

# Should all software be free?

---

## Preface

This is a report written as a part of a coursework given in the computing course at Roehampton University 09/10. This is one half of an assignment, where the other half was a debate where the two sides were discussed.

Some modifications have been made to the original report, but this only concerns the actual debate and the groups, not the main points of the report.

## Introduction

This report is written with background in a debate held where I was part of a team that were pro this notion: "This house believes that all software should be free". The report will be concentrated around the four freedoms the Free Software Foundation (FSF) has stated a piece of software must fulfil to be called "free software", and look at arguments from both sides. After debating these freedoms, I will try to suggest a solution both sides of the debate can partly accept. In the end I will also give my views on my own and my team's performance in the debate.

## Definitions

Before one can start discussing whether or not all software should be free we need to define what we mean with the term "free software". FSF describes free software as "software that respects our freedom" (Stallman, 2007b). And with that they mean that free software is not just software free of charge, but software that gives us the right to treat the software as our own, and makes us able to do whatever we want with it. They look at free software as not a question of pricing, but as a "political and ethical choice" (Free Software Foundation, 2010a).

It is important for this discussion to separate the free software community and the open source community. In most cases, open source software is free software, but free software is not limited to open source. Open source is a way of developing software, and free software is not limited to just concerning this type of software. Commercial developed software can also be distributed as free software, and not only as proprietary software (as most are today). FSF encourages re-distributors to "charge as much as they wish or can" for free software (Free Software Foundation, 2010b). And this fact might confuse many taking part in a discussion of free software, and make them assume that there is no money or future in free software.

Today, most proprietary software comes with a user license, i.e. you do not "buy" the software itself, but you pay for the right to use it. And in this license, you agree to obey many terms that limits you as a user. This can be limiting the numbers of computers to install the software on, limiting the use of the software, giving no access to the source of the software, and making the producer not liable for any loss the software may cause. The proprietary software manufacturers will argue strongly for this practice. Not only so that they can protect their product from unwanted snooping, but they will also argue that by letting the user only buy a license, the user is protected against any claims of intellectual property infringements. The user does not own the software, therefore the licensee will fight the allegations instead (Microsoft, 2008).

## The four freedoms of free software

### **Freedom 0 - The freedom to run the software for any purpose**

With this statement, FSF means that you should not be restricted when using the software (Free Software Foundation, 2009a), whether it is for the purpose (recreational, educational, commercial etc.), or the installation (e.g. installing it on as many computers as needed, or for as many users as needed). Software should be as natural as any other commodity you buy. You should not need to ask permission for any use of the software.

With a normal software license, you will be restricted. Especially when it comes to how many installations you can make. Most modern software is today equipped with an activation module, where you must register with the company to fully use it. To install it on a new machine, you must often deactivate your first installation. And the license is often personal, meaning that you can't let someone else use the software.

This is most of all an economic question for the software maker. They want a payment for each user, and do not allow sharing. This makes sense economically. And in fact ethically. If a company with one user spends £100 on software, the user expense is £100. But if a large company with 100 users buy the same software, and copies it on all the computers in the company, meaning that all the users can use one copy, the user expense is £1. Is this fair for the sole user in the first company? Most companies offer discounts when buying multiple licenses, and often offer student cheaper versions of their software. However, these versions are also highly limited. But it boils down to that they have a product, and everyone who uses it should pay.

### **Freedom 1 - The freedom to study how the program works, and change it to make it do what you wish**

When a user buys a software license today, he will only receive the executable files to install the program, not the source code for these files. With free software you actually get access to this source code, and the possibility to do modifications. This will enable you, if you want, to explore and see what the software is in fact doing to your property: the computer. However, most people will not look at the source code and try to change it, even if they had the chance. Most users do not have the knowledge or the wish to make changes to software, but it will apply to specialist users, e.g. computer students/professionals who want to learn or business users looking for a specific feature.

