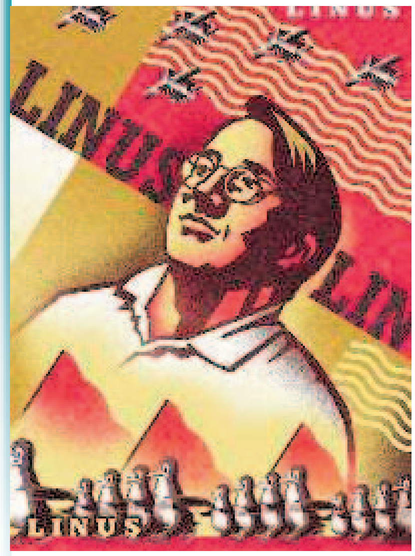
Free software is a matter of liberty, not price

HOW FREE IS FREE?



RICHARD SMEDLEY

Free Software (FS) and Open Source Software (OSS). Perhaps you have heard these terms used interchangably, perhaps even in opposition by two sides of a licensing argument. We attempt to tease apart the meanings of these two phrases and the purposes of the movements behind them and finish up with a brief look at software licenses for your project.

There are many definitions of Free and Open Source Software, so let's start with the Free software foundation (FSF): "Free software is software that comes with permission for anyone to use, copy, and distribute, either verbatim or with modifications, either gratis or for a fee. In particular, this means that source code must be available. *If it's not source, it's not software*."

Before we go deeper into this we need to journey back into the history of the FSF and the Free Software Movement. Writing software is like any other scientific endeavour – there is a process of discovery, then one of justification. The hypothesis, test conditions and results must be shared with other scientists, to see if the process is replicable. This is the justification stage. There must be sharing of the Source Code – the hypothesis and test conditions – for others to build upon each discovery and advance that area of science. Scientists have occasionally lapsed into secrecy due to strong rivalry, and some discoveries (Mendel and his vertically challenged peas) are made in total isolation, but ultimately science only advances through the sharing of ideas. Originally the world of Computer Science was just like this. Sharing the source was taken for granted. Outside circumstances lead to changes. These are thoroughly documented elsewhere by Eric Raymond (conventionally referred to as ESR), the semi-official anthropologist to the hacker community.

Like much of modern computing our story starts at MIT Artificial Intelligence Laboratory in the 1970s. Here the Lab's hackers had written the Incompatible Timesharing System (ITS) in assembler code to replace the Operating System (TOPS-10) supplied with its PDP-10 minicomputer by the manufacturer, Digital Equipment Corporation (DEC). It was in 1971 that Richard Stallman (conventionally referred to as RMS) joined the MIT AI Lab and became immersed in their culture of hacking and code sharing. It was 1971, coincidentally, when Ken Thompson and Dennis Ritchie won an internal contract at Bell Labs to produce an office-automation system using their recently developed Unix operating system. Three years later they had recoded it in C and ported it to several different machines. By 1983, when DEC cancelled plans for a follow-up to the PDP-10 range, Unix (usually running on a PDP-11 or Vax) was a strong alternative to the PDP-10/ TOPS-10 solution previously favoured by academia and research laboratories.

Cooperation is forbidden

Many people know the story of RMS being refused the source code for the control program of the Lab printer. This crystallised his opposition to the closing-off of source code into proprietary programs and he became a fierce opponent of the commercialisation of the Lab. In 1982 MIT AI Labs, having lost many of its original ITS team to new computer companies, went with DEC's own, non-free, timesharing OS for their new PDP-10. To use the OS's of the time one had to sign a nondisclosure agreement just to get an executable copy. Stallman was faced with a choice. He could ignore his principles, work with a system based on *not* sharing and helping members of the hacker community, or leave computing – which would have squandered his skills and training. RMS came up with a third option.

He decided to leave the AI Lab and found an organisation, the Free Software Foundation (FSF), to write a free operating system (OS), to encourage a worldwide community of co-operating hackers. The OS would be made compatible with Unix so that Unix users could easily switch to it. It would also have easy portability. Following a hacker tradition the self-recursive acronym GNU, for GNU's Not Unix, was chosen. In 1984 changes at AT&T meant Unix becoming a fully commercial project.

