach, resulting in the theft of approximately 21.5 million personnel records handled by the office.[210] The Office of Personnel Management hack has been described by federal officials as among the largest breaches of government data in the history of the United States.[211] Data targeted in the breach included personally identifiable information such as Social Security numbers, names, dates and places of birth, addresses, and fingerprints of current and former government employees as well as anyone who had undergone a government background check.[212][213] It is believed the hack was perpetrated by Chinese hackers.[214]

Ashley Madison breach

In July 2015, a hacker group is known as The Impact Team successfully breached the extramarital relationship website Ashley Madison, created by Avid Life Media. The group claimed that they had taken not only company data but user data as well. After the breach, The Impact Team dumped emails from the company's CEO, to prove their point, and threatened to dump customer data unless the website was taken down permanently.[215] When Avid Life Media did not take the site offline the group released two more compressed files, one 9.7GB and the second 20GB. After the second data dump, Avid Life Media CEO Noel Biderman resigned; but the website remained to function.

Colonial Pipeline ransomware attack

In June 2021, the cyber attack took down the largest fuel pipeline in the U.S. and led to shortages across the East Coast.[216]

Legal issues and global regulation

International legal issues of cyber attacks are complicated in nature. There is no global base of common rules to judge, and eventually punish, cybercrimes and cybercriminals - and where security firms or agencies do locate the cybercriminal behind the creation of a particular piece of malware or form of cyber attack, often the local authorities cannot take action due to lack of laws under which to prosecute.[217][218] Proving attribution for cybercrimes and cyberattacks is also a major problem for all law enforcement agencies. "Computer viruses switch from one country to another, from one jurisdiction to another – moving around the world, using the fact that we don't have the capability to globally police operations like this. So the Internet is as if someone [had] given free plane tickets to all the online criminals of the world."[217] The use of techniques such as dynamic DNS, fast flux and bullet proof servers add to the difficulty of investigation and enforcement.

Role of government

The role of the government is to make regulations to force companies and organizations to protect their systems, infrastructure and information from any cyberattacks, but also to protect its own national infrastructure such as the national power-grid.[219]

The government's regulatory role in cyberspace is complicated. For some, cyberspace was seen as a virtual space that was to remain free of government intervention, as can be seen in many of today's libertarian blockchain and bitcoin discussions.[220]

Many government officials and experts think that the government should do more and that there is a crucial need for improved regulation, mainly due to the failure of the private sector to solve efficiently the cybersecurity problem. R. Clarke said during a panel discussion at the RSA Security Conference in San Francisco, he believes that the "industry only responds when you threaten regulation. If the industry doesn't respond (to the threat), you have to follow through."[221] On the other hand, executives from the private sector agree that improvements are necessary, but think that government intervention would affect their ability to innovate efficiently. Daniel R. McCarthy analyzed this public-private partnership in cybersecurity and reflected on the role of cybersecurity in the broader constitution of political order.[222]

On 22 May 2020, the UN Security Council held its second ever informal meeting on cybersecurity to focus on cyber challenges to international peace. According to UN Secretary-General António Guterres, new technologies are too often used to violate rights.[223]

International actions

Many different teams and organizations exist, including:

- The Forum of Incident Response and Security Teams (FIRST) is the global association of CSIRTs.[224] The US-CERT, AT&T, Apple, Cisco, McAfee, Microsoft are all members of this international team.[225]
- The Council of Europe helps protect societies worldwide from the threat of cybercrime through the Convention on Cybercrime.[226]
- The purpose of the Messaging Anti-Abuse Working Group (MAAWG) is to bring the messaging industry together to work collaboratively and to successfully address the various forms of messaging abuse, such as spam, viruses, denial-of-service attacks and other messaging exploitations.[227] France Telecom, Facebook, AT&T, Apple, Cisco, Sprint are some of the members of the MAAWG.[228]
- ENISA : The European Network and Information Security Agency (ENISA) is an agency of the European Union with the objective to improve network and information security in the European Union.

Europe

On 14 April 2016, the European Parliament and the Council of the European Union adopted the General Data Protection Regulation (GDPR). The GDPR, which came into force on 25 May 2018, grants individuals within the European Union (EU) and the European Economic Area (EEA) the right to the protection of personal data. The regulation requires that any entity that processes personal data incorporate data protection by design and by default. It also requires that certain organizations appoint a Data Protection Officer (DPO).

The IT Security Association TeleTrusT exist in Germany since June 1986, which is an international competence network for IT security.

National actions

Computer emergency response teams

Most countries have their own computer emergency response team to protect network security.

Canada

Since 2010, Canada has had a cybersecurity strategy.[229][230] This functions as a counterpart document to the National Strategy and Action Plan for Critical Infrastructure.[231] The strategy has three main pillars: securing government systems, securing vital private cyber systems, and helping Canadians to be secure online.[230][231] There is also a Cyber Incident Management Framework to provide a coordinated response in the event of a cyber incident.[232][233]

The Canadian Cyber Incident Response Centre (CCIRC) is responsible for mitigating and responding to threats to Canada's critical infrastructure and cyber systems. It provides support to mitigate cyber threats, technical support to respond & recover from targeted cyber attacks, and provides online tools for members of Canada's critical infrastructure sectors.[234] It posts regular cybersecurity bulletins[235] & operates an online reporting tool where individuals and organizations can report a cyber incident.[236]

To inform the general public on how to protect themselves online, Public Safety Canada has partnered with STOP.THINK.CONNECT, a coalition of non-profit, private sector, and government organizations,[237] and launched the Cyber Security Cooperation Program.[238][239] They also run the GetCyberSafe portal for Canadian citizens, and Cyber Security Awareness Month during October.[240]

Public Safety Canada aims to begin an evaluation of Canada's cybersecurity strategy in early 2015.[231]

Australia

Australian federal government announced an $18.2 million investment to fortify the cybersecurity resilience of small and medium enterprises (SMEs) and enhance their capabilities in responding to cyber threats. This financial backing is an integral component of the 2023-2030 Australian Cyber Security Strategy (https://www.homeaffairs.gov.au/about-us/our-portfolios/cyber-security/strategy/2023-2030-aust ralian-cyber-security-strategy). A substantial allocation of $7.2 million is earmarked for the establishment of a voluntary cyber health check program, facilitating businesses in conducting a comprehensive and tailored self-assessment of their cybersecurity upskill.

This avant-garde health assessment serves as a diagnostic tool, enabling enterprises to ascertain the robustness of Australia's cyber security regulations (https://www.homeaffairs.gov.au/reports-and-pubs/files/strengthening-australias-cyber-security-submissions/nsw-young-lawyers.pdf). Furthermore, it affords them access to a repository of educational resources and materials, fostering the acquisition of skills necessary for an elevated cybersecurity posture. This groundbreaking initiative was jointly disclosed by Minister for Cyber Security Clare O'Neil and Minister for Small Business Julie Collins.[241]

India

Some provisions for cybersecurity have been incorporated into rules framed under the Information Technology Act 2000.[242]

The National Cyber Security Policy 2013 is a policy framework by the Ministry of Electronics and Information Technology (MeitY) which aims to protect the public and private infrastructure from cyberattacks, and safeguard "information, such as personal information (of web users), financial and banking information and sovereign data". CERT- In is the nodal agency which monitors the cyber threats in the country. The post of National Cyber Security Coordinator has also been created in the Prime Minister's Office (PMO).

The Indian Companies Act 2013 has also introduced cyber law and cybersecurity obligations on the part of Indian directors. Some provisions for cybersecurity have been incorporated into rules framed under the Information Technology Act 2000 Update in 2013.[243]

South Korea

Following cyberattacks in the first half of 2013, when the government, news media, television stations, and bank websites were compromised, the national government committed to the training of 5,000 new cybersecurity experts by 2017. The South Korean government blamed its northern counterpart for these attacks, as well as incidents that occurred in 2009, 2011,[244] and 2012, but Pyongyang denies the accusations.[245]

United States

Cyber Plan

The United States has its first fully formed cyber plan in 15 years, as a result of the release of this National Cyber plan.[246] In this policy, the US says it will: Protect the country by keeping networks, systems, functions, and data safe; Promote American wealth by building a strong digital economy and encouraging strong domestic innovation; Peace and safety should be kept by making it easier for the US to stop people from using computer tools for bad things, working with friends and partners to do this; and increase the United States' impact around the world to support the main ideas behind an open, safe, reliable, and compatible Internet.[247]

The new U.S. cyber strategy[248] seeks to allay some of those concerns by promoting responsible behavior in cyberspace, urging nations to adhere to a set of norms, both through international law and voluntary standards. It also calls for specific measures to harden U.S. government networks from attacks, like the June 2015 intrusion into the U.S. Office of Personnel Management (OPM), which compromised the records of about 4.2 million current and former government employees. And the strategy calls for the U.S. to continue to name and shame bad cyber actors, calling them out publicly for attacks when possible, along with the use of economic sanctions and diplomatic pressure.[249]

Legislation

The 1986 18 U.S.C. § 1030 (https://www.law.cornell.edu/uscode/text/18/1030), the Computer Fraud and Abuse Act is the key legislation. It prohibits unauthorized access or damage of *protected computers* as defined in 18 U.S.C. § 1030(e)(2) (https://www.law.cornell.edu/uscode/text/18/1030#e_2). Although various other measures have been proposed[250][251] – none have succeeded.

In 2013, executive order 13636 *Improving Critical Infrastructure Cybersecurity* was signed, which prompted the creation of the NIST Cybersecurity Framework.

In response to the Colonial Pipeline ransomware attack[252] President Joe Biden signed Executive Order 14028[253] on May 12, 2021, to increase software security standards for sales to the government, tighten detection and security on existing systems, improve information sharing and training, establish a Cyber Safety Review Board, and improve incident response.

Standardized government testing services

The General Services Administration (GSA) has standardized the *penetration test* service as a pre-vetted support service, to rapidly address potential vulnerabilities, and stop adversaries before they impact US federal, state and local governments. These services are commonly referred to as Highly Adaptive Cybersecurity Services (HACS).

Agencies

The Department of Homeland Security has a dedicated division responsible for the response system, risk management program and requirements for cybersecurity in the United States called the National Cyber Security Division.[254][255] The division is home to US-CERT operations and the National Cyber Alert System.[255] The National Cybersecurity and Communications Integration Center brings together government organizations responsible for protecting computer networks and networked infrastructure.[256]

The third priority of the FBI is to: "Protect the United States against cyber-based attacks and high-technology crimes",[257] and they, along with the National White Collar Crime Center (NW3C), and the Bureau of Justice Assistance (BJA) are part of the multi-agency task force, The Internet Crime Complaint Center, also known as IC3.[258]

In addition to its own specific duties, the FBI participates alongside non-profit organizations such as InfraGard.[259][260]

The Computer Crime and Intellectual Property Section (CCIPS) operates in the United States Department of Justice Criminal Division. The CCIPS is in charge of investigating computer crime and intellectual property crime and is specialized in the search and seizure of digital evidence in computers and networks.[261] In 2017, CCIPS published A Framework for a Vulnerability Disclosure Program for Online Systems to help organizations "clearly describe authorized vulnerability disclosure and discovery conduct, thereby substantially reducing the likelihood that such described activities will result in a civil or criminal violation of law under the Computer Fraud and Abuse Act (18 U.S.C. § 1030)."[262]

The United States Cyber Command, also known as USCYBERCOM, "has the mission to direct, synchronize, and coordinate cyberspace planning and operations to defend and advance national interests in collaboration with domestic and international partners."[263] It has no role in the protection of civilian networks.[264][265]

The U.S. Federal Communications Commission's role in cybersecurity is to strengthen the protection of critical communications infrastructure, to assist in maintaining the reliability of networks during disasters, to aid in swift recovery after, and to ensure that first responders have access to effective communications services.[266]

The Food and Drug Administration has issued guidance for medical devices,[267] and the National Highway Traffic Safety Administration[268] is concerned with automotive cybersecurity. After being criticized by the Government Accountability Office,[269] and following successful attacks on airports and claimed attacks on airplanes, the Federal Aviation Administration has devoted funding to securing systems on board the planes of private manufacturers, and the Aircraft Communications Addressing and Reporting System.[270] Concerns have also been raised about the future Next Generation Air Transportation System.[271]

The US Department of Defense (DoD) issued DoD Directive 8570 in 2004, supplemented by DoD Directive 8140, requiring all DoD employees and all DoD contract personnel involved in information assurance roles and activities to earn and maintain various industry Information Technology (IT) certifications in an effort to ensure that all DoD personnel involved in network infrastructure defense have minimum levels of IT industry recognized knowledge, skills and abilities (KSA). Andersson and Reimers (2019) report these certifications range from CompTIA's A+ and Security+ through the ICS2.org's CISSP, etc.[272]

Computer emergency readiness team

Computer emergency response team is a name given to expert groups that handle computer security incidents. In the US, two distinct organizations exist, although they do work closely together.

- US-CERT: part of the National Cyber Security Division of the United States Department of Homeland Security.[273]
- CERT/CC: created by the Defense Advanced Research Projects Agency (DARPA) and run by the Software Engineering Institute (SEI).

U.S. NRC, 10 CFR 73.54 Cybersecurity

In the context of U.S. nuclear power plants, the U.S. Nuclear Regulatory Commission (NRC) outlines cybersecurity requirements under 10 CFR Part 73, specifically in §73.54.[274]

NEI 08-09: Cybersecurity Plan for Nuclear Power Plants

The Nuclear Energy Institute's NEI 08-09 document, *Cyber Security Plan for Nuclear Power Reactors*,[275] outlines a comprehensive framework for cybersecurity in the nuclear power industry. Drafted with input from the U.S. NRC, this guideline is instrumental in aiding licensees to comply with the Code of Federal Regulations (CFR), which mandates robust protection of digital computers and equipment and communications systems at nuclear power plants against cyber threats.[276]

Modern warfare

There is growing concern that cyberspace will become the next theater of warfare. As Mark Clayton from *The Christian Science Monitor* wrote in a 2015 article titled "The New Cyber Arms Race":

> In the future, wars will not just be fought by soldiers with guns or with planes that drop bombs. They will also be fought with the click of a mouse a half a world away that unleashes carefully weaponized computer programs that disrupt or destroy critical industries like utilities, transportation, communications, and energy. Such attacks could also disable military networks that control the movement of troops, the path of jet fighters, the command and control of warships.[277]

This has led to new terms such as *cyberwarfare* and *cyberterrorism*. The United States Cyber Command was created in 2009[278] and many other countries have similar forces.

There are a few critical voices that question whether cybersecurity is as significant a threat as it is made out to be.[279][280][281]

Careers

Cybersecurity is a fast-growing field of IT concerned with reducing organizations' risk of hack or data breaches.[282] According to research from the Enterprise Strategy Group, 46% of organizations say that they have a "problematic shortage" of cybersecurity skills in 2016, up from 28% in 2015.[283] Commercial, government and non-governmental organizations all employ cybersecurity professionals. The fastest increases in demand for cybersecurity workers are in industries managing increasing volumes of consumer data such as finance, health care, and retail.[284] However, the use of the term *cybersecurity* is more prevalent in government job descriptions.[285]

Typical cybersecurity job titles and descriptions include:[286]

Security analyst

Analyzes and assesses vulnerabilities in the infrastructure (software, hardware, networks), investigates using available tools and countermeasures to remedy the detected vulnerabilities and recommends solutions and best practices. Analyzes and assesses damage to the data/infrastructure as a result of security incidents, examines available recovery tools and processes, and recommends solutions. Tests for compliance with security policies and procedures. May assist in the creation, implementation, or management of security solutions.

Security engineer

Performs security monitoring, security and data/logs analysis, and forensic analysis, to detect security incidents, and mount the incident response. Investigates and utilizes new technologies and processes to enhance security capabilities and implement improvements. May also review code or perform other security engineering methodologies.

Security architect

Designs a security system or major components of a security system, and may head a security design team building a new security system.[287]

Chief Information Security Officer (CISO)

A high-level management position responsible for the entire information security division/staff. The position may include hands-on technical work.[288]

Chief Security Officer (CSO)

A high-level management position responsible for the entire security division/staff. A newer position is now deemed needed as security risks grow.

Data Protection Officer (DPO)

A DPO is tasked with monitoring compliance with data protection laws (such as GDPR), data protection policies, awareness-raising, training, and audits.[289]

Security Consultant/Specialist/Intelligence

Broad titles that encompass any one or all of the other roles or titles tasked with protecting computers, networks, software, data or information systems against viruses, worms, spyware, malware, intrusion detection, unauthorized access, denial-of-service attacks, and an ever-increasing list of attacks by hackers acting as individuals or as part of organized crime or foreign governments.

Student programs are also available for people interested in beginning a career in cybersecurity.[290][291] Meanwhile, a flexible and effective option for information security professionals of all experience levels to keep studying is online security training, including webcasts.[292][293] A wide range of certified courses are also available.[294]

In the United Kingdom, a nationwide set of cybersecurity forums, known as the U.K Cyber Security Forum, were established supported by the Government's cybersecurity strategy[295] in order to encourage start-ups and innovation and to address the skills gap[296] identified by the U.K Government.

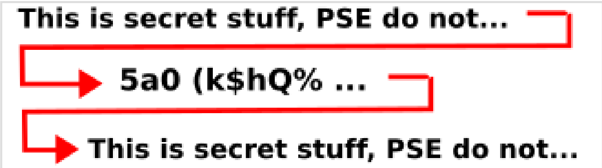
In Singapore, the Cyber Security Agency has issued a Singapore Operational Technology (OT) Cybersecurity Competency Framework (OTCCF). The framework defines emerging cybersecurity roles in Operational Technology. The OTCCF was endorsed by the Infocomm Media Development Authority (IMDA). It outlines the different OT cybersecurity job positions as well as the technical skills and core competencies necessary. It also depicts the many career paths available, including vertical and lateral advancement opportunities.[297]

Terminology

The following terms used with regards to computer security are explained below:

- Access authorization restricts access to a computer to a group of users through the use of authentication systems. These systems can protect either the whole computer, such as through an interactive login screen, or individual services, such as a FTP server. There are many methods for identifying and authenticating users, such as passwords, identification cards, smart cards, and biometric systems.
- Anti-virus software consists of computer programs that attempt to identify, thwart, and eliminate computer viruses and other malicious software (malware).
- Applications are executable code, so general corporate practice is to restrict or block users the power to install them; to install them only when there is a demonstrated need (e.g. software needed to perform assignments); to install only those which are known to be reputable (preferably with access to the computer code used to create the application), and to reduce the attack surface by installing as few as possible. They are typically run with least privilege, with a robust process in place to identify, test and install any released security patches or updates for them.
  - For example, programs can be installed into an individual user's account, which limits the program's potential access, as well as being a means control which users have specific exceptions to policy. In Linux, FreeBSD, OpenBSD, and other Unix-like operating systems there is an option to further restrict an application using chroot or other means of restricting the application to its own 'sandbox'. For example. Linux provides namespaces, and Cgroups to further restrict the access of an application to system resources.
  - Generalized security frameworks such as SELinux or AppArmor help administrators control access.
  - Java and other languages which compile to Java byte code and run in the Java virtual machine can have their access to other applications controlled at the virtual machine level.
  - Some software can be run in software containers which can even provide their own set of system libraries, limiting the software's, or anyone controlling it, access to the server's versions of the libraries.
- Authentication techniques can be used to ensure that communication end-points are who they say they are.
- Automated theorem proving and other verification tools can be used to enable critical algorithms and code used in secure systems to be mathematically proven to meet their specifications.
- Backups are one or more copies kept of important computer files. Typically, multiple copies will be kept at different locations so that if a copy is stolen or damaged, other copies will still exist.
- Capability and access control list techniques can be used to ensure privilege separation and

mandatory access control. Capabilities vs. ACLs discusses their use.
- Chain of trust techniques can be used to attempt to ensure that all software loaded has been certified as authentic by the system's designers.
- Confidentiality is the nondisclosure of information except to another authorized person.[298]
- Cryptographic techniques can be used to defend data in transit between systems, reducing the probability that the data exchange between systems can be intercepted or modified.
- Cyber attribution, is an attribution of cybercrime, i.e., finding who perpetrated a cyberattack.
- Cyberwarfare is an Internet-based conflict that involves politically motivated attacks on information and information systems. Such attacks can, for example, disable official websites and networks, disrupt or disable essential services, steal or alter classified data, and cripple financial systems.
- Data integrity is the accuracy and consistency of stored data, indicated by an absence of any alteration in data between two updates of a data record.[299]



Cryptographic techniques involve transforming information, scrambling it, so it becomes unreadable during transmission. The intended recipient can unscramble the message; ideally, eavesdroppers cannot.

- Encryption is used to protect the confidentiality of a message. Cryptographically secure ciphers are designed to make any practical attempt of breaking them infeasible. Symmetric-key ciphers are suitable for bulk encryption using shared keys, and public-key encryption using digital certificates can provide a practical solution for the problem of securely communicating when no key is shared in advance.
- Endpoint security software aids networks in preventing malware infection and data theft at network entry points made vulnerable by the prevalence of potentially infected devices such as laptops, mobile devices, and USB drives.[300]
- Firewalls serve as a gatekeeper system between networks, allowing only traffic that matches defined rules. They often include detailed logging, and may include intrusion detection and intrusion prevention features. They are near-universal between company local area networks and the Internet, but can also be used internally to impose traffic rules between networks if network segmentation is configured.
- A hacker is someone who seeks to breach defenses and exploit weaknesses in a computer system or network.
- Honey pots are computers that are intentionally left vulnerable to attack by crackers. They can be used to catch crackers and to identify their techniques.
- Intrusion-detection systems are devices or software applications that monitor networks or systems for malicious activity or policy violations.
- A microkernel is an approach to operating system design which has only the near-minimum amount of code running at the most privileged level – and runs other elements of the operating system such as device drivers, protocol stacks and file systems, in the safer, less privileged user space.
- Pinging. The standard ping application can be used to test if an IP address is in use. If it is, attackers may then try a port scan to detect which services are exposed.
- A port scan is used to probe an IP address for open ports to identify accessible network services and applications.
- A key logger is spyware that silently captures and stores each keystroke that a user types on the computer's keyboard.
- Social engineering is the use of deception to manipulate individuals to breach security.

- Logic bombs is a type of malware added to a legitimate program that lies dormant until it is triggered by a specific event.
- A unikernel is a computer program that runs on a minimalistic operating system where a single application is allowed to run (as opposed to a general purpose operating system where many applications can run at the same time). This approach to minimizing the attack surface is adopted mostly in cloud environments where software is deployed in virtual machines.
- Zero trust security means that no one is trusted by default from inside or outside the network, and verification is required from everyone trying to gain access to resources on the network.

History

Since the Internet's arrival and with the digital transformation initiated in recent years, the notion of cybersecurity has become a familiar subject in both our professional and personal lives. Cybersecurity and cyber threats have been consistently present for the last 60 years of technological change. In the 1970s and 1980s, computer security was mainly limited to academia until the conception of the Internet, where, with increased connectivity, computer viruses and network intrusions began to take off. After the spread of viruses in the 1990s, the 2000s marked the institutionalization of organized attacks such as distributed denial of service.[301] This led to the formalization of cybersecurity as a professional discipline.[302]

The April 1967 session organized by Willis Ware at the Spring Joint Computer Conference, and the later publication of the Ware Report, were foundational moments in the history of the field of computer security.[303] Ware's work straddled the intersection of material, cultural, political, and social concerns.[303]

A 1977 NIST publication[304] introduced the *CIA triad* of confidentiality, integrity, and availability as a clear and simple way to describe key security goals.[305] While still relevant, many more elaborate frameworks have since been proposed.[306][307]

However, in the 1970s and 1980s, there were no grave computer threats because computers and the internet were still developing, and security threats were easily identifiable. More often, threats came from malicious insiders who gained unauthorized access to sensitive documents and files. Although malware and network breaches existed during the early years, they did not use them for financial gain. By the second half of the 1970s, established computer firms like IBM started offering commercial access control systems and computer security software products.[308]

One of the earliest examples of an attack on a computer network was the computer worm Creeper written by Bob Thomas at BBN, which propagated through the ARPANET in 1971.[309] The program was purely experimental in nature and carried no malicious payload. A later program, Reaper, was created by Ray Tomlinson in 1972 and used to destroy Creeper.

Between September 1986 and June 1987, a group of German hackers performed the first documented case of cyber espionage.[310] The group hacked into American defense contractors, universities, and military base networks and sold gathered information to the Soviet KGB. The group was led by Markus Hess, who was arrested on 29 June 1987. He was convicted of espionage (along with two co-conspirators) on 15 Feb 1990.

In 1988, one of the first computer worms, called the Morris worm, was distributed via the Internet. It gained significant mainstream media attention.[311]

Netscape started developing the protocol SSL, shortly after the National Center for Supercomputing Applications (NCSA) launched Mosaic 1.0, the first web browser, in 1993.[312][313] Netscape had SSL version 1.0 ready in 1994, but it was never released to the public due to many serious security vulnerabilities.[312] However, in 1995, Netscape launched Version 2.0.[314]

The National Security Agency (NSA) is responsible for the protection of U.S. information systems and also for collecting foreign intelligence.[315] The agency analyzes commonly used software and system configurations to find security flaws, which it can use for offensive purposes against competitors of the United States.[316]

NSA contractors created and sold *click-and-shoot* attack tools to US agencies and close allies, but eventually, the tools made their way to foreign adversaries.[317] In 2016, NSAs own hacking tools were hacked, and they have been used by Russia and North Korea. NSA's employees and contractors have been recruited at high salaries by adversaries, anxious to compete in cyberwarfare. In 2007, the United States and Israel began exploiting security flaws in the Microsoft Windows operating system to attack and damage equipment used in Iran to refine nuclear materials. Iran responded by heavily investing in their own cyberwarfare capability, which it began using against the United States.[316]

Notable scholars

- Ross J. Anderson
- Annie Anton
- Adam Back
- Daniel J. Bernstein
- Matt Blaze
- Stefan Brands
- L. Jean Camp
- Lorrie Cranor
- Dorothy E. Denning
- Peter J. Denning
- Cynthia Dwork
- Chuck Easttom
- Deborah Estrin
- Joan Feigenbaum
- Ian Goldberg
- Shafi Goldwasser
- Lawrence A. Gordon
- Peter Gutmann
- Paul Kocher
- Monica S. Lam
- Butler Lampson
- Brian LaMacchia
- Susan Landau
- Carl Landwehr
- Kevin Mitnick
- Peter G. Neumann
- Susan Nycum
- Paul C. van Oorschot
- Fred Piper
- Roger R. Schell
- Bruce Schneier
- Dawn Song
- Gene Spafford
- Salvatore J. Stolfo
- Willis Ware
- Moti Yung

See also

- Attack tree – Conceptual diagrams showing how an asset, or target, might be attacked

- Bicycle attack – Method of discovering password length
- CAPTCHA – Test to determine whether a user is human
- Center for Internet Security – Nonprofit organization focused on cybersecurity
- Cloud computing security – Methods used to protect cloud based assets
- Comparison of antivirus software
- Content Disarm & Reconstruction – Policy-based removal of components
- Content Security Policy – Computer security standard to prevent cross-site scripting and related attacks
- Countermeasure (computer) – Process to reduce a security threat
- Cyber insurance – Information technology risk insurance
- Cyber self-defense – Protection of computer systems from information disclosure, theft or damage
- Cyberbiosecurity – Emerging field of computer security
- Cybersecurity information technology list
- Dancing pigs – Users' disregard for IT security
- Data security – Protection of digital data
- Defense strategy (computing) – Concept to reduce computer security risks
- Fault tolerance – Resilience of systems to component failures or errors
- Hardware security – Security architecture implemented in hardware
- Human–computer interaction (security) – Academic discipline studying the relationship between computer systems and their users
- Identity management – Technical and Policy systems to give users appropriate access
- Identity-based security – Access control by authenticated ID
- Information security awareness – part of information security that focuses on raising consciousness regarding potential risks of the rapidly evolving forms of information and the accompanying threats
- Internet privacy – Right or mandate of personal privacy concerning the internet
- Internet safety – Being aware of safety and security risks on the Internet
- Internet security – Branch of computer security
- IT risk – Any risk related to information technology
- IT security standards – Technology standards and techniques
- Cyber kill chain – Process of carrying out a cyberattack
- List of computer security certifications
- List of cyber warfare forces
- Open security – Open source approach to computer security
- Outline of computer security – Overview of and topical guide to computer security
- OWASP – Computer security organization
- Physical information security – Common ground of physical and information security
- Privacy software – Layer to protect users' privacy
- Security engineering – Process of incorporating security controls into an information system
- Security through obscurity – Reliance on design or implementation secrecy for security
- Software-defined perimeter – Method of enhancing computer security

References

1. Schatz, Daniel; Bashroush, Rabih; Wall, Julie (2017). "Towards a More Representative

Definition of Cyber Security" (https://commons.erau.edu/jdfsl/vol12/iss2/8/). *Journal of Digital Forensics, Security and Law*. **12** (2). ISSN 1558-7215 (https://search.worldcat.org/issn/1558-7215).

2. Computer security (https://www.britannica.com/EBchecked/topic/130682) at the *Encyclopædia Britannica*
3. Tate, Nick (7 May 2013). "Reliance spells end of road for ICT amateurs" (https://www.theaustralian.com.au/news/reliance-spells-end-of-road-for-ict-amateurs/news-story/6f84ad403b8721100f5957a472a945eb). *The Australian*.
4. Kianpour, Mazaher; Kowalski, Stewart; Øverby, Harald (2021). "Systematically Understanding Cybersecurity Economics: A Survey" (https://doi.org/10.3390%2Fsu132413677). *Sustainability*. **13** (24): 13677. Bibcode:2021Sust...1313677K (https://ui.adsabs.harvard.edu/abs/2021Sust...1313677K). doi:10.3390/su132413677 (https://doi.org/10.3390%2Fsu132413677). hdl:11250/2978306 (https://hdl.handle.net/11250%2F2978306). ISSN 2071-1050 (https://search.worldcat.org/issn/2071-1050).
5. Stevens, Tim (11 June 2018). "Global Cybersecurity: New Directions in Theory and Methods" (https://kclpure.kcl.ac.uk/portal/files/97261726/PaG_6_2_Global_Cybersecurity_New_Directions_in_Theory_and_Methods.pdf) (PDF). *Politics and Governance*. **6** (2): 1–4. doi:10.17645/pag.v6i2.1569 (https://doi.org/10.17645%2Fpag.v6i2.1569). Archived (https://web.archive.org/web/20190904151257/https://kclpure.kcl.ac.uk/portal/files/97261726/PaG_6_2_Global_Cybersecurity_New_Directions_in_Theory_and_Methods.pdf) (PDF) from the original on 4 September 2019.
6. "About the CVE Program" (https://www.cve.org/About/Overview). *www.cve.org*. Retrieved 12 April 2023.
7. Zlatanov, Nikola (3 December 2015). *Computer Security and Mobile Security Challenges* (https://www.researchgate.net/publication/298807979). Tech Security Conference At: San Francisco, CA.
8. "Ghidra" (https://web.archive.org/web/20200815201448/https://www.nsa.gov/resources/everyone/ghidra/). *nsa.gov*. 1 August 2018. Archived from the original (https://www.nsa.gov/resources/everyone/ghidra/) on 15 August 2020. Retrieved 17 August 2020.
9. Larabel, Michael (28 December 2017). "Syzbot: Google Continuously Fuzzing The Linux Kernel" (https://www.phoronix.com/scan.php?page=news_item&px=Syzbot-Linux-Kernel-Fuzzing/). *www.phoronix.com/*. Retrieved 25 March 2021.
10. "Cyber attacks on SMBs: Current Stats and How to Prevent Them" (https://www.crowdstrike.com/solutions/small-business/cyber-attacks-on-smbs/). *crowdstrike.com*. Retrieved 30 November 2023.
11. "Cyber security breaches survey 2023" (https://www.gov.uk/government/statistics/cyber-security-breaches-survey-2023/cyber-security-breaches-survey-2023). *GOV.UK*. Retrieved 30 November 2023.
12. "How cyber attacks work" (https://www.ncsc.gov.uk/information/how-cyber-attacks-work). *www.ncsc.gov.uk*. Retrieved 30 November 2023.
13. "What is a backdoor attack? Definition and prevention | NordVPN" (https://nordvpn.com/blog/backdoor-attack/). *nordvpn.com*. 30 November 2023. Retrieved 3 January 2024.
14. "What is a backdoor attack?" (https://www.mcafee.com/learn/backdoor-threat/). *McAfee*. 4 December 2023. Retrieved 4 December 2023.
15. "Denial of Service (DoS) guidance" (https://www.ncsc.gov.uk/collection/denial-service-dos-guidance-collection). *www.ncsc.gov.uk*. Retrieved 4 December 2023.
16. "Computer Security" (https://www.interelectronix.com/computer-security.html). *www.interelectronix.com*. Retrieved 30 November 2023.
17. "What Is a DMA Attack? Analysis & Mitigation" (https://www.kroll.com/en/insights/publications/cyber/what-is-dma-attack-understanding-mitigating-threat). *Kroll*. Retrieved 4 December 2023.

18. "What Are Eavesdropping Attacks?" (https://www.fortinet.com/resources/cyberglossary/eavesdropping). *Fortinet*. Retrieved 5 December 2023.
19. York, Dan (1 January 2010), York, Dan (ed.), "Chapter 3 – Eavesdropping and Modification" (https://www.sciencedirect.com/science/article/pii/B978159749547900003X), *Seven Deadliest Unified Communications Attacks*, Boston: Syngress, pp. 41–69, ISBN 978-1-59749-547-9, retrieved 5 December 2023
20. "What Are Eavesdropping Attacks & How To Prevent Them" (https://enterprise.verizon.com/resources/articles/s/what-are-eavesdropping-attacks/). *Verizon Enterprise*. Retrieved 5 December 2023.
21. "What is Malware? | IBM" (https://www.ibm.com/topics/malware). *www.ibm.com*. 14 April 2022. Retrieved 6 December 2023.
22. Bendovschi, Andreea (2015). "Cyber-Attacks – Trends, Patterns and Security Countermeasures" (https://doi.org/10.1016%2FS2212-5671%2815%2901077-1). *Procedia Economics and Finance*. **28**: 24–31. doi:10.1016/S2212-5671(15)01077-1 (https://doi.org/10.1016%2FS2212-5671%2815%2901077-1).
23. "What is malware?" (https://www.mcafee.com/en-gb/antivirus/malware.html). *McAfee*. Retrieved 30 November 2023.
24. "What is a man-in-the-middle attack and how can I protect my organization?" (https://www.verizon.com/business/resources/articles/s/what-is-a-man-in-the-middle-attack-and-how-can-i-protect-my-organization/). *verizon.com*.
25. "Multi-Vector Attacks Demand Multi-Vector Protection" (https://www.msspalert.com/analysis/multi-vector-attacks-demand-multi-vector-protection). *MSSP Alert*. 24 July 2017.
26. Millman, Renee (15 December 2017). "New polymorphic malware evades three-quarters of AV scanners" (https://www.scmagazineuk.com/new-polymorphic-malware-evades-three-quarters-of-av-scanners/article/718757/). *SC Magazine UK*.
27. Tounsi, Wiem (15 May 2019), Tounsi, Wiem (ed.), "What is Cyber Threat Intelligence and How is it Evolving?" (https://onlinelibrary.wiley.com/doi/10.1002/9781119618393.ch1), *Cyber-Vigilance and Digital Trust* (1 ed.), Wiley, pp. 1–49, doi:10.1002/9781119618393.ch1 (https://doi.org/10.1002%2F9781119618393.ch1), ISBN 978-1-78630-448-3, S2CID 187294508 (https://api.semanticscholar.org/CorpusID:187294508), retrieved 6 December 2023
28. "Identifying Phishing Attempts" (https://web.archive.org/web/20150913200707/http://www.case.edu/its/kba/its-kba-27196-phishing-attempt/). Case. Archived from the original (https://www.case.edu/its/kba/its-kba-27196-phishing-attempt/) on 13 September 2015. Retrieved 4 July 2016.
29. "Protect yourself from phishing – Microsoft Support" (https://support.microsoft.com/en-us/windows/protect-yourself-from-phishing-0c7ea947-ba98-3bd9-7184-430e1f860a44). *support.microsoft.com*. Retrieved 6 December 2023.
30. Lazarus, Ari (23 February 2018). "Phishers send fake invoices" (https://www.consumer.ftc.gov/blog/2018/02/phishers-send-fake-invoices). *Consumer Information*. Retrieved 17 February 2020.
31. "Email Security" (https://web.archive.org/web/20220522081555/https://www.trellix.com/en-us/platform/email-security.html). *Trellix*. 17 May 2022. Archived from the original (https://www.trellix.com/en-us/platform/email-security.html) on 22 May 2022. Retrieved 24 October 2022.
32. "What is Privilege Escalation? – CrowdStrike" (https://www.crowdstrike.com/cybersecurity-101/privilege-escalation/). *crowdstrike.com*. Retrieved 7 December 2023.
33. Spence, Aaron; Bangay, Shaun (June 2022). "Security beyond cybersecurity: side-channel attacks against non-cyber systems and their countermeasures" (https://link.springer.com/10.1007/s10207-021-00563-6). *International Journal of Information Security*. **21** (3): 437–453. doi:10.1007/s10207-021-00563-6 (https://doi.org/10.1007%2Fs10207-021-00563-6). ISSN 1615-5262 (https://search.worldcat.org/issn/1615-5262).

34. Arcos Sergio. "Social Engineering" (http://upcommons.upc.edu/pfc/bitstream/2099.1/12289/1/73827.pdf) (PDF). *upc.edu*. Archived (https://web.archive.org/web/20131203043630/http://upcommons.upc.edu/pfc/bitstream/2099.1/12289/1/73827.pdf) (PDF) from the original on 3 December 2013. Retrieved 16 April 2019.
35. Scannell, Kara (24 February 2016). "CEO email scam costs companies $2bn" (https://web.archive.org/web/20160623105523/http://www.ft.com/intl/cms/s/0/83b4e9be-db16-11e5-a72f-1e7744c66818.html#axzz41pN5YBV4). *Financial Times*. No. 25 February 2016. Archived from the original (https://www.ft.com/intl/cms/s/0/83b4e9be-db16-11e5-a72f-1e7744c66818.html#axzz41pN5YBV4) on 23 June 2016. Retrieved 7 May 2016.
36. "Bucks leak tax info of players, employees as result of email scam" (https://www.espn.com/nba/story/_/id/15615363/milwaukee-bucks-leak-tax-information-players-employees-result-email-scam). Associated Press. 20 May 2016. Archived (https://web.archive.org/web/20160520144908/http://espn.go.com/nba/story/_/id/15615363/milwaukee-bucks-leak-tax-informatiopn-players-employees-result-email-scam) from the original on 20 May 2016. Retrieved 20 May 2016.
37. "What is Spoofing? – Definition from Techopedia" (https://www.techopedia.com/definition/5398/spoofing). *techopedia.com*. Archived (https://web.archive.org/web/20160630134737/https://www.techopedia.com/definition/5398/spoofing) from the original on 30 June 2016. Retrieved 16 January 2022.
38. Butterfield, Andrew; Ngondi, Gerard Ekembe, eds. (21 January 2016). "spoofing" (http://www.oxfordreference.com/view/10.1093/acref/9780199688975.001.0001/acref-9780199688975-e-4987). *A Dictionary of Computer Science*. Oxford University Press. doi:10.1093/acref/9780199688975.001.0001 (https://doi.org/10.1093%2Facref%2F9780199688975.001.0001). ISBN 978-0-19-968897-5. Retrieved 8 October 2017.
39. Marcel, Sébastien; Nixon, Mark; Li, Stan, eds. (2014). *Handbook of Biometric Anti-Spoofing: Trusted Biometrics under Spoofing Attacks*. Advances in Computer Vision and Pattern Recognition. London: Springer. doi:10.1007/978-1-4471-6524-8 (https://doi.org/10.1007%2F978-1-4471-6524-8). ISBN 978-1-4471-6524-8. ISSN 2191-6594 (https://search.worldcat.org/issn/2191-6594). LCCN 2014942635 (https://lccn.loc.gov/2014942635). S2CID 27594864 (https://api.semanticscholar.org/CorpusID:27594864).
40. "80 to 0 in Under 5 Seconds: Falsifying a Medical Patient's Vitals" (https://www.trellix.com/en-us/about/newsroom/stories/research/80-to-0-in-under-5-seconds-falsifying-a-medical-patients-vitals.html). *www.trellix.com*. Retrieved 9 February 2023.
41. Gallagher, Sean (14 May 2014). "Photos of an NSA "upgrade" factory show Cisco router getting implant" (https://arstechnica.com/tech-policy/2014/05/photos-of-an-nsa-upgrade-factory-show-cisco-router-getting-implant/). *Ars Technica*. Archived (https://web.archive.org/web/20140804130416/http://arstechnica.com/tech-policy/2014/05/photos-of-an-nsa-upgrade-factory-show-cisco-router-getting-implant/) from the original on 4 August 2014. Retrieved 3 August 2014.
42. Intelligence, Microsoft Threat (11 November 2021). "HTML smuggling surges: Highly evasive loader technique increasingly used in banking malware, targeted attacks" (https://www.microsoft.com/en-us/security/blog/2021/11/11/html-smuggling-surges-highly-evasive-loader-technique-increasingly-used-in-banking-malware-targeted-attacks/). *Microsoft Security Blog*. Retrieved 7 December 2023.
43. "Obfuscated Files or Information: HTML Smuggling, Sub-technique T1027.006 – Enterprise | MITRE ATT&CK®" (https://attack.mitre.org/techniques/T1027/006/). *attack.mitre.org*. Retrieved 22 February 2023.
44. Lim, Joo S.; Chang, Shanton; Maynard, Sean; Ahmad, Atif (2009). "Exploring the Relationship between Organizational Culture and Information Security Culture" (http://ro.ecu.edu.au/ism/12). *Proceedings of the 7th Australian Information Security Management Conference*. Security Research Institute (SRI), Edith Cowan University. doi:10.4225/75/57B4065130DEF (https://doi.org/10.4225%2F75%2F57B4065130DEF).