The proprietary software vendors want to sell you a black box. They do not want you to look at the source, hidden behind a vale of copyrights and patents. Craig Mundie, Chief Technical Officer at Microsoft, said in 2004: "The one great principle of the software industry is: Never show anybody the source code to anything" (Moglen, 2004). It might be natural for companies like Microsoft and Adobe to want to protect their solutions for others. Products they have spent millions to develop or buy from other developers. IBM's "mistake" from the beginning of the 80's, letting other producers clone their computer setup so it would fit Microsoft's OS, is maybe still frightening many to protect what they got, since IBM lost market shares on that incident (O'Reilly, 2004). And most of Microsoft's revenue is still coming from the old Office, Windows and server combo (Silicon Alley Insider, 2010). It might be natural to think: Why risk that? And what do we gain?

Eben Moglen talks especially about Microsoft, when he suggest they might in fact get a better product, if they open up their source code. Open it up not only to the users, but to itself. He speculates that only about 2% of the people that work on Windows can actually build it. Meaning that 98% must build components without fully knowing the system they are building for (Moglen, 2004). They are protecting themselves into a lesser product. They can also receive help from the users. If a competent user have found an error, with access to the code, he can actually find the error and give them a solution rather than a new service tag (Free Software Foundation, 2009b)

Another argument for opening up source code comes from Thomas Larsen (2010), one of the authors of Yabasic. He argues the importance of free software in the context of legacy software. Historical important, or company important, software can be lost or marginalized when it's original authors stops upgrading and supporting the product, without publishing the source code.

On another note, making the source code of security software open, can potentially lead to hacker attacks. If a world with black hat hackers, this can mean that if a hacker learns how a system works, he learns how to work around said system.

### **Freedom 2 - The freedom to redistribute copies so you can help your neighbour.**

This freedom is linked tightly with freedom 0, but takes your software copy out of your house and into your neighbour's house (Free Software Foundation, 2009). File sharing. When you have the software, and your neighbour needs the software, it is your moral duty to help him out. Richard Stallman (2007a) claims that by not allowing sharing software with our neighbour, our civic duty is polluted.

Again, the proprietary companies opposing free software, is looking at this as a profit loss. As with freedom 0, they want payment from each user. And who should be allowed to use their software, is their decision. By not giving the user the software, but only a license, they reduce his choice massively. Re-distributing is license breach. That is, unless you uninstall your copy before handing over the copy to your neighbour.

### **Freedom 3 - The freedom to distribute copies of your modified versions to others**

When you make changes to a software (freedom 1), you can re-distribute these changes to other users. And, if you want to, charge for this distribution (Free Software Foundation, 2009a). Software should not have an owner (Stallman, 2007). When you let other install software you have developed, you give up your ownership. It is your duty to give the software on. And if others do changes to it, they can re-distribute it, and charge for their work. Even if they got the software for free. If it is morally right to charge for it, is another question.

From a proprietary company point of view, this is unacceptable. Besides the file sharing, they will see it like someone is exploiting their code to make a profit. As mentioned earlier, this is a product they have spent money on developing. Is it fair that someone else should make money of their source?

A thing that can benefit the proprietary companies here is trust. How can you, as an end-user, be sure that the software you are acquiring is working as it should? You can only do that by getting it directly from the source, the original maker. And with extended support agreements, they can give the customer safety, something they are not guaranteed from a 3rd party.

## **The future**

For today's proprietary software makers, it is money that stops them from entering the world of free software. Can they be guaranteed the same profit as today? Most likely not! But then again, that profit may be too high to start with. They will still sell software, but might start to evolve into offering extended services as well. But we know that quality sells. Even after the introduction of open source software as GIMP and OpenOffice, PhotoShop and MS Office have still the largest market shares.

But I do think these companies must start to look towards today's trend in the industry. The trend goes toward web services, and not desktop software, and on-line, the user is more and more in charge. If you look at Twitter, an on-line service worth billions, and many of their

services are in fact developed by their users. Biz Stone, creative director in Twitter, have stated that Twitter have 500 employees, but over 5000 user developers (Levy 2009). Maybe it is time to give the customer the benefit of the doubt, and embrace him as an asset, not as a potential money loss. Or should they in fact fight the trend? Josh Catone (2008) warns companies just using on-line services, since you are so dependent on a supplier. In addition, both the free software and open source movements must look forward to the changes. For instance web services like Google are built on open source Linux, but should you have access to Google's source code? Services are most like useless without the data they are using, and are that covered? (O'Reilly, 2004).