Having seen freely shared code taken up by commercial organisations and put in proprietary software, RMS worked on a license to protect users' freedom to "run, copy, distribute, study, change and improve the software." The FSF defined the *Four Freedoms*:

- The freedom to run the program, for any purpose (Freedom 0)
- The freedom to study how the program works, and adapt it to your needs (Freedom 1)
- Access to the source code is a precondition for this. The freedom to redistribute copies so you can help your neighbour (Freedom 2)
- The freedom to improve the program, and release your improvements to the public, so that the whole community benefits (Freedom 3)
- Access to the source code is a precondition for this.

To protect these freedoms the GNU General Public License (GPL) was developed. Defined as "a necessary evil", the GPL is the classic, "you are free to do what you like, as long as you do not remove freedom from others" license. The restrictions in the license only apply to those distributing modified forms of a GPL'ed program and are designed to pass on the same freedoms that you had with the code. In recent years the license has become the battleground between the worlds of proprietary and free software, we will return to this theme later with a look at available licenses. The GPL, and the freedoms it protects have become the standard against which other licenses are measured. This is in part due to one young programmer's decision to adopt the GPL for a particular piece of code he had written, something he has said is the smartest decision he made.

Enter the Penguin

In their first decade the FSF were quite successful at producing most of the unglamorous programs that go to make up an operating system - such as the linker, assembler, C library and so on. Following contemporary OS theory they were developing a microkernel - the HURD - and it was taking a long time. In 1991 this gap was filled by Linux.

Wanting to run a Unix-like system on his 386 PC, and dissatisfied with the shortcomings of Minix (a cheaply available academic OS), Helsinki University student Linus Torvalds developed a kernel using the GNU tools. He released the source code on the Internet and a group of hackers rapidly grew around the project. Within two years GNU/Linux had become a stable OS, competing with commercial Unices and attracting ports and new software. Free Software was now well and truly competitive.

Open Source

The growth of GNU/Linux, in comparison to freeBSD, as well as to proprietary software, is often attributed to the GPL license, which protects freedom. Although the GPL has always been anti-proprietary, it has never been *anti-commercial*, indeed it insists that there is no restriction on the commercial use of a piece of software.

Nevertheless a number of supporters of freedom and Free Software *are* anti-business, and whilst this has been no barrier to the business releasing their software under the GNU GPL, it was seen by a number of influential Linux figures as a barrier to the further growth of Linux.

Meeting on February 3rd 1998, in Palo Alto, California, to discuss the opportunity presented by Netscape's decision to open up the source to its browser; Eric Raymond, John 'Maddog' Hall and Larry Augustin (both of Linux International), Sam Ockman (of the Silicon Valley Linux User's Group), Todd Anderson and Chris Peterson (of the Foresight Institute) were looking to make a pragmatic case to businesses. Concerns of freedom, responsibility and ethical issues were seen as obstructive to getting businesses on board and a "better" term was sought. Peterson came up with "Open Source" and a new movement was born. Linus Torvalds, Bruce Perens (of Debian) and Phil Hughes (editor of Linux Journal) soon became involved and on the wave of publicity surrounding Netscape's announcements the term (and the movement) open source hit the mainstream, being profiled in the Economist and Forbes Magazine before the year was out.

But what about freedom?

Richard Stallman of the Free Software Foundation has been quite critical of the term "Open Source" and has covered a number of the above points on the FSF's Web pages. His strongest criticisms are reserved for companies who use their association with the Open Source movement to leverage proprietary (non-free and closed source) products, this has taken a number of forms. For example the featured speaker at a Linux trade show in late 1998 was an executive from a prominent software company that had decided to support Linux. Unfortunately, their form of support consists of releasing non-free software that works with the system - using the Free Software community as a market but not contributing to it. He said, "there is no way we will make our product open source, but perhaps we will make it 'internal' open source. If we allow our customer support staff to have access to the source code, they could fix bugs for the customers, and we could provide a better product and better service."