45. Reimers, Karl; Andersson, David (2017). *Post-secondary Education Network Security: the End User Challenge and Evolving Threats* (https://library.iated.org/view/REIMERS2017POS). ICERI2017 Proceedings. Vol. 1. IATED. pp. 1787–1796. doi:10.21125/iceri.2017.0554 (https://doi.org/10.21125%2Ficeri.2017.0554). ISBN 978-84-697-6957-7. ISSN 2340-1095 (https://search.worldcat.org/issn/2340-1095).
46. Verizon Data Breach Investigations Report 2020 (https://enterprise.verizon.com/resources/reports/2020-data-breach-investigations-report.pdf) (PDF). *verizon.com* (Report). Archived (https://web.archive.org/web/20200519161153/https://enterprise.verizon.com/resources/reports/2020-data-breach-investigations-report.pdf) (PDF) from the original on 19 May 2020. Retrieved 17 September 2021.
47. Schlienger, Thomas; Teufel, Stephanie (2003). "Information security culture-from analysis to change". *South African Computer Journal*. **31**: 46–52. hdl:10520/EJC27949 (https://hdl.handle.net/10520%2FEJC27949).
48. *Internet Security Glossary* (https://datatracker.ietf.org/doc/html/rfc2828). doi:10.17487/RFC2828 (https://doi.org/10.17487%2FRFC2828). RFC 2828 (https://datatracker.ietf.org/doc/html/rfc2828).
49. "CNSS Instruction No. 4009" (https://web.archive.org/web/20120227163121/http://www.cnss.gov/Assets/pdf/cnssi_4009.pdf) (PDF). 26 April 2010. Archived from the original (http://www.cnss.gov/Assets/pdf/cnssi_4009.pdf) (PDF) on 27 February 2012.
50. "InfosecToday Glossary" (http://www.infosectoday.com/Articles/Glossary.pdf) (PDF). Archived (https://web.archive.org/web/20141120041536/http://www.infosectoday.com/Articles/Glossary.pdf) (PDF) from the original on 20 November 2014.
51. "Cyber security design principles" (https://www.ncsc.gov.uk/collection/cyber-security-design-principles/cyber-security-design-principles). *www.ncsc.gov.uk*. Retrieved 11 December 2023.
52. "How the NCSC thinks about security architecture" (https://www.ncsc.gov.uk/blog-post/how-ncsc-thinks-about-security-architecture). *www.ncsc.gov.uk*. Retrieved 18 December 2023.
53. "Secure System Architecture and Design" (https://www.ukcybersecuritycouncil.org.uk/careers-and-learning/cyber-career-framework/secure-system-architecture-design). *UK Cyber Security Council*. 2024. Retrieved 4 January 2024.
54. "security architecture – Glossary | CSRC" (https://csrc.nist.gov/glossary/term/security_architecture). *csrc.nist.gov*. Retrieved 18 December 2023.
55. Jannsen, Cory. "Security Architecture" (http://www.techopedia.com/definition/72/security-architecture). *Techopedia*. Janalta Interactive Inc. Archived (https://web.archive.org/web/20141003064643/http://www.techopedia.com/definition/72/security-architecture) from the original on 3 October 2014. Retrieved 9 October 2014.
56. Oppliger, Rolf (1 May 1997). "Internet security: firewalls and beyond" (https://doi.org/10.1145%2F253769.253802). *Communications of the ACM*. **40** (5): 92–102. doi:10.1145/253769.253802 (https://doi.org/10.1145%2F253769.253802). ISSN 0001-0782 (https://search.worldcat.org/issn/0001-0782).
57. "How to Increase Cybersecurity Awareness" (https://www.isaca.org/resources/isaca-journal/issues/2019/volume-2/how-to-increase-cybersecurity-awareness). *ISACA*. Retrieved 25 February 2023.
58. Woodie, Alex (9 May 2016). "Why ONI May Be Our Best Hope for Cyber Security Now" (http://www.datanami.com/2016/05/09/oni-may-best-hope-cyber-security-now/). Archived (https://web.archive.org/web/20160820015812/https://www.datanami.com/2016/05/09/oni-may-best-hope-cyber-security-now/) from the original on 20 August 2016. Retrieved 13 July 2016.
59. Walkowski, Debbie (9 July 2019). "What Is The CIA Triad?" (https://www.f5.com/labs/articles/education/what-is-the-cia-triad.html). *F5 Labs*. Retrieved 25 February 2020.

60. "Knowing Value of Data Assets is Crucial to Cybersecurity Risk Management | SecurityWeek.Com" (https://www.securityweek.com/knowing-value-data-assets-crucial-cybersecurity-risk-management). *www.securityweek.com*. 3 December 2018. Retrieved 25 February 2020.
61. Foreman, Park (2009). *Vulnerability Management*. Boca Raton, Fla.: Auerbach Publications. p. 1. ISBN 978-1-4398-0150-5.
62. Johnson, A. (2018). *CCNA Cybersecurity Operations Companion Guide* (https://books.google.com/books?id=FxRbDwAAQBAJ&q=Vulnerabilities+can+be+discovered+with+a+vulnerability+scanner,+which+analyzes+a+computer+system+in+search+of+known+vulnerabilities&pg=SA5-PA83). Cisco Press. ISBN 978-0-13-516624-6.
63. Calder, Alan; Williams, Geraint (2014). *PCI DSS: A Pocket Guide* (3rd ed.). IT Governance Limited. ISBN 978-1-84928-554-4. "network vulnerability scans at least quarterly and after any significant change in the network"
64. Harrison, J. (2003). *Formal verification at Intel*. 18th Annual IEEE Symposium of Logic in Computer Science, 2003. Proceedings. pp. 45–54. doi:10.1109/LICS.2003.1210044 (https://doi.org/10.1109%2FLICS.2003.1210044). ISBN 978-0-7695-1884-8. S2CID 44585546 (https://api.semanticscholar.org/CorpusID:44585546).
65. Umrigar, Zerksis D.; Pitchumani, Vijay (1983). *Formal verification of a real-time hardware design* (http://portal.acm.org/citation.cfm?id=800667). Proceeding DAC '83 Proceedings of the 20th Design Automation Conference. IEEE Press. pp. 221–227. ISBN 978-0-8186-0026-5.
66. "Abstract Formal Specification of the seL4/ARMv6 API" (https://web.archive.org/web/20150521171234/https://sel4.systems/Docs/seL4-spec.pdf) (PDF). Archived from the original (https://sel4.systems/Docs/seL4-spec.pdf) (PDF) on 21 May 2015. Retrieved 19 May 2015.
67. Baumann, Christoph; Beckert, Bernhard; Blasum, Holger; Bormer, Thorsten. *Ingredients of Operating System Correctness? Lessons Learned in the Formal Verification of PikeOS* (https://web.archive.org/web/20110719110932/http://www-wjp.cs.uni-saarland.de/publikationen/Ba10EW.pdf) (PDF). Embedded World Conference, Nuremberg, Germany. Archived from the original (http://www-wjp.cs.uni-saarland.de/publikationen/Ba10EW.pdf) (PDF) on 19 July 2011.
68. Ganssle, Jack. "Getting it Right" (https://web.archive.org/web/20130504191958/http://www.ganssle.com/rants/gettingitright.htm). Archived from the original (http://www.ganssle.com/rants/gettingitright.htm) on 4 May 2013.
69. "Everything you need for a career as a SOC analyst" (https://www.cybersecurityjobsite.com/staticpages/10300/everything-you-need-for-a-career-as-a-soc-analyst/). *www.cybersecurityjobsite.com*. Retrieved 19 December 2023.
70. "Turn on 2-step verification (2SV)" (https://www.ncsc.gov.uk/collection/top-tips-for-staying-secure-online/activate-2-step-verification-on-your-email). *www.ncsc.gov.uk*. Retrieved 19 December 2023.
71. "NCSC's cyber security training for staff now available" (https://www.ncsc.gov.uk/blog-post/ncsc-cyber-security-training-for-staff-now-available). *www.ncsc.gov.uk*. Retrieved 19 December 2023.
72. Treglia, J.; Delia, M. (2017). *Cyber Security Inoculation*. NYS Cyber Security Conference, Empire State Plaza Convention Center, Albany, NY, 3–4 June.
73. "What is a license dongle?" (https://www.revenera.com/software-monetization/glossary/license-dongle). *www.revenera.com*. Retrieved 12 June 2024.
74. "Token-based authentication" (http://www.safenet-inc.com/multi-factor-authentication/authenticators/pki-usb-authentication/etoken-5200-token-based-authentication/). SafeNet.com. Archived (https://web.archive.org/web/20140320234026/http://www.safenet-inc.com/multi-factor-authentication/authenticators/pki-usb-authentication/etoken-5200-token-based-authentication/) from the original on 20 March 2014. Retrieved 20 March 2014.

75. "Lock and protect your Windows PC" (https://www.thewindowsclub.com/lock-protect-your-windows-pc-using-a-usb-drive-with-predator). TheWindowsClub.com. 10 February 2010. Archived (https://web.archive.org/web/20140320220321/http://www.thewindowsclub.com/lock-protect-your-windows-pc-using-a-usb-drive-with-predator) from the original on 20 March 2014. Retrieved 20 March 2014.
76. Greene, James (2012). "Intel Trusted Execution Technology: White Paper" (http://www.intel.com/content/dam/www/public/us/en/documents/white-papers/trusted-execution-technology-security-paper.pdf) (PDF). Intel Corporation. Archived (https://web.archive.org/web/20140611161421/http://www.intel.com/content/dam/www/public/us/en/documents/white-papers/trusted-execution-technology-security-paper.pdf) (PDF) from the original on 11 June 2014. Retrieved 18 December 2013.
77. "SafeNet ProtectDrive 8.4" (http://www.scmagazine.com/safenet-protectdrive-84/review/2596/). *SCMagazine.com*. 4 October 2008. Archived (https://web.archive.org/web/20140320220133/http://www.scmagazine.com/safenet-protectdrive-84/review/2596/) from the original on 20 March 2014. Retrieved 20 March 2014.
78. "Secure Hard Drives: Lock Down Your Data" (https://www.pcmag.com/article2/0,2817,2342798,00.asp). PCMag.com. 11 May 2009. Archived (https://web.archive.org/web/20170621202140/http://www.pcmag.com/article2/0,2817,2342798,00.asp) from the original on 21 June 2017.
79. Souppaya, Murugiah P.; Scarfone, Karen (2013). "Guidelines for Managing the Security of Mobile Devices in the Enterprise" (https://www.nist.gov/publications/guidelines-managing-security-mobile-devices-enterprise). *National Institute of Standards and Technology*. Special Publication (NIST SP). Gaithersburg, MD. doi:10.6028/NIST.SP.800-124r1 (https://doi.org/10.6028%2FNIST.SP.800-124r1).
80. "Access Control Statistics: Trends & Insights" (https://entrycare.com/access-control-statistics/). 23 February 2024. Retrieved 26 April 2024.
81. "Forget IDs, use your phone as credentials" (http://video.foxbusiness.com/v/2804966490001/forget-ids-use-your-phone-as-credentials/?playlist_id=937116503001#sp=show-clips). Fox Business Network. 4 November 2013. Archived (https://web.archive.org/web/20140320215829/http://video.foxbusiness.com/v/2804966490001/forget-ids-use-your-phone-as-credentials/?playlist_id=937116503001#sp=show-clips) from the original on 20 March 2014. Retrieved 20 March 2014.
82. "Direct memory access protections for Mac computers" (https://support.apple.com/guide/security/direct-memory-access-protections-seca4960c2b5/1/web/1). *Apple*. Retrieved 16 November 2022.
83. "Using IOMMU for DMA Protection in UEFI Firmware" (https://www.intel.com/content/dam/develop/external/us/en/documents/intel-whitepaper-using-iommu-for-dma-protection-in-uefi-820238.pdf) (PDF). Intel Corporation. Archived (https://web.archive.org/web/20211209062425/https://www.intel.com/content/dam/develop/external/us/en/documents/intel-whitepaper-using-iommu-for-dma-protection-in-uefi-820238.pdf) (PDF) from the original on 9 December 2021. Retrieved 16 November 2022.
84. Babaei, Armin; Schiele, Gregor; Zohner, Michael (26 July 2022). "Reconfigurable Security Architecture (RESA) Based on PUF for FPGA-Based IoT Devices" (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9331300). *Sensors*. **22** (15): 5577. Bibcode:2022Senso..22.5577B (https://ui.adsabs.harvard.edu/abs/2022Senso..22.5577B). doi:10.3390/s22155577 (https://doi.org/10.3390%2Fs22155577). ISSN 1424-8220 (https://search.worldcat.org/issn/1424-8220). PMC 9331300 (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9331300). PMID 35898079 (https://pubmed.ncbi.nlm.nih.gov/35898079).

85. Hassija, Vikas; Chamola, Vinay; Gupta, Vatsal; Jain, Sarthak; Guizani, Nadra (15 April 2021). "A Survey on Supply Chain Security: Application Areas, Security Threats, and Solution Architectures" (https://ieeexplore.ieee.org/document/9203862). *IEEE Internet of Things Journal*. **8** (8): 6222–6246. doi:10.1109/JIOT.2020.3025775 (https://doi.org/10.1109%2FJIOT.2020.3025775). ISSN 2327-4662 (https://search.worldcat.org/issn/2327-4662). S2CID 226767829 (https://api.semanticscholar.org/CorpusID:226767829).
86. "The Most Secure OS: What is the Safest OS Available?" (https://tech.co/antivirus-software/what-is-the-most-secure-os). *Tech.co*. Retrieved 19 December 2023.
87. Sanghavi, Alok (21 May 2010). "What is formal verification?". *EE Times_Asia*.
88. Ferraiolo, D.F. & Kuhn, D.R. (October 1992). "Role-Based Access Control" (http://csrc.nist.gov/groups/SNS/rbac/documents/ferraiolo-kuhn-92.pdf) (PDF). *15th National Computer Security Conference*: 554–563.
89. Sandhu, R; Coyne, EJ; Feinstein, HL; Youman, CE (August 1996). "Role-Based Access Control Models" (http://csrc.nist.gov/rbac/sandhu96.pdf) (PDF). *IEEE Computer*. **29** (2): 38–47. CiteSeerX 10.1.1.50.7649 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.50.7649). doi:10.1109/2.485845 (https://doi.org/10.1109%2F2.485845). S2CID 1958270 (https://api.semanticscholar.org/CorpusID:1958270).
90. Abreu, Vilmar; Santin, Altair O.; Viegas, Eduardo K.; Stihler, Maicon (2017). *A multi-domain role activation model* (https://secplab.ppgia.pucpr.br/files/papers/2017-1.pdf) (PDF). 2017 IEEE International Conference on Communications (ICC). IEEE Press. pp. 1–6. doi:10.1109/ICC.2017.7997247 (https://doi.org/10.1109%2FICC.2017.7997247). ISBN 978-1-4673-8999-0. S2CID 6185138 (https://api.semanticscholar.org/CorpusID:6185138).
91. A.C. O'Connor & R.J. Loomis (2002). *Economic Analysis of Role-Based Access Control* (http://csrc.nist.gov/groups/SNS/rbac/documents/20101219_RBAC2_Final_Report.pdf) (PDF). Research Triangle Institute. p. 145.
92. "Studies prove once again that users are the weakest link in the security chain" (https://www.csoonline.com/article/2137210/security-awareness/studies-prove-once-again-that-users-are-the-weakest-link-in-the-security-chain.html). *CSO Online*. 22 January 2014. Retrieved 8 October 2018.
93. "The Role of Human Error in Successful Security Attacks" (https://securityintelligence.com/the-role-of-human-error-in-successful-security-attacks/). *IBM Security Intelligence*. 2 September 2014. Retrieved 8 October 2018.
94. "90% of security incidents trace back to PEBKAC and ID10T errors" (https://www.computerworld.com/article/2910316/90-of-security-incidents-trace-back-to-pebkac-and-id10t-errors.html). *Computerworld*. 15 April 2015. Retrieved 8 October 2018.
95. "Protect your online banking with 2FA" (https://www.nzba.org.nz/2018/10/08/protect-your-online-banking-with-2fa/). *NZ Bankers Association*. 7 October 2018. Retrieved 7 September 2019.
96. "IBM Security Services 2014 Cyber Security Intelligence Index" (https://pcsite.co.uk/computer-security/IBM_Security_Services_2014_Cyber_Security_Intelligence_Index.pdf) (PDF). *PcSite*. 2014. Retrieved 9 October 2020.
97. Caldwell, Tracey (12 February 2013). "Risky business: why security awareness is crucial for employees" (https://www.theguardian.com/media-network/media-network-blog/2013/feb/12/business-cyber-security-risks-employees). *The Guardian*. Retrieved 8 October 2018.
98. "Developing a Security Culture" (https://web.archive.org/web/20181009013120/https://www.cpni.gov.uk/developing-security-culture). *CPNI – Centre for the Protection of National Infrastructure*. Archived from the original (https://www.cpni.gov.uk/developing-security-culture) on 9 October 2018. Retrieved 8 October 2018.
99. "Cyber Hygiene – ENISA" (https://www.enisa.europa.eu/publications/cyber-hygiene). Retrieved 27 September 2018.

100. Kaljulaid, Kersti (16 October 2017). "President of the Republic at the Aftenposten's Technology Conference" (https://president.ee/en/official-duties/speeches/13671-president-of-the-republic-at-the-aftenpostens-technology-conference/index.html). Retrieved 27 September 2018.
101. "Cyber security breaches survey 2023" (https://www.gov.uk/government/statistics/cyber-security-breaches-survey-2023/cyber-security-breaches-survey-2023). *GOV.UK*. Retrieved 27 December 2023.
102. Kuchler, Hannah (27 April 2015). "Security execs call on companies to improve 'cyber hygiene' " (https://ghostarchive.org/archive/20221210/https://www.ft.com/content/8468cfda-e9e3-11e4-a687-00144feab7de). *Financial Times*. Archived from the original (https://www.ft.com/content/8468cfda-e9e3-11e4-a687-00144feab7de) on 10 December 2022. Retrieved 27 September 2018.
103. "From AI to Russia, Here's How Estonia's President Is Planning for the Future" (https://www.wired.com/story/from-ai-to-russia-heres-how-estonias-president-is-planning-for-the-future/). *Wired*. Retrieved 28 September 2018.
104. "Professor Len Adleman explains how he coined the term "computer virus" " (https://www.welivesecurity.com/2017/11/01/professor-len-adleman-explains-computer-virus-term/). *WeLiveSecurity*. 1 November 2017. Retrieved 28 September 2018.
105. "Statement of Dr. Vinton G. Cerf" (https://www.jec.senate.gov/archive/Documents/Hearings/cerf22300.htm). *www.jec.senate.gov*. Retrieved 28 September 2018.
106. Promoting Good Cyber Hygiene Act of 2017 (https://www.congress.gov/bill/115th-congress/house-bill/3010) at Congress.gov
107. "Analysis | The Cybersecurity 202: Agencies struggling with basic cybersecurity despite Trump's pledge to prioritize it" (https://www.washingtonpost.com/news/powerpost/paloma/the-cybersecurity-202/2018/07/26/the-cybersecurity-202-agencies-struggling-with-basic-cybersecurity-despite-trump-s-pledge-to-prioritize-it/5b58a84e1b326b1e64695548/). *The Washington Post*. Retrieved 28 September 2018.
108. "Protected Voices" (https://www.fbi.gov/investigate/counterintelligence/foreign-influence/protected-voices). *Federal Bureau of Investigation*. Retrieved 28 September 2018.
109. Lin, Tom C. W. (3 July 2017). "The New Market Manipulation". *Emory Law Journal*. **66**: 1253. SSRN 2996896 (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2996896).
110. Lin, Tom C. W. (2016). "Financial Weapons of War". *Minnesota Law Review*. SSRN 2765010 (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2765010).
111. Cole, Jeffrey I.; Suman, Michael; Schramm, Phoebe; van Bel, Daniel; Lunn, B.; Maguire, Phyllisane; Hanson, Koran; Singh, Rajesh; Aquino, Jedrix-Sean; Lebo, Harlan (2000). The UCLA Internet report: Surveying the digital future (https://web.archive.org/web/20030423221926/http://ccp.ucla.edu/UCLA-Internet-Report-2000.pdf) (PDF). *ccp.ucla.edu* (Report). Archived from the original (http://ccp.ucla.edu/UCLA-Internet-Report-2000.pdf) (PDF) on 23 April 2003. Retrieved 15 September 2023.
112. Pagliery, Jose (18 November 2014). "Hackers attacked the U.S. energy grid 79 times this year" (https://money.cnn.com/2014/11/18/technology/security/energy-grid-hack/). *CNN Money*. Cable News Network. Archived (https://web.archive.org/web/20150218070238/https://money.cnn.com/2014/11/18/technology/security/energy-grid-hack) from the original on 18 February 2015. Retrieved 16 April 2015.
113. Neumann, P. G. (1997). *Computer Security in Aviation: Vulnerabilities, Threats, and Risks* (https://www.csl.sri.com/~neumann/air.html). International Conference on Aviation Safety and Security in the 21st Century, White House Commission on Safety and Security.
114. Dillingham, Gerald L. (20 September 2001). Aviation security: terrorist acts demonstrate urgent need to improve security at the nation's airports (https://rosap.ntl.bts.gov/view/dot/33937) (Report). United States. General Accounting Office.

115. "Air Traffic Control Systems Vulnerabilities Could Make for Unfriendly Skies [Black Hat] – SecurityWeek.Com" (http://www.securityweek.com/air-traffic-control-systems-vulnerabilities-could-make-unfriendly-skies-black-hat). 27 July 2012. Archived (https://web.archive.org/web/20150208070914/http://www.securityweek.com/air-traffic-control-systems-vulnerabilities-could-make-unfriendly-skies-black-hat) from the original on 8 February 2015.
116. "Hacker Says He Can Break into Airplane Systems Using In-Flight Wi-Fi" (https://www.npr.org/blogs/alltechconsidered/2014/08/04/337794061/hacker-says-he-can-break-into-airplane-systems-using-in-flight-wi-fi). *NPR*. 4 August 2014. Archived (https://web.archive.org/web/20150208072554/http://www.npr.org/blogs/alltechconsidered/2014/08/04/337794061/hacker-says-he-can-break-into-airplane-systems-using-in-flight-wi-fi) from the original on 8 February 2015. Retrieved 19 March 2020.
117. Finkle, Jim (4 August 2014). "Hacker says to show passenger jets at risk of cyber attack" (https://www.reuters.com/article/us-cybersecurity-hackers-airplanes-idUSKBN0G40WQ20140804). *Reuters*. Archived (https://web.archive.org/web/20151013061705/http://www.reuters.com/article/2014/08/04/us-cybersecurity-hackers-airplanes-idUSKBN0G40WQ20140804) from the original on 13 October 2015. Retrieved 21 November 2021.
118. Cesar, Alan (15 December 2023). "Online course bolsters cybersecurity in aviation" (https://engineering.purdue.edu/AAE/Aerogram/2023-2024/articles/41-cyber-course). *Aerogram*. Purdue University School of Aeronautics and Astronautics. Retrieved 9 January 2024.
119. "Pan-European Network Services (PENS) – Eurocontrol.int" (https://www.eurocontrol.int/articles/pan-european-network-services-pens). Archived (https://web.archive.org/web/20161212175606/https://www.eurocontrol.int/articles/pan-european-network-services-pens) from the original on 12 December 2016.
120. "Centralised Services: NewPENS moves forward – Eurocontrol.int" (https://www.eurocontrol.int/news/centralised-services-newpens-moves-forward). *Eurocontrol*. 17 January 2016. Archived (https://web.archive.org/web/20170319025329/https://www.eurocontrol.int/news/centralised-services-newpens-moves-forward) from the original on 19 March 2017.
121. "NextGen Data Communication" (https://web.archive.org/web/20150313110025/http://www.faa.gov/nextgen/update/progress_and_plans/data_comm/). FAA. Archived from the original (https://www.faa.gov/nextgen/update/progress_and_plans/data_comm/) on 13 March 2015. Retrieved 15 June 2017.
122. "e-Passports | Homeland Security" (https://www.dhs.gov/e-passports). *www.dhs.gov*. Retrieved 3 February 2023.
123. "The Australian ePassport. Australian Government Department of Foreign Affairs and Trade website" (https://web.archive.org/web/20150109033115/http://www.dfat.gov.au/dept/passports/). Archived from the original (http://www.dfat.gov.au/dept/passports/) on 9 January 2015. Retrieved 1 May 2023.
124. "Is Your Watch Or Thermostat A Spy? Cybersecurity Firms Are On It" (https://www.npr.org/blogs/alltechconsidered/2014/08/06/338334508/is-your-watch-or-thermostat-a-spy-cyber-security-firms-are-on-it). *NPR*. 6 August 2014. Archived (https://web.archive.org/web/20150211064650/http://www.npr.org/blogs/alltechconsidered/2014/08/06/338334508/is-your-watch-or-thermostat-a-spy-cyber-security-firms-are-on-it) from the original on 11 February 2015.
125. Kruse, CB; Smith, B; Vanderlinden, H; Nealand, A (21 July 2017). "Security Techniques for the Electronic Health Records" (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5522514). *Journal of Medical Systems*. **41** (8): 127. doi:10.1007/s10916-017-0778-4 (https://doi.org/10.1007%2Fs10916-017-0778-4). PMC 5522514 (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5522514). PMID 28733949 (https://pubmed.ncbi.nlm.nih.gov/28733949).
126. Backman, Melvin (18 September 2014). "Home Depot: 56 million cards exposed in breach" (https://money.cnn.com/2014/09/18/technology/security/home-depot-hack/). CNNMoney. Archived (https://web.archive.org/web/20141218221105/https://money.cnn.com/2014/09/18/technology/security/home-depot-hack/) from the original on 18 December 2014.

127. "Staples: Breach may have affected 1.16 million customers' cards" (http://fortune.com/2014/12/19/staples-cards-affected-breach/). *Fortune.com*. 19 December 2014. Archived (https://web.archive.org/web/20141221160612/http://fortune.com/2014/12/19/staples-cards-affected-breach/) from the original on 21 December 2014. Retrieved 21 December 2014.
128. "Target: 40 million credit cards compromised" (https://money.cnn.com/2013/12/18/news/companies/target-credit-card/index.html). *CNN*. 19 December 2013. Archived (https://web.archive.org/web/20171201035530/https://money.cnn.com/2013/12/18/news/companies/target-credit-card/index.html) from the original on 1 December 2017. Retrieved 29 November 2017.
129. Cowley, Stacy (2 October 2017). "2.5 Million More People Potentially Exposed in Equifax Breach" (https://www.nytimes.com/2017/10/02/business/equifax-breach.html). *The New York Times*. Archived (https://web.archive.org/web/20171201054900/https://www.nytimes.com/2017/10/02/business/equifax-breach.html) from the original on 1 December 2017. Retrieved 29 November 2017.
130. Finkle, Jim (23 April 2014). "Exclusive: FBI warns healthcare sector vulnerable to cyber attacks" (https://www.reuters.com/article/us-cybersecurity-healthcare-fbi-exclusiv-idUSBREA3M1Q920140423). *Reuters*. Archived (https://web.archive.org/web/20160604120725/http://www.reuters.com/article/us-cybersecurity-healthcare-fbi-exclusiv-idUSBREA3M1Q920140423) from the original on 4 June 2016. Retrieved 23 May 2016.
131. Seals, Tara (6 November 2015). "Lack of Employee Security Training Plagues US Businesses" (https://www.infosecurity-magazine.com/news/lack-of-employee-security-training/). *Infosecurity Magazine*. Archived (https://web.archive.org/web/20171109081033/https://www.infosecurity-magazine.com/news/lack-of-employee-security-training/) from the original on 9 November 2017. Retrieved 8 November 2017.
132. Bright, Peter (15 February 2011). "Anonymous speaks: the inside story of the HBGary hack" (https://arstechnica.com/tech-policy/news/2011/02/anonymous-speaks-the-inside-story-of-the-hbgary-hack.ars/). Arstechnica.com. Archived (https://web.archive.org/web/20110327045801/http://arstechnica.com/tech-policy/news/2011/02/anonymous-speaks-the-inside-story-of-the-hbgary-hack.ars) from the original on 27 March 2011. Retrieved 29 March 2011.
133. Anderson, Nate (9 February 2011). "How one man tracked down Anonymous – and paid a heavy price" (https://arstechnica.com/tech-policy/news/2011/02/how-one-security-firm-tracked-anonymousand-paid-a-heavy-price.ars/). Arstechnica.com. Archived (https://web.archive.org/web/20110329090824/http://arstechnica.com/tech-policy/news/2011/02/how-one-security-firm-tracked-anonymousand-paid-a-heavy-price.ars) from the original on 29 March 2011. Retrieved 29 March 2011.
134. Palilery, Jose (24 December 2014). "What caused Sony hack: What we know now" (https://money.cnn.com/2014/12/24/technology/security/sony-hack-facts/). *CNN Money*. Archived (https://web.archive.org/web/20150104195455/https://money.cnn.com/2014/12/24/technology/security/sony-hack-facts/) from the original on 4 January 2015. Retrieved 4 January 2015.
135. Cook, James (16 December 2014). "Sony Hackers Have Over 100 Terabytes Of Documents. Only Released 200 Gigabytes So Far" (http://www.businessinsider.com/the-sony-hackers-still-have-a-massive-amount-of-data-that-hasnt-been-leaked-yet-2014-12). *Business Insider*. Archived (https://web.archive.org/web/20141217204735/http://www.businessinsider.com/the-sony-hackers-still-have-a-massive-amount-of-data-that-hasnt-been-leaked-yet-2014-12) from the original on 17 December 2014. Retrieved 18 December 2014.
136. Lee, Timothy B. (18 January 2015). "The next frontier of hacking: your car" (https://www.vox.com/2015/1/18/7629603/car-hacking-dangers). *Vox*. Archived (https://web.archive.org/web/20170317212726/http://www.vox.com/2015/1/18/7629603/car-hacking-dangers) from the original on 17 March 2017.

137. Tracking & Hacking: Security & Privacy Gaps Put American Drivers at Risk (http://www.markey.senate.gov/imo/media/doc/2015-02-06_MarkeyReport-Tracking_Hacking_CarSecurity%202.pdf) (PDF) (Report). 6 February 2015. Archived (https://web.archive.org/web/20161109040112/http://www.markey.senate.gov/imo/media/doc/2015-02-06_MarkeyReport-Tracking_Hacking_CarSecurity%202.pdf) (PDF) from the original on 9 November 2016. Retrieved 4 November 2016.
138. "Cybersecurity expert: It will take a 'major event' for companies to take this issue seriously" (https://www.aol.com/article/news/2016/12/26/expert-warns-major-event-will-need-to-happen-for-cybersecurity/21632630/). *AOL.com*. 5 January 2017. Archived (https://web.archive.org/web/20170120180918/https://www.aol.com/article/news/2016/12/26/expert-warns-major-event-will-need-to-happen-for-cybersecurity/21632630/) from the original on 20 January 2017. Retrieved 22 January 2017.
139. "The problem with self-driving cars: who controls the code?" (https://www.theguardian.com/technology/2015/dec/23/the-problem-with-self-driving-cars-who-controls-the-code). *The Guardian*. 23 December 2015. Archived (https://web.archive.org/web/20170316152605/https://www.theguardian.com/technology/2015/dec/23/the-problem-with-self-driving-cars-who-controls-the-code) from the original on 16 March 2017. Retrieved 22 January 2017.
140. Checkoway, Stephen; McCoy, Damon; Kantor, Brian; Anderson, Danny; Shacham, Hovav; Savage, Stefan; Koscher, Karl; Czeskis, Alexei; Roesner, Franziska; Kohno, Tadayoshi (2011). *Comprehensive Experimental Analyses of Automotive Attack Surfaces* (http://www.autosec.org/pubs/cars-usenixsec2011.pdf) (PDF). SEC'11 Proceedings of the 20th USENIX conference on Security. Berkeley, California, US: USENIX Association. p. 6. Archived (https://web.archive.org/web/20150221064614/http://www.autosec.org/pubs/cars-usenixsec2011.pdf) (PDF) from the original on 21 February 2015.
141. Greenberg, Andy (21 July 2015). "Hackers Remotely Kill a Jeep on the Highway – With Me in It" (https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/). *Wired*. Archived (https://web.archive.org/web/20170119103855/https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/) from the original on 19 January 2017. Retrieved 22 January 2017.
142. "Hackers take control of car, drive it into a ditch" (https://www.independent.co.uk/news/science/hackers-remotely-carjack-jeep-from-10-miles-away-and-drive-it-into-ditch-10406554.html). *The Independent*. 22 July 2015. Archived (https://web.archive.org/web/20170202061247/http://www.independent.co.uk/news/science/hackers-remotely-carjack-jeep-from-10-miles-away-and-drive-it-into-ditch-10406554.html) from the original on 2 February 2017. Retrieved 22 January 2017.
143. "Tesla fixes software bug that allowed Chinese hackers to control car remotely" (https://www.telegraph.co.uk/technology/2016/09/21/tesla-fixes-software-bug-that-allowed-chinese-hackers-to-control/). *The Telegraph*. 21 September 2016. Archived (https://web.archive.org/web/20170202014932/http://www.telegraph.co.uk/technology/2016/09/21/tesla-fixes-software-bug-that-allowed-chinese-hackers-to-control/) from the original on 2 February 2017. Retrieved 22 January 2017.
144. Kang, Cecilia (19 September 2016). "Self-Driving Cars Gain Powerful Ally: The Government" (https://www.nytimes.com/2016/09/20/technology/self-driving-cars-guidelines.html). *The New York Times*. Archived (https://web.archive.org/web/20170214045032/https://www.nytimes.com/2016/09/20/technology/self-driving-cars-guidelines.html?_r=0) from the original on 14 February 2017. Retrieved 22 January 2017.
145. "Federal Automated Vehicles Policy" (https://www.transportation.gov/sites/dot.gov/files/docs/AV%20policy%20guidance%20PDF.pdf) (PDF). Archived (https://web.archive.org/web/20170121161404/https://www.transportation.gov/sites/dot.gov/files/docs/AV%20policy%20guidance%20PDF.pdf) (PDF) from the original on 21 January 2017. Retrieved 22 January 2017.
146. "Vehicle Cybersecurity" (https://www.nhtsa.gov/technology-innovation/vehicle-cybersecurity). *nhtsa.gov*. Retrieved 25 November 2022.
147. "Thales supplies smart driver license to 4 states in Mexico" (https://www.thalesgroup.com/en/markets/digital-identity-and-security/government/customer-cases/mexico). *Thales Group*.

148. "4 Companies Using RFID for Supply Chain Management" (https://www.atlasrfidstore.com/rfid-insider/4-companies-using-rfid-for-supply-chain-management). *atlasRFIDstore*. Retrieved 3 February 2023.
149. "The Cutting Edge of RFID Technology and Applications for Manufacturing and Distribution" (https://www.supplychainmarket.com/doc/the-cutting-edge-of-rfid-technology-and-appli-0001). *Supply Chain Market*.
150. Rahman, Mohammad Anwar; Khadem, Mohammad Miftaur; Sarder, MD. *Application of RFID in Supply Chain System*. Proceedings of the 2010 International Conference on Industrial Engineering and Operations Management Dhaka, Bangladesh, January 9 – 10, 2010. CiteSeerX 10.1.1.397.7831 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.397.7831).
151. "Gary McKinnon profile: Autistic 'hacker' who started writing computer programs at 14" (https://www.telegraph.co.uk/news/worldnews/northamerica/usa/4320901/Gary-McKinnon-profile-Autistic-hacker-who-started-writing-computer-programs-at-14.html). *The Daily Telegraph*. London. 23 January 2009. Archived (https://web.archive.org/web/20100602065423/http://www.telegraph.co.uk/news/worldnews/northamerica/usa/4320901/Gary-McKinnon-profile-Autistic-hacker-who-started-writing-computer-programs-at-14.html) from the original on 2 June 2010.
152. "Gary McKinnon extradition ruling due by 16 October" (https://www.bbc.co.uk/news/uk-19506090). *BBC News*. 6 September 2012. Archived (https://web.archive.org/web/20120906185731/http://www.bbc.co.uk/news/uk-19506090) from the original on 6 September 2012. Retrieved 25 September 2012.
153. *Mckinnon V Government of The United States of America and Another* (House of Lords 16 June 2008) ("15. ... alleged to total over $700,000"), Text (https://publications.parliament.uk/pa/ld200708/ldjudgmt/jd080730/mckinn-1.htm).
154. "Fresh Leak on US Spying: NSA Accessed Mexican President's Email" (https://web.archive.org/web/20151106193613/http://www.spiegel.de/international/world/nsa-hacked-email-account-of-mexican-president-a-928817.html). *SPIEGEL ONLINE*. 20 October 2013. Archived from the original (http://www.spiegel.de/international/world/nsa-hacked-email-account-of-mexican-president-a-928817.html) on 6 November 2015.
155. Sanders, Sam (4 June 2015). "Massive Data Breach Puts 4 Million Federal Employees' Records at Risk" (https://www.npr.org/sections/thetwo-way/2015/06/04/412086068/massive-data-breach-puts-4-million-federal-employees-records-at-risk). *NPR*. Archived (https://web.archive.org/web/20150605041629/http://www.npr.org/sections/thetwo-way/2015/06/04/412086068/massive-data-breach-puts-4-million-federal-employees-records-at-risk) from the original on 5 June 2015. Retrieved 5 June 2015.
156. Liptak, Kevin (4 June 2015). "U.S. government hacked; feds think China is the culprit" (http://www.cnn.com/2015/06/04/politics/federal-agency-hacked-personnel-management/). *CNN*. Archived (https://web.archive.org/web/20150606063139/http://www.cnn.com/2015/06/04/politics/federal-agency-hacked-personnel-management/) from the original on 6 June 2015. Retrieved 5 June 2015.
157. Gallagher, Sean. "Encryption "would not have helped" at OPM, says DHS official" (https://arstechnica.com/security/2015/06/encryption-would-not-have-helped-at-opm-says-dhs-official/). Archived (https://web.archive.org/web/20170624014751/https://arstechnica.com/security/2015/06/encryption-would-not-have-helped-at-opm-says-dhs-official/) from the original on 24 June 2017.
158. Davis, Michelle R. (19 October 2015). "Schools Learn Lessons From Security Breaches" (http://www.edweek.org/ew/articles/2015/10/21/lessons-learned-from-security-breaches.html). *Education Week*. Archived (https://web.archive.org/web/20160610130749/http://www.edweek.org/ew/articles/2015/10/21/lessons-learned-from-security-breaches.html) from the original on 10 June 2016. Retrieved 23 May 2016.

