

[\[https://gegen-kapital-und-nation.org/en/copyleft/\]](https://gegen-kapital-und-nation.org/en/copyleft/)

Free Property - On Social Criticism in the Form of a Software Licence

The open-source/free-software movement has quite a good reputation on the Left.¹ This is not simply because of the fact that open-source developers provide things for free which usually cost money, but also because the free-software movement often is regarded as an opposition or even a practical counter project to capitalist private property. Hence, this text investigates the apparent contradiction that a licence – an assertion of ownership – guarantees universal access, while being simultaneously adopted and promoted by multinational IT corporations for their own profit.

Intangible goods are different ...

Indeed, at least some people within the movement do seem to be bothered about property, at least where it specifically affects digital goods. Indeed, in terms of what they actually are, physical goods and so-called “intangible” goods differ.

If someone uses my bike, I cannot use it at the same time. Ideas, however, such as those expressed in this text, can be distributed and shared with others without ever running out of them. For example, we do not know less of the content of this text when the readers know more about it. But still: reading the text, comprehending it, finding mistakes, that we might have made, are intellectual efforts every time we accomplish them – activities that are both time consuming and full of preconditions, e.g. one is required to have learned how to read. Hence, distribution is not to be had entirely “free” and without any (basic) requirements. The text itself, however, and the information it contains bears the particular feature that it can be copied (and, by implication, transferred, displayed, made available, in short: used) any number of times. Once certain (basic) requirements are established (e.g. a computer is at hand, an Internet connection is up and running), it is fairly cheap to duplicate a file containing this text – the effort becomes close to zero at some point.

... and with them, property appears differently

It seems an ‘artificial’ and unnecessary restriction to stamp private property on ideas, files or other ‘containers of information’ milling about – for the single reason that one is used to copying those files. From this, first of all, it may be noted that the quality of being property is ascribed to things. It is not a characteristic inherent to them, i.e. necessarily or naturally ‘comes with’ things. Secondly, it is apparent that it is not *allowed* to make copies of some files, e.g. most music. It is illegal to distribute such files. With regard to files this seems, at first sight, rather absurd since their distribution neither changes nor damages their content. So, when it comes to ‘intellectual property’, property appears differently. Namely, it appears more obviously that state authority restricts its use through patent, copyright and other laws. This way it becomes very distinctly recognisable what property actually is – a barrier.

Moreover, scientific and technical results were products of collaboration long before the beginning of digital information processing. This is because even the smallest discovery or

invention is based on a host of other discoveries and inventions; so many that the respective originators only know a fraction of the sources from which their content derives. Mathematical findings are based on other mathematical findings, software is based on ideas found in other software packages or relies on those packages directly.² Thus, in order to make progress in research and development, access to what is already known is required. If nowadays intellectual property titles continuously are used and defended, i.e. if access and applicability of existing information is restricted by law, then this prevents the development of new ideas. Property appears as something arbitrarily separating that which essentially belongs together. Not only is property a barrier to access to existing things or knowledge, but is even a barrier to the discovery and development of new ones.

The absence of property relations as norm

The concept of open source emerged alongside the development of mainframes, personal computers and the Internet and it also pushed these developments forward. The starting point for the open-source movement was the acknowledgement of some particular qualities of digital goods, especially their lossless reproducibility and the implications for software development that come with this quality. The movement's protagonists knew how to take advantage of those qualities in their work and, hence, focused on their social requirements. It was a new phenomenon to concern oneself with this topic in the beginning of the field of computer science. From around the 1950s on, free access to and a de facto unrestricted use of all required information went without saying – at least with regard to software. This, anyhow, applied to people with the respective knowledge working at the relevant, well-equipped research institutions. Software simply was a free add-on that came with massive, expensive mainframes. Accordingly, it was openly distributed, studied and changed.

Only from the mid-1970s, a market for proprietary software developed, i.e. software that one is not allowed to freely modify and distribute. Companies such as Microsoft started doing business by selling software and especially licences granting the right to use this software.³ People such as Richard Stallman – founder of the GNU Project, the best-known free-software licence, the General Public License (GPL) – stepped up against this new movement in order to retain the status quo. Stallman and his colleagues developed software together and their demand was that others should be able to study, use and distribute their products. Indeed, from the standpoint of well-planned production of useful things, this is a sensible position.

Property – a standard for the world of physical things?