As a term, *Free Software* suffers from the dual meaning of the English word "free". Many other languages have separate words to cover "without cost" and "without restriction" and in parts of Europe FS is referred to as "Libre Software". However it is a simple matter to explain that it's about freedom - "think free speech rather than free beer". Open Source implies nothing about protecting freedom to run and distribute modified copies, and has lead to a plethora of different licenses (see Table) which allow access to the source code but place burdensome restrictions upon the use, modification or distribution of the software.

The Open Source Institute (OSI), as well as publishing a definition of Open Source - based upon Perens' Debian *Free Software Guidelines*, use the *OSI Certified Mark* on licenses they believe are compatible with their *Open Source Definition*. This, they say, is because "the term 'Open Source'... has become widely used and its meaning has lost some precision." Perens, on the OSI Web site: "To be Open Source, all of the terms below must be applied together, and in all cases. For example, they must be applied to derived versions of a program as well as the original program. It's not sufficient to apply some and not others, and it's not sufficient for the terms to only apply some of the time. After working through some particularly naive interpretations of the Open Source Definition, I feel tempted to add – this means you! "

The OS argument that to businesses appearance is everything and the word 'free' is a great obstruction to business involvement, is counterbalanced by OSS advocates Chris DiBona, Sam Ockman, and Mark Stone in the introduction to *Open Sources: Voices from the Revolution*. "The success of the open-source movement does not depend on businesses adopting it. It's not 'in the market' except in the sense that movement is in the bazaar. Nobody needs to buy it for it to succeed. The success of open source software depends on people taking pride in their work and in doing it right, and deriving their sense of worth from that. That the products are useful and desirable flows

Given free license

TL:Free Software License	FSF Approved	GPL-compatible	OSI Approved	Copyleft
GNU General Public License (GPL)	y	y	y	y
GNU Library or 'Lesser' Public License (LGPL)	y	y	y	weak
Guile/ GNU Ada	y	y	y	weak
X11/ Xfree86/ Cryptix	y	y	(y)	n
Original BSD	y	n	y	n
Modified BSD	y	y	y	n
The Artistic License	n	n	y	n
Clarified Artistic license	y	y	y	n
Netscape Javascript License	y	y	-	weak
Netscape Public License	y	n	-	n
Mozilla Public License v. 1.0 (MPL)	y	n	y	weak
Mozilla Public License 1.1 (MPL 1.1)	y	(y)	y	weak
Qt Public License (QPL)	y	n	y	n
IBM Public License	y	n	y	y
Sun Public License	y	n	-	n
Sun Community Source License	n	n	-	n
Sun Internet Standards Source License 1.0	y	n	y	weak
Sun Solaris Source Code (Foundation Release) 1.1	n	n	-	n
MITRE Collaborative Virtual Workspace License (CVW License)	y	y	y	y
Ricoh Source Code Public License	y	n	y	y
Python license (1.6a2 & earlier)	y	y	y	y
Python license (1.6b1 & later)	y	n	y	y
zlib/libpng license	y	y	y	n
Apache Software License	y	n	y	n
Zope Public License	y	n	-	n
Apple Public Source License (APSL)	n	n	-	n
Intel Open Source License	y	n	y	n
Jabber Open Source License	y	n	y	y

from the success of craftsmanship, not the other way around."

OSS and FS are not two factions of the same ideology, with the same enemy but different tactics. They are two different ideologies with different enemies but the same tactics and short-term goals. For Free Software advocates the enemy are restrictions upon freedom to share knowledge. For Open Source proponents the enemy is poorly written software, particularly products in a monopoly position with no likelihood of change.

There is no doubt that we owe a colossal debt to the Free Software Foundation for the GNU project and the GNU GPL as well as a whole generation of programmers brought up on the benefits of gcc and other free tools. However it is also true that some of the recent growth of GNU/Linux is due to corporate interest in the practical benefits of the Open Source idea and that a number of these corporations are unhappy with ideas about freedom. Unhappy enough that only the different emphasis of Open Source movement encouraged them to GPL their software. We have seen, however, that many corporations have little interest in the open source *community*, only in attaching themselves to the kudos of the Open Source tag. To remain free, Open Source needs constant explaining. For Free Software there is no doubt that it is about freedom, and the source should be always available. Sticking to purely free software may mean missing out on some tempting closed-source apps in the short-term, but the better quality software will continue to arrive.