159. "GE's Introduces ACUVision as a Single Panel Solution" (https://www.securityinfowatch.com/access-identity/access-control/press-release/10577631/ge-infrastructure-security-ges-introduces-acuvision-as-a-single-panel-solution). *www.securityinfowatch.com*. Security Info Watch. 11 August 2005. Retrieved 24 September 2019.
160. "Internet of Things Global Standards Initiative" (http://www.itu.int/en/ITU-T/gsi/iot/Pages/default.aspx). *ITU*. Archived (https://web.archive.org/web/20150626125229/http://www.itu.int/en/ITU-T/gsi/iot/Pages/default.aspx) from the original on 26 June 2015. Retrieved 26 June 2015.
161. Singh, Jatinder; Pasquier, Thomas; Bacon, Jean; Ko, Hajoon; Eyers, David (2015). "Twenty Cloud Security Considerations for Supporting the Internet of Things" (https://dash.harvard.edu/bitstream/1/35349952/1/iot-2016.pdf) (PDF). *IEEE Internet of Things Journal*. **3** (3): 269–284. doi:10.1109/JIOT.2015.2460333 (https://doi.org/10.1109%2FJIOT.2015.2460333). S2CID 4732406 (https://api.semanticscholar.org/CorpusID:4732406).
162. Clearfield, Chris. "Why The FTC Can't Regulate The Internet Of Things" (https://www.forbes.com/sites/chrisclearfield/2013/09/18/why-the-ftc-cant-regulate-the-internet-of-things/). *Forbes*. Archived (https://web.archive.org/web/20150627090938/http://www.forbes.com/sites/chrisclearfield/2013/09/18/why-the-ftc-cant-regulate-the-internet-of-things/) from the original on 27 June 2015. Retrieved 26 June 2015.
163. "Internet of Things: Science Fiction or Business Fact?" (https://hbr.org/resources/pdfs/comm/verizon/18980_HBR_Verizon_IoT_Nov_14.pdf) (PDF). *Harvard Business Review*. Archived (https://web.archive.org/web/20150317052909/https://hbr.org/resources/pdfs/comm/verizon/18980_HBR_Verizon_IoT_Nov_14.pdf) (PDF) from the original on 17 March 2015. Retrieved 4 November 2016.
164. Vermesan, Ovidiu; Friess, Peter. "Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems" (http://www.internet-of-things-research.eu/pdf/Converging_Technologies_for_Smart_Environments_and_Integrated_Ecosystems_IERC_Book_Open_Access_2013.pdf) (PDF). River Publishers. Archived (https://web.archive.org/web/20161012010519/http://www.internet-of-things-research.eu/pdf/Converging_Technologies_for_Smart_Environments_and_Integrated_Ecosystems_IERC_Book_Open_Access_2013.pdf) (PDF) from the original on 12 October 2016. Retrieved 4 November 2016.
165. Clearfield, Chris (20 June 2013). "Rethinking Security for the Internet of Things" (http://blogs.hbr.org/2013/06/rethinking-security-for-the-in/). *Harvard Business Review*. Archived (https://web.archive.org/web/20130920145534/http://blogs.hbr.org/2013/06/rethinking-security-for-the-in/) from the original on 20 September 2013.
166. "Hotel room burglars exploit critical flaw in electronic door locks" (https://arstechnica.com/security/2012/11/hotel-room-burglars-exploit-critical-flaw-in-electronic-door-locks/). *Ars Technica*. 26 November 2012. Archived (https://web.archive.org/web/20160514002208/http://arstechnica.com/security/2012/11/hotel-room-burglars-exploit-critical-flaw-in-electronic-door-locks/) from the original on 14 May 2016. Retrieved 23 May 2016.
167. "Hospital Medical Devices Used As Weapons in Cyberattacks" (http://www.darkreading.com/vulnerabilities---threats/hospital-medical-devices-used-as-weapons-in-cyberattacks/d/d-id/1320751). *Dark Reading*. 6 August 2015. Archived (https://web.archive.org/web/20160529002947/http://www.darkreading.com/vulnerabilities---threats/hospital-medical-devices-used-as-weapons-in-cyberattacks/d/d-id/1320751) from the original on 29 May 2016. Retrieved 23 May 2016.
168. Kirk, Jeremy (17 October 2012). "Pacemaker hack can deliver deadly 830-volt jolt" (http://www.computerworld.com/article/2492453/malware-vulnerabilities/pacemaker-hack-can-deliver-deadly-830-volt-jolt.html). *Computerworld*. Archived (https://web.archive.org/web/20160604201841/http://www.computerworld.com/article/2492453/malware-vulnerabilities/pacemaker-hack-can-deliver-deadly-830-volt-jolt.html) from the original on 4 June 2016. Retrieved 23 May 2016.

169. "How Your Pacemaker Will Get Hacked" (http://www.thedailybeast.com/articles/2014/11/17/how-your-pacemaker-will-get-hacked.html). *The Daily Beast*. Kaiser Health News. 17 November 2014. Archived (https://web.archive.org/web/20160520155616/http://www.thedailybeast.com/articles/2014/11/17/how-your-pacemaker-will-get-hacked.html) from the original on 20 May 2016. Retrieved 23 May 2016.
170. Leetaru, Kalev. "Hacking Hospitals And Holding Hostages: Cybersecurity In 2016" (https://www.forbes.com/sites/kalevleetaru/2016/03/29/hacking-hospitals-and-holding-hostages-cybersecurity-in-2016/). *Forbes*. Archived (https://web.archive.org/web/20161229104021/http://www.forbes.com/sites/kalevleetaru/2016/03/29/hacking-hospitals-and-holding-hostages-cybersecurity-in-2016/) from the original on 29 December 2016. Retrieved 29 December 2016.
171. "Cyber-Angriffe: Krankenhäuser rücken ins Visier der Hacker" (http://www.wiwo.de/technologie/digitale-welt/cyber-angriffe-krankenhaeuser-ruecken-ins-visier-der-hacker/14946040.html). Wirtschafts Woche. 7 December 2016. Archived (https://web.archive.org/web/20161229101724/http://www.wiwo.de/technologie/digitale-welt/cyber-angriffe-krankenhaeuser-ruecken-ins-visier-der-hacker/14946040.html) from the original on 29 December 2016. Retrieved 29 December 2016.
172. "Hospitals keep getting attacked by ransomware – Here's why" (http://www.businessinsider.com/hospital-ransomware-hack-2016-5). *Business Insider*. Archived (https://web.archive.org/web/20161229101247/http://www.businessinsider.com/hospital-ransomware-hack-2016-5) from the original on 29 December 2016. Retrieved 29 December 2016.
173. "MedStar Hospitals Recovering After 'Ransomware' Hack" (https://www.nbcnews.com/news/us-news/medstar-hospitals-recovering-after-ransomware-hack-n548121). *NBC News*. 31 March 2016. Archived (https://web.archive.org/web/20161229103355/https://www.nbcnews.com/news/us-news/medstar-hospitals-recovering-after-ransomware-hack-n548121) from the original on 29 December 2016. Retrieved 29 December 2016.
174. Pauli, Darren. "US hospitals hacked with ancient exploits" (https://www.theregister.co.uk/2016/06/28/medjack/). *The Register*. Archived (https://web.archive.org/web/20161116141207/http://www.theregister.co.uk/2016/06/28/medjack) from the original on 16 November 2016. Retrieved 29 December 2016.
175. Pauli, Darren. "Zombie OS lurches through Royal Melbourne Hospital spreading virus" (https://www.theregister.co.uk/2016/01/19/melbourne_hospital_pathology_wing_splattered_by_virus/). *The Register*. Archived (https://web.archive.org/web/20161229101019/http://www.theregister.co.uk/2016/01/19/melbourne_hospital_pathology_wing_splattered_by_virus/) from the original on 29 December 2016. Retrieved 29 December 2016.
176. "Hacked Lincolnshire hospital computer systems 'back up' " (https://www.bbc.com/news/uk-england-humber-37849746). *BBC News*. 2 November 2016. Archived (https://web.archive.org/web/20161229101819/http://www.bbc.com/news/uk-england-humber-37849746) from the original on 29 December 2016. Retrieved 29 December 2016.
177. "Lincolnshire operations cancelled after network attack" (https://www.bbc.com/news/uk-england-humber-37822084). *BBC News*. 31 October 2016. Archived (https://web.archive.org/web/20161229101209/http://www.bbc.com/news/uk-england-humber-37822084) from the original on 29 December 2016. Retrieved 29 December 2016.
178. "Legion cyber-attack: Next dump is sansad.nic.in, say hackers" (http://indianexpress.com/article/technology/tech-news-technology/legion-hacking-no-political-agenda-just-computer-geeks-says-hacker-4423167/). *The Indian Express*. 12 December 2016. Archived (https://web.archive.org/web/20161229100631/http://indianexpress.com/article/technology/tech-news-technology/legion-hacking-no-political-agenda-just-computer-geeks-says-hacker-4423167/) from the original on 29 December 2016. Retrieved 29 December 2016.

179. "Former New Hampshire Psychiatric Hospital Patient Accused Of Data Breach" (http://boston.cbslocal.com/2016/12/27/former-patient-accused-data-breech-new-hampshire-psychiatric-hospital/). CBS Boston. 27 December 2016. Archived (https://web.archive.org/web/20170929233237/http://boston.cbslocal.com/2016/12/27/former-patient-accused-data-breech-new-hampshire-psychiatric-hospital/) from the original on 29 September 2017. Retrieved 29 December 2016.
180. "Texas Hospital hacked, affects nearly 30,000 patient records" (http://www.healthcareitnews.com/news/texas-hospital-hacked-affects-nearly-30000-patient-records). Healthcare IT News. 4 November 2016. Archived (https://web.archive.org/web/20161229171117/http://www.healthcareitnews.com/news/texas-hospital-hacked-affects-nearly-30000-patient-records) from the original on 29 December 2016. Retrieved 29 December 2016.
181. Becker, Rachel (27 December 2016). "New cybersecurity guidelines for medical devices tackle evolving threats" (https://www.theverge.com/2016/12/27/14095166/fda-guidance-medical-device-cybersecurity-cyberattack-hacking-guidelines). *The Verge*. Archived (https://web.archive.org/web/20161228210257/http://www.theverge.com/2016/12/27/14095166/fda-guidance-medical-device-cybersecurity-cyberattack-hacking-guidelines) from the original on 28 December 2016. Retrieved 29 December 2016.
182. "Postmarket Management of Cybersecurity in Medical Devices" (https://www.fda.gov/downloads/MedicalDevices/DeviceRegulationandGuidance/GuidanceDocuments/UCM482022.pdf) (PDF). *Food and Drug Administration*. 28 December 2016. Archived (https://web.archive.org/web/20161229102808/https://www.fda.gov/downloads/MedicalDevices/DeviceRegulationandGuidance/GuidanceDocuments/UCM482022.pdf) (PDF) from the original on 29 December 2016. Retrieved 29 December 2016.
183. Brandt, Jaclyn (18 June 2018). "D.C. distributed energy proposal draws concerns of increased cybersecurity risks" (https://dailyenergyinsider.com/featured/13110-d-c-distributed-4:.energy-proposal-draws-concerns-of-increased-cybersecurity-risks/). *Daily Energy Insider*. Retrieved 4 July 2018.
184. "Current Releases - The Open Mobile Alliance" (http://www.openmobilealliance.org/Technical/release_program/scws_v1_0.aspx). *openmobilealliance.org*.
185. Cashell, B.; Jackson, W. D.; Jickling, M.; Webel, B. (2004). The Economic Impact of Cyber-Attacks (https://sgp.fas.org/crs/misc/RL32331.pdf) (PDF) (Report). Washington DC: Congressional Research Service, Government, and Finance Division. RL32331.
186. Gordon, Lawrence; Loeb, Martin (November 2002). "The Economics of Information Security Investment". *ACM Transactions on Information and System Security*. **5** (4): 438–457. doi:10.1145/581271.581274 (https://doi.org/10.1145%2F581271.581274). S2CID 1500788 (https://api.semanticscholar.org/CorpusID:1500788).
187. Sanger, David E.; Barnes, Julian E. (20 December 2021). "U.S. and Britain Help Ukraine Prepare for Potential Russian Cyberassault" (https://www.nytimes.com/2021/12/20/us/politics/russia-ukraine-cyberattacks.html). *The New York Times*. ISSN 0362-4331 (https://search.worldcat.org/issn/0362-4331). Retrieved 4 December 2023.
188. "Cyber-Attack Against Ukrainian Critical Infrastructure | CISA" (https://www.cisa.gov/news-events/ics-alerts/ir-alert-h-16-056-01). *www.cisa.gov*. 20 July 2021. Retrieved 4 December 2023.
189. Han, Chen; Dongre, Rituja (2014). "Q&A. What Motivates Cyber-Attackers?" (https://doi.org/10.22215%2Ftimreview%2F838). *Technology Innovation Management Review*. **4** (10): 40–42. doi:10.22215/timreview/838 (https://doi.org/10.22215%2Ftimreview%2F838). ISSN 1927-0321 (https://search.worldcat.org/issn/1927-0321).
190. Chermick, Steven; Freilich, Joshua; Holt, Thomas (April 2017). "Exploring the Subculture of Ideologically Motivated Cyber-Attackers". *Journal of Contemporary Criminal Justice*. **33** (3): 212–233. doi:10.1177/1043986217699100 (https://doi.org/10.1177%2F1043986217699100). S2CID 152277480 (https://api.semanticscholar.org/CorpusID:152277480).

191. Anderson, Ross (2020). *Security engineering: a guide to building dependable distributed systems* (3rd ed.). Indianapolis, IN: John Wiley & Sons. ISBN 978-1-119-64281-7. OCLC 1224516855 (https://search.worldcat.org/oclc/1224516855).
192. "The Leading Cloud Recruiting Software" (https://www.icims.com/gc/incident-response-procedures/). iCIMS. Retrieved 13 March 2021.
193. Wilcox, S. and Brown, B. (2005) 'Responding to Security Incidents – Sooner or Later Your Systems Will Be Compromised', *Journal of Health Care Compliance*, 7(2), pp. 41–48
194. Jonathan Zittrain, 'The Future of The Internet', Penguin Books, 2008
195. Information Security (https://fas.org/irp/gao/aim96084.htm) Archived (https://web.archive.org/web/20160306140354/http://fas.org/irp/gao/aim96084.htm) 6 March 2016 at the Wayback Machine. United States Department of Defense, 1986
196. "The TJX Companies, Inc. Victimized by Computer System Intrusion; Provides Information to Help Protect Customers" (http://www.businesswire.com/news/tjx/20070117005971/en) (Press release). The TJX Companies, Inc. 17 January 2007. Archived (https://web.archive.org/web/20120927014805/http://www.businesswire.com/news/tjx/20070117005971/en) from the original on 27 September 2012. Retrieved 12 December 2009.
197. Largest Customer Info Breach Grows (http://www.myfoxtwincities.com/myfox/pages/Home/Detail?contentId=2804836&version=3&locale=EN-US&layoutCode=TSTY&pageId=1.1.1) Archived (https://web.archive.org/web/20070928041047/http://www.myfoxtwincities.com/myfox/pages/Home/Detail?contentId=2804836&version=3&locale=EN-US&layoutCode=TSTY&pageId=1.1.1) 28 September 2007 at the Wayback Machine. MyFox Twin Cities, 29 March 2007.
198. "The Stuxnet Attack On Iran's Nuclear Plant Was 'Far More Dangerous' Than Previously Thought" (http://www.businessinsider.com/stuxnet-was-far-more-dangerous-than-previous-thought-2013-11). *Business Insider*. 20 November 2013. Archived (https://web.archive.org/web/20140509020404/http://www.businessinsider.com/stuxnet-was-far-more-dangerous-than-previous-thought-2013-11) from the original on 9 May 2014.
199. Reals, Tucker (24 September 2010). "Stuxnet Worm a U.S. Cyber-Attack on Iran Nukes?" (https://www.cbsnews.com/news/stuxnet-worm-a-us-cyber-attack-on-iran-nukes/). *CBS News*. Archived (https://web.archive.org/web/20131016133651/http://www.cbsnews.com/8301-501465_162-20017507-501465.html) from the original on 16 October 2013.
200. Zetter, Kim (17 February 2011). "Cyberwar Issues Likely to Be Addressed Only After a Catastrophe" (https://www.wired.com/threatlevel/2011/02/cyberwar-issues-likely-to-be-addressed-only-after-a-catastrophe). *Wired*. Archived (https://web.archive.org/web/20110218154415/http://www.wired.com/threatlevel/2011/02/cyberwar-issues-likely-to-be-addressed-only-after-a-catastrophe/) from the original on 18 February 2011. Retrieved 18 February 2011.
201. Carroll, Chris (18 October 2011). "Cone of silence surrounds U.S. cyberwarfare" (http://www.stripes.com/news/cone-of-silence-surrounds-u-s-cyberwarfare-1.158090). Stars and Stripes. Archived (https://web.archive.org/web/20120307021747/http://www.stripes.com/news/cone-of-silence-surrounds-u-s-cyberwarfare-1.158090) from the original on 7 March 2012. Retrieved 30 October 2011.
202. Bumgarner, John (27 April 2010). "Computers as Weapons of War" (https://web.archive.org/web/20111219174833/http://www.crows.org/images/stories/pdf/IOI/IO%20Journal_Vol2Iss2_0210.pdf) (PDF). IO Journal. Archived from the original (http://www.crows.org/images/stories/pdf/IOI/IO%20Journal_Vol2Iss2_0210.pdf) (PDF) on 19 December 2011. Retrieved 30 October 2011.

203. Greenwald, Glenn (6 June 2013). "NSA collecting phone records of millions of Verizon customers daily" (https://www.theguardian.com/world/2013/jun/06/nsa-phone-records-verizon-court-order). *The Guardian*. Archived (https://web.archive.org/web/20130816045641/http://www.theguardian.com/world/2013/jun/06/nsa-phone-records-verizon-court-order) from the original on 16 August 2013. Retrieved 16 August 2013. "Exclusive: Top secret court order requiring Verizon to hand over all call data shows scale of domestic surveillance under Obama"
204. Seipel, Hubert. "Transcript: ARD interview with Edward Snowden" (https://www.freesnowden.is/fr/2014/01/27/video-ard-interview-with-edward-snowden/). *La Foundation Courage*. Archived (https://web.archive.org/web/20140714174333/https://www.freesnowden.is/fr/2014/01/27/video-ard-interview-with-edward-snowden/) from the original on 14 July 2014. Retrieved 11 June 2014.
205. Newman, Lily Hay (9 October 2013). "Can You Trust NIST?" (https://spectrum.ieee.org/can-you-trust-nist). *IEEE Spectrum*. Archived (https://web.archive.org/web/20160201095426/https://spectrum.ieee.org/telecom/security/can-you-trust-nist) from the original on 1 February 2016.
206. "NIST Removes Cryptography Algorithm from Random Number Generator Recommendations" (https://www.nist.gov/itl/csd/sp800-90-042114.cfm). *National Institute of Standards and Technology*. 21 April 2014.
207. "New Snowden Leak: NSA Tapped Google, Yahoo Data Centers" (http://mashable.com/2013/10/30/nsa-google-yahoo-data-centers/) Archived (https://web.archive.org/web/20140709131535/http://mashable.com/2013/10/30/nsa-google-yahoo-data-centers/) 9 July 2014 at the Wayback Machine, 31 October 2013, Lorenzo Franceschi-Bicchierai, mashable.com
208. Riley, Michael; Elgin, Ben; Lawrence, Dune; Matlack, Carol (17 March 2014). "Target Missed Warnings in Epic Hack of Credit Card Data" (https://web.archive.org/web/20150127015928/http://www.businessweek.com/articles/2014-03-13/target-missed-alarms-in-epic-hack-of-credit-card-data). *Businessweek*. Archived from the original (http://www.businessweek.com/articles/2014-03-13/target-missed-alarms-in-epic-hack-of-credit-card-data) on 27 January 2015.
209. Rosenblatt, Seth (6 November 2014). "Home Depot says 53 million emails stolen" (https://www.cnet.com/news/53-million-emails-stolen-in-home-depot-breach/). *CNET*. CBS Interactive. Archived (https://web.archive.org/web/20141209035159/http://www.cnet.com/news/53-million-emails-stolen-in-home-depot-breach/) from the original on 9 December 2014.
210. "Millions more Americans hit by government personnel data hack" (https://www.reuters.com/article/us-cybersecurity-usa-idUSKCN0PJ2M420150709). *Reuters*. 9 July 2017. Archived (https://web.archive.org/web/20170228005352/http://www.reuters.com/article/us-cybersecurity-usa-idUSKCN0PJ2M420150709) from the original on 28 February 2017. Retrieved 25 February 2017.
211. Barrett, Devlin (4 June 2015). "U.S. Suspects Hackers in China Breached About four (4) Million People's Records, Officials Say" (https://www.wsj.com/articles/u-s-suspects-hackers-in-china-behind-government-data-breach-sources-say-1433451888). *The Wall Street Journal*. Archived (https://web.archive.org/web/20150604215718/http://www.wsj.com/articles/u-s-suspects-hackers-in-china-behind-government-data-breach-sources-say-1433451888) from the original on 4 June 2015.
212. Risen, Tom (5 June 2015). "China Suspected in Theft of Federal Employee Records" (https://web.archive.org/web/20150606064331/http://www.usnews.com/news/articles/2015/06/05/china-suspected-in-theft-of-federal-employee-records). *U.S. News & World Report*. Archived from the original (https://www.usnews.com/news/articles/2015/06/05/china-suspected-in-theft-of-federal-employee-records) on 6 June 2015.
213. Zengerle, Patricia (19 July 2015). "Estimate of Americans hit by government personnel data hack skyrockets" (https://www.reuters.com/article/us-cybersecurity-usa-idUSKCN0PJ2M420150709). *Reuters*. Archived (https://web.archive.org/web/20150710075449/http://www.reuters.com/article/2015/07/09/us-cybersecurity-usa-idUSKCN0PJ2M420150709) from the original on 10 July 2015.

214. Sanger, David (5 June 2015). "Hacking Linked to China Exposes Millions of U.S. Workers" (https://www.nytimes.com/2015/06/05/us/breach-in-a-federal-computer-system-exposes-personnel-data.html). *The New York Times*. Archived (https://web.archive.org/web/20150605135158/http://www.nytimes.com/2015/06/05/us/breach-in-a-federal-computer-system-exposes-personnel-data.html) from the original on 5 June 2015.
215. Mansfield-Devine, Steve (1 September 2015). "The Ashley Madison affair". *Network Security*. **2015** (9): 8–16. doi:10.1016/S1353-4858(15)30080-5 (https://doi.org/10.1016%2FS1353-4858%2815%2930080-5).
216. Turton, W.; Mehrotra, K. (4 June 2021). "Hackers Breached Colonial Pipeline Using Compromised Password" (https://www.bloomberg.com/news/articles/2021-06-04/hackers-breached-colonial-pipeline-using-compromised-password). Bloomberg L.P. Retrieved 3 December 2023.
217. "Mikko Hypponen: Fighting viruses, defending the net" (http://www.ted.com/talks/mikko_hypponen_fighting_viruses_defending_the_net.html). TED. Archived (https://web.archive.org/web/20130116010603/http://www.ted.com/talks/mikko_hypponen_fighting_viruses_defending_the_net.html) from the original on 16 January 2013.
218. "Mikko Hypponen – Behind Enemy Lines" (https://www.youtube.com/watch?v=0TMFRO66Wv4). Hack in the Box Security Conference. 9 December 2012. Archived (https://web.archive.org/web/20161125075257/https://www.youtube.com/watch?v=0TMFRO66Wv4) from the original on 25 November 2016.
219. "Ensuring the Security of Federal Information Systems and Cyber Critical Infrastructure and Protecting the Privacy of Personally Identifiable Information" (http://www.gao.gov/highrisk/protecting_the_federal_government_information_systems/why_did_study). Government Accountability Office. Archived (https://web.archive.org/web/20151119221200/http://www.gao.gov/highrisk/protecting_the_federal_government_information_systems/why_did_study) from the original on 19 November 2015. Retrieved 3 November 2015.
220. King, Georgia (23 May 2018). "The Venn diagram between libertarians and crypto bros is so close it's basically a circle" (https://qz.com/1284178/almost-half-of-cryptocurrency-and-bitcoin-bros-identify-as-libertarian/). *Quartz*.
221. Kirby, Carrie (24 June 2011). "Former White House aide backs some Net regulation / Clarke says government, industry deserve 'F' in cyber security" (https://www.sfgate.com/business/article/Former-White-House-aide-backs-some-Net-regulation-2729985.php). *The San Francisco Chronicle*.
222. McCarthy, Daniel (11 June 2018). "Privatizing Political Authority: Cybersecurity, Public-Private Partnerships, and the Reproduction of Liberal Political Order" (https://www.cogitatiopress.com/politicsandgovernance/article/download/1335/800). *Politics and Governance*. **6** (2): 5–12. doi:10.17645/pag.v6i2.1335 (https://doi.org/10.17645%2Fpag.v6i2.1335).
223. "It's Time to Treat Cybersecurity as a Human Rights Issue" (https://www.hrw.org/news/2020/05/26/its-time-treat-cybersecurity-human-rights-issue). *Human Rights Watch*. 26 May 2020. Retrieved 26 May 2020.
224. "FIRST Mission" (https://www.first.org/about/mission/). FIRST. Retrieved 6 July 2018.
225. "FIRST Members" (https://www.first.org/members/). FIRST. Retrieved 6 July 2018.
226. "European council" (http://www.coe.int/t/DGHL/cooperation/economiccrime/cybercrime/default_en.asp). Archived (https://web.archive.org/web/20141203223358/http://www.coe.int/t/DGHL/cooperation/economiccrime/cybercrime/default_en.asp) from the original on 3 December 2014.
227. "MAAWG" (http://www.maawg.org/about_maawg). Archived (https://web.archive.org/web/20140923153548/http://www.maawg.org/about_maawg) from the original on 23 September 2014.
228. "MAAWG" (http://www.maawg.org/about/roster). Archived (https://web.archive.org/web/20141017165203/http://www.maawg.org/about/roster) from the original on 17 October 2014.

229. "Government of Canada Launches Canada's Cyber Security Strategy" (http://www.marketwired.com/press-release/government-of-canada-launches-canadas-cyber-security-strategy-1328661.htm). *Market Wired*. 3 October 2010. Archived (https://web.archive.org/web/20141102175904/http://www.marketwired.com/press-release/government-of-canada-launches-canadas-cyber-security-strategy-1328661.htm) from the original on 2 November 2014. Retrieved 1 November 2014.
230. "Canada's Cyber Security Strategy" (http://www.publicsafety.gc.ca/cnt/rsrcs/pblctns/cbr-scrt-strtgy/index-eng.aspx). *Public Safety Canada*. Government of Canada. Archived (https://web.archive.org/web/20141102175701/http://www.publicsafety.gc.ca/cnt/rsrcs/pblctns/cbr-scrt-strtgy/index-eng.aspx) from the original on 2 November 2014. Retrieved 1 November 2014.
231. "Action Plan 2010–2015 for Canada's Cyber Security Strategy" (http://www.publicsafety.gc.ca/cnt/rsrcs/pblctns/ctn-pln-cbr-scrt/index-eng.aspx). *Public Safety Canada*. Government of Canada. Archived (https://web.archive.org/web/20141102173436/http://www.publicsafety.gc.ca/cnt/rsrcs/pblctns/ctn-pln-cbr-scrt/index-eng.aspx) from the original on 2 November 2014. Retrieved 3 November 2014.
232. "Cyber Incident Management Framework For Canada" (http://www.publicsafety.gc.ca/cnt/rsrcs/pblctns/cbr-ncdnt-frmwrk/index-eng.aspx#_Toc360619104). *Public Safety Canada*. Government of Canada. Archived (https://web.archive.org/web/20141102213822/http://www.publicsafety.gc.ca/cnt/rsrcs/pblctns/cbr-ncdnt-frmwrk/index-eng.aspx#_Toc360619104) from the original on 2 November 2014. Retrieved 3 November 2014.
233. "Action Plan 2010–2015 for Canada's Cyber Security Strategy" (http://www.publicsafety.gc.ca/cnt/rsrcs/pblctns/ctn-pln-cbr-scrt/index-eng.aspx). *Public Safety Canada*. Government of Canada. Archived (https://web.archive.org/web/20141102173436/http://www.publicsafety.gc.ca/cnt/rsrcs/pblctns/ctn-pln-cbr-scrt/index-eng.aspx) from the original on 2 November 2014. Retrieved 1 November 2014.
234. "Canadian Cyber Incident Response Centre" (http://www.publicsafety.gc.ca/cnt/ntnl-scrt/cbr-scrt/ccirc-ccric-eng.aspx). *Public Safety Canada*. Archived (https://web.archive.org/web/20141008035436/http://www.publicsafety.gc.ca/cnt/ntnl-scrt/cbr-scrt/ccirc-ccric-eng.aspx) from the original on 8 October 2014. Retrieved 1 November 2014.
235. "Cyber Security Bulletins" (http://www.publicsafety.gc.ca/cnt/rsrcs/cybr-ctr/index-eng.aspx). *Public Safety Canada*. Archived (https://web.archive.org/web/20141008194739/http://www.publicsafety.gc.ca/cnt/rsrcs/cybr-ctr/index-eng.aspx) from the original on 8 October 2014. Retrieved 1 November 2014.
236. "Report a Cyber Security Incident" (http://www.publicsafety.gc.ca/cnt/ntnl-scrt/cbr-scrt/rprt-eng.aspx). *Public Safety Canada*. Government of Canada. Archived (https://web.archive.org/web/20141111212708/http://www.publicsafety.gc.ca/cnt/ntnl-scrt/cbr-scrt/rprt-eng.aspx) from the original on 11 November 2014. Retrieved 3 November 2014.
237. "Government of Canada Launches Cyber Security Awareness Month With New Public Awareness Partnership" (http://www.marketwired.com/press-release/government-canada-launches-cyber-security-awareness-month-with-new-public-awareness-1706660.htm). *Market Wired*. Government of Canada. 27 September 2012. Archived (https://web.archive.org/web/20141103225408/http://www.marketwired.com/press-release/government-canada-launches-cyber-security-awareness-month-with-new-public-awareness-1706660.htm) from the original on 3 November 2014. Retrieved 3 November 2014.
238. "Cyber Security Cooperation Program" (http://www.publicsafety.gc.ca/cnt/ntnl-scrt/cbr-scrt/cprtn-prgrm/index-eng.aspx). *Public Safety Canada*. Archived (https://web.archive.org/web/20141102184754/http://www.publicsafety.gc.ca/cnt/ntnl-scrt/cbr-scrt/cprtn-prgrm/index-eng.aspx) from the original on 2 November 2014. Retrieved 1 November 2014.
239. "Cyber Security Cooperation Program" (http://www.publicsafety.gc.ca/cnt/ntnl-scrt/cbr-scrt/cprtn-prgrm/index-eng.aspx). *Public Safety Canada*. 16 December 2015. Archived (https://web.archive.org/web/20141102184754/http://www.publicsafety.gc.ca/cnt/ntnl-scrt/cbr-scrt/cprtn-prgrm/index-eng.aspx) from the original on 2 November 2014.

240. "GetCyberSafe" (http://www.getcybersafe.gc.ca/index-eng.aspx). *Get Cyber Safe*. Government of Canada. Archived (https://web.archive.org/web/20141111210737/http://www.getcybersafe.gc.ca/index-eng.aspx) from the original on 11 November 2014. Retrieved 3 November 2014.
241. "Australian federal government announces cybersecurity support for SMBs","2023-2030 Australian Cyber Security Strategy" (https://www.homeaffairs.gov.au/about-us/our-portfolios/cyber-security/strategy/2023-2030-australian-cyber-security-strategy). Retrieved 22 November 2023.
242. "Need for proper structure of PPPs to address specific cyberspace risks" (http://www.orfonline.org/cyfy-event/need-for-proper-structure-of-ppps-to-address-specific-cyberspace-risks/). Archived (https://web.archive.org/web/20171113165123/http://www.orfonline.org/cyfy-event/need-for-proper-structure-of-ppps-to-address-specific-cyberspace-risks/) from the original on 13 November 2017.
243. "National Cyber Safety and Security Standards(NCSSS)-Home" (https://web.archive.org/web/20180219150958/https://www.ncdrc.res.in/). *www.ncdrc.res.in*. Archived from the original (https://www.ncdrc.res.in/) on 19 February 2018. Retrieved 19 February 2018.
244. "South Korea seeks global support in cyber attack probe". *BBC Monitoring Asia Pacific*. 7 March 2011.
245. Jun, Kwanwoo (23 September 2013). "Seoul Puts a Price on Cyberdefense" (https://blogs.wsj.com/korearealtime/2013/09/23/seoul-puts-a-price-on-cyberdefense/). *The Wall Street Journal*. Dow Jones & Company, Inc. Archived (https://web.archive.org/web/20130925102342/http://blogs.wsj.com/korearealtime/2013/09/23/seoul-puts-a-price-on-cyberdefense/) from the original on 25 September 2013. Retrieved 24 September 2013.
246. White, House (March 2023). "National security strategy" (https://bidenwhitehouse.archives.gov/wp-content/uploads/2023/03/National-Cybersecurity-Strategy-2023.pdf) (PDF). No. March 2032. white house. US gov.
247. Adil, Sajid (16 October 2023). "Do You Know About Biggest Cybersecurity Threats In 2023?" (https://cybernexguard.com/do-you-know-about-biggest-cybersecurity-threats-in-2023/). *Cybernexguard*. Adil Sajid. Retrieved 18 December 2023.
248. Adil, Sajid (September 2018). "National Cyber Strategy of the United States of America" (https://digital.library.unt.edu/ark:/67531/metadc1259394/). *University Libraries UNT Digital Library*. Retrieved 18 December 2023.
249. Adil, Sajid (September 2018). "Do You Know About Biggest Cybersecurity Threats In 2023?" (https://digital.library.unt.edu/ark:/67531/metadc1259394/#collections). *University Libraries UNT Digital Library*. Retrieved 18 December 2023.
250. International Cybercrime Reporting and Cooperation Act (https://www.congress.gov/bill/111th-congress/house-bill/4962) at Congress.gov
251. "111th Congress, 2nd Session" (https://web.archive.org/web/20120120040012/http://hsgac.senate.gov/public/index.cfm?FuseAction=Files.View&FileStore_id=4ee63497-ca5b-4a4b-9bba-04b7f4cb0123). Archived from the original on 20 January 2012.
252. "Biden Adviser On Cyber Threats And The New Executive Order To Combat Them" (https://www.npr.org/2021/05/13/996617560/biden-advisor-on-cyber-threats-and-the-new-executive-order-to-combat-them). *NPR*.
253. Executive Order on Improving the Nation's Cybersecurity (https://bidenwhitehouse.archives.gov/briefing-room/presidential-actions/2021/05/12/executive-order-on-improving-the-nations-cybersecurity/) (full text)
254. "National Cyber Security Division" (https://web.archive.org/web/20080611210347/https://www.dhs.gov/xabout/structure/editorial_0839.shtm). U.S. Department of Homeland Security. Archived from the original (https://www.dhs.gov/xabout/structure/editorial_0839.shtm) on 11 June 2008. Retrieved 14 June 2008.

255. "FAQ: Cyber Security R&D Center" (http://www.cyber.st.dhs.gov/faq.html). U.S. Department of Homeland Security S&T Directorate. Archived (https://web.archive.org/web/20081006042850/http://www.cyber.st.dhs.gov/faq.html) from the original on 6 October 2008. Retrieved 14 June 2008.
256. AFP-JiJi, "U.S. boots up cybersecurity center", 31 October 2009.
257. "Federal Bureau of Investigation – Priorities" (https://www.fbi.gov/about-us/quick-facts). Federal Bureau of Investigation. Archived (https://web.archive.org/web/20160711053557/https://www.fbi.gov/about-us/quick-facts) from the original on 11 July 2016.
258. "Internet Crime Complaint Center (IC3) – Home" (https://www.ic3.gov/default.aspx). Archived (https://web.archive.org/web/20111120021742/http://www.ic3.gov/default.aspx) from the original on 20 November 2011.
259. "Infragard, Official Site" (http://www.infragard.net/). *Infragard*. Archived (https://web.archive.org/web/20100909051004/http://www.infragard.net/) from the original on 9 September 2010. Retrieved 10 September 2010.
260. "Robert S. Mueller, III – InfraGard Interview at the 2005 InfraGard Conference" (https://web.archive.org/web/20110617004540/http://www.infragard.net/media/files/dir_med.mov). *Infragard (Official Site) – "Media Room"*. Archived from the original (http://www.infragard.net/media/files/dir_med.mov) on 17 June 2011. Retrieved 9 December 2009.
261. "CCIPS" (https://www.cybercrime.gov/). 25 March 2015. Archived (https://web.archive.org/web/20060823173821/http://www.cybercrime.gov/) from the original on 23 August 2006.
262. "A Framework for a Vulnerability Disclosure Program for Online Systems" (https://www.justice.gov/criminal-ccips/page/file/983996/download). Cybersecurity Unit, Computer Crime & Intellectual Property Section Criminal Division U.S. Department of Justice. July 2017. Retrieved 9 July 2018.
263. "Mission and Vision" (https://www.cybercom.mil/About/Mission-and-Vision/). *www.cybercom.mil*. Retrieved 20 June 2020.
264. William J. Lynn, III (12 November 2009). *Remarks at the Defense Information Technology Acquisition Summit* (https://www.defense.gov/speeches/speech.aspx?speechid=1399) (Speech). Washington D.C. Archived (https://web.archive.org/web/20100415113237/http://www.defense.gov/speeches/speech.aspx?speechid=1399) from the original on 15 April 2010. Retrieved 10 July 2010.
265. Shachtman, Noah (23 September 2010). "Military's Cyber Commander Swears: "No Role" in Civilian Networks" (https://web.archive.org/web/20101106032102/http://www.brookings.edu/opinions/2010/0923_military_internet_shachtman.aspx). *brookings.edu*. Archived from the original (http://www.brookings.edu/opinions/2010/0923_military_internet_shachtman.aspx) on 6 November 2010.
266. "FCC Cybersecurity" (https://web.archive.org/web/20100527095750/http://www.fcc.gov/pshs/emergency-information/cybersecurity.html). FCC. Archived from the original (https://www.fcc.gov/pshs/emergency-information/cybersecurity.html) on 27 May 2010. Retrieved 3 December 2014.
267. "Cybersecurity for Medical Devices and Hospital Networks: FDA Safety Communication" (https://www.fda.gov/MedicalDevices/Safety/AlertsandNotices/ucm356423.htm). *Food and Drug Administration*. Archived (https://web.archive.org/web/20160528153847/https://www.fda.gov/medicaldevices/safety/alertsandnotices/ucm356423.htm) from the original on 28 May 2016. Retrieved 23 May 2016.
268. "Automotive Cybersecurity – National Highway Traffic Safety Administration (NHTSA)" (https://web.archive.org/web/20160525195552/http://www.nhtsa.gov/Research/Crash+Avoidance/Automotive+Cybersecurity). Archived from the original (https://www.nhtsa.gov/Research/Crash+Avoidance/Automotive+Cybersecurity) on 25 May 2016. Retrieved 23 May 2016.

269. Air Traffic Control: FAA Needs a More Comprehensive Approach to Address Cybersecurity As Agency Transitions to NextGen (http://www.gao.gov/products/GAO-15-370) (Report). U. S. Government Accountability Office. 14 April 2015. Archived (https://web.archive.org/web/20160613150636/http://www.gao.gov/products/GAO-15-370) from the original on 13 June 2016. Retrieved 23 May 2016.
270. Sternstein, Aliya (4 March 2016). "FAA Working on New Guidelines for Hack-Proof Planes" (http://www.nextgov.com/cybersecurity/2016/03/faa-has-started-shaping-cybersecurity-regulations/126449/). *Nextgov*. Archived (https://web.archive.org/web/20160519181332/http://www.nextgov.com/cybersecurity/2016/03/faa-has-started-shaping-cybersecurity-regulations/126449/) from the original on 19 May 2016. Retrieved 23 May 2016.
271. Elias, Bart (18 June 2015). "Protecting Civil Aviation from Cyberattacks" (https://www.fas.org/sgp/crs/homesec/IN10296.pdf) (PDF). Archived (https://web.archive.org/web/20161017100306/https://www.fas.org/sgp/crs/homesec/IN10296.pdf) (PDF) from the original on 17 October 2016. Retrieved 4 November 2016.
272. Anderson, David; Reimers, Karl (2019). *CYBER SECURITY EMPLOYMENT POLICY AND WORKPLACE DEMAND IN THE U.S. GOVERNMENT*. EDULEARN19 Proceedings. Vol. 1. IATED. pp. 7858–7866. doi:10.21125/edulearn.2019.1914 (https://doi.org/10.21125%2Fedulearn.2019.1914). ISBN 978-84-09-12031-4. ISSN 2340-1117 (https://search.worldcat.org/issn/2340-1117).
273. Verton, Dan (28 January 2004). "DHS launches national cyber alert system" (http://www.computerworld.com/securitytopics/security/story/0,10801,89488,00.html). *Computerworld*. IDG. Archived (https://web.archive.org/web/20050831162039/http://www.computerworld.com/securitytopics/security/story/0,10801,89488,00.html) from the original on 31 August 2005. Retrieved 15 June 2008.
274. Details can be found in 10 CFR 73.54, Protection of digital computer and communication systems and networks (https://www.ecfr.gov/current/title-10/section-73.54).
275. *Cyber Security Plan for Nuclear Power Reactors (https://www.nrc.gov/docs/ML1011/ML101180437.pdf)* - Nuclear Energy Institute
276. Refer to **NEI 08-09 (https://www.nrc.gov/docs/ML1011/ML101180437.pdf)** for more details.
277. Clayton, Mark (7 March 2011). "The new cyber arms race" (http://www.csmonitor.com/USA/Military/2011/0307/The-new-cyber-arms-race). *The Christian Science Monitor*. Archived (https://web.archive.org/web/20150416090310/http://www.csmonitor.com/USA/Military/2011/0307/The-new-cyber-arms-race) from the original on 16 April 2015. Retrieved 16 April 2015.
278. Nakashima, Ellen (13 September 2016). "Obama to be urged to split cyberwar command from NSA" (https://archive.today/20161012083815/https://www.washingtonpost.com/world/national-security/obama-to-be-urged-to-split-cyberwar-command-from-the-nsa/2016/09/12/0ad09a22-788f-11e6-ac8e-cf8e0dd91dc7_story.html). *The Washington Post*. Archived from the original (https://www.washingtonpost.com/world/national-security/obama-to-be-urged-to-split-cyberwar-command-from-the-nsa/2016/09/12/0ad09a22-788f-11e6-ac8e-cf8e0dd91dc7_story.html) on 12 October 2016. Retrieved 15 June 2017.
279. Overland, Indra (1 March 2019). "The geopolitics of renewable energy: Debunking four emerging myths" (https://doi.org/10.1016%2Fj.erss.2018.10.018). *Energy Research & Social Science*. **49**: 36–40. Bibcode:2019ERSS...49...36O (https://ui.adsabs.harvard.edu/abs/2019ERSS...49...36O). doi:10.1016/j.erss.2018.10.018 (https://doi.org/10.1016%2Fj.erss.2018.10.018). hdl:11250/2579292 (https://hdl.handle.net/11250%2F2579292). ISSN 2214-6296 (https://search.worldcat.org/issn/2214-6296).
280. Maness, Ryan C.; Valeriano, Brandon (11 June 2018). "How We Stopped Worrying about Cyber Doom and Started Collecting Data" (https://doi.org/10.17645%2Fpag.v6i2.1368). *Politics and Governance*. **6** (2): 49–60. doi:10.17645/pag.v6i2.1368 (https://doi.org/10.17645%2Fpag.v6i2.1368). hdl:10945/60589 (https://hdl.handle.net/10945%2F60589). ISSN 2183-2463 (https://search.worldcat.org/issn/2183-2463).