### **Personal opinion**

I do not agree with the free software foundation in all they are saying. But I do like what they are doing. I think however, that the two sides in this debate can get a little closer. Software licenses should not be as limiting as they are today. I mean that the use of the software should be up to the user's discretion. The company should not control his use of the software. And it should be clearer what the software is in fact doing with a computer when it is operating. But I still don't think a user should be given total freedom over copying and re-distributing the software. Software is still a commodity, and just because it is easy to copy this commodity, I do not think it is morally right to do so. I'm still out on giving away the source code. And I think it is important to decide if software patents are valid, or not, before the battle can continue.

# Bibliography

---

Catone, J, 2008 *The argument against software as a service*,  
<http://www.sitepoint.com/blogs/2008/12/06/the-argument-against-saas/>, last accessed 10/02-2010

Free Software Foundation, 2009 *The free software definition*,  
<http://www.gnu.org/philosophy/free-sw.html>, last accessed 10/02-2010

Free Software Foundation, 2009 *Free software is more reliable!*,  
<http://www.gnu.org/software/reliability.html>, last accessed 10/02-2010

Free Software Foundation, 2010a *What is free software*, <http://www.fsf.org/resources/what-is-fs>, last accessed 10/02-2010

Free Software Foundation, 2010b *Selling free software*,  
<http://www.gnu.org/philosophy/selling.html>, last accessed 10/02-2010

Larsen, T, 2010 *An argument for free software*,  
<http://yabasic.basicprogramming.org/blog/2010/01/argument-for-free-software.html>, last accessed 10/02-2010

Levy, S, 2009 *How will Twitter grow up?*, Wired UK, 12.09, pp. 58-64

Microsoft, 2008 *Microsoft Intellectual Property Indemnification*,  
<http://www.microsoft.com/iplicensing/IPindemnification.aspx>, last accessed 10/02-2010

Moglen, E, 2007 *Freeing the mind: Free software and the end of proprietary software*, speech at St Joseph College, Bangalore, 03/06-2007,  
[http://en.wikisource.org/wiki/Freeing\\_the\\_Mind:\\_Free\\_Software\\_and\\_the\\_end\\_of\\_proprietary\\_culture](http://en.wikisource.org/wiki/Freeing_the_Mind:_Free_Software_and_the_end_of_proprietary_culture), last accessed 10/02-2010

Open Options, 2005 *Arguments about open software*,  
[http://www.netc.org/openoptions/pros\\_cons/comparing.html](http://www.netc.org/openoptions/pros_cons/comparing.html), last accessed 10/02-2010

O'Reilly, T, 2004 *Open Source Paradigm Shift*,  
[http://tim.oreilly.com/articles/paradigmshift\\_0504.html](http://tim.oreilly.com/articles/paradigmshift_0504.html), last accessed 10/02-2010

Silicon Alley Insider, 2010 *Microsoft Operating Profit By Division*,  
<http://link.businessinsider.com/view/aqm.239/c360bccb>, last accessed 10/02-2010

Stallman, R, 2002 *Software Patents - Obstacles to software development*, speech at Cambridge 25/03-2002, <http://www.cl.cam.ac.uk/~mgk25/stallman-patents.html>, last accessed 10/02-2010

Author: Jonas Knutsen  
Date: 11/02 - 2010

Available at [www.jonasknutsen.com](http://www.jonasknutsen.com)

Stallman, R, 2007a *Why Software Should Not Have Owners*,  
<http://www.gnu.org/philosophy/why-free.html>, last accessed 10/02-2010

Stallman, R, 2007b *Why Open Source Misses The Point of Free Software*,  
<http://www.gnu.org/philosophy/open-source-misses-the-point.html>, last accessed 10/02-2010