At the moment there is little practical difference between FS and OS. If people cease to value the freedom of their software, will that always be the case?To protect the freedoms discussed above the GNU project uses copyright law to enforce the freedoms of the GPL. As well as declaring the right for anyone to run, copy, modify and distribute modified copies of the software, it refuses these rights to anyone who seeks to add restrictions of their own. This reversal of the traditional use of copyright law has been named *copyleft*, following a quip scribbled on the envelope of a letter to RMS "copyleft – all rights reserved". Some FS licenses do not protect software from future restrictions, many programs under these non-copyleft licenses have been absorbed into proprietary code, with further development not returned to the community.

The GPL has been covered earlier, so we turn to look at the LGPL. Originally the GNU *Library General Public License* was conceived for tactical reasons. The GPL does not allow a non-free piece of code to be linked to it, as there were already many C libraries the GNU C library was given a special license so that it would be more widely used, leading to the rapid spread of the GNU tools. For specialist libraries, such as the GNU Readline, developed to provide the Bash shell with command line editing, the GPL is more appropriate at it gives an advantage to free software (i.e. only free software can link to it). As people were beginning to LPGL their libraries as a matter of course the LPGL was renamed the GNU Lesser General Public License, to give a less misleading impression. The licenses for Guile and the GNU Ada compiler are similar.

There are many free licenses that are incompatible with the GPL, due to restrictions on use or modification of the software - see table. However the largest group of GPL-compatible free licenses to "rival" the GPL is the X11/ BSD type license.

Do what thou wilt

In the 1980s the many competing windowing systems for Unix were vanquished by the X11 windowing system. This was licensed under permissive (non-copyleft) terms, which gave the user permission to do what they liked with the code, but placed no restrictions upon taking the code and making it proprietary. Thus commercial Unix vendors soon each had their own proprietary X11 version. If your aim is many users for your standard, then this is a useful license. However it does not protect the freedom of future users of the software.

The modified BSD license, under which the "other free Unix-like operating system" – freeBSD (& its close relatives) is released, is a similar permissive license. As BSD-licensed code can be linked with any (proprietary or not) code, some developers and firms see this as a big advantage. Earlier versions contained an advertising clause, insisting on credit for earlier authors being placed in advertisements for modified versions, which resulted in some advertisements containing 70 or more credits. It is the permissive nature of the license that has helped to attract Apple to use freeBSD as the heart of its soon-to-be-released OSX.

If you really want to investigate the minutiae of all the other free software licenses the list at the FSF Web site is a good place to start. However my personal advice is that your time would be much better spent coding! ■

The author

Richard Smedley is an organic gardener by training, an engineer by temperament and a writer because he has to pay the rent.

Info

The Free Software Foundation
http://www.fsf.org
The GNU project
http://www.gnu.org
Open Source Initiative
http://www.opensource.org/
O'Reilly open source network - OSS news site
http://opensource.oreilly.com/
Open Sources: Voices from the Open Source Revolution, various authors, O'Reilly 1999. ISBN 1-56592-582-3
Eric Raymond's histories of hackers and the Open Source movement
http://www.tuxedo.org/

Yet more licenses...

Other GPL compatible FS licenses include: iMatrix Standard Function Library; W3C Software Notice and License; Berkeley Database License (as published 1999-09-12).
Other GPL incompatible FS licenses include: Arphic Public License (no incompatibility when used for fonts); OpenLDAP License; Phorum License; LaTeX Project Public License; Netizen Open Source License (NOSL), Version 1.0; Interbase Public License, Version 1.0; Freetype License; Open Compatibility License; PHP License, Version 2.02 [this is used for PHP4, PHP3 is dual licensed under the GPL].
Non-free: Plan9 license; Open Public License.