281. Maness, Ryan C.; Valeriano, Brandon (25 March 2015). "The Impact of Cyber Conflict on International Interactions". *Armed Forces & Society*. **42** (2): 301–323. doi:10.1177/0095327x15572997 (https://doi.org/10.1177%2F0095327x15572997). ISSN 0095-327X (https://search.worldcat.org/issn/0095-327X). S2CID 146145942 (https://api.semanticscholar.org/CorpusID:146145942).
282. Bullard, Brittany (2016). *Style and Statistics: The Art of Retail Analytics* (https://onlinelibrary.wiley.com/doi/book/10.1002/9781119271260). Wiley. doi:10.1002/9781119271260.ch8 (https://doi.org/10.1002%2F9781119271260.ch8). ISBN 978-1-119-27031-7.
283. Oltsik, Jon (18 March 2016). "Cybersecurity Skills Shortage Impact on Cloud Computing" (https://web.archive.org/web/20160323042705/http://www.networkworld.com/article/3045801/security/cybersecurity-skills-shortage-impact-on-cloud-computing.html). *Network World*. Archived from the original (http://www.networkworld.com/article/3045801/security/cybersecurity-skills-shortage-impact-on-cloud-computing.html) on 23 March 2016. Retrieved 23 March 2016.
284. Robinson, Terry (30 May 2018). "Why is a Degree in Cyber Security one of the Best?" (https://web.archive.org/web/20211010052542/https://www.degreequery.com/why-is-a-degree-in-cyber-security-one-of-the-best/). *DegreeQuery.com*. Archived from the original (https://www.degreequery.com/why-is-a-degree-in-cyber-security-one-of-the-best/) on 10 October 2021. Retrieved 10 October 2021.
285. de Silva, Richard (11 October 2011). "Government vs. Commerce: The Cyber Security Industry and You (Part One)" (https://www.defenceiq.com/defence-technology/articles/the-cyber-security-industry-and-you/). Defence IQ. Archived (https://web.archive.org/web/20140424200253/http://www.defenceiq.com/defence-technology/articles/the-cyber-security-industry-and-you/) from the original on 24 April 2014. Retrieved 24 April 2014.
286. "Department of Computer Science" (https://web.archive.org/web/20130603085633/http://www.cs.gwu.edu/academics/graduate_programs/master/cybersecurity/cybersecurity-jobs). Archived from the original (http://www.cs.gwu.edu/academics/graduate_programs/master/cybersecurity/cybersecurity-jobs) on 3 June 2013. Retrieved 30 April 2013.
287. "About Cyber Security architect" (https://www.cisa.gov/security-architect). *cisa.gov*. 1 August 2021. Retrieved 1 January 2022.
288. "How to become a Chief Information Security Officer (CISO)?" (https://cybersecuritycareer.org/chief-information-security-officer-ciso/). *cybersecuritycareer.org*. 1 August 2021. Retrieved 4 January 2022.
289. "Data Protection Officers" (https://ico.org.uk/for-organisations/guide-to-data-protection/guide-to-the-general-data-protection-regulation-gdpr/accountability-and-governance/data-protection-officers/). *ico.org.uk*. January 2021.
290. "Student Cybersecurity Resources" (https://niccs.cisa.gov/formal-education/students-launch-your-cyber-career). NICCS (US National Initiative for Cybercareers and Studies). Archived (https://web.archive.org/web/20201105234726/https://niccs.cisa.gov/formal-education/students-launch-your-cyber-career) from the original on 5 November 2020.
291. "Current Job Opportunities at DHS" (https://www.dhs.gov/join-dhs-cybersecurity). U.S. Department of Homeland Security. Archived (https://web.archive.org/web/20130502135412/http://www.dhs.gov/join-dhs-cybersecurity) from the original on 2 May 2013. Retrieved 5 May 2013.
292. "Cybersecurity Training & Exercises" (https://www.dhs.gov/cybersecurity-training-exercises). U.S. Department of Homeland Security. 12 May 2010. Archived (https://web.archive.org/web/20150107111146/http://www.dhs.gov/cybersecurity-training-exercises) from the original on 7 January 2015. Retrieved 9 January 2015.
293. "Cyber Security Awareness Free Training and Webcasts" (https://msisac.cisecurity.org/resources/videos/free-training.cfm). MS-ISAC (Multi-State Information Sharing & Analysis Center). Archived (https://web.archive.org/web/20150106064140/http://msisac.cisecurity.org/resources/videos/free-training.cfm) from the original on 6 January 2015. Retrieved 9 January 2015.

294. "DoD Approved 8570 Baseline Certifications" (https://web.archive.org/web/20161021073353/http://iase.disa.mil/iawip/Pages/iabaseline.aspx). *iase.disa.mil*. Archived from the original (http://iase.disa.mil/iawip/Pages/iabaseline.aspx) on 21 October 2016. Retrieved 19 June 2017.
295. "The UK Cyber Security Strategy: Report on Progress and Forward Plans December 2014" (https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/386093/The_UK_Cyber_Security_Strategy_Report_on_Progress_and_Forward_Plans_-_De___.pdf) (PDF). United Kingdom Cabinet Office. Archived (https://web.archive.org/web/20180418230804/https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/386093/The_UK_Cyber_Security_Strategy_Report_on_Progress_and_Forward_Plans_-_De___.pdf) (PDF) from the original on 18 April 2018. Retrieved 20 August 2021.
296. "Cyber skills for a vibrant and secure UK" (https://www.gov.uk/government/news/cyber-skills-for-a-vibrant-and-secure-uk). *GOV.UK*.
297. "Singapore Operational Technology (OT) Cybersecurity Competency Framework" (https://web.archive.org/web/20211016185633/https://www.csa.gov.sg/News/Press-Releases/singapore-operational-technology-cybersecurity-competency-framework). *Cyber Security Agency* (Press release). 8 October 2021. Archived from the original (https://www.csa.gov.sg/News/Press-Releases/singapore-operational-technology-cybersecurity-competency-framework) on 16 October 2021. Retrieved 23 October 2021.
298. "Confidentiality" (http://medical-dictionary.thefreedictionary.com/confidentiality). Retrieved 31 October 2011.
299. "Data Integrity" (http://www.businessdictionary.com/definition/data-integrity.html). Archived (https://web.archive.org/web/20111106055944/http://www.businessdictionary.com/definition/data-integrity.html) from the original on 6 November 2011. Retrieved 31 October 2011.
300. "Endpoint Security" (http://www.webopedia.com/TERM/E/endpoint_security.html). 10 November 2010. Archived (https://web.archive.org/web/20140316021605/http://www.webopedia.com/TERM/E/endpoint_security.html) from the original on 16 March 2014. Retrieved 15 March 2014.
301. "A Brief History of the Cybersecurity Profession" (https://www.isaca.org/resources/news-and-trends/industry-news/2022/a-brief-history-of-the-cybersecurity-profession). *ISACA*. Retrieved 13 October 2023.
302. "One step ahead in computing security" (https://www.rit.edu/news/one-step-ahead-computing-security). *RIT*. Retrieved 13 October 2023.
303. Misa, Thomas J. (2016). "Computer Security Discourse at RAND, SDC, and NSA (1958-1970)" (https://dl.acm.org/doi/10.1109/MAHC.2016.48). *IEEE Annals of the History of Computing*. **38** (4): 12–25. doi:10.1109/MAHC.2016.48 (https://doi.org/10.1109%2FMAHC.2016.48). S2CID 17609542 (https://api.semanticscholar.org/CorpusID:17609542).
304. Neumann, A. J.; Statland, N.; Webb, R. D. (1977). "Post-processing audit tools and techniques" (https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nbsspecialpublication500-19.pdf) (PDF). *nist.gov*. US Department of Commerce, National Bureau of Standards. pp. 11–3–11–4. Archived (https://web.archive.org/web/20161010044638/http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nbsspecialpublication500-19.pdf) (PDF) from the original on 10 October 2016. Retrieved 19 June 2020.
305. Irwin, Luke (5 April 2018). "How NIST can protect the CIA triad, including the often overlooked 'I' – integrity" (https://blog.itgovernanceusa.com/blog/how-nist-can-protect-the-cia-triad-including-the-often-overlooked-i-integrity). *www.itgovernanceusa.com*. Retrieved 16 January 2021.
306. Perrin, Chad (30 June 2008). "The CIA Triad" (https://www.techrepublic.com/blog/security/the-cia-triad/488). *techrepublic.com*. Retrieved 31 May 2012.

307. Stoneburner, G.; Hayden, C.; Feringa, A. (2004). Engineering Principles for Information Technology Security (http://csrc.nist.gov/publications/nistpubs/800-27A/SP800-27-RevA.pdf) (PDF) (Report). csrc.nist.gov. doi:10.6028/NIST.SP.800-27rA (https://doi.org/10.6028%2FNIST.SP.800-27rA). Archived (https://web.archive.org/web/20041012074937/http://csrc.nist.gov/publications/nistpubs/800-27A/SP800-27-RevA.pdf) (PDF) from the original on 12 October 2004. *Note: this document has been superseded by later versions.*
308. Yost, Jeffrey R. (April 2015). "The Origin and Early History of the Computer Security Software Products Industry" (https://ieeexplore.ieee.org/document/7116464). *IEEE Annals of the History of Computing*. **37** (2): 46–58. doi:10.1109/MAHC.2015.21 (https://doi.org/10.1109%2FMAHC.2015.21). ISSN 1934-1547 (https://search.worldcat.org/issn/1934-1547). S2CID 18929482 (https://api.semanticscholar.org/CorpusID:18929482).
309. "A Brief History of Computer Viruses & What the Future Holds" (https://www.kaspersky.com/resource-center/threats/a-brief-history-of-computer-viruses-and-what-the-future-holds). *www.kaspersky.com*. 19 April 2023. Retrieved 12 June 2024.
310. "First incident of cyber-espionage" (https://www.guinnessworldrecords.com/world-records/612868-first-incident-of-cyber-espionage). *Guinness World Records*. Retrieved 23 January 2024.
311. FBI News (2 November 2018). "The Morris Worm - 30 Years Since First Major Attack on the Internet" (https://www.fbi.gov/news/stories/morris-worm-30-years-since-first-major-attack-on-internet-110218). *fbi.gov*. Retrieved 23 January 2024.
312. Boncella, Robert J (April 2004). Bidgoli, Hossein (ed.). *The Internet Encyclopedia, Volume 2* (2nd ed.). Wiley. p. 262. ISBN 978-0-471-68996-6.
313. "1993: Mosaic Launches and the Web is Set Free" (https://webdevelopmenthistory.com/1993-mosaic-launches-and-the-web-is-set-free/). *Web Development History*. 8 December 2021.
314. "Web Design Museum - Netscape Navigator 2.0" (https://www.webdesignmuseum.org/web-design-history/netscape-navigator-2-0-1995). 10 March 2023. Retrieved 4 December 2023.
315. Nakashima, Ellen (26 January 2008). "Bush Order Expands Network Monitoring: Intelligence Agencies to Track Intrusions" (https://www.washingtonpost.com/wp-dyn/content/article/2008/01/25/AR2008012503261_pf.html). *The Washington Post*. Retrieved 8 February 2021.
316. Perlroth, Nicole (7 February 2021). "How the U.S. Lost to Hackers" (https://ghostarchive.org/archive/20211228/https://www.nytimes.com/2021/02/06/technology/cyber-hackers-usa.html). *The New York Times*. Archived from the original (https://www.nytimes.com/2021/02/06/technology/cyber-hackers-usa.html) on 28 December 2021. Retrieved 9 February 2021.
317. Perlroth, Nicole; Sanger, David; Shane, Scott (6 May 2019). "How Chinese Spies Got the N.S.A.'s Hacking Tools, and Used Them for Attacks" (https://www.nytimes.com/2019/05/06/us/politics/china-hacking-cyber.html). *The New York Times*. Retrieved 18 October 2024.

Further reading

- Branch, Jordan (24 September 2020). "What's in a Name? Metaphors and Cybersecurity". *International Organization*. **75** (1). Cambridge University Press (CUP): 39–70. doi:10.1017/s002081832000051x (https://doi.org/10.1017%2Fs002081832000051x). ISSN 0020-8183 (https://search.worldcat.org/issn/0020-8183). S2CID 224886794 (https://api.semanticscholar.org/CorpusID:224886794).
- Costigan, Sean; Hennessy, Michael (2016). *Cybersecurity: A Generic Reference Curriculum* (https://www.nato.int/nato_static_fl2014/assets/pdf/pdf_2016_10/20161025_1610-cybersecurity-curriculum.pdf) (PDF). NATO. ISBN 978-92-845-0196-0. Archived (https://web.archive.org/web/20170310031437/http://www.nato.int/nato_static_fl2014/assets/pdf/pdf_2016_10/20161025_1610-cybersecurity-curriculum.pdf) (PDF) from the original on 10 March 2017.

- Fuller, Christopher J (11 June 2018). "The Roots of the United States' Cyber (In)Security" (https://eprints.soton.ac.uk/407741/3/Fuller_Roots_of_Cyber_Insecurity_clean_images.doc) (DOC). *Diplomatic History*. **43** (1). Oxford University Press (OUP): 157–185. doi:10.1093/dh/dhy038 (https://doi.org/10.1093%2Fdh%2Fdhy038). ISSN 0145-2096 (https://search.worldcat.org/issn/0145-2096).
- Bob, Yonah Jeremy (21 August 2021). "Ex-IDF cyber intel. official reveals secrets behind cyber offense" (https://www.jpost.com/israel-news/ex-idf-cyber-intel-official-how-to-carry-out-a-cyber-offense-attack-677173). *The Jerusalem Post*.
- Kim, Peter (2014). *The Hacker Playbook: Practical Guide To Penetration Testing*. Seattle: CreateSpace Independent Publishing Platform. ISBN 978-1-4949-3263-3.
- Lee, Newton (2015). *Counterterrorism and Cybersecurity: Total Information Awareness* (2nd ed.). Springer. ISBN 978-3-319-17243-9.
- Montagnani, Maria Lillà; Cavallo, Mirta Antonella (2018). "Cybersecurity and Liability in a Big Data World" (https://revistas.ucp.pt/index.php/mclawreview/article/view/325). *Market and Competition Law Review*. **2** (2). Elsevier BV: 71–98. doi:10.2139/ssrn.3220475 (https://doi.org/10.2139%2Fssrn.3220475). ISSN 1556-5068 (https://search.worldcat.org/issn/1556-5068). S2CID 216704215 (https://api.semanticscholar.org/CorpusID:216704215). SSRN 3220475 (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3220475).
- Shariati, Marzieh; Bahmani, Faezeh; Shams, Fereidoon (2011). "Enterprise information security, a review of architectures and frameworks from interoperability perspective" (https://doi.org/10.1016%2Fj.procs.2010.12.089). *Procedia Computer Science*. **3**. Elsevier BV: 537–543. doi:10.1016/j.procs.2010.12.089 (https://doi.org/10.1016%2Fj.procs.2010.12.089). ISSN 1877-0509 (https://search.worldcat.org/issn/1877-0509).
- Singer, P. W.; Friedman, Allan (2014). *Cybersecurity and Cyberwar: What Everyone Needs to Know*. Oxford University Press. ISBN 978-0-19-991811-9.
- Wu, Chwan-Hwa (John); Irwin, J. David (2013). *Introduction to Computer Networks and Cybersecurity*. Boca Raton: CRC Press. ISBN 978-1-4665-7213-3.
- Cybersecurity Best Practices | Cybersecurity and Infrastructure Security Agency CISA. (n.d.). Retrieved April 24, 2024, from https://www.cisa.gov/topics/cybersecurity-best-practices
- Sztyber-Betley, A., Syfert, M., Kościelny, J. M., & Górecka, Z. (2023). Controller Cyber-Attack Detection and Isolation †: Sensors (14248220). Sensors (14248220), 23(5), 2778. doi:10.3390/s23052778 (https://doi.org/10.3390%2Fs23052778)

Tivoization

Tivoization (/ˌtiːvoʊɪˈzeɪʃən, -aɪ-/) is the practice of designing hardware that incorporates software under the terms of a copyleft software license like the GNU General Public License (GNU GPL), but uses hardware restrictions or digital rights management (DRM) to prevent users from running modified versions of the software on that hardware. Richard Stallman of the Free Software Foundation (FSF) coined the term in reference to TiVo's use of GNU GPL licensed software on the TiVo brand digital video recorders (DVR), which actively block modified software by design.[1][2] Stallman believes this practice denies users some of the freedom that the GNU GPL was designed to protect.[3] The FSF refers to tivoized hardware as "proprietary tyrants".[4]

The Free Software Foundation explicitly forbade tivoization in version 3 of the GNU General Public License. However, although version 3 has been adopted by many software projects, the authors of the Linux kernel have notably declined to move from version 2 to version 3.

Background

TiVo's software incorporates the Linux kernel and GNU software, both of which are licensed under version 2 of the GNU General Public License (GPLv2). GPLv2 requires distributors to make the corresponding source code available to each person who receives the software. One goal of this requirement is to allow users of GPL-covered software to modify the software to better suit their purposes.[5]

Richard Stallman of the Free Software Foundation asserts that TiVo circumvented the GPL's goal by making their products run programs only if the program's digital signature matches those authorized by the manufacturer of the TiVo.[6] While TiVo has complied with the GPL v2 requirement to release the source code for others to modify, any modified software will not run on TiVo's hardware.

Bradley Kuhn of the Software Freedom Conservancy disputes Stallman's narrative. Kuhn asserts that TiVo did not strictly forbid software replacement, but TiVo's proprietary software was intentionally designed to not function if any open-source components were replaced, which consequently required the user to find fully open-source alternatives to the proprietary software. In Kuhn's view, TiVo did not tivoize, the GPLv2 was already sufficient to prevent tivoization, and the intent of the GPLv3 was to add an additional, unnecessary requirement that proprietary software continue to function.[7]

GNU GPLv3

In 2006, the Free Software Foundation (FSF) decided to combat TiVo's technical system of blocking users from running modified software. The FSF subsequently developed a third version of the GNU General Public License (GPLv3) which was designed to include language which prohibited this activity.[8]

According to Eben Moglen, "the license should prohibit technical means of evasion of its rules, with the same clarity that it prohibits legal evasion of its rules."[9]

The second draft of the GPLv3 attempted to clarify the rules regarding tivoization.[10] However, some Linux kernel developers were still concerned that this draft might still prohibit beneficial uses of digital signatures.[11] Stallman and the Free Software Foundation attempted to respond to some of these concerns by stating that the GPLv3 allows private digital signatures for security purposes while still preventing tivoization.

In the third and fourth discussion drafts of the GPLv3, released March 28, 2007 and May 31, 2007, respectively, the anti-tivoization clause was limited so as not to apply when the software is distributed to a business.[12] Thus, medical devices and voting machines would not be covered. The final, official GPLv3 was published on June 29, 2007, with no major changes in respect to tivoization relative to the fourth draft.

Linus Torvalds said he was "pretty pleased" with the new draft's stance on DRM.[13] However, he still does not support relicensing the Linux kernel under GPLv3, stating that:[14]

> Stallman calls it "tivoization", but that's a word he has made up, and a term I find offensive, so I don't choose to use it. It's offensive because Tivo never did anything wrong, and the FSF even acknowledged that. The fact [is] that they do their hardware and have some DRM issues with the content producers and thus want to protect the integrity of that hardware. The kernel license covers the *kernel*. It does not cover boot loaders and hardware, and as far as I'm concerned, people who make their own hardware can design them any which way they want. Whether that means "booting only a specific kernel" or "sharks with lasers", I don't care.

The GPLv3's new license provisions were acknowledged by TiVo in its April 2007 SEC filing: "If the currently proposed version of GPLv3 is widely adopted, we may be unable to incorporate future enhancements to the GNU/Linux operating system into our software, which could adversely affect our business."[15]

Outcome

The Linux kernel, which is included in the operating system of TiVo-branded hardware, is still distributed under the terms of the GPLv2. The kernel has not been changed to use GPLv3[16] because the kernel maintainers have generally perceived the GPLv3 to be overly restrictive,[17][18][19] although some kernel developers, such as Alan Cox,[20] have expressed divergent opinions. In any case, offering the Linux kernel under a different license would likely be infeasible because of its very large number of copyright holders. Unlike most GPL software, the kernel is licensed only under GPLv2 without the wording "or, at your option, any later version", therefore the explicit agreement of all copyright holders would be required to license the kernel as a whole under a new version.[21]

Some other projects widely used in tivoized embedded systems, such as BusyBox, have also declined to move to GPLv3.[22]

See also

- Vendor lock-in
- Defective by Design

References

1. "Frequently Asked Questions about the GNU Licenses" (https://www.gnu.org/licenses/gpl-faq.html#Tivoization). Archived (https://web.archive.org/web/20161229211715/https://www.gnu.org/licenses/gpl-faq.html) from the original on December 29, 2016. Retrieved March 17, 2015. GNU.org Frequently Asked Questions about the GNU Licenses
2. "A Quick Guide to GPLv3" (https://www.gnu.org/licenses/quick-guide-gplv3.html). Archived (https://web.archive.org/web/20161229211618/https://www.gnu.org/licenses/quick-guide-gplv3.html) from the original on December 29, 2016. Retrieved March 17, 2015. A Quick Guide to GPLv3
3. "[Info-gplv3] GPLv3 Update #2" (https://web.archive.org/web/20061026084218/http://gplv3.fsf.org/pipermail/info-gplv3/2006-February/000001.html). *fsf.org*. Archived from the original (http://gplv3.fsf.org/pipermail/info-gplv3/2006-February/000001.html) on October 26, 2006. Retrieved October 2, 2015.
4. "Proprietary Tyrants" (http://www.gnu.org/proprietary/proprietary-tyrants.html). Free Software Foundation. Archived (https://web.archive.org/web/20150924024955/http://www.gnu.org/proprietary/proprietary-tyrants.html) from the original on September 24, 2015. Retrieved January 28, 2023.
5. "The Free Software Definition" (https://www.gnu.org/philosophy/free-sw.html). Archived (https://web.archive.org/web/20230127163705/https://www.gnu.org/philosophy/free-sw.html) from the original on January 27, 2023. Retrieved January 28, 2023. "...The freedom to study how the program works, and adapt it to your needs..."
6. "Using large disks with TiVo" (https://web.archive.org/web/20120206023943/http://www.gratisoft.us/tivo/bigdisk.html). *gratisoft.us*. Archived from the original (http://www.gratisoft.us/tivo/bigdisk.html) on February 6, 2012. Retrieved October 2, 2015.
7. Kuhn, Bradley (July 23, 2021). " "Tivoization" and Your Right to Install Under Copyleft" (https://sfconservancy.org/blog/2021/jul/23/tivoization-and-the-gpl-right-to-install/). *Conservancy Blog*. Retrieved April 6, 2023.
8. "Richard Stallman explains the new GPL provisions to block "tivoisation" " (https://fsfe.org/activities/gplv3/brussels-rms-transcript#tivoisation). Archived (https://web.archive.org/web/20221006035111/https://fsfe.org/activities/gplv3/brussels-rms-transcript#tivoisation) from the original on October 6, 2022. Retrieved January 28, 2023.
9. "Eben Moglen, speaking about GPLv3 in Barcelona" (https://fsfe.org/activities/gplv3/barcelona-moglen-transcript.en.html#drm). Archived (https://web.archive.org/web/20230112031229/https://fsfe.org/activities/gplv3/barcelona-moglen-transcript.en.html#drm) from the original on January 12, 2023. Retrieved January 28, 2023.
10. "Opinion on Digital Restrictions Management" (http://gplv3.fsf.org/drm-dd2.html). *fsf.org*. Archived (https://web.archive.org/web/20060819170543/http://gplv3.fsf.org/drm-dd2.html) from the original on August 19, 2006. Retrieved October 2, 2015.
11. Bottomley, James E.J.; Chehab, Mauro Carvalho; Gleixner, Thomas; Hellwig, Christoph; Jones, Dave; Kroah-Hartman, Greg; Luck, Tony; Morton, Andrew; Myklebust, Trond; Woodhouse, David (September 22, 2006). "GPLv3 Position Statement" (https://groups.google.com/group/linux.kernel/msg/36ad030397f575f5). *google.com*. Archived (https://web.archive.org/web/20211202200652/https://groups.google.com/g/linux.kernel/c/IdRykVVBrEg/m/9XX1lwMDrTYJ) from the original on December 2, 2021. Retrieved October 2, 2015.

12. brett (June 26, 2007). "GPLv3 FAQ, with explanation of Section 6's limits |" (http://gplv3.fsf.org/dd3-faq). Archived (https://web.archive.org/web/20070606082643/http://gplv3.fsf.org/dd3-faq) from the original on June 6, 2007. Retrieved January 28, 2023.
13. "Torvalds 'pretty pleased' about new GPL 3 draft" (https://archive.today/20120713115859/http://news.cnet.com/8301-10784_3-6171300-7.html). CNet news.com. Archived from the original (http://news.com/2061-10795_3-6171300.html) on July 13, 2012.
14. Torvalds, Linus (June 13, 2007). "Re: Dual-Licensing Linux Kernel with GPL V2 and GPL V3" (https://lkml.org/lkml/2007/6/13/289). *Linux Kernel Mailing List*. Archived (https://web.archive.org/web/20070617234347/http://lkml.org/lkml/2007/6/13/289) from the original on June 17, 2007. Retrieved January 28, 2023.
15. "InformationWeek: TiVo Warns Investors New Open Source License Could Hurt Business" (https://www.informationweek.com/software/tivo-warns-investors-new-open-source-license-could-hurt-business). June 1, 2007. Archived (https://web.archive.org/web/20220727084236/https://www.informationweek.com/software/tivo-warns-investors-new-open-source-license-could-hurt-business) from the original on July 27, 2022.
16. Torvalds, Linus. "COPYING" (https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/tree/COPYING?id=b3358a116c25267446a45e4164cb2b8e98276da0). kernel.org. blob ca442d313d86dc67e0a2e5d584b465bd382cbf5c. Archived (https://web.archive.org/web/20230128083313/https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/tree/COPYING?id=b3358a116c25267446a45e4164cb2b8e98276da0) from the original on January 28, 2023. Retrieved August 13, 2013. "*Also note that the only valid version of the GPL as far as the kernel is concerned is _this_ particular version of the license (ie v2, not v2.2 or v3.x or whatever), unless explicitly otherwise stated.*"
17. Bottomley, James E.J.; Chehab, Mauro Carvalho; Gleixner, Thomas; Hellwig, Christoph; Jones, Dave; Kroah-Hartman, Greg; Luck, Tony; Morton, Andrew; Myklebust, Trond; Woodhouse, David (September 15, 2006). "Kernel developers' position on GPLv3 – The Dangers and Problems with GPLv3" (https://lwn.net/Articles/200422/). LWN.net. Archived (https://web.archive.org/web/20060925172439/http://lwn.net/Articles/200422/) from the original on September 25, 2006. Retrieved March 11, 2015. "The current version (Discussion Draft 2) of GPLv3 on first reading fails the necessity test of section 1 on the grounds that there's no substantial and identified problem with GPLv2 that it is trying to solve. However, a deeper reading reveals several other problems with the current FSF draft: 5.1 DRM Clauses [...] 5.2 Additional Restrictions Clause [...] 5.3 Patents Provisions [...]since the FSF is proposing to shift all of its projects to GPLv3 and apply pressure to every other GPL licensed project to move, we foresee the release of GPLv3 portends the Balkanisation of the entire Open Source Universe upon which we rely."
18. *Linus Torvalds says GPL v3 violates everything that GPLv2 stood for* (https://www.youtube.com/watch?v=PaKIZ7gJlRU). *Debconf 2014*. Portland. September 4, 2014. Archived (https://web.archive.org/web/20161119072821/https://www.youtube.com/watch?v=PaKIZ7gJlRU) from the original on November 19, 2016. Retrieved March 11, 2015.
19. "fa.linux.kernel: Linus Torvalds: Flame Linus to a crisp!" (https://groups.google.com/g/fa.linux.kernel/c/L5NRD_ONkIk/m/Vps_c1wLyiQJ). *google.com*. April 24, 2003. Archived (https://web.archive.org/web/20201227002332/https://groups.google.com/g/fa.linux.kernel/c/L5NRD_ONkIk) from the original on December 27, 2020. Retrieved December 27, 2020.
20. "UK Linux guru backs GPL 3" (https://web.archive.org/web/20090428180852/http://news.zdnet.co.uk/software/0,1000000121,39249913,00.htm). *ZDNet*. January 31, 2006. Archived from the original (http://news.zdnet.co.uk/software/0,1000000121,39249913,00.htm) on April 28, 2009. Retrieved October 2, 2015.
21. Mark P. Lindhout (October 16, 2006). "(About GPLv3) Can the Linux Kernel Relicense? — Ciarán's free software notes" (http://blogs.fsfe.org/ciaran/?p=58). *fsfe.org*. Archived (https://web.archive.org/web/20090223105632/http://blogs.fsfe.org/ciaran/?p=58) from the original on February 23, 2009. Retrieved October 2, 2015.

22. "Busy busy busybox" (https://lwn.net/Articles/202106/). *LWN*. Archived (https://web.archive.org/web/20061105234312/http://lwn.net/Articles/202106/) from the original on November 5, 2006. Retrieved January 28, 2023.

Further reading

- Free Software Foundation definition of "Tivoization" (https://www.gnu.org/licenses/gpl-faq.html#Tivoization)
- An Introduction to Tivoization (http://www.linfo.org/tivoization.html) by The Linux Information Project (LINFO)
- Stallman, Torvalds, Moglen share views on DRM and GPLv3 (http://www.mail-archive.com/blinuxforum@lists.freearchive.org/msg00592.html) – by Shashank Sharma August 9, 2006, from Newsforge
- TiVo's GPL-required modifications (http://www.tivo.com/linux/) to the Linux kernel, glibc, and other software
- Richard Stallman's explanations of "tivoisation":
  - Italy, March 2006 (https://fsfe.org/activities/gplv3/torino-rms-transcript.en.html#drm)
  - Brazil, April 2006 (https://fsfe.org/activities/gplv3/fisl-rms-transcript.en.html#tivoisation)
  - Spain, June 2006 (https://fsfe.org/activities/gplv3/barcelona-rms-transcript.en.html#tivoisation)
  - India, August 2006 (https://fsfe.org/activities/gplv3/bangalore-rms-transcript.en.html#tivoisation)
  - Japan, November 2006 (https://fsfe.org/activities/gplv3/tokyo-rms-transcript#tivoisation)
  - Stallman and Bruce Perens debating about GPL and tivoization (on Wikimedia Commons)
- Eben Moglen's explanations:
  - Spain, June 2006 (https://fsfe.org/activities/gplv3/barcelona-moglen-transcript.en.html#drm)
- Ciarán O'Riordan's explanations:
  - Japan, November 2006 (https://fsfe.org/activities/gplv3/tokyo-ciaran-transcript#tivoisation-and-freedom)
  - Tivoisation explained – implementation and harms (http://fsfe.org/en/fellows/ciaran/ciaran_s_free_software_notes/tivoisation_explained_implementation_and_harms) (article)
- " "Using a hypervisor to reconcile GPL and proprietary embedded code" article" (https://archive.today/20130104020016/http://linuxdevices.com/articles/AT3899346728.html). Archived from the original (http://linuxdevices.com/articles/AT3899346728.html) on January 4, 2013. Retrieved August 30, 2007. with a focus on Tivoization

Trusted Computing

Trusted Computing (**TC**) is a technology developed and promoted by the Trusted Computing Group.[1] The term is taken from the field of trusted systems and has a specialized meaning that is distinct from the field of confidential computing.[2] With Trusted Computing, the computer will consistently behave in expected ways, and those behaviors will be enforced by computer hardware and software.[1] Enforcing this behavior is achieved by loading the hardware with a unique encryption key that is inaccessible to the rest of the system and the owner.

TC is controversial as the hardware is not only secured for its owner, but also against its owner, leading opponents of the technology like free software activist Richard Stallman to deride it as "treacherous computing",[3][4] and certain scholarly articles to use scare quotes when referring to the technology.[5][6]

Trusted Computing proponents such as International Data Corporation,[7] the Enterprise Strategy Group[8] and Endpoint Technologies Associates[9] state that the technology will make computers safer, less prone to viruses and malware, and thus more reliable from an end-user perspective. They also state that Trusted Computing will allow computers and servers to offer improved computer security over that which is currently available. Opponents often state that this technology will be used primarily to enforce digital rights management policies (imposed restrictions to the owner) and not to increase computer security.[3][10]:23

Chip manufacturers Intel and AMD, hardware manufacturers such as HP and Dell, and operating system providers such as Microsoft include Trusted Computing in their products if enabled.[11][12] The U.S. Army requires that every new PC it purchases comes with a Trusted Platform Module (TPM).[13][14] As of July 3, 2007, so does virtually the entire United States Department of Defense.[15]

Key concepts

Trusted Computing encompasses six key technology concepts, of which all are required for a fully Trusted system, that is, a system compliant to the TCG specifications:

1. Endorsement key
2. Secure input and output
3. Memory curtaining / protected execution
4. Sealed storage
5. Remote attestation
6. Trusted Third Party (TTP)

Endorsement key

The endorsement key is a 2048-bit RSA public and private key pair that is created randomly on the chip at manufacture time and cannot be changed. The private key never leaves the chip, while the public key is used for attestation and for encryption of sensitive data sent to the chip, as occurs during the

TPM_TakeOwnership command.[16]

This key is used to allow the execution of secure transactions: every Trusted Platform Module (TPM) is required to be able to sign a random number (in order to allow the owner to show that he has a genuine trusted computer), using a particular protocol created by the Trusted Computing Group (the direct anonymous attestation protocol) in order to ensure its compliance of the TCG standard and to prove its identity; this makes it impossible for a software TPM emulator with an untrusted endorsement key (for example, a self-generated one) to start a secure transaction with a trusted entity. The TPM should be designed to make the extraction of this key by hardware analysis hard, but tamper resistance is not a strong requirement.

Memory curtaining

Memory curtaining extends common memory protection techniques to provide full isolation of sensitive areas of memory—for example, locations containing cryptographic keys. Even the operating system does not have full access to curtained memory. The exact implementation details are vendor specific.

Sealed storage

Sealed storage protects private information by binding it to platform configuration information including the software and hardware being used. This means the data can be released only to a particular combination of software and hardware. Sealed storage can be used for DRM enforcing. For example, users who keep a song on their computer that has not been licensed to be listened will not be able to play it. Currently, a user can locate the song, listen to it, and send it to someone else, play it in the software of their choice, or back it up (and in some cases, use circumvention software to decrypt it). Alternatively, the user may use software to modify the operating system's DRM routines to have it leak the song data once, say, a temporary license was acquired. Using sealed storage, the song is securely encrypted using a key bound to the trusted platform module so that only the unmodified and untampered music player on his or her computer can play it. In this DRM architecture, this might also prevent people from listening to the song after buying a new computer, or upgrading parts of their current one, except after explicit permission of the vendor of the song.

Remote attestation

Remote attestation allows changes to the user's computer to be detected by authorized parties. For example, software companies can identify unauthorized changes to software, including users modifying their software to circumvent commercial digital rights restrictions. It works by having the hardware generate a certificate stating what software is currently running. The computer can then present this certificate to a remote party to show that unaltered software is currently executing. Numerous remote attestation schemes have been proposed for various computer architectures, including Intel,[17] RISC-V,[18] and ARM.[19]

Remote attestation is usually combined with public-key encryption so that the information sent can only be read by the programs that requested the attestation, and not by an eavesdropper.

To take the song example again, the user's music player software could send the song to other machines, but only if they could attest that they were running an authorized copy of the music player software. Combined with the other technologies, this provides a more restricted path for the music: encrypted I/O prevents the user from recording it as it is transmitted to the audio subsystem, memory locking prevents it from being dumped to regular disk files as it is being worked on, sealed storage curtails unauthorized access to it when saved to the hard drive, and remote attestation prevents unauthorized software from accessing the song even when it is used on other computers. To preserve the privacy of attestation responders, Direct Anonymous Attestation has been proposed as a solution, which uses a group signature scheme to prevent revealing the identity of individual signers.

Proof of space (PoS) have been proposed to be used for malware detection, by determining whether the L1 cache of a processor is empty (e.g., has enough space to evaluate the PoSpace routine without cache misses) or contains a routine that resisted being evicted.[20][21]

Trusted third party

Known applications

The Microsoft products Windows Vista, Windows 7, Windows 8 and Windows RT make use of a Trusted Platform Module to facilitate BitLocker Drive Encryption.[22] Other known applications with runtime encryption and the use of secure enclaves include the Signal messenger[23] and the e-prescription service ("E-Rezept")[24] by the German government.

Possible applications

Digital rights management

Trusted Computing would allow companies to create a digital rights management (DRM) system which would be very hard to circumvent, though not impossible. An example is downloading a music file. Sealed storage could be used to prevent the user from opening the file with an unauthorized player or computer. Remote attestation could be used to authorize play only by music players that enforce the record company's rules. The music would be played from curtained memory, which would prevent the user from making an unrestricted copy of the file while it is playing, and secure I/O would prevent capturing what is being sent to the sound system. Circumventing such a system would require either manipulation of the computer's hardware, capturing the analogue (and thus degraded) signal using a recording device or a microphone, or breaking the security of the system.

New business models for use of software (services) over Internet may be boosted by the technology. By strengthening the DRM system, one could base a business model on renting programs for a specific time periods or "pay as you go" models. For instance, one could download a music file which could only be played a certain number of times before it becomes unusable, or the music file could be used only within a certain time period.

Preventing cheating in online games

Trusted Computing could be used to combat cheating in online games. Some players modify their game copy in order to gain unfair advantages in the game; remote attestation, secure I/O and memory curtaining could be used to determine that all players connected to a server were running an unmodified copy of the software.[25]

Verification of remote computation for grid computing

Trusted Computing could be used to guarantee participants in a grid computing system are returning the results of the computations they claim to be instead of forging them. This would allow large scale simulations to be run (say a climate simulation) without expensive redundant computations to guarantee malicious hosts are not undermining the results to achieve the conclusion they want.[26]

Criticism

The Electronic Frontier Foundation and the Free Software Foundation criticize that trust in the underlying companies is not deserved and that the technology puts too much power and control into the hands of those who design systems and software. They also state that it may cause consumers to lose anonymity in their online interactions, as well as mandating technologies Trusted Computing opponents say are unnecessary. They suggest Trusted Computing as a possible enabler for future versions of mandatory access control, copy protection, and DRM.

Some security experts, such as Alan Cox[27] and Bruce Schneier,[28] have spoken out against Trusted Computing, believing it will provide computer manufacturers and software authors with increased control to impose restrictions on what users are able to do with their computers. There are concerns that Trusted Computing would have an anti-competitive effect on the IT market.[10]

There is concern amongst critics that it will not always be possible to examine the hardware components on which Trusted Computing relies, the Trusted Platform Module, which is the ultimate hardware system where the core 'root' of trust in the platform has to reside.[10] If not implemented correctly, it presents a security risk to overall platform integrity and protected data. The specifications, as published by the Trusted Computing Group, are open and are available for anyone to review. However, the final implementations by commercial vendors will not necessarily be subjected to the same review process. In addition, the world of cryptography can often move quickly, and that hardware implementations of algorithms might create an inadvertent obsolescence. Trusting networked computers to controlling authorities rather than to individuals may create digital imprimaturs.

