Free software

From MTU LUG wiki

Free software, **software libre** or **libre software** is software that can be used, studied, and modified without restriction, and which can be copied and redistributed in modified or unmodified form either without restriction, or with minimal restrictions only to ensure that further recipients can also do these things and that manufacturers of consumer-facing hardware allow user modifications to their hardware. Free software is available gratis (free of charge) in most cases.

In practice, for software to be distributed as free software, the human-readable form of the program (the source code) must be made available to the recipient along with a notice granting the above permissions. Such a notice either is a "free software license", or a notice that the source code is released into the public domain.

The free software movement was conceived in 1983 by Richard Stallman to satisfy the need for and to give the benefit of "software freedom" to computer users. The Free Software Foundation was founded in 1985 to provide the organizational structure which Stallman correctly foresaw would be necessary to advance his Free Software ideas.

From 1998 onward, alternative terms for free software came into use. The most common are "software libre", "free and open source software" ("FOSS") and "free, libre and open source software" ("FLOSS"). The "Software Freedom Law Center" was founded in 2005 to protect and advance FLOSS The antonym of free software is "proprietary software" or "non-free software". Commercial software may be either free software or proprietary software, contrary to a popular misconception that "commercial software" is a synonym for "proprietary software". (An example of commercial free software is Red Hat Linux.)

Free software, which may or may not be distributed free of charge, is distinct from "freeware" which, by definition, does not require payment for use. The authors or copyright holders of freeware may retain all rights to the software; it is not necessarily permissible to reverse engineering, modify, or redistribute freeware.

Freeware is provided to end-users at no cost, but free software provides more benefits than simply delivering a no-cost product--indeed, for the end-user, there may be circumstances where the monetary cost of acquiring free software exceeds the cost of freeware.

Since free software may be freely redistributed it is generally available at little or no cost. Free software business models are usually based on adding value such as applications, support, training, customization, integration, or certification. At the same time, some business models which work with proprietary software are not compatible with free software, such as those that depend on a user paying for a license in order to lawfully use a software product.

Contents

History

In the 1950s, 1960s, and 1970s, it was normal for computer users to have the freedoms that are provided by free software. Software was commonly shared by individuals who used computers and by hardware manufacturers who were glad that people were making software that made their hardware useful. Organizations of users and suppliers were formed to facilitate the exchange of software, see, for example, SHARE. By the late 1960s change was inevitable: 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 anticompetitive.

In 1983, Richard Stallman, longtime member of the hacker (programmer subculture) 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. 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.

The economic viability of free software has been recognised by large corporations such as IBM, Red Hat, and Sun Microsystems. 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. Also, 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.

Naming

The FSF recommends using the term "free software" rather than "open source software" because, they state in a paper on Free Software philosophy, the latter term and the associated marketing campaign focuses on the technical issues of software development, avoiding the issue of user freedoms. "Libre" is used to avoid the ambiguity of the word "free".

Definition

The first formal definition of free software was published by FSF in February 1986. 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:

- Freedom 0: The freedom to run 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 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.

Freedoms 1 and 3 require source code to be available because studying and modifying software without its source code is highly impractical.

Thus, free software means that user have the freedom to cooperate with whom they choose, and to control the software they use. To summarize this into a remark distinguishing freedom software from zero price software, Richard Stallman said: "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

In the late 90s, other groups published their own definitions which describe an almost identical set of software. The most notable are Debian Free Software Guidelines published in 1997, 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 make software based on their source code, and then release the modified result as proprietary software. Their view is that this permissive approach is more free. The Kerberos (protocol), X.org, and Apache License software licenses are substantially similar in intent and implementation. All of these software packages originated in academic institutions interested in wide technology transfer University of California, MIT, and University of Illinois at Urbana-Champaign.

Examples of free software

The Free Software Directory maintains a large database of free software packages. Some of the best-known examples include the Linux Kernel, the BSD and GNU/Linux operating systems, the GNU Compiler Collection and GNU C Library; the MySQL relational database; the Apache HTTP Server 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 OpenOffice.org office suite; and the TeX and LaTeX typesetting systems.