Cryptographer Ross Anderson, University of Cambridge, has great concerns that:[10]

> TC can support remote censorship [...] In general, digital objects created using TC systems remain under the control of their creators, rather than under the control of the person who owns the machine on which they happen to be stored [...] So someone who writes a paper that a court decides is defamatory can be compelled to censor it — and the software company that wrote the word processor could be ordered to do the deletion if she refuses. Given such possibilities, we can expect TC to be used to suppress everything from pornography to writings that criticize political leaders.

He goes on to state that:

> [...] software suppliers can make it much harder for you to switch to their competitors' products. At a simple level, Word could encrypt all your documents using keys that only Microsoft products have access to; this would mean that you could only read them using Microsoft products, not with any competing word processor. [...]

> The [...] most important benefit for Microsoft is that TC will dramatically increase the costs of switching away from Microsoft products (such as Office) to rival products (such as OpenOffice). For example, a law firm that wants to change from Office to OpenOffice right now merely has to install the software, train the staff and convert their existing files. In five years' time, once they have received TC-protected documents from perhaps a thousand different clients, they would have to get permission (in the form of signed digital certificates) from each of these clients in order to migrate their files to a new platform. The law firm won't in practice want to do this, so they will be much more tightly locked in, which will enable Microsoft to hike its prices.

Anderson summarizes the case by saying:

> The fundamental issue is that whoever controls the TC infrastructure will acquire a huge amount of power. Having this single point of control is like making everyone use the same bank, or the same accountant, or the same lawyer. There are many ways in which this power could be abused.

Digital rights management

One of the early motivations behind trusted computing was a desire by media and software corporations for stricter DRM technology to prevent users from freely sharing and using potentially copyrighted or private files without explicit permission. An example could be downloading a music file from a band: the band's record company could come up with rules for how the band's music can be used. For example, they might want the user to play the file only three times a day without paying additional money. Also, they could use remote attestation to only send their music to a music player that enforces their rules: sealed storage would prevent the user from opening the file with another player that did not enforce the restrictions. Memory curtaining would prevent the user from making an unrestricted copy of the file while it is playing, and secure output would prevent capturing what is sent to the sound system.

Users unable to modify software

A user who wanted to switch to a competing program might find that it would be impossible for that new program to read old data, as the information would be "locked in" to the old program. It could also make it impossible for the user to read or modify their data except as specifically permitted by the software.

Users unable to exercise legal rights

The law in many countries allows users certain rights over data whose copyright they do not own (including text, images, and other media), often under headings such as fair use or public interest. Depending on jurisdiction, these may cover issues such as whistleblowing, production of evidence in court, quoting or other small-scale usage, backups of owned media, and making a copy of owned material for personal use on other owned devices or systems. The steps implicit in trusted computing have the practical effect of preventing users exercising these legal rights.[3]

Users vulnerable to vendor withdrawal of service

A service that requires external validation or permission - such as a music file or game that requires connection with the vendor to confirm permission to play or use - is vulnerable to that service being withdrawn or no longer updated. A number of incidents have already occurred where users, having purchased music or video media, have found their ability to watch or listen to it suddenly stop due to vendor policy or cessation of service,[29][30][31] or server inaccessibility,[32] at times with no compensation.[33] Alternatively in some cases the vendor refuses to provide services in future which leaves purchased material only usable on the present -and increasingly obsolete- hardware (so long as it lasts) but not on any hardware that may be purchased in future.[29]

Users unable to override

Some opponents of Trusted Computing advocate "owner override": allowing an owner who is confirmed to be physically present to allow the computer to bypass restrictions and use the secure I/O path. Such an override would allow remote attestation to a user's specification, e.g., to create certificates that say Internet Explorer is running, even if a different browser is used. Instead of preventing software change, remote attestation would indicate when the software has been changed without owner's permission.

Trusted Computing Group members have refused to implement owner override.[34] Proponents of trusted computing believe that owner override defeats the trust in other computers since remote attestation can be forged by the owner. Owner override offers the security and enforcement benefits to a machine owner, but does not allow them to trust other computers, because their owners could waive rules or restrictions on their own computers. Under this scenario, once data is sent to someone else's computer, whether it be a diary, a DRM music file, or a joint project, that other person controls what security, if any, their computer will enforce on their copy of those data. This has the potential to undermine the applications of trusted computing to enforce DRM, control cheating in online games and attest to remote computations for grid computing.

Loss of anonymity

Because a Trusted Computing equipped computer is able to uniquely attest to its own identity, it will be possible for vendors and others who possess the ability to use the attestation feature to zero in on the identity of the user of TC-enabled software with a high degree of certainty.

Such a capability is contingent on the reasonable chance that the user at some time provides user-identifying information, whether voluntarily, indirectly, or simply through inference of many seemingly benign pieces of data. (e.g. search records, as shown through simple study of the AOL search records leak[35]). One common way that information can be obtained and linked is when a user registers a computer just after purchase. Another common way is when a user provides identifying information to the website of an affiliate of the vendor.

While proponents of TC point out that online purchases and credit transactions could potentially be more secure as a result of the remote attestation capability, this may cause the computer user to lose expectations of anonymity when using the Internet.

Critics point out that this could have a chilling effect on political free speech, the ability of journalists to use anonymous sources, whistle blowing, political blogging and other areas where the public needs protection from retaliation through anonymity.

The TPM specification offers features and suggested implementations that are meant to address the anonymity requirement. By using a third-party Privacy Certification Authority (PCA), the information that identifies the computer could be held by a trusted third party. Additionally, the use of direct anonymous attestation (DAA), introduced in TPM v1.2, allows a client to perform attestation while not revealing any personally identifiable or machine information.

The kind of data that must be supplied to the TTP in order to get the trusted status is at present not entirely clear, but the TCG itself admits that "attestation is an important TPM function with significant privacy implications".[36] It is, however, clear that both static and dynamic information about the user computer may be supplied (Ekpubkey) to the TTP (v1.1b),[37] it is not clear what data will be supplied to the “verifier” under v1.2. The static information will uniquely identify the endorser of the platform, model, details of the TPM, and that the platform (PC) complies with the TCG specifications . The dynamic information is described as software running on the computer.[37] If a program like Windows is registered in the user's name this in turn will uniquely identify the user. Another dimension of privacy infringing capabilities might also be introduced with this new technology; how often you use your programs might be possible information provided to the TTP. In an exceptional, however practical situation, where a user purchases a pornographic movie on the Internet, the purchaser nowadays, must accept the fact that he has to provide credit card details to the provider, thereby possibly risking being identified. With the new technology a purchaser might also risk someone finding out that he (or she) has watched this pornographic movie 1000 times. This adds a new dimension to the possible privacy infringement. The extent of data that will be supplied to the TTP/Verifiers is at present not exactly known, only when the technology is implemented and used will we be able to assess the exact nature and volume of the data that is transmitted.

TCG specification interoperability problems

Trusted Computing requests that all software and hardware vendors will follow the technical specifications released by the Trusted Computing Group in order to allow interoperability between different trusted software stacks. However, since at least mid-2006, there have been interoperability problems between the TrouSerS trusted software stack (released as open source software by IBM) and Hewlett-Packard's stack.[38] Another problem is that the technical specifications are still changing, so it is unclear which is the standard implementation of the trusted stack.

Shutting out of competing products

People have voiced concerns that trusted computing could be used to keep or discourage users from running software created by companies outside of a small industry group. Microsoft has received a great deal of bad press surrounding their Palladium software architecture, evoking comments such as "Few pieces of vaporware have evoked a higher level of fear and uncertainty than Microsoft's Palladium",

"Palladium is a plot to take over cyberspace", and "Palladium will keep us from running any software not personally approved by Bill Gates".[39] The concerns about trusted computing being used to shut out competition exist within a broader framework of consumers being concerned about using bundling of products to obscure prices of products and to engage in anti-competitive practices.[5] Trusted Computing is seen as harmful or problematic to independent and open source software developers.[40]

Trust

In the widely used public-key cryptography, creation of keys can be done on the local computer and the creator has complete control over who has access to it, and consequentially their own security policies.[41] In some proposed encryption-decryption chips, a private/public key is permanently embedded into the hardware when it is manufactured,[42] and hardware manufacturers would have the opportunity to record the key without leaving evidence of doing so. With this key it would be possible to have access to data encrypted with it, and to authenticate as it.[43] It is trivial for a manufacturer to give a copy of this key to the government or the software manufacturers, as the platform must go through steps so that it works with authenticated software.

Therefore, to trust anything that is authenticated by or encrypted by a TPM or a Trusted computer, an end user has to trust the company that made the chip, the company that designed the chip, the companies allowed to make software for the chip, and the ability and interest of those companies not to compromise the whole process.[44] A security breach breaking that chain of trust happened to a SIM card manufacturer Gemalto, which in 2010 was infiltrated by US and British spies, resulting in compromised security of cellphone calls.[45]

It is also critical that one be able to trust that the hardware manufacturers and software developers properly implement trusted computing standards. Incorrect implementation could be hidden from users, and thus could undermine the integrity of the whole system without users being aware of the flaw.[46]

Hardware and software support

Since 2004, most major manufacturers have shipped systems that have included Trusted Platform Modules, with associated BIOS support.[47] In accordance with the TCG specifications, the user must enable the Trusted Platform Module before it can be used.

The Linux kernel has included trusted computing support since version 2.6.13, and there are several projects to implement trusted computing for Linux. In January 2005, members of Gentoo Linux's "crypto herd" announced their intention of providing support for TC—in particular support for the Trusted Platform Module.[48] There is also a TCG-compliant software stack for Linux named TrouSerS (http://trousers.sourceforge.net/), released under an open source license. There are several open-source projects that facilitate the use of confidential computing technology, including EGo (https://github.com/edgelesssys/ego), EdgelessDB and MarbleRun from Edgeless Systems, as well as Enarx, which originates from security research at Red Hat.

Some limited form of trusted computing can be implemented on current versions of Microsoft Windows with third-party software. Major cloud providers such as Microsoft Azure,[49] AWS[50] and Google Cloud Platform[51] have virtual machines with trusted computing features available. With the Intel Software

Guard Extension (SGX) and AMD Secure Encrypted Virtualization (SEV) processors, there is hardware available for runtime memory encryption and remote attestation features.[52]

The Intel Classmate PC (a competitor to the One Laptop Per Child) includes a Trusted Platform Module.[53]

PrivateCore vCage software can be used to attest x86 servers with TPM chips.

Mobile T6 secure operating system simulates the TPM functionality in mobile devices using the ARM TrustZone technology.[54]

Samsung smartphones come equipped with Samsung Knox that depend on features like Secure Boot, TIMA, MDM, TrustZone and SE Linux.[55]

See also

- Glossary of legal terms in technology
- Next-Generation Secure Computing Base (formerly known as Palladium)
- Trusted Network Connect
- Trusted Platform Module
- Unified Extensible Firmware Interface § Secure Boot
- Web Environment Integrity

References

1. Chris Mitchell (2005). *Trusted Computing* (https://books.google.com/books?id=9iriBw2AuToC). IET. ISBN 978-0-86341-525-8.
2. "What is the Confidential Computing Consortium?" (https://confidentialcomputing.io/). *Confidential Computing Consortium*. Retrieved 20 May 2022.
3. Stallman, Richard. "Can You Trust Your Computer?" (https://www.gnu.org/philosophy/can-you-trust.html). *gnu.org*. Retrieved 12 August 2013.
4. scl-paullauria (2017-01-23). "Trust me, I'm a computer" (https://www.scl.org/3835-trust-me-i-m-a-computer/). *Society for Computers & Law*. Retrieved 2024-04-03.
5. Anderson, Ross (November 15, 2004). Camp, L. Jean; Lewis, Stephen (eds.). *Cryptography and Competition Policy - Issues with 'Trusted Computing', in Economics of Information Security* (https://doi.org/10.1007/1-4020-8090-5_3). Vol. 12. Springer US. pp. 35–52. doi:10.1007/1-4020-8090-5_3 (https://doi.org/10.1007%2F1-4020-8090-5_3) – via Springer Link.
6. "F. Stajano, "Security for whom? The shifting security assumptions of pervasive computing", *Lecture notes in computer science*, vol. 2609, pp. 16-27, 2003" (http://www.cl.cam.ac.uk/~fms27/papers/2003-stajano-shifting.pdf) (PDF).
7. Rau, Shane (February 2006). "The Trusted Computing Platform Emerges as Industry's First Comprehensive Approach to IT Security" (https://www.trustedcomputinggroup.org/news/Industry_Data/IDC_448_Web.pdf) (PDF). *IDC Executive Brief*. International Data Corporation. Retrieved 2007-02-07.

8. Oltsik, Jon (January 2006). "Trusted Enterprise Security: How the Trusted Computing Group (TCG) Will Advance Enterprise Security" (https://www.trustedcomputinggroup.org/news/Industry_Data/ESG_White_Paper.pdf) (PDF). *White Paper*. Enterprise Strategy Group. Retrieved 2007-02-07.
9. Kay, Roger L. (2006). "How to Implement Trusted Computing: A Guide to Tighter Enterprise Security" (https://www.trustedcomputinggroup.org/news/Industry_Data/Implementing_Trusted_Computing_RK.pdf) (PDF). Endpoint Technologies Associates. Retrieved 2007-02-07.
10. Anderson, Ross (August 2003). " 'Trusted Computing' Frequently Asked Questions: TC / TCG / LaGrande / NGSCB / Longhorn / Palladium / TCPA Version 1.1" (http://www.cl.cam.ac.uk/~rja14/tcpa-faq.html). Retrieved 2007-02-07.
11. "Enhancing IT Security with Trusted Computing Group standards" (http://www.dell.com/downloads/global/power/ps4q06-20070160-tcg.pdf) (PDF). *Dell Power Solutions*. November 2006. p. 14. Retrieved 2006-02-07. "TPMs [Trusted Platform Modules] from various semiconductor vendors are included on enterprise desktop and notebook systems from Dell and other vendors"
12. "Trusted Platform Module Services in Windows Vista" (https://web.archive.org/web/20070515072944/http://www.microsoft.com/whdc/system/platform/pcdesign/TPM_secure.mspx). *Windows Hardware Development Central*. Microsoft. 2005-04-25. Archived from the original (http://www.microsoft.com/whdc/system/platform/pcdesign/TPM_secure.mspx) on 2007-05-15. Retrieved 2007-02-07. "Windows Vista provides a set of services for applications that use TPM technologies."
13. Lemos, Robert (2006-07-28). "U.S. Army requires trusted computing" (http://www.securityfocus.com/brief/265). Security Focus. Retrieved 2007-02-07.
14. "Army CIO/G-6 500-day plan" (https://web.archive.org/web/20061018034556/http://www.army.mil/ciog6/news/500Day2006Update.pdf) (PDF). U.S. Army. October 2006. Archived from the original (http://www.army.mil/ciog6/news/500Day2006Update.pdf) (PDF) on October 18, 2006. Retrieved 2007-02-07. "Strategic goal n. 3, 'deliver a joint netcentric information that enables warfighter decision superiority' "
15. encryption of unclassified data (http://iase.disa.mil/policy-guidance/dod-dar-tpm-decree07-03-07.pdf) Archived (https://web.archive.org/web/20070927060332/http://iase.disa.mil/policy-guidance/dod-dar-tpm-decree07-03-07.pdf) 2007-09-27 at the Wayback Machine
16. Safford, David (2003-08-01). "Take Control of TCPA" (http://www.linuxjournal.com/article/6633). *Linux Journal*. Retrieved 2007-02-07.
17. Johnson, Simon (2016). *Intel Software Guard Extensions: EPID Provisioning and Attestation Services* (https://software.intel.com/content/dam/develop/public/us/en/documents/ww10-2016-sgx-provisioning-and-attestation-final.pdf) (PDF). Intel. Retrieved 14 May 2021.
18. Shepherd, Carlton; Markantonakis, Konstantinos; Jaloyan, Georges-Axel (2021). *LIRA-V: Lightweight Remote Attestation for Constrained RISC-V Devices*. IEEE Security and Privacy Workshops. IEEE. arXiv:2102.08804 (https://arxiv.org/abs/2102.08804).
19. Abera, Tigist (2016). *C-FLAT: Control-Flow Attestation for Embedded Systems Software* (https://dl.acm.org/doi/abs/10.1145/2976749.2978358). CCS '16. ACM. pp. 743–754. doi:10.1145/2976749.2978358 (https://doi.org/10.1145%2F2976749.2978358). ISBN 9781450341394. S2CID 14663076 (https://api.semanticscholar.org/CorpusID:14663076). Retrieved 14 May 2021.
20. Jakobsson, Markus; Stewart, Guy (2013). *Mobile Malware: Why the Traditional AV Paradigm is Doomed, and How to Use Physics to Detect Undesirable Routines*. Black Hat USA.
21. Markus Jakobsson Secure Remote Attestation (https://eprint.iacr.org/2018/031.pdf) *Cryptology ePrint Archive.* Retrieved January 8, 2018.
22. Ferguson, Niels (August 2006). "AES-CBC + Elephant: A Disk Encryption Algorithm for Windows Vista" (http://download.microsoft.com/download/0/2/3/0238acaf-d3bf-4a6d-b3d6-0a0be4bbb36e/BitLockerCipher200608.pdf) (PDF). Microsoft TechNet. Retrieved 2007-02-07.

23. "Scaling secure enclave environments with Signal and Azure confidential computing" (https://customers.microsoft.com/en-us/story/1374464612401582154-signal-nonprofit-azure-security). *Microsoft Customers Stories*. Retrieved 2022-02-09.
24. Mutzbauer, Julia (2 February 2021). "Confidential Computing soll Patientendaten schützen" (https://www.healthcare-computing.de/confidential-computing-soll-patientendaten-schuetzen-a-996680/). *www.healthcare-computing.de* (in German). Retrieved 2022-02-09.
25. Bin Xiao (2007). *Autonomic and Trusted Computing: 4th International Conference, ATC 2007, Hong Kong, China, July 11-13, 2007, Proceedings* (https://books.google.com/books?id=cUhpq98Zb8AC&pg=PA124). Springer Science & Business Media. p. 124. ISBN 978-3-540-73546-5.
26. Mao, Wenbo Jin, Hai and Martin, Andrew (2005-06-07). "Innovations for Grid Security From Trusted Computing" (https://web.archive.org/web/20060822043633/http://www.hpl.hp.com/personal/Wenbo_Mao/research/tcgridsec.pdf) (PDF). Archived from the original (http://www.hpl.hp.com/personal/Wenbo_Mao/research/tcgridsec.pdf) (PDF) on 2006-08-22. Retrieved 2007-02-07.
27. Marson, Ingrid (2006-01-27). "Trusted Computing comes under attack" (https://www.zdnet.com/article/trusted-computing-comes-under-attack/). *ZDNet*. Retrieved 2021-09-12.
28. Schneier, Bruce (2002-08-15). "Palladium and the TCPA" (http://www.schneier.com/crypto-gram-0208.html#1). *Crypto-Gram Newsletter*. Retrieved 2007-02-07.
29. Cheng, Jacqui (2008-04-22). "DRM sucks redux: Microsoft to nuke MSN Music DRM keys" (https://arstechnica.com/information-technology/2008/04/drm-sucks-redux-microsoft-to-nuke-msn-music-drm-keys/). *Ars Technica*. Retrieved 2014-05-31.
30. "Yahoo! DRM servers going away" (http://www.fudzilla.com/home/item/3495-yahoo-drm-servers-going-away?tmpl=component&print=1). Fudzilla.com. 2008-07-29. Retrieved 2014-05-31.
31. Fisher, Ken (2007-08-13). "Google selleth then taketh away, proving the need for DRM circumvention" (https://arstechnica.com/tech-policy/2007/08/google-selleth-then-taketh-away-proving-the-need-for-drm-circumvention/). *Ars Technica*. Retrieved 2014-05-31.
32. Fister, Mister (26 March 2010). "Ubisoft Offers Free Goodies as Compensation f - Video Game News, Videos and File Downloads for PC and Console Games at" (http://www.shacknews.com/article/62995/ubisoft-offers-free-goodies-as). Shacknews.com. Retrieved 2014-05-31.
33. Bangeman, Eric (2007-11-07). "Major League Baseball's DRM change strikes out with fans" (https://arstechnica.com/uncategorized/2007/11/major-league-baseballs-drm-change-strikes-out-with-fans/). *Ars Technica*. Retrieved 2014-05-31.
34. Schoen, Seth (2003-12-01). "Give TCPA an Owner Override" (http://www.linuxjournal.com/article/7055). *Linux Journal*. Retrieved 2007-02-07.
35. "A Face Is Exposed for AOL Searcher No. 4417749" (https://www.nytimes.com/2006/08/09/technology/09aol.html?pagewanted=all&_r=0). *The New York Times*. 2006-08-09. Retrieved 2013-05-10.
36. TPM version 1.2 specifications changes, 16.04.04
37. TPM v1.2 specification changes, 2004
38. "1.7 - I've taken ownership of my TPM under another OS..." (http://trousers.sourceforge.net/faq.html#1.7) *TrouSerS FAQ*. Retrieved 2007-02-07.
39. Felten, E.W. (May 2003). "Understanding trusted computing: will its benefits outweigh its drawbacks?". *IEEE Security & Privacy*. **1** (3): 60–62. doi:10.1109/MSECP.2003.1203224 (https://doi.org/10.1109%2FMSECP.2003.1203224).
40. Oppliger, R.; Rytz, R. (2005). "Does Trusted Computing Remedy Computer Security Problems?" (https://ieeexplore.ieee.org/document/1423956). *IEEE Security and Privacy Magazine*. **3** (2): 16–19. doi:10.1109/MSP.2005.40 (https://doi.org/10.1109%2FMSP.2005.40). S2CID 688158 (https://api.semanticscholar.org/CorpusID:688158).

41. "IEEE P1363: Standard Specifications For Public-Key Cryptography", Retrieved March 9, 2009. (http://grouper.ieee.org/groups/1363/) Archived (https://web.archive.org/web/20141201024245/http://grouper.ieee.org/groups/1363/) December 1, 2014, at the Wayback Machine
42. Garfinkel, Tal; Pfaff, Ben; Chow, Jim; Rosenblum, Mendel; Boneh, Dan (October 19, 2003). "Terra: a virtual machine-based platform for trusted computing" (https://doi.org/10.1145/945445.945464). Association for Computing Machinery. pp. 193–206. doi:10.1145/945445.945464 (https://doi.org/10.1145%2F945445.945464). S2CID 156799 (https://api.semanticscholar.org/CorpusID:156799) – via ACM Digital Library.
43. These are the functions of the private key in the RSA algorithm (http://www.di-mgt.com.au/rsa_alg.html)
44. Sullivan, Nick (27 December 2016). "Deploying TLS 1.3: the great, the good and the bad (33c3)" (https://www.youtube.com/watch?time_continue=1533&v=0opakLwtPWk). *media.ccc.de*. YouTube. Retrieved 30 July 2018.
45. "The Great SIM Heist: How Spies Stole the Keys to the Encryption Castle" (https://firstlook.org/theintercept/2015/02/19/great-sim-heist). *firstlook.org*. 2015-02-19. Retrieved 2015-02-27.
46. Seth Schoen, "Trusted Computing: Promise and Risk", *COSPA Knowledge Base: Comparison, selection, & suitability of OSS*, April 11th, 2006. (http://pascal.case.unibz.it/handle/2038/871) Archived (https://web.archive.org/web/20090319043100/http://pascal.case.unibz.it/handle/2038/871) 2009-03-19 at the Wayback Machine
47. Tony McFadden (March 26, 2006). "TPM Matrix" (https://web.archive.org/web/20070426034219/http://www.tonymcfadden.net/tpmvendors_arc.html). Archived from the original (http://www.tonymcfadden.net/tpmvendors_arc.html) on April 26, 2007. Retrieved 2006-05-05.
48. "Trusted Gentoo" (https://lwn.net/Articles/121386/). *Gentoo Weekly Newsletter*. January 31, 2005. Retrieved 2006-05-05.
49. "Azure Confidential Computing – Protect Data-In-Use | Microsoft Azure" (https://azure.microsoft.com/en-us/solutions/confidential-compute/). *azure.microsoft.com*. Retrieved 2022-02-09.
50. "What is AWS Nitro Enclaves? - AWS" (https://docs.aws.amazon.com/enclaves/latest/user/nitro-enclave.html). *docs.aws.amazon.com*. Retrieved 2022-02-09.
51. "Confidential Computing" (https://cloud.google.com/confidential-computing). *Google Cloud*. Retrieved 2022-02-09.
52. "Cryptography and privacy: protecting private data" (https://www.ericsson.com/en/blog/2021/7/cryptography-and-privacy-protecting-private-data). *www.ericsson.com*. 2021-07-08. Retrieved 2022-02-09.
53. Intel (December 6, 2006). "Product Brief: Classmate PC" (http://download.intel.com/intel/worldahead/pdf/classmatepc_productbrief.pdf?iid=worldahead+ac_cmpc_pdf) (PDF). Retrieved 2007-01-13.
54. "T6: TrustZone Based Trusted Kernel" (http://www.trustkernel.org). Retrieved 2015-01-12.
55. "Samsung Newsroom" (https://news.samsung.com/global/editorial-protecting-your-mobile-with-samsung-knox). Retrieved 2018-03-07.

External links

- Trusted Computing Group website (http://www.trustedcomputinggroup.org)

Wikipedia
The Free Encyclopedia

Fork (software development)

In software development, a **fork** is a codebase that is created by duplicating an existing codebase and, generally, is subsequently modified independently of the original. Software built from a fork initially has identical behavior as software built from the original code, but as the source code is increasingly modified, the resulting software tends to have increasingly different behavior compared to the original. A fork is a form of branching, but generally involves storing the forked files separately from the original; not in the repository. Reasons for forking a codebase include user preference, stagnated or discontinued development of the original software or a schism in the developer community.[1] Forking proprietary software (such as Unix) is prohibited by copyright law without explicit permission, but free and open-source software, by definition, may be forked without permission.

A timeline chart showing the evolution of Linux distributions, with each split in the diagram being called "a fork"

Etymology

The word *fork* has been used to mean "to divide in branches, go separate ways" as early as the 14th century.[2]

In the context of software development, *fork* was used in the sense of creating a revision control **branch** by Eric Allman as early as 1980, in the context of Source Code Control System:[3]

> Creating a branch "forks off" a version of the program.

The term was in use on Usenet by 1983 for the process of creating a subgroup to move topics of discussion to.[4]

Although *fork* is not known to have been used in the sense of a community schism during the origins of Lucid Emacs (now XEmacs) (1991) or the Berkeley Software Distributions (BSDs) (1993–1994), Russ Nelson used the term *shattering* in this sense in 1993 (attributing it to John Gilmore).[5] In 1995, *fork* was used to describe the XEmacs split,[6] and was an understood usage in the GNU Project by 1996.[7]

The word is used similarly for the fork() system call which causes a running process to split in two – typically, to allow them to perform different tasks in parallel.[8]

Forking of free and open-source software

Free and open-source software may be legally forked without prior approval of those currently developing, managing, or distributing the software per both The Free Software Definition and The Open Source Definition:[9]

> The freedom to distribute copies of your modified versions to others (freedom 3). 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.
>
> —The Free Software Definition[10]

> 3. Derived Works: The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.
>
> —The Open Source Definition[11]

In free software, forks often result from a schism over different goals or personality clashes. In a fork, both parties assume nearly identical code bases, but typically only the larger group, or whoever controls the web site, will retain the full original name and the associated user community. Thus, there is a reputation penalty associated with forking.[9] The relationship between the different teams can be cordial or very bitter. On the other hand, a *friendly fork* or a *soft fork* is a fork that does not intend to compete, but wants to eventually merge with the original.

Eric S. Raymond, in his essay *Homesteading the Noosphere*,[12] stated that "The most important characteristic of a fork is that it spawns competing projects that cannot later exchange code, splitting the potential developer community". He notes in the Jargon File:[13]

> Forking is considered a Bad Thing—not merely because it implies a lot of wasted effort in the future, but because forks tend to be accompanied by a great deal of strife and acrimony between the successor groups over issues of legitimacy, succession, and design direction. There is serious social pressure against forking. As a result, major forks (such as the Gnu-Emacs/XEmacs split, the fissioning of the 386BSD group into three daughter projects, and the short-lived GCC/EGCS split) are rare enough that they are remembered individually in hacker folklore.

David A. Wheeler notes[9] four possible outcomes of a fork, with examples:

1. The death of the fork. This is by far the most common case. It is easy to declare a fork, but considerable effort to continue independent development and support.
2. A re-merging of the fork (*e.g.*, egcs becoming "blessed" as the new version of GNU Compiler Collection.)
3. The death of the original (*e.g.* the X.Org Server succeeding and XFree86 dying.)
4. Successful branching, typically with differentiation (*e.g.*, OpenBSD and NetBSD.)

Distributed revision control (DVCS) tools have popularised a less emotive use of the term "fork", blurring the distinction with "branch".[14] With a DVCS such as Mercurial or Git, the normal way to contribute to a project, is to first create a personal branch of the repository, independent of the main repository, and later seek to have your changes integrated with it. Sites such as GitHub, Bitbucket and Launchpad provide free DVCS hosting expressly supporting independent branches, such that the technical, social and financial barriers to forking a source code repository are massively reduced, and GitHub uses "fork" as its term for this method of contribution to a project.

Forks often restart version numbering from numbers typically used for initial versions of programs like 0.0.1, 0.1, or 1.0 even if the original software was at another version such as 3.0, 4.0, or 5.0. An exception is sometimes made when the forked software is designed to be a drop-in replacement for the original project, *e.g.* MariaDB for MySQL[15] or LibreOffice for OpenOffice.org.

The BSD licenses permit forks to become proprietary software, and copyleft proponents say that commercial incentives thus make proprietisation almost inevitable. (Copyleft licenses can, however, be circumvented via dual-licensing with a proprietary grant in the form of a Contributor License Agreement.) Examples include macOS (based on the proprietary NeXTSTEP and the open source FreeBSD), Cedega and CrossOver (proprietary forks of Wine, though CrossOver tracks Wine and contributes considerably), EnterpriseDB (a fork of PostgreSQL, adding Oracle compatibility features[16]), Supported PostgreSQL with their proprietary ESM storage system,[17] and Netezza's[18] proprietary highly scalable derivative of PostgreSQL. Some of these vendors contribute back changes to the community project, while some keep their changes as their own competitive advantages.

Forking proprietary software

In proprietary software, the copyright is usually held by the employing entity, not by the individual software developers. Proprietary code is thus more commonly forked when the owner needs to develop two or more versions, such as a windowed version and a command line version, or versions for differing operating systems, such as a word processor for IBM PC compatible machines and Macintosh computers. Generally, such internal forks will concentrate on having the same look, feel, data format, and behavior between platforms so that a user familiar with one can also be productive or share documents generated on the other. This is almost always an economic decision to generate a greater market share and thus pay back the associated extra development costs created by the fork.

A notable proprietary fork not of this kind is the many varieties of proprietary Unix—almost all derived from AT&T Unix under license and all called "Unix", but increasingly mutually incompatible.[19] *See* Unix wars.

See also

- Custom software – Software that is specially developed for some specific organization or other user
- Downstream (software development) – Concept in computer science
- Duplicate code – piece of source code that occurs more than once in the same environment
- Group decision-making – Subset of Decision Science

- List of software forks
- Modding – Customization of a product by the end user
- Modular programming – Software design technique
- Personalization – Using technology to accommodate the differences between individuals
- ROM Hacking – Editing technique for video games
- Source port – Altering a game engine's source code to allow it to run on incompatible systems
- Team effectiveness – A team's ability to accomplish their goals or objectives

References

1. "Schism", with its connotations, is a common usage, e.g.
   - "the Lemacs/FSFmacs schism" (https://www.jwz.org/doc/lemacs.html) Archived (https://web.archive.org/web/20091130093142/http://www.jwz.org/doc/lemacs.html) 30 November 2009 at the Wayback Machine (Jamie Zawinski, 2000)
   - "Behind the KOffice split" (https://lwn.net/Articles/419822/) Archived (https://web.archive.org/web/20130706094238/http://lwn.net/Articles/419822/) 6 July 2013 at the Wayback Machine (Joe Brockmeier, *Linux Weekly News*, 2010-12-14)
   - "Copyright assignment – once bitten, twice shy" (http://www.h-online.com/open/features/Copyright-assignment-Once-bitten-twice-shy-1049631.html) Archived (https://web.archive.org/web/20120330153250/http://www.h-online.com/open/features/Copyright-assignment-Once-bitten-twice-shy-1049631.html) 30 March 2012 at the Wayback Machine (Richard Hillesley, *H-Online*, 2010-08-06)
   - "Forking is a feature" (http://dashes.com/anil/2010/09/forking-is-a-feature.html) Archived (https://web.archive.org/web/20120229032536/http://dashes.com/anil/2010/09/forking-is-a-feature.html) 29 February 2012 at the Wayback Machine (Anil Dash, 2010-09-10)
   - "The Great Software Schism" (http://www.linuxjournal.com/node/1000101) Archived (https://web.archive.org/web/20120106065841/http://www.linuxjournal.com/node/1000101) 6 January 2012 at the Wayback Machine (Glyn Moody, *Linux Journal*, 2006-09-28)
   - "To Fork Or Not To Fork: Lessons From Ubuntu and Debian" (http://mako.cc/writing/to_fork_or_not_to_fork.html) Archived (https://web.archive.org/web/20120226160810/http://mako.cc/writing/to_fork_or_not_to_fork.html) 26 February 2012 at the Wayback Machine (Benjamin Mako Hill, 2005).
2. Entry 'fork' in Online Etymology Dictionary (http://www.etymonline.com/index.php?term=fork) Archived (https://web.archive.org/web/20120525165727/http://www.etymonline.com/index.php?term=fork) 25 May 2012 at the Wayback Machine
3. Allman, Eric. "An Introduction to the Source Code Control System." (http://sccs.sourceforge.net/man/sccs.me.html) Archived (https://web.archive.org/web/20141106144859/http://sccs.sourceforge.net/man/sccs.me.html) 6 November 2014 at the Wayback Machine Project Ingres, University of California at Berkeley, 1980.
4. Can somebody fork off a "net.philosophy"? (https://groups.google.com/group/net.misc/browse_thread/thread/b0e9f8531558b7e9/1cc726d9e9e05ebd?q=fork#1cc726d9e9e05ebd) (John Gilmore, net.misc, 18 January 1983)
5. Shattering — good or bad? (https://groups.google.com/group/gnu.misc.discuss/browse_thread/thread/1bba70b8f8676c43/309504a5a51dd0f0) (Russell Nelson, gnu.misc.discuss, 1 October 1993)
6. Re: Hey Franz: 32K Windows SUCK!!!!! (https://groups.google.com/group/cu.cs.macl.info/browse_thread/thread/ed9e5ff9cb21c359/4ce930d2fd1271eb?q=%22fork+of%22+xemacs#4ce930d2fd1271eb) (Bill Dubuque, cu.cs.macl.info, 21 September 1995)

7. Lignux? (https://groups.google.com/group/gnu.misc.discuss/browse_thread/thread/5d7529d865d4d9ca/69ebc4771f32bc45?q=fork+xemacs&utoken=jCqifzMAAABlYGNE0Z8-NzB8d_a6gkmMuNtzuV4zLfLK8bEgnt4z8g27cB9GA0FI9e6hHUM__C_J8dUPGZeQNP0WkVA49NN0) (Marcus G. Daniels, gnu.misc.discuss, 7 June 1996)
8. "The term fork is derived from the POSIX standard for operating systems: the system call used so that a process generates a copy of itself is called fork()." Robles, Gregorio; González-Barahona, Jesús M. (2012). *A Comprehensive Study of Software Forks: Dates, Reasons and Outcomes* (http://flosshub.org/sites/flosshub.org/files/paper_0.pdf) (PDF). OSS 2012 The Eighth International Conference on Open Source Systems. doi:10.1007/978-3-642-33442-9_1 (https://doi.org/10.1007%2F978-3-642-33442-9_1). Archived (https://web.archive.org/web/20131202221721/http://flosshub.org/sites/flosshub.org/files/paper_0.pdf) (PDF) from the original on 2 December 2013. Retrieved 20 October 2012.
9. Why Open Source Software / Free Software (OSS/FS, FLOSS, or FOSS)? Look at the Numbers!: Forking (http://www.dwheeler.com/oss_fs_why.html#forking) Archived (https://web.archive.org/web/20060405112628/http://www.dwheeler.com/oss_fs_why.html) 5 April 2006 at the Wayback Machine (David A. Wheeler)
10. Stallman, Richard. "The Free Software Definition" (https://www.gnu.org/philosophy/free-sw.html). Free Software Foundation. Archived (https://web.archive.org/web/20131014132149/http://www.gnu.org/philosophy/free-sw.html) from the original on 14 October 2013. Retrieved 15 October 2013.
11. "The Open Source Definition" (http://opensource.org/docs/osd). The Open Source Initiative. 7 July 2006. Archived (https://web.archive.org/web/20131015144021/http://opensource.org/docs/osd) from the original on 15 October 2013. Retrieved 15 October 2013.
12. Raymond, Eric S. (15 August 2002). "Promiscuous Theory, Puritan Practice" (http://www.catb.org/~esr/writings/cathedral-bazaar/homesteading/ar01s03.html). *catb.org*. Archived (https://web.archive.org/web/20061006010031/http://www.catb.org/~esr/writings/cathedral-bazaar/homesteading/ar01s03.html) from the original on 6 October 2006.
13. Forked (http://catb.org/jargon/html/F/forked.html) Archived (https://web.archive.org/web/20111108171515/http://catb.org/jargon/html/F/forked.html) 8 November 2011 at the Wayback Machine (Jargon File), first added to v4.2.2 (http://magic-cookie.co.uk/jargon/jarg422/) Archived (https://web.archive.org/web/20120114192615/http://magic-cookie.co.uk/jargon/jarg422/) 14 January 2012 at the Wayback Machine, 20 August 2000)
14. *e.g.* Willis, Nathan (15 January 2015). "An "open governance" fork of Node.js" (https://lwn.net/Articles/628527/). *LWN.net*. Archived (https://web.archive.org/web/20150421055059/http://lwn.net/Articles/628527/) from the original on 21 April 2015. Retrieved 15 January 2015. "Forks are a natural part of the open development model—so much so that GitHub famously plasters a "fork your own copy" button on almost every page." See also Nyman, Linus (2015). *Understanding Code Forking in Open Source Software* (PhD). Hanken School of Economics. p. 57. hdl:10138/153135 (https://hdl.handle.net/10138%2F153135). "Where practitioners have previously had rather narrow definitions of a fork, [...] the term now appears to be used much more broadly. Actions that would traditionally have been called a branch, a new distribution, code fragmentation, a pseudo-fork, etc. may all now be called forks by some developers. This appears to be in no insignificant part due to the broad definition and use of the term fork by GitHub."
15. Forked a project, where do my version numbers start? (http://programmers.stackexchange.com/questions/31551/forked-a-project-where-do-my-version-numbers-start) Archived (https://web.archive.org/web/20110826152252/http://programmers.stackexchange.com/questions/31551/forked-a-project-where-do-my-version-numbers-start) 26 August 2011 at the Wayback Machine
16. EnterpriseDB (http://www.enterprisedb.com) Archived (https://web.archive.org/web/20061113052033/http://www.enterprisedb.com/) 13 November 2006 at the Wayback Machine

17. Fujitsu Supported PostgreSQL (http://www.fastware.com.au/docs/FujitsuSupportedPostreSQLWhitePaperV1_02.pdf) Archived (https://web.archive.org/web/20060820144738/http://fastware.com.au/docs/FujitsuSupportedPostreSQLWhitePaperV1_02.pdf) 20 August 2006 at the Wayback Machine
18. Netezza (http://www.netezza.com/media/2004/Weblog.htm) Archived (https://web.archive.org/web/20061113210650/http://www.netezza.com/media/2004/Weblog.htm) 13 November 2006 at the Wayback Machine
19. Fear of forking (http://linuxmafia.com/faq/Licensing_and_Law/forking.html) Archived (https://web.archive.org/web/20121217044712/http://linuxmafia.com/faq/Licensing_and_Law/forking.html) 17 December 2012 at the Wayback Machine – An essay about forking in free software projects, by Rick Moen

External links

- Right to Fork (http://meatballwiki.org/wiki/RightToFork) at Meatball Wiki
- A PhD examining forking: (Nyman, 2015) (http://helda.helsinki.fi/handle/10138/153135) Archived (https://web.archive.org/web/20230716095029/http://helda.helsinki.fi/handle/10138/153135) 16 July 2023 at the Wayback Machine "Understanding Code Forking in Open Source Software – An examination of code forking, its effect on open source software, and how it is viewed and practiced by developers"

GNU Manifesto

The ***GNU Manifesto*** is a call-to-action by Richard Stallman encouraging participation and support of the GNU Project's goal in developing the GNU free computer operating system. The GNU Manifesto was published in March 1985 in *Dr. Dobb's Journal of Software Tools*.[1] It is held in high regard within the free software movement as a fundamental philosophical source.[2][3][4][5][6][7]

The full text is included with GNU software such as Emacs, and is publicly available.[8]

GNU logo

Background

Some parts of the *GNU Manifesto* began as an announcement of the GNU Project posted by Richard Stallman on September 27, 1983, in form of an email on Usenet newsgroups.[9] The project's aim was to give computer users freedom and control over their computers by collaboratively developing and providing software that is based on Stallman's idea of software freedom (although the written definition had not existed until February 1986).[10] The manifesto was written as a way to familiarize more people with these concepts, and to find more support in form of work, money, programs and hardware.

The *GNU Manifesto* possessed its name and full written form in 1985 but was updated in minor ways in 1987.[8]

Summary

The *GNU Manifesto* opens with an explanation of what the GNU Project is, and what is the current, at the time, progress in creation of the GNU operating system. The system, although based on, and compatible with Unix, is meant by the author to have many improvements over it, which are listed in detail in the manifesto.

One of the major driving points behind the GNU project, according to Stallman, was the rapid (at the time) trend toward Unix and its various components becoming proprietary (i.e. closed-source and non-libre) software.[11]

The manifesto lays a philosophical basis for launching the project, and importance of bringing it to fruition — proprietary software is a way to divide users, who are no longer able to help each other. Stallman refuses to write proprietary software as a sign of solidarity with them.

The author provides many reasons for why the project and software freedom is beneficial to users, although he agrees that its wide adoption will make the work of programmers less profitable.

A large part of the *GNU Manifesto* is focused on rebutting possible objections to GNU Project's goals. They include the programmer's need to make a living, the issue of advertising and distributing free software, and the perceived need of a profit incentive.

Inspired by GNU Manifesto

Throughout history, the *GNU Manifesto* has inspired various other UNIX-related manifestos. Based on it, 10 years later, a popular magazine Linux Focus released its manifesto.[12] 20 years later, a popular illustrations publisher has published their Linux developer manifesto.[13]

See also

Free and open-source software portal

- History of free and open-source software
- Open Letter to Hobbyists

References

1. Stallman, Richard (March 1985). "The GNU Manifesto" (http://www.math.utah.edu/ftp/pub/tex/bib/toc/dr-dobbs-1980.html#10(3):March:1985). *Dr. Dobb's Journal*. **10** (3): 30. Retrieved 2011-10-18.
2. Bustillos, Maria (2015-03-17). "The GNU Manifesto Turns Thirty" (https://www.newyorker.com/business/currency/the-gnu-manifesto-turns-thirty). *The New Yorker*. ISSN 0028-792X (https://search.worldcat.org/issn/0028-792X). Retrieved 2019-10-07.
3. "Trisquel GNU/Linux flies the flag for software freedom" (https://web.archive.org/web/20191007143507/https://www.computerworld.com.au/article/460211/trisquel_gnu_linux_flies_flag_software_freedom/). *Computerworld*. Archived from the original (https://www.computerworld.com.au/article/460211/trisquel_gnu_linux_flies_flag_software_freedom/) on 2019-10-07. Retrieved 2019-10-07.
4. "LWN: Interview with Richard M. Stallman" (https://lwn.net/2002/features/rms.php3). *lwn.net*. Retrieved 2019-10-07.
5. "Developer interview: DOS is (long) dead, long live FreeDOS" (https://web.archive.org/web/20191007150522/https://www.computerworld.com.au/article/452826/developer_interview_dos_long_dead_long_live_freedos/). *Computerworld*. Archived from the original (https://www.computerworld.com.au/article/452826/developer_interview_dos_long_dead_long_live_freedos/) on 2019-10-07. Retrieved 2019-10-07.
6. "CNN - Apple warms up to open source community - June 16, 1999" (http://www.cnn.com/TECH/computing/9906/16/apple.idg/). *www.cnn.com*. Retrieved 2019-10-07.
7. "Red Hat: open source genesis, to mainstreaming revelations - Open Source Insider" (https://www.computerweekly.com/blog/Open-Source-Insider/Red-Hat-open-source-genesis-to-mainstreaming-revelations). *www.computerweekly.com*. Retrieved 2019-10-07.
8. Stallman, Richard (March 1985). "The GNU Manifesto" (https://www.gnu.org/gnu/manifesto.html). GNU Project. Retrieved 2019-02-09.
9. Stallman, Richard. "Initial announcement of the GNU Project" (https://www.gnu.org/gnu/initial-announcement.en.html). *www.gnu.org*. Retrieved 2019-02-07.

10. Stallman, Richard M. (February 1986). "GNU's Bulletin, Volume 1 Number 1" (https://www.gnu.org/bulletins/bull1.txt). *Gnu.org*. p. 8. Retrieved 2019-02-08.
11. Armstrong, Alex (2015-03-25). "GNU Manifesto Published Thirty Years Ago" (https://www.i-programmer.info/news/82-heritage/8403-gnu-manifesto-published-thirty-years-ago.html). *I Programmer*. Retrieved 2019-10-07.
12. "Linux focus manifesto" (http://www.linuxfocus.org/English/LF-manifesto.html). *linuxfocus.org*. April 2023. Retrieved 2023-04-12.
13. "Linux developer manifesto" (https://www.devshirt.club/manifesto/linux/). *devshirt.club*. April 2023. Retrieved 2023-04-12.

External links

- GNU Manifesto (https://www.gnu.org/gnu/manifesto.html)

Microsoft Open Specification Promise

The **Microsoft Open Specification Promise** (or **OSP**) is a promise by Microsoft, published in September 2006, to not assert its patents, in certain conditions, against implementations of a certain list of specifications.[1]

The OSP is not a licence, but rather a covenant not to sue.[2] It promises protection but does not grant any rights.

The OSP is limited to implementations to the extent that they conform to those specifications. This allows for conformance to be partial. So if an implementation follows the specification for some aspects, and deviates in other aspects, then the Covenant Not to Sue applies only to the implementation's aspects which follow the specification.[3]

Relations with free software and open source projects

The protections granted by the OSP are independent to the licence of implementations. There is disagreement as to whether the conditions of the OSP can be fulfilled by free software and open source projects, and whether they thus gain any protection from the OSP.

An article in Cover Pages quotes Lawrence Rosen, an attorney and lecturer at Stanford Law School, as saying,

> "I'm pleased that this OSP is compatible with free and open-source licenses."[4]

Linux vendor Red Hat's stance, as communicated by lawyer Mark Webbink in 2006, is:

> "Red Hat believes that the text of the OSP gives sufficient flexibility to implement the listed specifications in software licensed under free and open-source licenses. We commend Microsoft's efforts to reach out to representatives from the open source community and solicit their feedback on this text, and Microsoft's willingness to make modifications in response to our comments."[5][6]

Standards lawyer Andy Updegrove said in 2006 the Open Specification Promise was

> "what I consider to be a highly desirable tool for facilitating the implementation of open standards, in particular where those standards are of interest to the open source community."[7]

However, the Software Freedom Law Center, a law firm for free software and open source software, has warned of problems with the OSP for use in free software and open source software projects. In a published analysis of the promise it states that[8]

> "...it permits implementation under free software licenses so long as the resulting code isn't used freely."[9]

Their analysis warned of a possible inconsistency with GPL.[9] This applies specifically to the patent promise scope being limited to conforming implementations of covered specifications only.

Effectively when an implementer owns a patent and builds that patent technology in GPL3 licensed code, the implementer grants those first party patent rights downline to all re-users of that code.[10] When the code is reused, the OSP only applies as long as the reuse of that code is limited to implementing the covered specifications.

Other patent promises with similar limitations include IBM's Interoperability Specifications Pledge (https://web.archive.org/web/20090531052839/http://www-03.ibm.com/linux/ossstds/isplist.html) (ISP) and Sun Microsystems' OpenDocument Patent Statement (http://www.oasis-open.org/committees/office/ipr.php). This means, for example, that use of the required Sun patented StarOffice-related technology for OpenDocument should be protected by the Sun Covenant, but reuse of the code with the patented technology for non-OpenDocument implementations is no longer protected by the related Sun covenant.

For this reason the SFLC has stated:

> "The OSP cannot be relied upon by GPL developers for their implementations not because its provisions conflict with GPL, but because it does not provide the freedom that the GPL requires."[9]

The SFLC specifically point out:

- new versions of listed specifications could be issued at any time by Microsoft, and be excluded from the OSP.
- any code resulting from an implementation of one of the covered specifications could not safely be used outside the very limited field of use defined by Microsoft in the OSP.[9]

The Microsoft OSP itself mentions the GPL in two of its FAQs. In one it says,

> "we can't give anyone a legal opinion about how our language relates to the GPL or other OSS licenses".

In another, it specifically only mentions the "developers, distributors, and users of Covered Implementations", so excluding downstream developers, distributors, and users of code later derived from these "Covered Implementations"[9] and it specifically does not mention which version of the GPL is addressed, leading some commentators to conclude that the current GPLv3 may be excluded.[11]

> Q: I am a developer/distributor/user of software that is licensed under the GPL, does the Open Specification Promise apply to me?
> A: Absolutely, yes. The OSP applies to developers, distributors, and users of Covered Implementations without regard to the development model that created such implementations, or the type of copyright licenses under which they are distributed, or the business model of distributors/implementers. The OSP provides the assurance that Microsoft

will not assert its Necessary Claims against anyone who make, use, sell, offer for sale, import, or distribute any Covered Implementation under any type of development or distribution model, including the GPL.[12]

Licensed technologies

Technologies on which the Open Specification Promise applies are:[1][13]

Web Services

- Devices Profile for Web Services (DPWS)
- Identity Selector Interoperability Profile v1.0
- Identity Selector Interoperability Profile v1.5
- Open Data Protocol (OData)
- Remote Shell Web Services Protocol
- SOAP
- SOAP 1.1 Binding for MTOM 1.0
- SOAP MTOM / XOP
- SOAP-over-UDP
- Web Single Sign-On Interoperability Profile
- Web Single Sign-On Metadata Exchange Protocol
- WS-Addressing
- WS-Addressing End Point References and Identity
- WS-AtomicTransaction
- WS-BusinessActivity
- WS-Coordination
- WS-Discovery
- WSDL
- WSDL 1.1 Binding Extension for SOAP 1.2
- WS-Enumeration
- WS-Eventing
- WS-Federation
- WS-Federation Active Requestor Profile
- WS-Federation Passive Requestor Profile
- WS-I Basic Profile
- WS-Management
- WS-Management Catalog
- WS-MetadataExchange
- WS-Policy
- WS-PolicyAttachment
- WS-ReliableMessaging
- WS-RM Policy
- WS-SecureConversation
- WS-Security: Kerberos Binding

- WS-Security: Kerberos Token Profile
- WS-Security: Rights Expression Language (REL) Token Profile
- WS-Security: SAML Token profile
- WS-Security: SOAP Message Security
- WS-Security: UsernameToken Profile
- WS-Security: X.509 Certificate Token Profile
- WS-SecurityPolicy
- WS-Transfer
- WS-Trust

Web

- OpenService Format Specification (a.o. Accelerator)
- Web Slice Format Specification introduced with Internet Explorer 8
- XML Search Suggestions Format Specification

Virtualization Specifications

- Virtual Hard Disk (VHD) Image Format Specification
- Microsoft Application Virtualization File Format Specification v1
- Hyper-V Functional Specification[14]

Security

- RFC 4406 – Sender ID: Authenticating E-Mail
- RFC 4408 – Sender Policy Framework: Authorizing Use of Domains in "Mail From"
- RFC 4407 – Purported Responsible Address in E-Mail Messages
- RFC 4405 – SMTP Service Extension for Indicating the Responsible Submitter of an E-Mail Message
- RFC 7208 – Sender Policy Framework (SPF) for Authorizing Use of Domains in Email[15]
- U-Prove Cryptographic Specification V1.0
- U-Prove Technology Integration into the Identity Metasystem V1.0

Office file formats

XML file formats

- Office 2003 XML Reference Schemas
- Office Open XML 1.0 – Ecma-376
- Office Open XML ISO/IEC 29500:2008
- OpenDocument Format for Office Applications v1.0 OASIS
- OpenDocument Format for Office Applications v1.0 ISO/IEC 26300:2006
- OpenDocument Format for Office Applications v1.1 OASIS

Binary file formats

- Word 97-2007 Binary File Format (.doc) Specification
- PowerPoint 97-2007 Binary File Format (.ppt) Specification
- Excel 97-2007 Binary File Format (.xls) Specification
- Excel 2007 Binary File Format (.xlsb) Specification
- Office Drawing 97-2007 Binary Format Specification

Structure specifications

- [MS-DOC]: Word Binary File Format (.doc) Structure Specification
- [MS-PPT]: PowerPoint Binary File Format (.ppt) Structure Specification
- [MS-XLS]: Excel Binary File Format (.xls) Structure Specification
- [MS-XLSB]: Excel Binary File Format (.xlsb) Structure Specification
- [MS-ODRAW]: Office Drawing Binary File Format Structure Specification
- [MS-CTDOC]: Word Custom Toolbar Binary File Format Structure Specification
- [MS-CTXLS]: Excel Custom Toolbar Binary File Format Structure Specification
- [MS-OFORMS]: Office Forms Binary File Format Structure Specification
- [MS-OGRAPH]: Office Graph Binary File Format Structure Specification
- [MS-OSHARED]: Office Common Data Types and Objects Structure Specification
- [MS-OVBA]: Office VBA File Format Structure Specification
- [MS-OFFCRYPTO]: Office Document Cryptography Structure Specification

Windows compound formats

- [MS-CFB] Windows Compound Binary File Format Specification

Graphics formats

- Windows Metafile Format (.wmf) Specification
- Ink Serialized Format (ISF) Specification
- JPEG XR (.jxr) Format

Microsoft computer languages

- [MS-XAML]: XAML Object Mapping Specification 2006 (Draft v0.1)
- [MS-XAML]: XAML Object Mapping Specification 2006 (v1.0)
- [MS-WPFXV]: WPF XAML Vocabulary Specification 2006 (Draft v0.1)
- [MS-WPFXV]: WPF XAML Vocabulary Specification 2006 (v1.0)
- [MS-SLXV]: Silverlight XAML Vocabulary Specification 2008 (Draft v0.9)

Robotics

- Decentralized Software Services Protocol – DSSP/1.0

Synchronization

- FeedSync v1.0, v1.0.1

Windows Rally Technologies

- Windows Connect Now – UFD and Windows Vista
- Windows Connect Now – UFD for Windows XP

Published protocols

In Microsoft's list of covered protocols there are many third-party protocols which Microsoft did not create but for which they imply they have patents which are necessary for implementation:

- AppleTalk
- [MC-BUP]: Background Intelligent Transfer Service (BITS) Upload Protocol Specification
- [MC-CCFG]: Server Cluster: Configuration (ClusCfg) Protocol Specification
- [MC-COMQC]: Component Object Model Plus (COM+) Queued Components Protocol Specification
- [MC-FPSEWM]: FrontPage Server Extensions: Website Management Specification
- [MC-SMP]: Session Multiplex Protocol Specification
- [MC-SQLR]: SQL Server Resolution Protocol Specification
- 1394 Serial Bus Protocol 2
- IBM NetBIOS Extended User Interface (NetBEUI) v 3.0
- IEC 61883-1
- IEEE 1284 – Interface - Parallel
- IEEE 802.1x - 2004
- Infrared Data Association (IrDA) Published Standards
- Intel Preboot Execution Environment (PXE)
- Novell Internetwork Packet Exchange (IPX)
- Novell Sequenced Packet Exchange (SPX)
- Novell Service Advertising Protocol (SAP)
- RFC 1001 (https://www.rfc-editor.org/rfc/rfc1001) and RFC 1002 (https://www.rfc-editor.org/rfc/rfc1002) – NetBIOS over TCP (NETBT)
- Serial Line Internet Protocol (SLIP, RFC 1055 (https://www.rfc-editor.org/rfc/rfc1055))
- RFC 1058 (https://www.rfc-editor.org/rfc/rfc1058), RFC 1723 (https://www.rfc-editor.org/rfc/rfc1723), and RFC 2453 (https://www.rfc-editor.org/rfc/rfc2453) – Routing Information Protocol 1.0, 2.0 (RIP)
- RFC 1112 (https://www.rfc-editor.org/rfc/rfc1112), RFC 2236 (https://www.rfc-editor.org/rfc/rfc2236), and RFC 3376 (https://www.rfc-editor.org/rfc/rfc3376) – Internet Group Management Protocol (IGMP) v1, v2, and v3

- RFC 1155 (https://www.rfc-editor.org/rfc/rfc1155), RFC 1157 (https://www.rfc-editor.org/rfc/rfc1157), RFC 1213 (https://www.rfc-editor.org/rfc/rfc1213), RFC 1289 (https://www.rfc-editor.org/rfc/rfc1289), RFC 1901 (https://www.rfc-editor.org/rfc/rfc1901), RFC 1902 (https://www.rfc-editor.org/rfc/rfc1902), RFC 1903 (https://www.rfc-editor.org/rfc/rfc1903), RFC 1904 (https://www.rfc-editor.org/rfc/rfc1904), RFC 1905 (https://www.rfc-editor.org/rfc/rfc1905), RFC 1906 (https://www.rfc-editor.org/rfc/rfc1906), RFC 1907 (https://www.rfc-editor.org/rfc/rfc1907), and RFC 1908 (https://www.rfc-editor.org/rfc/rfc1908): Simple Network Management Protocol v2 (SNMP)
- RFC 1179 (https://www.rfc-editor.org/rfc/rfc1179) – Line Printer Daemon (LPD)
- RFC 1191 (https://www.rfc-editor.org/rfc/rfc1191), RFC 1323 (https://www.rfc-editor.org/rfc/rfc1323), RFC 2018 (https://www.rfc-editor.org/rfc/rfc2018), and RFC 2581 (https://www.rfc-editor.org/rfc/rfc2581) – TCP/IP Extensions
- RFC 1256 (https://www.rfc-editor.org/rfc/rfc1256) – ICMP Router Discovery Messages
- RFC 1258 (https://www.rfc-editor.org/rfc/rfc1258) and RFC 1282 (https://www.rfc-editor.org/rfc/rfc1282) – Remote LOGIN (rlogin)
- RFC 1332 (https://www.rfc-editor.org/rfc/rfc1332) and RFC 1877 (https://www.rfc-editor.org/rfc/rfc1877) – Internet Protocol Control Protocol (IPCP)
- RFC 1334 (https://www.rfc-editor.org/rfc/rfc1334) – Password Authentication Protocol (PAP)
- RFC 1393 (https://www.rfc-editor.org/rfc/rfc1393) – Traceroute
- RFC 1436 (https://www.rfc-editor.org/rfc/rfc1436) – Internet Gopher
- RFC 1483 (https://www.rfc-editor.org/rfc/rfc1483), RFC 1755 (https://www.rfc-editor.org/rfc/rfc1755), and RFC 2225 (https://www.rfc-editor.org/rfc/rfc2225) – Internet Protocol over Asynchronous Transfer Mode (IP over ATM)
- RFC 1510 (https://www.rfc-editor.org/rfc/rfc1510) and RFC 1964 (https://www.rfc-editor.org/rfc/rfc1964) – Kerberos Network Authentication Service (v5)
- RFC 1552 (https://www.rfc-editor.org/rfc/rfc1552) – PPP Internetwork Packet Exchange Control Protocol (IPXCP)
- RFC 1661 (https://www.rfc-editor.org/rfc/rfc1661) – Point-to-Point Protocol (PPP)
- RFC 1739 (https://www.rfc-editor.org/rfc/rfc1739) Section 2.2 – Packet Internet Groper (ping)
- RFC 1889 (https://www.rfc-editor.org/rfc/rfc1889) and RFC 3550 (https://www.rfc-editor.org/rfc/rfc3550) – Real-Time Transport Protocol (RTP)
- RFC 1939 (https://www.rfc-editor.org/rfc/rfc1939) and RFC 1734 (https://www.rfc-editor.org/rfc/rfc1734) – Post Office Protocol, v3 (POP3)
- RFC 1962 (https://www.rfc-editor.org/rfc/rfc1962) – Compression Control Protocol (CCP)
- RFC 1990 (https://www.rfc-editor.org/rfc/rfc1990) – Multilink Protocol (MP)
- RFC 1994 (https://www.rfc-editor.org/rfc/rfc1994) – MD5 Challenge Handshake Authentication Protocol (MD5-CHAP)
- RFC 2097 (https://www.rfc-editor.org/rfc/rfc2097) – NetBIOS Frames Control Protocol (NBFCP)
- RFC 2118 (https://www.rfc-editor.org/rfc/rfc2118) – Microsoft Point-to-Point Compression (MPPC)
- RFC 2125 (https://www.rfc-editor.org/rfc/rfc2125) – Bandwidth Allocation Protocol (BAP)
- RFC 2131 (https://www.rfc-editor.org/rfc/rfc2131), RFC 2132 (https://www.rfc-editor.org/rfc/rfc2132), and RFC 3361 (https://www.rfc-editor.org/rfc/rfc3361) – Dynamic Host Configuration Protocol (DHCP)
- RFC 2205 (https://www.rfc-editor.org/rfc/rfc2205), RFC 2209 (https://www.rfc-editor.org/rfc/rfc2209), and RFC 2210 (https://www.rfc-editor.org/rfc/rfc2210) – Resource Reservation Setup (RSVP)

- RFC 2222 (https://www.rfc-editor.org/rfc/rfc2222) – Simple Authentication and Security Layer (SASL)
- RFC 2225 (https://www.rfc-editor.org/rfc/rfc2225) – Asynchronous Transfer Mode
- Server Message Block
- Sun Microsystems Remote Procedure Call (SunRPC)
- T.120
- Tabular Data Stream (TDS) v7.1, 7.2, 7.3
- Universal Plug and Play (UPnP)
- Universal Serial Bus (USB) Revision 2.0

See also

- Microsoft
- Glossary of patent law terms

References

1. "Microsoft Open Specification Promise" (https://learn.microsoft.com/en-us/openspecs/dev_center/ms-devcentlp/1c24c7c8-28b0-4ce1-a47d-95fe1ff504bc). *Microsoft*. Retrieved 2007-11-02.
2. "Microsoft's Open Specification Promise Eases Web Services Patent Concerns" (http://xml.coverpages.org/ni2006-09-12-a.html). Retrieved 2015-04-18.
3. Richard Wilder (Associate General Counsel for Intellectual Property Policy at Microsoft) (2008-07-25). "The OSP and You" (http://port25.technet.com/archive/2008/07/25/osp.aspx). Port 25 (Microsoft's open source portal). Retrieved 2008-07-25. {{cite web}}: |author= has generic name (help)
4. "Microsoft's Open Specification Promise Eases Web Services Patent Concerns" (http://xml.coverpages.org/ni2006-09-12-a.html). xml.coverpages.org. 2006-09-12.
5. "Microsoft Open Specification Promise" (http://www.microsoft.com/interop/osp/default.mspx#EOHAC). *Microsoft*.
6. "Microsoft promises to hang patent fire over web services" (http://www.computerbuyer.co.uk/buyer/news/93789/microsoft-promises-to-hang-patent-fire-over-web-services.html). 2006-09-12.
7. Peter Galli (2006-09-12). "Microsoft Promises Not to Sue over Web Services Specs" (http://www.devsource.com/c/a/Architecture/Microsoft-Promises-Not-to-Sue-over-Web-Services-Specs/).
8. "Software Freedom Law Center Publishes Analysis of Microsoft's Open Specification Promise" (http://www.softwarefreedom.org/news/2008/mar/12/osp-gpl/). Software Freedom Law Center. March 12, 2008.
9. "Microsoft's Open Specification Promise: No Assurance for GPL" (http://www.softwarefreedom.org/resources/2008/osp-gpl.html). Software Freedom Law Center. 2008-03-12. Retrieved 2008-03-30.
10. "GNU General Public License (Version 3, 29 June 2007)" (http://www.fsf.org/licensing/licenses/gpl.html). Free Software Foundation. 2008-02-12. See section 11 for patents.

11. Bilodeau, J-F (March 19, 2008). "Three Things Microsoft Should Do" (https://web.archive.org/web/20100227233629/http://farstarnet.com/jfbilodeau/labels/fud.html). Archived from the original (http://farstarnet.com/jfbilodeau/labels/fud.html) on February 27, 2010. Retrieved 2009-09-16. "Use the GNU GPL 3: Most free software is released under the GNU GPL 2 or 3, which is incompatible with Microsoft's OSP (Open Specification Promise) and the Ms-PL (Microsoft Public License). This is not an accident. Microsoft does not want software written using their technology to spread to other platform. Again, it's vendor lock-in. If Microsoft truly wanted to work with the Open Source community, they should abandon the OSP and the Ms-PL for the GPL, or another OSI certified license."
12. Microsoft Open Specification Promise - Frequently Asked Questions (https://msdn.microsoft.com/en-us/openspecifications/dn646765#OSPFAQ)
13. Microsoft Office Binary (doc, xls, ppt) File Formats (http://www.microsoft.com/interop/docs/officebinaryformats.mspx) Archived (https://web.archive.org/web/20080430103542/http://www.microsoft.com/interop/docs/officebinaryformats.mspx) April 30, 2008, at the Wayback Machine
14. Hypervisor Functional Specification (http://www.microsoft.com/downloads/details.aspx?FamilyID=91e2e518-c62c-4ff2-8e50-3a37ea4100f5&displaylang=en)
15. "[MS-DEVCENTLP]: Open Specifications Dev Center" (http://www.microsoft.com/openspecifications/en/us/programs/osp/default.aspx#security). *www.microsoft.com*. Retrieved 2014-05-08.

External links

- Open Specification Promise (https://learn.microsoft.com/en-us/openspecs/dev_center/ms-devcentlp/1c24c7c8-28b0-4ce1-a47d-95fe1ff504bc) — Microsoft page describing the OSP and listing the specifications covered by it.
- Analysis of OSP (http://www.consortiuminfo.org/standardsblog/article.php?story=20060912140103877) by standards lawyer Andy Updegrove
- Analysis of OSP (http://www.softwarefreedom.org/resources/2008/osp-gpl.html) by Software Freedom Law Center. Rebuttal (http://blogs.technet.com/gray_knowlton/archive/2008/03/13/a-disappointing-surprise-from-the-sflc.aspx) by Gray Knowlton, group product manager for Microsoft Office.
- Open Specifications (https://learn.microsoft.com/en-us/openspecs/main/ms-openspeclp/3589baea-5b22-48f2-9d43-f5bea4960ddb) — Documentation for the covered specifications.

Wikipedia
The Free Encyclopedia

Open-core model

The **open-core model** is a business model for the monetization of commercially produced open-source software. The open-core model primarily involves offering a "core" or feature-limited version of a software product as free and open-source software, while offering "commercial" versions or add-ons as proprietary software.[1][2] The term was coined by Andrew Lampitt in 2008.[3][4]

GitLab Community Edition

The concept of open-core software has proven to be controversial, as many developers do not consider the business model to be true open-source software. Despite this, open-core models are used by many open-source software companies.[5]

Use of contributor license agreements

Some open-core products require their contributors to sign a contributor license agreement, which either dictates that the copyright of all contributions to the product become the property of its owner, or that the product's owner is given an unlimited, non-exclusive license to use the contributions, but the authors retain copyright ownership. In an open-core scenario, these agreements are typically meant to allow the commercial owner of the product (which in some cases, is ultimately the copyright holder to all of its code, regardless of its original author) to simultaneously market versions of the product under open-source and non-free licenses. This is in contrast with more traditional uses of CLAs, which are meant solely to allow the steward of an open-source project to defend and protect the copyrights of its contributors, or to guarantee that the code will only ever be made available under open-source terms (thus protecting it from becoming open core).[6][7][8]

Examples

- Kafka, a data streaming service under the Apache 2.0 license, is the open-source core to the company, Confluent, which issues a Confluent Community License, a source-available license that governs additional features in the Confluent Platform.[9]
- Cassandra, an open-source database under the Apache 2.0 license, is the core to the company, Datastax, which issues enterprise subscription license for additional management and security features inside DataStax Enterprise.[10]
- Instructure's Canvas learning management software.
- Oracle's MySQL database software is dual-licensed under a proprietary license, and the GNU General Public License (GPL); proprietary versions offer additional features and enterprise support plans.[11]

- Oracle VM VirtualBox is GNU GPL-licensed, but some features, such as encryption and remote display, require Oracle's closed-source Extension Pack.[12]
- Elastic's core, which includes Elasticsearch, Kibana, Logstash and Beats, was under an Apache 2.0 license, while additional plugins are distributed under Elastic's own proprietary license.[13] In January 2021, Elastic re-licensed its software under the non-free Server Side Public License and Elastic License, which restrict use of the software as part of managed services, and circumvention of software locks on premium features.[14] This means it is no longer open core, but source available-software.
- Eucalyptus, private cloud software, has a proprietary enterprise edition which provides additional features.[15][16][17]
- IntelliJ IDEA CE (Community Edition) is licensed under the Apache License, while IDEA Ultimate Edition is trialware.
- GitLab CE (Community Edition) is under an MIT-style open source license,[18] while GitLab EE (Enterprise Edition) is under a commercial license.[19]
- Neo4j CE (Community Edition) is licensed under GPL version 3, while Neo4j EE (Enterprise Edition) is under a commercial license, providing additional features including clustering and hot backups.
- Seldon Core, a machine learning platform under the Apache 2.0 license, is the core to the company Seldon, which provides Seldon Deploy under a commercial license.[20]

Restrictions on use in services

A new variation of the practice emerged in 2018 among several open core products intended for server-side use, seeking to control use of the product as part of a service offered to a customer. These practices, in particular, target incorporation of the software into proprietary services by cloud application service providers such as Amazon Web Services, but with what vendors perceive to be inadequate compensation or contributions back to the upstream software in return.[21][22]

MongoDB changed its license from the GNU Affero General Public License (a variation of the GPL which requires that the software's source code be offered to those who use it over a network) to a modified version titled the "Server Side Public License" (SSPL), where the source code of the entire service (including, without limitation, all code needed for another user to run an instance of the service themselves) must be released under the SSPL if it incorporates an SSPL-licensed component (unlike the AGPL, where this provision only applies to the copyrighted work that is licensed under the AGPL).[23] Bruce Perens, co-author of The Open Source Definition, argued that the SSPL violated its requirement for an open source license to not place restrictions on software distributed alongside the licensed software.[21] The Open Source Initiative (OSI) ruled that the SSPL violates the Open Source Definition and is therefore not a free software license, as the provision discriminates against commercial users.[24] Debian, Fedora, and Red Hat Enterprise Linux pulled MongoDB from their distributions after the license change, considering the new license to be in violation of their licensing policies.[23][25]

Redis Labs made its Redis plugins subject to the "Commons Clause", a restriction on sale of the software on top of the existing Apache License terms. After criticism, this was changed in 2019 to the "Redis Source Available License", a non-free license which forbids sale of the software as part of "a database, a caching engine, a stream processing engine, a search engine, an indexing engine or an ML/DL/AI serving engine".[26][22][27] The last versions of the modules licensed solely under the Apache License were forked and are maintained by community members under the GoodFORM project.[21] Redis itself later followed suit in 2024, switching from a BSD-styled license to dual-licensing under the SSPL and Redis Source

Available License; in 2025, the company reinstated a free and open source license by also allowing use under the AGPL, citing that forks (including those created in response to the fork, such as Valkey) had differentiated themselves enough to allow Redis to "compete on product".[28]

A similar move was made when HashiCorp switched to the non-free Business Source License (BSL) on its products, including Terraform, which received the Linux Foundation-backed fork OpenTofu.[29]

In September 2024, WP Engine—a hosting provider that uses the free and open source WordPress software—began to face criticism from Matt Mullenweg—the founder of the project's corporate sponsor Automattic, and owner of the competitor WordPress.com. During a presentation and blog post, he criticized WP Engine over inadequate upstream contributions, disabling of features, private equity funding, and trademark dilution of the "WP" prefix. He called the company a "cancer" to WordPress, and called for a boycott of its services.[30] WP Engine sent a cease and desist to Automattic demanding the removal of the comments, stating that they operated within the WordPress Foundation trademark usage guidelines, and that Automattic had been demanding "significant percentage of its gross revenues" in licensing fees.[31] While WordPress is licensed under the GNU General Public License, Mullenweg began to enforce restrictions against WP Engine by banning it from any services hosted under the WordPress.org domain, including automatic updates and the ability to download plug-ins and themes from within the software. The trademark guidelines were also modified to cover use of "WP".[32] In October 2024, WP Engine formally filed a lawsuit against Automattic for defamation and extortion.[33]

See also

Free and open-source software portal

- Freemium
- Shareware

References

1. Riehle, Dirk (2009). "The Commercial Open Source Business Model" (http://dirkriehle.com/publications/2009/the-commercial-open-source-business-model/). *Value Creation in e-Business Management*. Springer Verlag. pp. 18–30.
2. Wasserman, Anthony I. (2011). "How the Internet transformed the software industry" (https://doi.org/10.1007%2Fs13174-011-0019-x). *Journal of Internet Services and Applications*. **2** (1): 11–22. doi:10.1007/s13174-011-0019-x (https://doi.org/10.1007%2Fs13174-011-0019-x). ISSN 1867-4828 (https://search.worldcat.org/issn/1867-4828). "Some companies have only a single version of their software, while others follow an "open core" model, providing a community release of the core version, and offering proprietary premium features using a commercial license."
3. Lampitt, Andrew (29 August 2008). "Open-Core Licensing (OCL): Is this Version of the Dual License Open Source Business Model the New Standard?" (https://alampitt.typepad.com/lampitt_or_leave_it/2008/08/open-core-licen.html). Retrieved 21 January 2024.
4. Phipps, Simon (July 2012). *Open Source Strategies for the Enterprise*. O'Reilly Media. ISBN 978-1-4493-4117-6.

5. Germain, Jack M. (15 April 2009). "Open Core Debate: The Battle for a Business Model" (https://linuxinsider.com/story/open-core-debate-the-battle-for-a-business-model-66807.html). *Linux Insider*. Retrieved 28 March 2013.
6. Phipps, Simon (21 June 2013). "MySQL mistake is a wake-up call on open source ownership" (https://www.infoworld.com/article/2078812/mysql-mistake-is-a-wake-up-call-on-open-source-ownership.html). *InfoWorld*. Retrieved 11 September 2015.
7. "FSFE welcomes KDE's adoption of the Fiduciary Licence Agreement (FLA)" (https://ev.kde.org/2008/08/22/fsfe-welcomes-kde-s-fiduciary-license-agreement/). *KDE*. 22 August 2008.
8. "6.1 Copyright Papers" (https://www.gnu.org/prep/maintain/html_node/Copyright-Papers.html). *gnu.org*. Retrieved 3 January 2011.
9. "Confluent Community License FAQ" (https://www.confluent.io/confluent-community-license-faq). *Confluent*. Retrieved 9 September 2019.
10. "Product Specific License Terms | DataStax" (https://www.datastax.com/terms/product-specific-license-terms). *DataStax: Active Everywhere, Every Cloud | Hybrid Cloud | Apache Cassandra | NoSQL*. 21 February 2018. Retrieved 9 September 2019.
11. Hillesley, Richard. "Open core or dual licensing? The example of MySQL" (http://www.h-online.com/open/features/Open-core-or-dual-licensing-The-example-of-MySQL-1367824.html). *The H*. Retrieved 11 September 2015.
12. "Licensing: Frequently Asked Questions" (https://www.virtualbox.org/wiki/Licensing_FAQ). Oracle Corporation. Retrieved 21 July 2024.
13. Woodie, Alex (12 March 2019). "War Unfolding for Control of Elasticsearch" (https://www.datanami.com/2019/03/12/search-war-unfolding-for-control-of-elasticsearch/). *Datanami*. Retrieved 9 September 2019.
14. "FAQ on 2021 License Change | Elastic" (https://www.elastic.co/pricing/faq/licensing). *www.elastic.co*. Retrieved 26 May 2021.
15. Bort, Julie (18 April 2012). "This Startup That Angered A Lot Of Open Source Fans Just Got $30 Million In Funding" (http://www.businessinsider.com/the-startup-that-ticked-off-a-lot-of-people-just-got-30-million-in-funding-2012-4). Business Insider. Retrieved 19 February 2016. "It was one of the first commercial companies to champion a concept called "open core." "
16. Bort, Julie (22 June 2010). "Marten Mickos says open source doesn't have to be fully open" (https://www.networkworld.com/article/756296/opensource-subnet-marten-mickos-says-open-source-doesn-t-have-to-be-fully-open.html). Network World. Retrieved 19 February 2016. " "We deliver a fully functional cloud with Eucalyptus software. You can download it on a GPL v3 license. But, additionally, we provide enterprise features only if you pay for them ... it's open core," he says."
17. Jackson, Jacob (25 August 2010). "Eucalyptus Strengthens Its Back End" (http://www.pcworld.com/article/204092/article.html). PCWorld. Retrieved 19 February 2016. "To make money, Eucalyptus Systems uses an open-core business model, offering one version of the software free through an open-source license and selling a commercial version with support and additional features ..."
18. "CONTRIBUTING.md · master · GitLab.org / GitLab Community Edition" (https://gitlab.com/gitlab-org/gitlab-ce/blob/master/CONTRIBUTING.md). *GitLab*. Retrieved 5 June 2018.
19. "GitLab Enterprise Edition license change" (https://about.gitlab.com/2014/02/11/gitlab-ee-license-change/). *GitLab*. 11 February 2014. Retrieved 5 June 2018.
20. *Seldon Core: Blazing Fast, Industry-Ready ML* (https://github.com/SeldonIO/seldon-core/blob/6467d7780be545e046471230b9a7376edadce10e/LICENSE), Seldon, 13 June 2022, retrieved 13 June 2022
21. Gilbertson, Scott (16 October 2019). "In 2019, multiple open source companies changed course—is it the right move?" (https://arstechnica.com/information-technology/2019/10/is-the-software-world-taking-too-much-from-the-open-source-community/). *Ars Technica*. Retrieved 16 October 2019.

22. Finley, Klint (31 July 2019). "When Open Source Software Comes With a Few Catches" (https://www.wired.com/story/when-open-source-software-comes-with-catches/). *Wired*. ISSN 1059-1028 (https://search.worldcat.org/issn/1059-1028). Retrieved 1 August 2019.
23. Vaughan-Nichols, Steven J. (16 January 2019). "MongoDB "open-source" Server Side Public License rejected" (https://www.zdnet.com/article/mongodb-open-source-server-side-public-license-rejected/). *ZDNet*. Retrieved 17 January 2019.
24. OSI Board of Directors (19 January 2021). "The SSPL is Not an Open Source License" (https://opensource.org/node/1099). Open Source Initiative. Retrieved 23 January 2021.
25. "MongoDB's licensing changes led Red Hat to drop the database from the latest version of its server OS" (https://www.geekwire.com/2019/mongodbs-licensing-changes-led-red-hat-drop-database-latest-version-server-os/). *GeekWire*. 16 January 2019. Retrieved 17 January 2019.
26. Vaughan-Nichols, Steven J. "Redis Labs drops Commons Clause for a new license" (https://www.zdnet.com/article/redis-labs-drops-commons-clause-for-a-new-license/). *ZDNet*. Retrieved 1 August 2019.
27. Baer, Tony (16 October 2018). "It's MongoDB's turn to change its open source license" (https://www.zdnet.com/article/its-mongodbs-turn-to-change-its-open-source-license/). *ZDNet*. Retrieved 1 August 2019.
28. "Redis bets big on an open source return" (https://www.infoworld.com/article/3975620/redis-bets-big-on-an-open-source-return.html). *InfoWorld*. Retrieved 17 May 2025.
29. Miller, Ron (20 September 2023). "Terraform fork gets renamed OpenTofu, and joins Linux Foundation" (https://techcrunch.com/2023/09/20/terraform-fork-gets-a-new-name-opentofu-and-joins-linux-foundation/). *TechCrunch*. Retrieved 15 October 2023.
30. Sawers, Paul (22 September 2024), "Matt Mullenweg calls WP Engine a 'cancer to WordPress' and urges community to switch providers" (https://techcrunch.com/2024/09/22/matt-mullenweg-calls-wp-engine-a-cancer-to-wordpress-and-urges-community-to-switch-providers/), *TechCrunch*, archived (https://web.archive.org/web/20241001215144/https://techcrunch.com/2024/09/22/matt-mullenweg-calls-wp-engine-a-cancer-to-wordpress-and-urges-community-to-switch-providers/) from the original on 1 October 2024, retrieved 24 September 2024
31. Mehta, Ivan (24 September 2024). "WP Engine sends cease-and-desist letter to Automattic over Mullenweg's comments" (https://techcrunch.com/2024/09/23/wp-engine-sends-cease-and-desist-letter-to-automattic-over-mullenwegs-comments/). *TechCrunch*. Retrieved 15 October 2024.
32. Mehta, Ivan (26 September 2024). "WordPress.org bans WP Engine, blocks it from accessing its resources" (https://techcrunch.com/2024/09/25/wordpress-org-bans-wp-engine-blocks-it-from-accessing-its-resources/). *TechCrunch*. Retrieved 15 October 2024.
33. Mehta, Ivan (3 October 2024), "WP Engine sues WordPress co-creator Mullenweg and Automattic, alleging abuse of power" (https://techcrunch.com/2024/10/02/wp-engine-sues-automattic-and-wordpress-co-founder-matt-mullenweg/), *TechCrunch*, retrieved 4 October 2024

External links

- Bradley M. Kuhn (16 October 2009). " "Open Core" Is the New Shareware" (http://ebb.org/bkuhn/blog/2009/10/16/open-core-shareware.html).
- Simon Phipps (29 June 2010). "Open Core Is Bad For You" (https://web.archive.org/web/20110128014410/http://blogs.computerworlduk.com/simon-says/2010/06/open-core-is-bad-for-you). ComputerWorldUK. Archived from the original (http://blogs.computerworlduk.com/simon-says/2010/06/open-core-is-bad-for-you) on 28 January 2011.