Free software licenses

All free software licenses must grant people 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.

The majority of free software uses a small set of licenses. The most popular of these licenses are:

- the GNU General Public License
- the GNU Lesser General Public License
- the BSD License
- the Mozilla Public License
- the MIT License
- the Apache 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 licensees can exist which the FSF has not heard about, or considered important enough to write about. So it's possible for a license to be free and not in the FSF list. However, the OSI list is prescriptive: they only list licenses that have been submitted, considered and approved. This formal process of approval is what defines a license as Open Source. Thus, it's not possible for a license to be Open Source and not on the OSI approved list.

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 doesn't 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.

It is rare that a license is announced as being in-compliance by either FSF or OSI guidelines and not vice versa (the Netscape Public License used for early versions of Mozilla being an exception, as well as the ASA Open Source Agreement).

Permissive and copyleft licenses

The FSF categorizes licenses in the following ways:

- Public domain software – the copyright has expired, the work was not copyrighted or the author has released the software onto the public domain. Since public-domain software lacks copyright protection, it may be freely incorporated into any work, whether proprietary or free.
- 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 proprietary ones.
- Copyleft licenses, the GPL being the most prominent. The author retains copyright and permits redistribution and modification provided 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.

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 which lack fixes for known security flaws, at least until a fix is available.

Free software advocates say that this method is biased by counting more vulnerabilities for the free software, since its source code is accessible and its community is more forthcoming about what problems exist,

Commercial viability and adoption

Free software played a part in the development of the Internet, the World Wide Web and the infrastructure of dot-com companies.

Under the free software business model, free software vendors may charge a fee for distribution and offer pay support and software customization services. Proprietary software uses a different business model, where a customer of the proprietary software pays a fee for a license to use the software. 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.

Free software is generally available at no cost and can result in permanently lower costs compared to proprietary software. 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.

A report by Standish Group says that adoption of open source has caused a drop in revenue to the proprietary software industry by about $60 billion per year.

Controversies

Binary blobs

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 to effectively 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 GNU/Linux distribution with all the binary blobs removed. The project received support from the Free Software Foundation.

BitKeeper

Larry McVoy invited high-profile free software projects to use his proprietary versioning system, BitKeeper, free of charge, in order to attract paying users. In 2002, Linux coordinator Linus Torvalds decided to use BitKeeper to develop the Linux kernel, a free software project, claiming no free software alternative met his needs. This controversial decision drew criticism from several sources, including the Free Software Foundation's founder Richard Stallman.

Following the apparent reverse engineering of BitKeeper's protocols, McVoy withdrew permission for gratis use by free software projects, leading the Linux kernel community to develop a free software replacement called Git.

Patent deals

In November 2006, the Microsoft and Novell software corporations announced a controversial partnership involving, among other things, patent protection for some customers of Novell under certain conditions.

External links

- Software Freedom Law Center (http://www.softwarefreedom.org)
- The Free Software Definition (http://www.gnu.org/philosophy/free-sw.html)
- Transcripts about Free Software (http://fsfe.org/transcripts)
- Free Software Magazine (http://www.freesoftwaremagazine.com)
- Free cultural works definition (http://freedomdefined.org/Definition)
- Why Open Source Software / Free Software (OSS/FS)? Look at the Numbers! (http://www.dwheeler.com/oss_fs_why.html), analysis of the advantages of OSS/FS by David A. Wheeler.
- FLOSSWorld - Free/Libre/Open-Source Software: Worldwide impact study (http://www.flossworld.org/index.php)
- Software Freedom: An Introduction (http://www.teak.cc/softfree/software-freedom.html)
- Decoding Liberation: The Promise of Free and Open Source Software (http://www.sci.brooklyn.cuny.edu/~bcfoss/DL)
- Free Software Definition at The Linux Information Project (http://www.linfo.org/free_software.html)
- Open Source Enters the Mainstream According to Findings from the Actuate Annual Open Source Survey for 2008 (http://www.actuate.com/company/news/press-releases-resources.asp?ArticleId=13847)
- FSF/UNESCO Free Software Directory (http://directory.fsf.org)