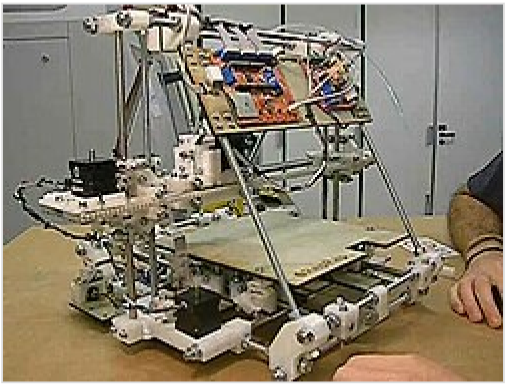
- Brian Prentice (31 March 2010). "Open-Core: The Emperor's New Clothes" (https://web.archive.org/web/20110606151309/http://blogs.gartner.com/brian_prentice/2010/03/31/open-core-the-emperors-new-clothes/). Gartner. Archived from the original (http://blogs.gartner.com/brian_prentice/2010/03/31/open-core-the-emperors-new-clothes/) on 6 June 2011.

Wikipedia
The Free Encyclopedia

Open-source hardware

The "open source hardware" logo proposed by OSHWA, one of the main defining organizations

Open-source hardware (**OSH**, **OSHW**) consists of physical artifacts of technology designed and offered by the open-design movement. Both free and open-source software (FOSS) and open-source hardware are created by this open-source culture movement and apply a like concept to a variety of components. It is sometimes, thus, referred to as **free and open-source hardware** (**FOSH**), meaning that the design is easily available ("open") and that it can be used, modified and shared freely ("free"). The term usually means that information about the hardware is easily discerned so that others can make it – coupling it closely to the maker movement.[1] Hardware design (i.e. mechanical drawings, schematics, bills of material, PCB layout data, HDL source code[2] and integrated circuit layout data), in addition to the software that drives the hardware, are all released under free/libre terms. The original sharer gains feedback and potentially improvements on the design from the FOSH community. There is now significant evidence that such sharing can drive a high return on investment for the scientific community.[3]


The RepRap Mendel general-purpose 3D printer with the ability to make copies of most of its own structural parts

It is not enough to merely use an open-source license; an open source product or project will follow open source principles, such as modular design and community collaboration.[4][5][6]

Since the rise of reconfigurable programmable logic devices, sharing of logic designs has been a form of open-source hardware. Instead of the schematics, hardware description language (HDL) code is shared. HDL descriptions are commonly used to set up system-on-a-chip systems either in field-programmable gate arrays (FPGA) or directly in application-specific integrated circuit (ASIC) designs. HDL modules, when distributed, are called semiconductor intellectual property cores, also known as IP cores.

Open-source hardware also helps alleviate the issue of proprietary device drivers for the free and open-source software community, however, it is not a pre-requisite for it, and should not be confused with the concept of open documentation for proprietary hardware, which is already sufficient for writing FLOSS device drivers and complete operating systems.[7][8] The difference between the two concepts is that OSH includes both the instructions on how to replicate the hardware itself as well as the information on communication protocols that the software (usually in the form of device drivers) must use in order to communicate with the hardware (often called register documentation, or open documentation for hardware[7]), whereas open-source-friendly proprietary hardware would only include the latter without including the former.

History

The first hardware-focused "open source" activities were started around 1997 by Bruce Perens, creator of the Open Source Definition, co-founder of the Open Source Initiative, and a ham radio operator. He launched the Open Hardware Certification Program, which had the goal of allowing hardware manufacturers to self-certify their products as open.[9][10]

Shortly after the launch of the Open Hardware Certification Program, David Freeman announced the Open Hardware Specification Project (OHSpec), another attempt at licensing hardware components whose interfaces are available publicly and of creating an entirely new computing platform as an alternative to proprietary computing systems.[11] In early 1999, Sepehr Kiani, Ryan Vallance and Samir Nayfeh joined efforts to apply the open-source philosophy to machine design applications. Together they established the Open Design Foundation (ODF) [12] as a non-profit corporation and set out to develop an Open Design Definition. However, most of these activities faded out after a few years.

A "Free Hardware" organization, known as FreeIO, was started in the late 1990s by Diehl Martin, who also launched a FreeIO website in early 2000. In the early to mid 2000s, FreeIO was a focus of free/open hardware designs released under the GNU General Public License. The FreeIO project advocated the concept of Free Hardware and proposed four freedoms that such hardware provided to users, based on the similar freedoms provided by free software licenses.[13] The designs gained some notoriety due to Martin's naming scheme in which each free hardware project was given the name of a breakfast food such as Donut, Flapjack, Toast, etc. Martin's projects attracted a variety of hardware and software developers as well as other volunteers. Development of new open hardware designs at FreeIO ended in 2007 when Martin died of pancreatic cancer but the existing designs remain available from the organization's website.[14]

By the mid 2000s open-source hardware again became a hub of activity due to the emergence of several major open-source hardware projects and companies, such as OpenCores, RepRap (3D printing), Arduino, Adafruit, SparkFun, and Open Source Ecology. In 2007, Perens reactivated the openhardware.org website, but it's currently (February 2025) inactive.

openhardware.org logo (2013)

Following the Open Graphics Project, an effort to design, implement, and manufacture a free and open 3D graphics chip set and reference graphics card, Timothy Miller suggested the creation of an organization to safeguard the interests of the Open Graphics Project community. Thus, Patrick McNamara founded the Open Hardware Foundation (OHF) in 2007.[15]

The Tucson Amateur Packet Radio Corporation (TAPR), founded in 1982 as a non-profit organization of amateur radio operators with the goals of supporting R&D efforts in the area of amateur digital communications, created in 2007 the first open hardware license, the TAPR Open Hardware License. The OSI president Eric S. Raymond expressed some concerns about certain aspects of the OHL and decided to not review the license.[16]

Around 2010 in context of the Freedom Defined project, the *Open Hardware Definition* was created as collaborative work of many[17] and is accepted as of 2016 by dozens of organizations and companies.[18]

In July 2011, CERN (European Organization for Nuclear Research) released an open-source hardware license, CERN OHL. Javier Serrano, an engineer at CERN's Beams Department and the founder of the Open Hardware Repository, explained: "By sharing designs openly, CERN expects to improve the quality of designs through peer review and to guarantee their users – including commercial companies – the freedom to study, modify and manufacture them, leading to better hardware and less duplication of efforts".[19] While initially drafted to address CERN-specific concerns, such as tracing the impact of the organization's research, in its current form it can be used by anyone developing open-source hardware.[20]

Following the 2011 Open Hardware Summit, and after heated debates on licenses and what constitutes open-source hardware, Bruce Perens abandoned the OSHW Definition and the concerted efforts of those involved with it.[21] Openhardware.org, led by Bruce Perens, promotes and identifies practices that meet all the combined requirements of the Open Source Hardware Definition, the Open Source Definition, and the Four Freedoms of the Free Software Foundation[22] Since 2014 openhardware.org is not online and seems to have ceased activity.[23]

The Open Source Hardware Association (OSHWA) at oshwa.org acts as hub of open-source hardware activity of all genres, while cooperating with other entities such as TAPR, CERN, and OSI. The OSHWA was established as an organization in June 2012 in Delaware and filed for tax exemption status in July 2013.[24] After some debates about trademark interferences with the OSI, in 2012 the OSHWA and the OSI signed a co-existence agreement.[25][26]

OSHWA logo

The FOSSi Foundation is founded in 2015 as a UK-based non-profit to promote and protect the open source silicon chip movement, roughly a year after the official release of RISC-V architecture.[27]

The Free Software Foundation has suggested an alternative "free hardware" definition derived from the Four Freedoms.[28][29]

Forms of open-source hardware

The term *hardware* in open-source hardware has been historically used in opposition to the term *software* of open-source software. That is, to refer to the electronic hardware on which the software runs (see previous section). However, as more and more non-electronic hardware products are made open source (for example WikiHouse, OpenBeam or Hovalin), this term tends to be used back in its broader sense of "physical product". The field of open-source hardware has been shown to go beyond electronic hardware and to cover a larger range of product categories such as machine tools, vehicles and medical equipment.[30] In that sense, *hardware* refers to any form of tangible product, be it electronic hardware, mechanical hardware, textile or even construction hardware. The Open Source Hardware (OSHW) Definition 1.0 defines hardware as "tangible artifacts — machines, devices, or other physical things".[31]

Explainer video for Open Science Hardware

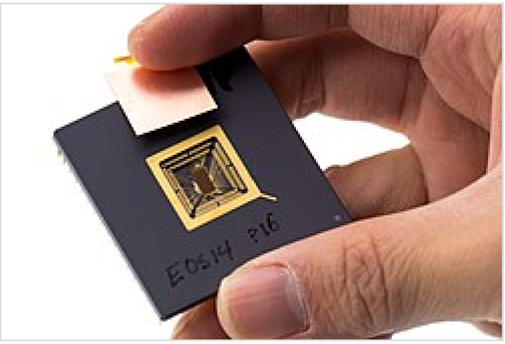
Electronics

Electronics is one of the most popular types of open-source hardware. PCB based designs can be published similarly to software as CAD files, which users can send directly to PCB fabrication companies to receive hardware in the mail. Alternatively, users can obtain components and solder them together themselves.

There are many companies that provide large varieties of open-source electronics such as Sparkfun, Adafruit, and Seeed. In addition, there are NPOs and companies that provide a specific open-source electronic component such as the Arduino electronics prototyping platform. There are many examples of specialty open-source electronics such as low-cost voltage and current GMAW open-source 3-D printer monitor[32][33] and a robotics-assisted mass spectrometry assay platform.[34][35] Open-source electronics finds various uses, including automation of chemical procedures.[36][37]

Chip design

Open Standard chip designs are now common. OpenRISC (2000 - LGPL / GPL), OpenSparc (2005 - GPLv2), and RISC-V (2010 - Open Standard, free to implement for non-commercial purposes), are examples of free to use instruction set architecture.



RISC-V processor prototype, January 2013

OpenCores is a large library of standard chip design subcomponents which can be combined into larger designs.

Complete open source software stacks and shuttle fabrication services are now available which can take OSH chip designs from hardware description languages to masks and ASIC fabrication on maker-scale budgets.[38]

Mechanics

Purely mechanical OSH designs include mechanical components, machine tools, and vehicles. Open Source Ecology is a large project which seeks to develop a complete ecosystem of mechanical tools and components which aim to be able to replicate themselves.

Open-source vehicles have also been developed including bicycles like XYZ Space Frame Vehicles and cars such as the Tabby OSVehicle.

Mechatronics

Most OSH systems combine elements of electronics and mechanics to form mechatronics systems. A large range of open-source mechatronic products have been developed, including machine tools, musical instruments, and medical equipment.[30]

Examples of open-source machine tools include 3D printers such as RepRap, Prusa, and Ultimaker, 3D printer filament extruders such as polystruder[39] XR PRO as well as the laser cutter Lasersaur.

Examples of open source medical equipment include open-source ventilators, the echostethoscope echOpen (co-founded by Mehdi Benchoufi, Olivier de Fresnoye, Pierre Bourrier and Luc Jonveaux[40]), and a wide range of prosthetic hands listed in the review study by Ten Kate *et.al.*[41] (e.g. OpenBionics' Prosthetic Hands).

Robotics

Open source robotics combines open source hardware mechatronics with open source AI and control software. Due to the mixture of hardware and software it serves as a particularly active area for open source ideas to move between them.

Other

Examples of open-source hardware products can also be found to a lesser extent in construction (Wikihouse), textile (Kit Zéro Kilomètres), and firearms (3D printed firearm, Defense Distributed).

Licenses

Rather than creating a new license, some open-source hardware projects use existing, free and open-source software licenses.[42] These licenses may not accord well with patent law.[43]

Later, several new licenses were proposed, designed to address issues specific to hardware design.[44] In these licenses, many of the fundamental principles expressed in open-source software (OSS) licenses have been "ported" to their counterpart hardware projects. New hardware licenses are often explained as the "hardware equivalent" of a well-known OSS license, such as the GPL, LGPL, or BSD license.

Despite superficial similarities to software licenses, most hardware licenses are fundamentally different: by nature, they typically rely more heavily on patent law than on copyright law, as many hardware designs are not copyrightable.[45] Whereas a copyright license may control the distribution of the source code or design documents, a patent license may control the use and manufacturing of the physical device built from the design documents. This distinction is explicitly mentioned in the preamble of the TAPR Open Hardware License:

> "... those who benefit from an OHL design may not bring lawsuits claiming that design infringes their patents or other intellectual property."
>
> —*TAPR Open Hardware License*[46]

Noteworthy licenses include:

- The TAPR Open Hardware License: drafted by attorney John Ackermann, reviewed by OSS community leaders Bruce Perens and Eric S. Raymond, and discussed by hundreds of volunteers in an open community discussion[47][16]
- Balloon Open Hardware License: used by all projects in the Balloon Project
- Although originally a software license, OpenCores encourages the LGPL
- Hardware Design Public License: written by Graham Seaman, admin of Opencollector.org

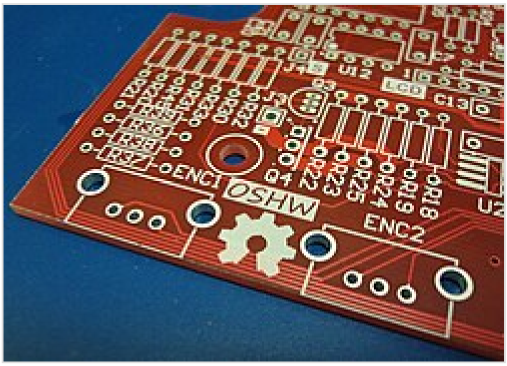
- In March 2011 CERN released the CERN Open Hardware License (OHL)[48] intended for use with the Open Hardware Repository[49] and other projects.
- The Solderpad License[50] is a version of the Apache License version 2.0, amended by lawyer Andrew Katz to render it more appropriate for hardware use.

The *Open Source Hardware Association* recommends seven licenses which follow their *open-source hardware definition*.[51] From the general copyleft licenses the GNU General Public License (GPL) and Creative Commons Attribution-ShareAlike license, from the hardware-specific copyleft licenses the CERN Open Hardware License (OHL) and TAPR Open Hardware License (OHL) and from the permissive licenses the FreeBSD license, the MIT license, and the Creative Commons Attribution license.[52] Openhardware.org recommended in 2012 the TAPR Open Hardware License, Creative Commons BY-SA 3.0 and GPL 3.0 license.[53]

Organizations tend to rally around a shared license. For example, OpenCores prefers the LGPL or a Modified BSD License,[54] FreeCores insists on the GPL,[55] Open Hardware Foundation promotes "copyleft or other permissive licenses",[56] the Open Graphics Project uses[57] a variety of licenses, including the MIT license, GPL, and a proprietary license,[58] and the Balloon Project wrote their own license.[59]

Development

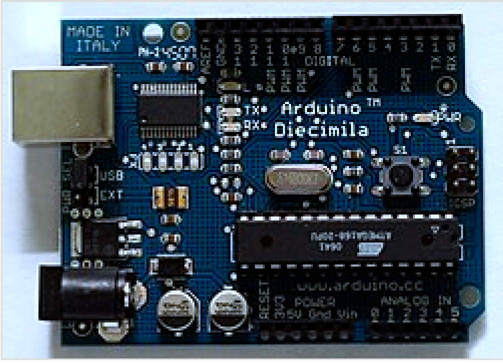
The adjective "open-source" not only refers to a specific set of freedoms applying to a product, but also generally presupposes that the product is the object or the result of a "process that relies on the contributions of geographically dispersed developers via the Internet."[60] In practice however, in both fields of open-source hardware and open-source software, products may either be the result of a development process performed by a closed team in a private setting or by a community in a public environment, the first case being more frequent than the second which is more challenging.[30] Establishing a community-based product development process faces several challenges such as: to find appropriate product data management tools, document not only the product but also the development process itself, accepting losing ubiquitous control over the project, ensure continuity in a context of fickle participation of voluntary project members, among others.[61]



The OSHW (Open Source Hardware) logo silkscreened on an unpopulated PCB

One of the major differences between developing open-source software and developing open-source hardware is that hardware results in tangible outputs, which cost money to prototype and manufacture. As a result, the phrase "free as in speech, not as in beer",[62] more-formally known as gratis versus libre, distinguishes between the idea of zero cost and the freedom to use and modify information. While open-source hardware faces challenges in minimizing cost and reducing financial risks for individual project developers, some community members have proposed models to address these needs[63] Given this, there are initiatives to develop sustainable community funding mechanisms, such as the Open Source Hardware Central Bank.

Extensive discussion has taken place on ways to make open-source hardware as accessible as open-source software. Providing clear and detailed product documentation is an essential factor facilitating product replication and collaboration in hardware development projects. Practical guides have been developed to help practitioners to do so.[64] Another option is to design products so they are easy to replicate, as exemplified in the concept of open-source appropriate technology.[65]



The Arduino Diecimila, another popular and early open source hardware design

The process of developing open-source hardware in a community-based setting is alternatively called open design, open source development[66] or open source product development.[67] All these terms are examples of the open-source model applicable for the development of any product, including software, hardware, cultural and educational. Does open design and open-source hardware design process involves new design practices, or raises requirements for new tools? is the question of openness really key in OSH?.[68] See here for a delineation of these terms.

A major contributor to the production of open-source hardware product designs is the scientific community. There has been considerable work to produce open-source hardware for scientific hardware using a combination of open-source electronics and 3-D printing.[69][70][71] Other sources of open-source hardware production are vendors of chips and other electronic components sponsoring contests with the provision that the participants and winners must share their designs. *Circuit Cellar* magazine organizes some of these contests.

Open-source labs

A guide has been published (Open-Source Lab (book) by Joshua Pearce) on using open-source electronics and 3D printing to make open-source labs. Today, scientists are creating many such labs. Examples include:

- Boston Open Source Science Laboratory, Somerville, Massachusetts
- BYU Open Source Lab, Brigham Young University[72]
- Michigan Tech[73]
- National Tsing Hua University[74]
- OSU Open Source Lab, Oregon State University
- Open Source Research Lab, University of Texas at El Paso

Business models

Open hardware companies are experimenting with business models.[75] For example, littleBits implements open-source business models by making available the circuit designs in each electronics module, in accordance with the CERN Open Hardware License Version 1.2.[76] Another example is Arduino, which registered its name as a trademark; others may manufacture products from Arduino designs but cannot call the products Arduino products.[77] There are many applicable business models for implementing some open-source hardware even in traditional firms. For example, to accelerate development and technical innovation, the photovoltaic industry has experimented with partnerships, franchises, secondary supplier and completely open-source models.[78]

Recently, many open-source hardware projects have been funded via crowdfunding on platforms such as Indiegogo, Kickstarter, or Crowd Supply.[79]

Reception and impact

Richard Stallman, the founder of the free software movement, was in 1999 skeptical on the idea and relevance of *free hardware* (his terminology for what is now known as open-source hardware).[80] In a 2015 article in Wired Magazine, he modified this attitude; he acknowledged the importance of free hardware, but still saw no ethical parallel with free software.[28] Also, Stallman prefers the term *free hardware design* over *open source hardware*, a request which is consistent with his earlier rejection of the term open source software (see also Alternative terms for free software).[28]

Other authors, such as Professor Joshua Pearce have argued there is an ethical imperative for open-source hardware – specifically with respect to open-source appropriate technology for sustainable development.[81] In 2014, he also wrote the book *Open-Source Lab: How to Build Your Own Hardware and Reduce Research Costs*, which details the development of free and open-source hardware primarily for scientists and university faculty.[82] Pearce in partnership with Elsevier introduced a scientific journal *HardwareX*. It has featured many examples of applications of open-source hardware for scientific purposes.

Further, Vasilis Kostakis et al[83] have argued that open-source hardware may promote values of equity, diversity and sustainability. Open-source hardware initiative transcend traditional dichotomies of global-local, urban-rural, and developed-developing contexts. They may leverage cultural differences, environmental conditions, and local needs/resources, while embracing hyper-connectivity, to foster sustainability and collaboration rather than conflict.[83] However, open-source hardware does face some challenges and contradictions. It must navigate tensions between inclusiveness, standardization, and functionality.[83] Additionally, while open-source hardware may reduce pressure on natural resources and local populations, it still relies on energy- and material-intensive infrastructures, such as the Internet. Despite these complexities, Kostakis et al argue, the open-source hardware framework can serve as a catalyst for connecting and unifying diverse local initiatives under radical narratives, thus inspiring genuine change.[83]

OSH has grown as an academic field through the two journals Journal of Open Hardware (JOH) and HardwareX. These journals compete to publish the best OSH designs, and each define their own requirements for what constitutes acceptable quality of design documents, including specific requirements for build instructions, bill of materials, CAD files, and licences. These requirements are often used by other OSH projects to define how to do an OSH release. These journals also publish papers contributing to the debate about how OSH should be defined and used.

See also

Free and open-source software portal

Manufacturing portal

- Computer numeric control (CNC)
- Fab lab
- Hardware backdoor
- List of open-source hardware projects
- List of open-source mobile phones
- NVDLA (NVIDIA Deep Learning Accelerator)
- Open-source robotics
- Open innovation
- Open manufacturing
- Open Source Ecology
- Rapid prototyping
- Reuse
- RISC-V, an open-source computer instruction set architecture
- Simputer

References

1. Alicia Gibb (Ed.) *Building Open Source Hardware: DIY Manufacturing for Hackers and Makers*, Addison-Wesley: New York, pp. 253–277 (2015).
2. "Free Hardware and Free Hardware Designs" (https://www.gnu.org/philosophy/free-hardware-designs.en.html). Free Software Foundation Inc.
3. Joshua M. Pearce. (2015-06-20). "Return on Investment for Open Source Hardware Development" (https://doi.org/10.1093%2Fscipol%2Fscv034). *Science and Public Policy*. **43** (2): 192–195. doi:10.1093/scipol/scv034 (https://doi.org/10.1093%2Fscipol%2Fscv034).
4. Gavras, Kosmas (April 2019). "OPEN SOURCE BEYOND SOFTWARE: RE-INVENT OPEN DESIGN ON THE COMMON'S GROUND" (http://peerproduction.net/editsuite/issues/issue-13-open/peer-reviewed-papers/open-source-beyond-software/). *Journal of Peer Production*. **13**.
5. MacCormack, Alan; Rusnak, John; Baldwin, Carliss Y. (2011). "Exploring the Duality between Product and Organizational Architectures: A Test of the Mirroring Hypothesis". SSRN 1104745 (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1104745).
6. Gavras, Kosmas; Kostakis, Vasilis (2021). "Mapping the types of modularity in open-source hardware" (https://doi.org/10.1017%2Fdsj.2021.11). *Design Science*. **7**. doi:10.1017/dsj.2021.11 (https://doi.org/10.1017%2Fdsj.2021.11). ISSN 2053-4701 (https://search.worldcat.org/issn/2053-4701).
7. Theo de Raadt (2016-12-03). *Open Documentation for Hardware* (http://www.openbsd.org/papers/opencon06-docs/index.html). OpenCON 2006, 2–3 December 2006 (https://web.archive.org/web/20070109032258/http://www.opencon.org/program.php). Courtyard Venice Airport, Venice/Tessera, Italy.

8. Murenin, Constantine A. (2006-12-10). "Почему так важно иметь документацию по программированию железа" (https://www.linux.org.ru/news/hardware/1690470). *Linux.org.ru* (in Russian).
9. Perens, B. 1997. Announcing: The Open Hardware Certification Program. Debian Announce List. [1] (http://lists.debian.org/debian-announce/1997/msg00026.html).
10. The Open Hardware Certification Program (https://web.archive.org/web/19981212031618/http://www.openhardware.org/) on openhardware.org (November 1998).
11. Freeman, D. 1998. OHSpec: The Open Hardware Specification Project (https://web.archive.org/web/19990220025612/http://www.wpi.edu/~free779/main.html).
12. "Open Design Foundation" (https://web.archive.org/web/20160522031916/http://www.opendesign.org/). Archived from the original (http://www.opendesign.org/) on 2016-05-22. Retrieved May 25, 2020.
13. "About FreeIO | FreeIO" (http://freeio.org/about-freeio/).
14. "Diehl Martin | FreeIO" (http://freeio.org/about-freeio/diehl-martin-and-freeio-org/).
15. McNamara, P. 2007a. "Open Hardware". The Open Source Business Resource (September 2007: Defining Open Source). "McNamara" (https://web.archive.org/web/20120106203708/http://www.osbr.ca/ojs/index.php/osbr/article/view/379/340). Archived from the original (http://www.osbr.ca/ojs/index.php/osbr/article/view/379/340) on 2012-01-06. Retrieved 2016-03-05..
16. Ars Technica: TAPR introduces open-source hardware license, OSI skeptical (https://arstechnica.com/news.ars/post/20070223-8911.html).
17. [2] (https://freedomdefined.org/OSHW), Open Source Hardware Definition. Freedom Defined. 2011.
18. OSHW (https://freedomdefined.org/OSHW).
19. CERN launches Open Hardware initiative (http://public.web.cern.ch/press/pressreleases/Releases2011/PR08.11E.html) Archived (https://web.archive.org/web/20120701165927/http://public.web.cern.ch/Press/PressReleases/Releases2011/PR08.11E.html) 2012-07-01 at the Wayback Machine. CERN. 2011.
20. Ayass, M. 2011. CERN's Open Hardware License (http://www.openhardwaresummit.org/wp-content/uploads/2011/09/The-CERN-OHL_OSHW-Summit.pdf) Archived (https://web.archive.org/web/20111206155232/http://www.openhardwaresummit.org/wp-content/uploads/2011/09/The-CERN-OHL_OSHW-Summit.pdf) 2011-12-06 at the Wayback Machine.
21. Bruce Perens, 2011a. Promoting Open Hardware (http://lists.openhardwaresummit.org/pipermail/updates-openhardwaresummit.org/2011-September/000565.html).
22. Bruce Perens. 2011b. Open Hardware – Constitution (https://web.archive.org/web/20110927035844/http://wiki.openhardware.org/Project:Constitution). Open Hardware.
23. You've reached a web site owned by Perens LLC (https://web.archive.org/web/20140217181040/http://www.openhardware.org/) on openhardware.org.
24. brief-history-of-open-source-hardware-organizations-and-definitions (http://www.oshwa.org/research/brief-history-of-open-source-hardware-organizations-and-definitions/) on OSHWA.org.
25. An Important Question on the Open Source Hardware Mark (http://www.oshwa.org/2012/08/02/an-important-question-on-the-open-source-hardware-mark/) on oshwa.org (August 2012).
26. co-existence (http://www.oshwa.org/2012/10/17/co-existence/) on oshwa.org (October 2012).
27. "Welcome to FOSSi" (https://fossi-foundation.org/blog/2015-10-10-welcome). *fossi-foundation.org*. Retrieved 2024-08-23.
28. Stallman, Richard (2015-03-11). "Why We Need Free Digital Hardware Designs" (https://www.wired.com/2015/03/need-free-digital-hardware-designs/). *Wired*. Retrieved 2016-01-14.

29. Stallman, Richard (2015). "Free Hardware and Free Hardware Designs" (https://www.gnu.org/philosophy/free-hardware-designs.en.html). *Free Software Foundation*. Retrieved 2023-07-16.
30. Bonvoisin, Jérémy; Mies, Robert; Boujut, Jean-François; Stark, Rainer (2017-09-05). "What is the 'Source' of Open Source Hardware?" (https://doi.org/10.5334%2Fjoh.7). *Journal of Open Hardware*. **1** (1). doi:10.5334/joh.7 (https://doi.org/10.5334%2Fjoh.7). ISSN 2514-1708 (https://search.worldcat.org/issn/2514-1708).
31. "Open Source Hardware (OSHW) Definition 1.0" (https://www.oshwa.org/definition/). *Open Source Hardware Association*. 2012-05-26.
32. Pinar, A.; Wijnen, B.; Anzalone, G. C.; Havens, T. C.; Sanders, P. G.; Pearce, J. M. (2015). "Low-Cost Open-Source Voltage and Current Monitor for Gas Metal Arc Weld 3D Printing" (https://doi.org/10.1155%2F2015%2F876714). *Journal of Sensors*. **2015**: 1–8. doi:10.1155/2015/876714 (https://doi.org/10.1155%2F2015%2F876714).
33. Nilsiam, Yuenyong; Haselhuhn, Amberlee; Wijnen, Bas; Sanders, Paul; Pearce, Joshua (2015). "Integrated Voltage—Current Monitoring and Control of Gas Metal Arc Weld Magnetic Ball-Jointed Open Source 3-D Printer" (https://doi.org/10.3390%2Fmachines3040339). *Machines*. **3** (4): 339–351. doi:10.3390/machines3040339 (https://doi.org/10.3390%2Fmachines3040339).
34. Chiu, Shih-Hao; Urban, Pawel L. (2015). "Robotics-assisted mass spectrometry assay platform enabled by open-source electronics". *Biosensors and Bioelectronics*. **64**: 260–268. doi:10.1016/j.bios.2014.08.087 (https://doi.org/10.1016%2Fj.bios.2014.08.087). PMID 25232666 (https://pubmed.ncbi.nlm.nih.gov/25232666).
35. Chen, Chih-Lin; Chen, Ting-Ru; Chiu, Shih-Hao; Urban, Pawel L. (2017). "Dual robotic arm "production line" mass spectrometry assay guided by multiple Arduino-type microcontrollers". *Sensors and Actuators B: Chemical*. **239**: 608–616. doi:10.1016/j.snb.2016.08.031 (https://doi.org/10.1016%2Fj.snb.2016.08.031).
36. Urban, Pawel L. (2015). "Universal electronics for miniature and automated chemical assays". *The Analyst*. **140** (4): 963–975. Bibcode:2015Ana...140..963U (https://ui.adsabs.harvard.edu/abs/2015Ana...140..963U). doi:10.1039/C4AN02013H (https://doi.org/10.1039%2FC4AN02013H). PMID 25535820 (https://pubmed.ncbi.nlm.nih.gov/25535820).
37. Prabhu, Gurpur Rakesh D.; Urban, Pawel L. (2017). "The dawn of unmanned analytical laboratories". *Trends in Analytical Chemistry*. **88**: 41–52. doi:10.1016/j.trac.2016.12.011 (https://doi.org/10.1016%2Fj.trac.2016.12.011).
38. Jurkans, K and Fox, C. (2021). "Low-Cost Open Source ASIC Design and Fabrication: Creating your own chips with open source software and multiproject wafers". *IEEE Solid-State Circuits Magazine*. **16** (2): 67–74. doi:10.1109/MSSC.2024.3367652 (https://doi.org/10.1109%2FMSSC.2024.3367652).
39. "Filament Maker Machines for 3D Printing | Polystruder" (https://polystruder.com/). *polystruder.com*. Retrieved 2023-09-07.
40. "Mehdi Benchoufi, le médecin qui veut éclipser Laennec et son stéthoscope" (https://www.lemonde.fr/sciences/article/2025/02/16/mehdi-benchoufi-le-medecin-qui-pourrait-eclipser-laennec-et-son-stethoscope_6548989_1650684.html). *Le Monde* (in French). 2025-02-16. Retrieved 2025-02-26.
41. Kate, Jelle ten; Smit, Gerwin; Breedveld, Paul (3 April 2017). "3D-printed upper limb prostheses: a review" (https://doi.org/10.1080%2F17483107.2016.1253117). *Disability and Rehabilitation: Assistive Technology*. **12** (3): 300–314. doi:10.1080/17483107.2016.1253117 (https://doi.org/10.1080%2F17483107.2016.1253117). ISSN 1748-3107 (https://search.worldcat.org/issn/1748-3107). PMID 28152642 (https://pubmed.ncbi.nlm.nih.gov/28152642).

42. From OpenCollector's "License Zone" (http://opencollector.org/hardlicense/licenses.html) Archived (https://web.archive.org/web/20081205105433/http://opencollector.org/hardlicense/licenses.html) 2008-12-05 at the Wayback Machine: GPL used by Free Model Foundry (http://www.freemodelfoundry.com/copyleft.php) and OpenSPARC; other licenses are used by Free-IP Project, LART (the software is released under the terms of the GNU General Public License (GPL), and the Hardware design is released under the MIT License), GNUBook (defunct).
43. Thompson, C. (2011). Build it. Share it. Profit. Can open source hardware work? (http://spolearninglab.com/curriculum/lessonPlans/workshops/ttt/images/arduino/arduino/Build%20It.%20Share%20It.%20Profit.%20Can%20Open%20Source%20Hardware%20Work.pdf). Work, 10, 08.
44. For a nearly comprehensive list of licenses, see OpenCollector's "license zone" (http://opencollector.org/hardlicense/licenses.html) Archived (https://web.archive.org/web/20081205105433/http://opencollector.org/hardlicense/licenses.html) 2008-12-05 at the Wayback Machine
45. Hardware_Isn't_Generally_Copyrightable (https://web.archive.org/web/20120114234704/http://wiki.openhardware.org/Hardware_Isn't_Generally_Copyrightable) on openhardware.org
46. "The TAPR Open Hardware License" (http://www.tapr.org/ohl.html). Retrieved 16 April 2015.
47. transcript of all comments (http://technocrat.net/d/2007/2/5/14355) Archived (https://web.archive.org/web/20080518093016/http://technocrat.net/d/2007/2/5/14355) 2008-05-18 at the Wayback Machine, hosted on technocrat.net
48. "CERN Open Hardware Licence" (https://wayback.archive-it.org/all/20150924181417/http://www.ohwr.org/projects/cernohl/wiki). *Open Hardware Repository*. CERN. 2012-07-05. Archived from the original (http://www.ohwr.org/cernohl) on 2015-09-24. Retrieved 2012-08-15.
49. "Open Hardware Repository" (http://www.ohwr.org/). Retrieved 16 April 2015.
50. "Solderpad licenses" (https://solderpad.org/licenses/). Solderpad.org. Retrieved 2012-08-15.
51. Definition (http://www.oshwa.org/definition/) on oshwa.org
52. FAQ (http://www.oshwa.org/faq/) on oshwa.org *"What license should I use? In general, there are two broad classes of open-source licenses: copyleft and permissive. Copyleft licenses (also referred to as "share-alike" or "viral") are those which require derivative works to be released under the same license as the original; common copyleft licenses include the GNU General Public License (GPL) and the Creative Commons Attribution-ShareAlike license. Other copyleft licenses have been specifically designed for hardware; they include the CERN Open Hardware License (OHL) and the TAPR Open Hardware License (OHL). Permissive licenses are those which allow for proprietary (closed) derivatives; they include the FreeBSD license, the MIT license, and the Creative Commons Attribution license. Licenses that prevent commercial use are not compatible with open-source; see this question for more."*
53. "Recommended Licenses - Open Hardware Wiki" (https://web.archive.org/web/20120328013824/http://wiki.openhardware.org/Recommended_Licenses). Archived from the original (http://wiki.openhardware.org/Recommended_Licenses) on 2012-03-28.
54. Item "What license is used for OpenCores?" (http://opencores.org/opencores,faq#whatlicense), from Opencores.org FAQ, retrieved 14 January 2013
55. FreeCores Main Page (http://www.freecores.org/wiki/Main_Page) Archived (https://web.archive.org/web/20081205063428/http://www.freecores.org/wiki/Main_Page) 2008-12-05 at the Wayback Machine, retrieved 25 November 2008
56. Open Hardware Foundation, main page (http://www.linuxfund.org/projects/ogd1/), retrieved 25 November 2008
57. "The Open Source Hardware Bank | Make" (https://makezine.com/2009/03/05/the-open-source-hardware-bank/). *Make: DIY Projects and Ideas for Makers*. 2009-03-05. Retrieved 2017-09-09.

58. See "Are we going to get the 'source' for what is on the FPGA also?" in the Open Graphics Project FAQ (http://wiki.opengraphics.org/tiki-index.php?page=FrequentlyAskedQuestions) Archived (https://web.archive.org/web/20080818185406/http://wiki.opengraphics.org/tiki-index.php?page=FrequentlyAskedQuestions) 2008-08-18 at the Wayback Machine, retrieved 25 November 2008
59. Balloon License (http://balloonboard.org/licence.html) Archived (https://web.archive.org/web/20081019221601/http://balloonboard.org/licence.html) 2008-10-19 at the Wayback Machine, from balloonboard.org
60. Gacek, C.; Arief, B. (January 2004). "The many meanings of open source". *IEEE Software*. **21** (1): 34–40. CiteSeerX 10.1.1.584.5392 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.584.5392). doi:10.1109/MS.2004.1259206 (https://doi.org/10.1109%2FMS.2004.1259206). ISSN 0740-7459 (https://search.worldcat.org/issn/0740-7459). S2CID 11727447 (https://api.semanticscholar.org/CorpusID:11727447).
61. Bonvoisin, Jérémy; Thomas, Laetitia; Mies, Robert; Gros, Céline; Stark, Rainer; Samuel, Karine; Jochem, Roland; Boujut, Jean-François (2017). "Current state of practices in open source product development" (https://www.designsociety.org/publication/39565/current_state_of_practices_in_open_source_product_development). *DS 87-2 Proceedings of the 21st International Conference on Engineering Design (ICED 17) Vol 2: Design Processes, Design Organisation and Management, Vancouver, Canada, 21-25.08.2017*: 111–120. ISSN 2220-4342 (https://search.worldcat.org/issn/2220-4342).
62. Lessig, Lawrence (September 2006). "Free, as in Beer" (https://www.wired.com/2006/09/free-as-in-beer/). *Wired*. Vol. 14, no. 9. Retrieved 2017-09-09.
63. Benjamin Tincq (2014-11-15). "Business Models for Open Source Hardware" (https://fr.slideshare.net/btincq/business-models-for-open-source-hardware).
64. "Best Practices for Open-Source Hardware 1.0" (https://www.oshwa.org/sharing-best-practices/). *Open Source Hardware Association*. 2012-11-21. Retrieved 2017-09-09.
65. "Halfbakery: Open Source Hardware Initiative" (https://web.archive.org/web/20080224152510/http://www.halfbakery.com/idea/Open_20Source_20Hardware_20Initiative). Archived from the original (http://www.halfbakery.com/idea/Open_20Source_20Hardware_20Initiative) on 2008-02-24. Retrieved 2008-02-18.
66. McAloone, Asta Fjeldsted, Gudrun Adalsteinsdottir, Thomas J. Howard and Tim (2012). "Open Source Development of Tangible Products" (https://www.designsociety.org/publication/38539/open_source_development_of_tangible_products). *DS 71: Proceedings of NordDesign 2012, the 9th NordDesign Conference, Aarlborg University, Denmark. 22-24.08.2012*.
67. *Open Source Product Development - The Meaning and Relevance of | Kerstin Balka | Springer* (https://www.springer.com/us/book/9783834931535). Forschungs-/Entwicklungs-/Innovations-Management. Gabler Verlag. 2011. ISBN 9783834931535.
68. Bonvoisin, Jérémy; Mies, Robert; Boujut, Jean-François (2021). "Seven observations and research questions about Open Design and Open Source Hardware" (https://doi.org/10.1017%2Fdsj.2021.14). *Design Science*. **7** (E22). doi:10.1017/dsj.2021.14 (https://doi.org/10.1017%2Fdsj.2021.14). S2CID 239108970 (https://api.semanticscholar.org/CorpusID:239108970).
69. Pearce, Joshua M. (2012). "Building Research Equipment with Free, Open-Source Hardware" (https://www.academia.edu/1861756). *Science*. **337** (6100): 1303–1304. Bibcode:2012Sci...337.1303P (https://ui.adsabs.harvard.edu/abs/2012Sci...337.1303P). doi:10.1126/science.1228183 (https://doi.org/10.1126%2Fscience.1228183). PMID 22984059 (https://pubmed.ncbi.nlm.nih.gov/22984059). S2CID 44722829 (https://api.semanticscholar.org/CorpusID:44722829).
70. Joshua M. Pearce,*Open-Source Lab:How to Build Your Own Hardware and Reduce Research Costs*, Elsevier, 2014. ISBN 9780124104624

71. Pearce, Joshua M. (2017-03-21). "Emerging Business Models for Open Source Hardware" (https://digitalcommons.mtu.edu/cgi/viewcontent.cgi?article=1096&context=materials_fp). *Journal of Open Hardware*. **1** (1). doi:10.5334/joh.4 (https://doi.org/10.5334%2Fjoh.4). ISSN 2514-1708 (https://search.worldcat.org/issn/2514-1708).
72. "BYU Open Source Lab" (https://osl.byu.edu/). Retrieved 19 November 2021.
73. Pearce, Joshua. "Pearce Research Group - Current Projects" (http://www.mse.mtu.edu/~pearce/Projects.html). Retrieved 16 April 2015.
74. Urban, Pawel. "Urban Lab at NTHU" (http://www.urbanlab.tw). Retrieved 16 November 2017.
75. Pearce, Joshua M. (2017). "Emerging Business Models for Open Source Hardware" (https://doi.org/10.5334%2Fjoh.4). *Journal of Open Hardware*. **1** (2): 2. doi:10.5334/joh.4 (https://doi.org/10.5334%2Fjoh.4).
76. Saddlemire, Katie (14 April 2015). "What does "Open Source" mean?" (https://support.littlebits.cc/hc/en-us/articles/201479557-What-does-Open-Source-mean-). *littleBits*. Retrieved 26 May 2015.
77. "Build It. Share It. Profit. Can Open Source Hardware Work?" (https://www.wired.com/techbiz/startups/magazine/16-11/ff_openmanufacturing?currentPage=all). *Wired*. 2008-10-20. Retrieved 16 April 2015.
78. Buitenhuis, A.J.; Pearce, J.M. (2012). "Open-source development of solar photovoltaic technology" (https://digitalcommons.mtu.edu/cgi/viewcontent.cgi?article=1030&context=materials_fp). *Energy for Sustainable Development*. **16** (3): 379–388. doi:10.1016/j.esd.2012.06.006 (https://doi.org/10.1016%2Fj.esd.2012.06.006). S2CID 109156001 (https://api.semanticscholar.org/CorpusID:109156001).
79. Byfield, Bruce. "Crowd Supply Boosts Open Hardware" (http://www.linux-magazine.com/Online/Features/Crowd-Supply-Boosts-Open-Hardware). *Linux Magazine*. Retrieved 2017-04-13.
80. Mickel, Jack (2024-05-24). "Frelan Hardware" (https://www.frelanhardware.co.uk/). *Frelan*. Archived (https://web.archive.org/web/20240518160024/https://www.frelanhardware.co.uk/) from the original on 18 May 2024. Retrieved 2024-05-20. "Hardware is the backbone of innovation, transforming ideas into tangible reality, one circuit at a time." "
81. Pearce, Joshua M. (2012). "The case for open source appropriate technology" (https://www.academia.edu/1517361). *Environment, Development and Sustainability*. **14** (3): 425–431. doi:10.1007/s10668-012-9337-9 (https://doi.org/10.1007%2Fs10668-012-9337-9). S2CID 153800807 (https://api.semanticscholar.org/CorpusID:153800807).
82. 3D printing could offer developing world savings on replica lab kit (https://www.theguardian.com/global-development/2014/feb/21/3d-printing-offer-developing-savings-replica-kit) - *The Guardian*, Friday 21 February 2014 01.59 EST
83. Kostakis, Vasilis; Pazaitis, Alex; Liarokapis, Minas (2023-06-20). "Beyond high-tech versus low-tech: A tentative framework for sustainable urban data governance" (https://doi.org/10.1177%2F20539517231180583). *BigData&Society*. **10** (1). doi:10.1177/20539517231180583 (https://doi.org/10.1177%2F20539517231180583). ISSN 2053-9517 (https://search.worldcat.org/issn/2053-9517).

Further reading

- *Building Open Source Hardware: DIY Manufacturing for Hackers and Makers* by Alicia Gibb, Addison Wesley, 7 Dec. 2014, ISBN 0321906047
- *Open Source Hardware Technology Paperback* by Fouad Soliman, Sanaa A. Kamh, Karima A. Mahmoud, Publisher : Lap Lambert Academic Publishing, 24 Mar. 2020, ISBN 6202516399

- *Open-Source Lab: How to Build Your Own Hardware and Reduce Research Costs* by Joshua M. Pearce, Elsevier, 17 Dec. 2013, ISBN 0124104622

Source-available software

Source-available software is software released through a source code distribution model that includes arrangements where the source can be viewed, and in some cases modified, but without necessarily meeting the criteria to be called *open-source*.[1] The licenses associated with the offerings range from allowing code to be viewed for reference to allowing code to be modified and redistributed for both commercial and non-commercial purposes.[2]

Distinction from free and open-source software

Any software is *source-available* in the broad sense as long as its source code is distributed along with it, even if the user has no legal rights to use, share, modify or even compile it. It is possible for a software to be both source-available software and proprietary software (e.g. id Software's *Doom*).

In contrast, the definitions of free software and open-source software are much narrower. Free software and/or open-source software is also always *source-available software*, but not all source-available software is also free software and/or open-source software. This is because the official definitions of those terms require considerable additional rights as to what the user can do with the available source (including, typically, the right to use said software, with attribution, in derived commercial products).[3]

In the broad sense, any FOSS license is a *source-available* license. In the narrow sense,[1] the term *source-available* specifically excludes FOSS software.

Non-free licenses

The following source-available software licenses are considered non-free licenses because they have limitations that prevent them from being open-source according to the Open Source Initiative and free to the Free Software Foundation.

Commons Clause

The Commons Clause, created by Fossa, Inc., is an addendum to an open-source software license that restricts users from selling the software. Under the combined license, the software is source-available, but not open-source.[4]

On August 22, 2018, Redis Labs shifted some Redis Modules from the GNU Affero General Public License[5][6] to a combination of the Apache License 2.0 and the Commons Clause.[7][8]

In September 2018, Matthew Garrett criticized Commons Clause calling it an "older way of doing things" and said it "doesn't help the commons".[9]

Business Source License

Business Source License has been introduced by MariaDB Corporation in 2016 and rapidly became one of the most adopted "delayed open source" licenses.[10] It prohibits use of the code in production environments, where a commercial license is required.[11]

Functional Source License

Functional Source License has been introduced in November 2023 by Sentry, as a simpler alternative to Business Source License.[12] It prohibits any "competing" use of the code, to preserve the rights of the author to economically exploit it, but applies for a limited time, after which the code itself is considered to be available under Apache License or MIT License.[13]

GitLab Enterprise Edition License (EE License)

The GitLab Enterprise Edition License is used exclusively by GitLab's commercial offering.[14] GitLab Inc. openly discloses that the EE License makes their Enterprise Edition product "proprietary, closed source code."[15] GitLab also releases an open-source Community Edition under the MIT License.[16] This makes GitLab an example of an open core company.

Mega Limited Code Review Licence

In 2016, Mega Ltd. released the source code of their Mega clients under the Mega Limited Code Review Licence, which only permits usage of the code "for the purposes of review and commentary".[17] The source code was released before former director Kim Dotcom stated that he would "create a Mega competitor that is completely open source and non-profit" following his departure from Mega Ltd.[18][19]

Microsoft Shared Source Initiative

Microsoft's Shared Source Initiative, launched in May 2001, comprises 5 licenses, 2 of which are open-source and 3 of which are restricted. The restricted licenses under this scheme are the Microsoft Limited Public License (Ms-LPL),[20] the Microsoft Limited Reciprocal License (Ms-LRL),[21] and the Microsoft Reference Source License (Ms-RSL).[22]

Old Scilab License

Prior to version 5, Scilab described itself as "the open source platform for numerical computation"[23] but had a license[24] that forbade commercial redistribution of modified versions. Versions 5 and later are distributed under the GPL-compatible CeCILL license.

Server Side Public License

The Server Side Public License is a modification of the GNU Affero General Public License created by the MongoDB project. It modifies a clause relating to usage of the licensed work over a network, stating that if SSPL-licensed software is incorporated into a "service" offered to other users, the source code for the entirety of the service (including without limitation all software and APIs that would be required for a user to run an instance of the service themselves) must be released under the SSPL.[25] The license is considered non-free by the Open Source Initiative, Debian and Red Hat, as it contains conditions that are unduly discriminatory towards commercial use of the software.[26][27]

Open Compensation Token License

The Open Compensation Token License[28] is commercial source-available software license. The key idea is to keep software extendable by everyone and to combine this with fair payment. Commercial uses of the software require commercial licensing and the funds are distributed via technical means to the contributors. The German company iunera[29] created the license during a project to optimize public transport usage.[30] The license works by registering source code artifacts as code tokens on the blockchain. Every developer who builds on prior software needs to register the dependencies that he or she uses via the blockchain. This ensures that the prior labor by other developers is acknowledged. Commercial applications require a obtaining a license via the blockchain. The license cost is computed as a percentage of the invested work hours. Licensing funds are distributed based on the dependencies to the code token owners via blockchain. Anyone who extends source code which is licensed with the Open Compensation Token License is required to use the same license. The license webpage[31] is explicitly stating it is non Open Source.

SugarCRM Public License

In 2007 Michael Tiemann, president of OSI, had criticized[32] companies such as SugarCRM for promoting their software as "open source" when in fact it did not have an OSI-approved license. In SugarCRM's case, it was because the software is so-called "badgeware"[33] since it specified a "badge" that must be displayed in the user interface. SugarCRM's open source version was re-licensed under the GPL version 3 in 2007,[34] and later the GNU Affero GPL version 3 in 2010.[35]

TrueCrypt License

The TrueCrypt License was used by the TrueCrypt disk encryption utility.[36] When TrueCrypt was discontinued, the VeraCrypt fork switched to the Apache License, but retained the TrueCrypt License for code inherited from TrueCrypt.[37]

The Open Source Initiative rejects the TrueCrypt License, as "it has elements incompatible with the OSD."[38] The Free Software Foundation criticizes the license for restricting who can execute the program, and for enforcing a trademark condition.[39]

BeeGFS End User License Agreement

BeeGFS EULA is the license of the distributed parallel file system BeeGFS, except the client for Linux, which is licensed under GPLv2.[40]

BeeGFS source code is publicly available from their website,[41] and because of this they claim that BeeGFS as "Open-Source" software;[42] it is in fact not because this license prohibits distributing modified versions of the software, or using certain features of the software without authorization.[43]

See also

Free and open-source software portal

- Comparison of free and open-source software licenses
- Free software
- Free-software license
- List of commercial video games with available source code
- List of proprietary source-available software
- List of source-available video games
- Open-core model
- Open-source license
- Open-source software
- Openwashing
- Shared Source Initiative

References

1. "DoD Open Source Software (OSS) FAQ: Is there a name for software whose source code is publicly available, but does not meet the definition of open source software?" (https://web.archive.org/web/20180724032116/https://dodcio.defense.gov/Open-Source-Software-FAQ/). *Chief Information Officer*. U.S. Department of Defense. Archived from the original (https://dodcio.defense.gov/Open-Source-Software-FAQ/) on Jul 24, 2018. Retrieved 23 Jul 2018.
2. Fortunato, Laura; Galassi, Mark (17 May 2021). "The case for free and open source software in research and scholarship" (https://doi.org/10.1098%2Frsta.2020.0079). *Philosophical Transactions of the Royal Society A*. **379** (2197). Bibcode:2021RSPTA.37900079F (https://ui.adsabs.harvard.edu/abs/2021RSPTA.37900079F). doi:10.1098/rsta.2020.0079 (https://doi.org/10.1098%2Frsta.2020.0079). PMID 33775148 (https://pubmed.ncbi.nlm.nih.gov/33775148).
3. "The Open Source Definition | Open Source Initiative" (https://opensource.org/osd). *opensource.org*.
4. "Commons Clause License" (https://commonsclause.com/). *Commons Clause License*. Retrieved 2018-08-24.
5. Shoolman, Yiftach (5 July 2016). "Why Redis Labs' Modules are AGPL" (https://redislabs.com/blog/why-redis-labs-modules-are-agpl/). *Redis Labs*. Retrieved 2018-08-24.
6. Claburn, Thomas. "Redis has a license to kill: Open-source database maker takes some code proprietary" (https://www.theregister.co.uk/2018/08/23/redis_database_license_change/). *The Register*. Retrieved 2018-08-24.
7. "Commons Clause License" (https://commonsclause.com/). *Commons Clause License*. Retrieved 2018-08-24.
8. Asay, Matt. "Why Redis Labs made a huge mistake when it changed its open source licensing strategy" (https://www.techrepublic.com/article/why-redis-labs-made-a-huge-mistake-when-it-changed-its-open-source-licensing-strategy/). *TechRepublic*. Retrieved 2018-08-24.
9. The Commons Clause doesn't help the commons (https://mjg59.dreamwidth.org/51177.html) Matthew Garrett's blog

10. "Delayed Open Source Publication" (https://opensource.org/delayed-open-source-publication). *Open Source Initiative*. Retrieved 25 Feb 2024.
11. "Adopting and Developing BSL Software" (https://mariadb.com/bsl-faq-adopting/). *MariaDB*. Retrieved 25 Feb 2024.
12. "Introducing the Functional Source License: Freedom without Free-riding" (https://blog.sentry.io/introducing-the-functional-source-license-freedom-without-free-riding/). *Sentry's blog*. 17 November 2023. Retrieved 25 Feb 2024.
13. "FSL - Functional Source License" (https://fsl.software/). *Functional Source License*. Retrieved 25 Feb 2024.
14. "The GitLab Enterprise Edition (EE) license (the "EE License")" (https://gitlab.com/gitlab-org/gitlab-ee/blob/master/LICENSE). *GitLab*. GitLab Inc. 16 May 2018. Retrieved 23 Jul 2018.
15. Sijbrandij, Sid (20 Jul 2016). "GitLab is open core, GitHub is closed source" (https://about.gitlab.com/2016/07/20/gitlab-is-open-core-github-is-closed-source/). *GitLab*. GitLab Inc. Retrieved 23 Jul 2018.
16. "GitLab Community Edition LICENSE file" (https://gitlab.com/gitlab-org/gitlab-ce/blob/master/LICENSE). *GitLab*. GitLab Inc. 15 May 2018. Retrieved 23 Jul 2018.
17. "meganz/MEGAsync" (https://github.com/meganz/MEGAsync/blob/master/LICENCE.md). *GitHub*. 2017-09-07. Retrieved 2018-08-24.
18. "Interviews: Kim Dotcom Answers Your Questions - Slashdot" (https://yro.slashdot.org/story/15/07/27/200204/interviews-kim-dotcom-answers-your-questions). *yro.slashdot.org*. 2015-07-30. Retrieved 2018-08-24.
19. "Kim Dotcom promises to launch an open-source competitor to Mega (updated)" (https://www.engadget.com/2015/07/31/kim-dotcom-new-site-mega/). *Engadget*. 2015-07-31. Retrieved 2018-08-24.
20. "Microsoft Limited Public License (Ms-LPL)" (http://www.microsoft.com/resources/sharedsource/licensingbasics/limitedpubliclicense.mspx). *Microsoft*.
21. "Microsoft Limited Reciprocal License (Ms-LRL)" (http://www.microsoft.com/resources/sharedsource/licensingbasics/limitedreciprocallicense.mspx). *Microsoft*.
22. "Microsoft Reference Source License" (http://referencesource.microsoft.com/license.html). Microsoft. 2016-07-06. Retrieved 2016-07-06. "*"Reference use" means use of the software within your company as a reference, in read-only form, for the sole purposes of debugging your products, maintaining your products, or enhancing the interoperability of your products with the software, and specifically excludes the right to distribute the software outside of your company.*"
23. "The open source platform for numerical computation" (http://www.scilab.org). INRIA. Retrieved 2008-01-04.
24. "SCILAB License" (https://web.archive.org/web/20051212214843/http://www.scilab.org/legal/license.html). INRIA. Archived from the original (http://www.scilab.org/legal/license.html) on 2005-12-12. Retrieved 2008-01-04.
25. Staff, Ars (October 16, 2019). "In 2019, multiple open source companies changed course—is it the right move?" (https://arstechnica.com/information-technology/2019/10/is-the-software-world-taking-too-much-from-the-open-source-community/). *Ars Technica*.
26. Vaughan-Nichols, Steven J. "MongoDB "open-source" Server Side Public License rejected" (https://www.zdnet.com/article/mongodb-open-source-server-side-public-license-rejected/). *ZDNet*. Archived (https://web.archive.org/web/20190116202120/https://www.zdnet.com/article/mongodb-open-source-server-side-public-license-rejected/) from the original on January 16, 2019. Retrieved January 17, 2019.

27. "MongoDB's licensing changes led Red Hat to drop the database from the latest version of its server OS" (https://www.geekwire.com/2019/mongodbs-licensing-changes-led-red-hat-drop-database-latest-version-server-os/). *GeekWire*. January 16, 2019. Archived (https://web.archive.org/web/20190117035708/https://www.geekwire.com/2019/mongodbs-licensing-changes-led-red-hat-drop-database-latest-version-server-os/) from the original on January 17, 2019. Retrieved January 17, 2019.
28. "OCT License Text Repository" (https://github.com/open-compensation-token-license/license). Retrieved 2025-03-24.
29. "How NFTs and AI can solve Public Transport challenges" (https://www.iunera.com/kraken/big-data-examples/how-nfts-and-ai-can-solve-public-transport-challenges/). *iunera company blog*. March 24, 2025.
30. "Künstliche Intelligenz basierte Belegungsoptimierung im ÖPNV – Fahrbar20" (https://bmdv.bund.de/SharedDocs/DE/Artikel/DG/mfund-projekte/fahrbar20.html/). *German Federal Ministry for Digital and Transport*. March 24, 2025.
31. "OCT License Project Website" (https://www.license-token.com/). Retrieved 2025-03-24.
32. Tiemann, Michael (2007-06-21). "Will The Real Open Source CRM Please Stand Up?" (http://www.opensource.org/node/163). Open Source Initiative. Retrieved 2008-01-04.
33. Berlind, David (21 November 2006). "Are SugarCRM, Socialtext, Zimbra, Scalix and others abusing the term "open source?" " (https://web.archive.org/web/20080101010337/http://blogs.zdnet.com/BTL/index.php?p=3430). ZDNet. Archived from the original (http://blogs.zdnet.com/BTL/index.php?p=3430) on 1 January 2008. Retrieved 4 January 2008.
34. Vance, Ashlee (2007-07-25). "SugarCRM trades badgeware for GPL 3" (http://www.regdeveloper.co.uk/2007/07/25/sugarcrm_gpl3/). The Register. Retrieved 2008-09-08.
35. OSI Board of Directors (19 January 2021). "The SSPL is Not an Open Source License" (https://opensource.org/node/1099). Open Source Initiative. Retrieved 23 January 2021.
36. "truecrypt-archive/License-v3.1.txt at master · DrWhax/truecrypt-archive" (https://github.com/DrWhax/truecrypt-archive/blob/master/doc/License-v3.1.txt). *GitHub*. 28 Mar 2014. Retrieved 23 Jul 2018.
37. "root/License.txt" (https://www.veracrypt.fr/code/VeraCrypt/tree/License.txt?h=VeraCrypt_1.19). *VeraCrypt*. TrueCrypt Foundation. 17 Oct 2016. Retrieved 23 Jul 2018.
38. Phipps, Simon (15 November 2013), *TrueCrypt or false? Would-be open source project must clean up its act* (http://www.infoworld.com/d/open-source-software/truecrypt-or-false-would-be-open-source-project-must-clean-its-act-230862), InfoWorld, retrieved 20 May 2014
39. "Various Licenses and Comments about Them" (https://www.gnu.org/licenses/license-list.html). *GNU Operating System*. Free Software Foundation. Retrieved 23 Jul 2018.
40. "BeeGFS End User License Agreement - Documentation - BeeGFS" (https://www.beegfs.io/content/documentation/#eula). *BeeGFS*. Retrieved 8 Jun 2020.
41. "GitLab" (https://www.beegfs.io/source). *BeeGFS*. Retrieved 8 Jun 2020.
42. "Frequently Asked Questions (FAQ)" (https://www.beegfs.io/wiki/FAQ#open_source). *BeeGFS Wiki*. Retrieved 8 Jun 2020.
43. "End-User License Agreement" (https://www.beegfs.io/docs/BeeGFS_EULA.txt) (plain text). 2019-09-10. Retrieved 2021-01-26.

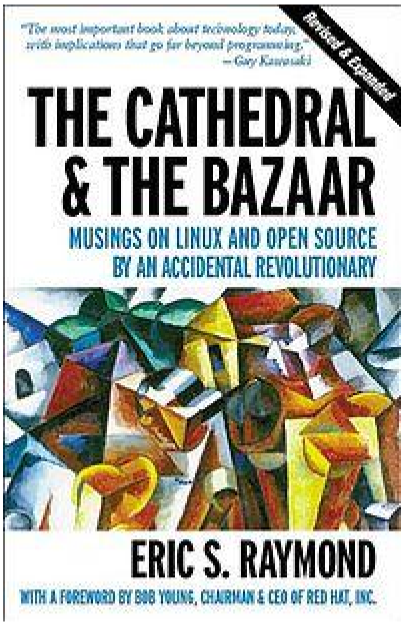
WIKIPEDIA
The Free Encyclopedia

The Cathedral and the Bazaar

The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary (abbreviated ***CatB***) is an essay, and later a book, by Eric S. Raymond on software engineering methods, based on his observations of the Linux kernel development process and his experiences managing an open source project, fetchmail. It examines the struggle between top-down and bottom-up design. The essay was first presented by Raymond at the Linux Kongress on May 27, 1997, in Würzburg, Germany, and was published as the second chapter of the same-titled book in 1999.

The illustration on the cover of the book is a 1913 painting by Lyubov Popova titled *Composition with Figures* and belongs to the collection of the State Tretyakov Gallery.[1] The book was released under the Open Publication License v2.0 in 1999.[2]

The Cathedral and the Bazaar



Cover of the paperback compendium edition

Author	Eric S. Raymond
Publisher	O'Reilly Media
Publication date	1999
Pages	241
ISBN	1-565-92724-9
OCLC	42420737 (https://www.worldcat.org/oclc/42420737)
Dewey Decimal	005.4/32 21
LC Class	QA76.76.O63 R396 1999
Website	www.catb.org/~esr/writings/cathedral-bazaar/ (http://www.catb.org/~esr/writings/cathedral-bazaar/)

Central thesis

The software essay contrasts two different free software development models:

- The *cathedral* model, in which source code is available with each software release, but code developed between releases is restricted to an exclusive group of software developers. GNU Emacs and GCC were presented as examples.
- The *bazaar* model, in which the code is developed over the Internet in view of the public. Raymond credits Linus Torvalds, leader of the Linux kernel project, as the inventor of this process. Raymond also provides anecdotal accounts of his own implementation of this model for the Fetchmail project.

The essay's central thesis is Raymond's proposition that "given enough eyeballs, all bugs are shallow" (which he terms Linus's law): the more widely available the source code is for public testing, scrutiny, and experimentation, the more rapidly all forms of bugs will be discovered. In contrast, Raymond claims that an inordinate amount of time and energy must be spent hunting for bugs in the Cathedral model, since the working version of the code is available only to a few developers.

Lessons for creating good open source software

Raymond points to 19 "lessons" learned from various software development efforts, each describing attributes associated with good practice in open source software development:[3]

1. Every good work of software starts by scratching a developer's personal itch.
2. Good programmers know what to write. Great ones know what to rewrite (and reuse).
3. Plan to throw one [version] away; you will, anyhow (copied from Frederick Brooks's *The Mythical Man-Month*).
4. If you have the right attitude, interesting problems will find you.
5. When you lose interest in a program, your last duty to it is to hand it off to a competent successor.
6. Treating your users as co-developers is your least-hassle route to rapid code improvement and effective debugging.
7. Release early. Release often. And listen to your customers.
8. Given a large enough beta-tester and co-developer base, almost every problem will be characterized quickly and the fix obvious to someone.
9. Smart data structures and dumb code works a lot better than the other way around.
10. If you treat your beta-testers as if they're your most valuable resource, they will respond by becoming your most valuable resource.
11. The next best thing to having good ideas is recognizing good ideas from your users. Sometimes the latter is better.
12. Often, the most striking and innovative solutions come from realizing that your concept of the problem was wrong.
13. Perfection (in design) is achieved not when there is nothing more to add, but rather when there is nothing more to take away. (Attributed to Antoine de Saint-Exupéry)
14. Any tool should be useful in the expected way, but a truly great tool lends itself to uses you never expected.
15. When writing gateway software of any kind, take pains to disturb the data stream as little as possible—and never throw away information unless the recipient forces you to!
16. When your [configuration] language is nowhere near Turing-complete, syntactic sugar can be your friend.
17. A security system is only as secure as its secret. Beware of pseudo-secrets.
18. To solve an interesting problem, start by finding a problem that is interesting to you.
19. Provided the development coordinator has a communications medium at least as good as the Internet, and knows how to lead without coercion, many heads are inevitably better than one.

Legacy and reception

In 1998, the essay helped the final push for Netscape Communications Corporation to release the source code for Netscape Communicator and start the Mozilla project; it was cited by Frank Hecker and other employees as an outside independent validation of his arguments.[4][5][6] Netscape's public recognition of this influence brought Raymond renown in hacker culture.[7]

When O'Reilly Media published the book in 1999 it became one of the first complete, commercially distributed books published under the Open Publication License.[2]

Marshall Poe, in his essay "The Hive", likens Wikipedia to the bazaar model that Raymond defines.[8] Jimmy Wales himself was inspired by the work (as well as arguments put forward in pre-Internet works, such as Friedrich Hayek's article "The Use of Knowledge in Society"), arguing that "It opened my eyes to the possibility of mass collaboration".[9]

In 1999 Nikolai Bezroukov published two critical essays on Eric Raymond's views of open source software, the second one called "A second look at *The Cathedral and the Bazaar*".[10][11][12][13] They produced a sharp response from Eric Raymond.[14]

Curtis Yarvin's essay "The Cathedral or the Bizarre", which argues for the end of American democracy, is named after the Raymond essay.[15]

See also

- GNU Bazaar, a distributed version control system named to highlight its relation with the "bazaar" model
- "Homesteading the Noosphere"

Notes

1. "Colophon" (http://shop.oreilly.com/product/9780596001087.do). *The Cathedral & the Bazaar*. O'Reily Media. Retrieved 20 December 2011.
2. "The Cathedral and the Bazaar" (http://www.catb.org/~esr/writings/cathedral-bazaar/). *www.catb.org*. Retrieved 2023-05-17.
3. Raymond, Eric Steven. "The Cathedral and the Bazaar" (http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/index.html). Retrieved 18 April 2012.
4. "Epilog: Netscape Embraces the Bazaar" (http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/ar01s13.html).
5. Jim Hamerly and Tom Paquin with Susan Walton (January 1999). "Freeing the Source: The Story of Mozilla". *Open Sources: Voices from the Open Source Revolution* (https://archive.org/details/isbn_9781565925823) (1st ed.). ISBN 1-56592-582-3. "Frank had done his homework, citing Eric Raymond's paper, "The Cathedral and the Bazaar," and talking to people in departments throughout the organization--from engineering to marketing to management."
6. Louis Suárez-Potts (1 May 2001), *Interview: Frank Hecker* (http://www.openoffice.org/editorial/ec1May.html), openoffice.org, "(Since it always gets mentioned in relation to Netscape's Mozilla decision, I should also note that Eric Raymond's paper "The Cathedral and the Bazaar" was referenced by me and others who were lobbying Netscape's management. In my opinion the paper's importance in the context of Netscape's decision was mainly that it provided some independent validation of ideas that were already being actively discussed and promoted within Netscape. If you've ever tried to promote a proposal within your organization, then you may have discovered that it's somewhat easier to do this if you can point to someone outside the organization who's saying the same thing.)"

7. Sam Williams (30 November 2011). *Free as in Freedom [Paperback]: Richard Stallman's Crusade for Free Software* (https://books.google.com/books?id=BB68Ql7ZY_AC&pg=PA161). "O'Reilly Media, Inc.". p. 161. ISBN 978-1-4493-2464-3. "When Netscape CEO Jim Barksdale cited Raymond's 'Cathedral and the Bazaar' essay as a major influence upon the company's decision, the company instantly elevated Raymond to the level of hacker celebrity. Determined not to squander the opportunity, Raymond traveled west to deliver interviews, advise Netscape executives, and take part in the eventual party celebrating the publication of Netscape Navigator's source code."
8. Poe, Marshall (September 2006). "The Hive" (https://www.theatlantic.com/magazine/archive/2006/09/the-hive/305118/). *The Atlantic*. Retrieved 2012-07-05.
9. Schiff, Stacy. "Annals of Information" (http://www.newyorker.com/archive/2006/07/31/060731fa_fact). *The New Yorker*. Retrieved 4 July 2014.
10. Karl Eugen Kurbel (23 June 2008). *The Making of Information Systems: Software Engineering and Management in a Globalized World* (https://books.google.com/books?id=ggVaezlfOCcC&pg=PA222). Springer. pp. 222–. ISBN 978-3-540-79260-4. Retrieved 15 October 2012.
11. Bezroukov, Open source software development as a special type of academic research: Critique of vulgar Raymondism (https://firstmonday.org/ojs/index.php/fm/article/view/696/606)" Accessed 23 May 2019.
12. Bezroukov, A second look at *The Cathedral and the Bazaar* (https://firstmonday.org/ojs/index.php/fm/article/view/708/618) Accessed 23 May 2019.
13. Jan Bergstra; Mark Burgess (19 December 2007). *Handbook of Network and System Administration* (https://books.google.com/books?id=NUoZ7fKOITQC&pg=PA202). Elsevier. pp. 202–. ISBN 978-0-444-52198-9. Retrieved 15 October 2012.
14. Eric S. Raymond, "Response to Nikolai Bezroukov (http://catb.org/~esr/writings/response-to-bezroukov.html)"
15. Yarvin, Curtis (31 March 2022). "The Cathedral or the Bizarre" (https://www.tabletmag.com/sections/news/articles/the-cathedral-or-the-bizarre). *Tablet*. Retrieved 13 October 2024.

References

- Raymond, Eric S. (1999). *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. O'Reilly Media. ISBN 1-56592-724-9.

External links

- Official website (http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/)
- Roberts, Russ (January 19, 2009). "Eric Raymond on Hacking, Open Source, and *The Cathedral and the Bazaar*" (http://www.econtalk.org/archives/_featuring/eric_raymond/). *EconTalk*. Library of Economics and Liberty.

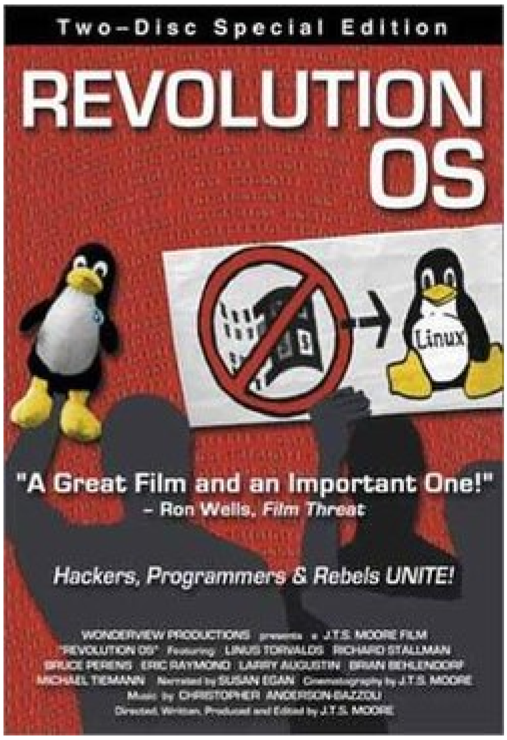
WIKIPEDIA
The Free Encyclopedia

Revolution OS

Revolution OS is a 2001 documentary film that traces the twenty-year history of GNU, Linux, open source, and the free software movement.

Directed by J. T. S. Moore, the film features interviews with prominent hackers and entrepreneurs including Richard Stallman, Michael Tiemann, Linus Torvalds, Larry Augustin, Eric S. Raymond, Bruce Perens, Frank Hecker and Brian Behlendorf.

Revolution OS



Promotional poster for two disc edition of *Revolution OS*

Directed by	J. T. S. Moore
Written by	J. T. S. Moore
Produced by	J. T. S. Moore
Starring	Richard Stallman Linus Torvalds Eric S. Raymond Bruce Perens
Edited by	J. T. S. Moore
Music by	Christopher Anderson-Bazzoli
Release date	2001
Running time	85 min
Country	United States
Language	English

Synopsis

The film begins with glimpses of Raymond, a Linux IPO, Torvalds, the idea of Open Source, Perens, Stallman, then sets the historical stage in the early days of hackers and computer hobbyists when code was shared freely. It discusses how change came in 1978 as Microsoft co-founder Bill Gates, in his Open Letter to Hobbyists, pointedly prodded hobbyists to pay up. Stallman relates his struggles with proprietary software vendors at the MIT Artificial Intelligence Lab, leading to his departure to focus on the development of free software, and the GNU Project.

Torvalds describes the development of the Linux kernel, the GNU/Linux naming controversy, Linux's further evolution, and its commercialization.

Raymond and Stallman clarify the philosophy of free software versus communism and capitalism, as well as the development stages of Linux.

Michael Tiemann discusses meeting Stallman in 1987, getting an early version of Stallman's GCC, and founding Cygnus Solutions.

Larry Augustin describes combining GNU software with a normal PC to create a Unix-like workstation at one third the price and twice the power of a Sun workstation. He relates his early dealings with venture capitalists, the eventual capitalization and commodification of Linux for his own company, VA Linux, and its IPO.

Brian Behlendorf, one of the original developers of the Apache HTTP Server, explains that he started to exchange patches for the NCSA web server daemon HTTPd with other developers, which led to the release of "a patchy" web server, dubbed Apache.

Frank Hecker of Netscape discusses the events leading up to Netscape's executives releasing the source code for Netscape's browser, one of the signal events which made open source a force to be reckoned with by business executives, the mainstream media, and the public at large.[1] This point was validated further after the film's release as the Netscape source code eventually became the Firefox web browser, reclaiming a large percentage of market share from Microsoft's Internet Explorer.

The film also documents the scope of the first full-scale LinuxWorld Summit conference, with appearances by Linus Torvalds and Larry Augustin on the keynote stage.

Much of the footage for the film was shot in Silicon Valley.

Screenings

The film appeared in several film festivals including South by Southwest, the Atlanta Film and Video Festival, Boston Film Festival, and Denver International Film Festival; it won Best Documentary at both the Savannah Film and Video Festival and the Kudzu Film Festival.

Quotes

> I bumped into him (Craig Mundie of Microsoft) in an elevator. I looked at his badge and said, "Ah, I see you work for Microsoft."
>
> He looked back at me and said, "Oh yeah, and what do you do?"
>
> And I thought he seemed just some sort of a tad dismissive, I mean here is the archetypal guy in a suit looking at a scruffy hacker. . . so I gave him the thousand yard stare and said, "I'm your worst nightmare."
>
> —Eric S. Raymond

> Giving the Linus Torvalds award to the Free Software Foundation is sort of like giving the Han Solo award to the Rebel Fleet.
>
> —Richard Stallman

> . . . and I realised he (Steve Ballmer) had read my document and understood it, and was now telling the press about this. Now, if you're like just a guy on the net who's not doing this for a job at all and you sort of write a manifesto and it spreads out through the world, and a year later the Vice President of Microsoft is talking about that, you'd think you were on drugs, wouldn't you? But that's what *really* happened.
>
> —Bruce Perens

> Think of Richard Stallman as the great philosopher and think of me as the engineer.
>
> —Linus Torvalds

Reception

Every review noted the historical significance of the information, and those that noticed found the production values high, but the presentation of history mainly too dry, even resembling a lecture. Ron Wells of *Film Threat* found the film important, worthwhile, and well thought out for explaining the principles of the free software and open source concepts. Noting its failure to represent on camera any debate with representatives of the proprietary software camp, Wells gave the film 4 of 5 stars.[2] *TV Guide* rated the film 3 of 4 stars: "surprisingly exciting", "fascinating" and "sharp looking" with a good soundtrack.[3] *Daily Variety* saw the film as "targeted equally at the techno-illiterate and the savvy-hacker crowd;" educating and patting one group on the head, and canonizing the other, but strong enough for an "enjoyable" recommendation.[4]

On the negative side, *The New York Times* faulted the film's one-sidedness, found its reliance on jargon "fairly dense going", and gave no recommendation.[5] *Internet Reviews* found it "a didactic and dull documentary glorifying software anarchy. Raging against Microsoft and Sun. . .", lacking follow-through on Red Hat and VALinux stock (in 2007, at 2% of peak value), with "lots of talking heads".[6] *Toxicuniverse.com* noted "Revolution OS blatantly serves as infomercial and propaganda. Bearded throwback to the sixties, hacker Richard Stallman serves as the movement's spiritual leader while Scandinavian Linus Torvalds acts as its mild mannered chief engineer (as developer of the Linux kernel)."[7]

To Tim Lord, reviewing for *Slashdot*, the film is interesting and worthy of viewing, with some misgivings: it is "about the growth of the free software movement, and its eventual co-option by the open source movement. . . it was supposed to be about Linux and its battle about Microsoft, but the movie is quickly hijacked by its participants." The film "lacks the staple of documentaries: scenes with multiple people that are later analyzed individually by each of the participants" (or indeed, much back-and-forth at all). Linux itself and its benefits are notably missing, and, "[w]e are never shown anyone using Linux, except for unhappy users at an Installfest." The debate over Linux vs Windows is missing, showing the origin of the OS only as a response to proprietary and expensive Sun and DEC software and hardware, and its growth solely due to the Apache web server. And Lord notes that the film shows, but does not challenge Torvalds or Stallman about their equally disingenuous remarks about the "Linux" vs "GNU/Linux" naming issue.[8]

See also

- *The Code* – another documentary film about Linux
- *Pirates of Silicon Valley*
- Open source
- Linux
- Free software movement
- Copyleft

- *The Cathedral and the Bazaar*

References

1. Community Articles: Interview: Frank Hecker (http://www.openoffice.org/editorial/ec1May.html) Openoffice.org. Louis Suárez-Potts, May 1, 2001. Retrieved 2007-04-22.
2. Revolution OS (https://www.filmthreat.com/index.php?section=reviews&Id=1820) *Film Threat*. Ron Wells, February 21, 2002. Retrieved 2007-04-23.
3. Review - Revolution OS (https://tvguide.com/movies/revolution-os/review/135794) *TV Guide.com.* 2002. Retrieved 2007-04-22.
4. Revolution OS (https://www.variety.com/review/VE1117917117.html?categoryid=31&cs=1&p=0) Review. *Daily Variety*. Scott Foundas, March 1, 2002. Retrieved 2007-04-22.
5. Heroes and Villains on a Cyberbattlefield (https://movies2.nytimes.com/mem/movies/review.html?_r=2&title1=&title2=Revolution%20Os%20%28Movie%29&reviewer=Stephen%20Holden&v_id=261234&pdate=&partner=Rotten%20Tomatoes&oref=slogin&oref=login) Archived (https://web.archive.org/web/20151017031844/http://movies2.nytimes.com/mem/movies/review.html?_r=2&title1=&title2=Revolution%20Os%20%28Movie%29&reviewer=Stephen%20Holden&v_id=261234&pdate=&partner=Rotten%20Tomatoes&oref=slogin&oref=login) 2015-10-17 at the Wayback Machine *New York Times*. Stephen Holden, February 22, 2002. Retrieved 2007-04-22
6. Revolution OS (2001) (https://www.imdb.com/reviews/325/32591.html) *InternetReviews.com*. Steve Rhodes. Retrieved 2007-04-22.
7. Introduction to Linux 101 (http://www.toxicuniverse.com/review.php?rid=10005579) *ToxicUniverse.com*. John Nesbit, April 07, 2004. Retrieved 2007-04-22.
8. Revolution OS Review (http://slashdot.org/article.pl?sid=02/04/19/1255213) *Slashdot.org.* Tim Lord *(timothy)*, April 19, 2002. Retrieved 2007-04-22.

External links

- Official website (http://www.revolution-os.com/)
- *Revolution OS* (https://www.imdb.com/title/tt0308808/) at IMDb
- *Revolution OS* (http://slashdot.org/article.pl?sid=02/04/19/1255213) Slashdot (20 April 2002)
- *Revolution OS* (https://www.rottentomatoes.com/m/Revolution_OS) at Rotten Tomatoes
- *Revolution OS (https://www.metacritic.com/movie/revolution-os)* at Metacritic