The open-source/free-software movement started off with the GNU Project. It is important to this movement today that property relating to intangible goods has to play an inferior or different role than property regarding other, i.e. material, things. The reason for this – according to this movement – is to be found in the particularity of intangible goods themselves.

For example, the German Pirate Party – as other Pirate Parties concerned with issues at the crossroad of democracy and the digital life – writes in its manifesto, “Systems that obstruct or prevent the reproduction of works on a technical level (‘copy protection’, ‘DRM’, etc.) *artificially reduce* their availability in order to turn a free good into an economical good. The *creation of artificial shortage for mere economical interests* appears to us as amoral; therefore we reject this procedure. [...] It is our conviction that the non-commercial reproduction and use of

works should be *natural*; and that the interests of most originators are not negatively affected by this – despite contrary statements of particular interest groups.”⁴

With regard to digital goods, the members of the Pirate Party complain that by means of a title of ownership access to information is “artificially” prevented, which goes against information’s “natural” feature of being copyable: “information wants to be free”. At the same time, they see no reason to make the same claim for material things. According to the logic of the party’s political programme, those are “economical goods” quite by themselves. An assumption that seems so self-evident to the authors that they do not explicitly mention it.

The GNU Project, on the contrary, explicitly addresses the assumed distinction between non-material and material: “Our ideas and intuitions about property for material objects are about whether it is right to take an object away from someone else. They don’t directly apply to making a copy of something. But the owners ask us to apply them anyway. [...] But people in general are only likely to feel any sympathy with the natural rights claims for two reasons. One reason is an *overstretched analogy with material objects*. When I cook spaghetti, I do object if someone else eats it, because then I cannot eat it. His action hurts me exactly as much as it benefits him; only one of us can eat the spaghetti, so the question is, which one? The smallest distinction between us is enough to tip the ethical balance. But whether you run or change a program I wrote affects you directly and me only indirectly. Whether you give a copy to your friend affects you and your friend much more than it affects me. I shouldn’t have the power to tell you not to do these things. No one should.”⁵

However, this distinction between material and non-material goods is not correct.

1) The GNU Project claims that a difference between spaghetti and a program is that the former can only be consumed by one person, while the latter can be used by indefinitely many people. Hence, for the GNU Project the former implies the need for private property while the latter does not. Yet, under the regime of property it does not matter whether an owner actually uses her stuff or not. When people think about property in material goods, they have their personal belongings in mind, things they need more or less regularly. But this is not the main point of private property – the way it works is much more far reaching and fundamental. For example, squatted houses get evicted to stand empty again, pieces of woodland are fenced in by their owners even if they live hundreds of miles away or supermarkets lock their bins to prevent people from dumpster diving. The question whether someone could make use of something is subordinate to ownership, not the other way around. Property applies no matter whether the owner or someone else, e.g. in return for payment, uses it. Making successful claims to an absolute disposal over wealth of whatever kind and whatever quantity regardless of neediness – this is private property. Regardless of material or intangible goods, the regime of property does not care who wants to use what and how. Whereas it is true that only one person can eat one’s fill given only one serving of spaghetti, under the regime of private property to own spaghetti is the condition for eating them, but the desire to eat them does not establish ownership. So, in this respect the material vs. non-material distinction is wrong.

2) In one respect though, need does play a role, namely a negative one. Property in a machine indicates the exclusion of third parties from using that machine. One cannot enter into an ownership relation with a machine because a machine is not eligible for a legal relationship. It is the same with a disc containing a copy of a Windows operating system on it. One is not allowed to install it merely because this disc lies around somewhere unused. The particular function of a title of ownership for the owner is strictly that others may not use her property without her

consent, even though they might want to and perhaps even be physically able to do so. What friends of free software notice and highlight with regard to digital goods could also be observed with regard to ordinary material things: it is a fact that property is a relationship between people in regard to things, but not immediately between things and people. If no one else is there, it does not really matter what belongs to me or what I simply use. This only becomes relevant when others want to have access, too. Property is a barrier between those who want to use a thing and the thing itself, between need and the means to satisfy it. The guarantee for property in material things does not exist despite but because people want, need, require them. To own bread and all the more to own a bread factory is significant because other people are hungry. Otherwise, what would be the point of guaranteeing the right of exclusive disposal?

3) Furthermore, with respect to reproducibility a rigorous contrast, material vs. intangible, does not exist either. It is possible to produce things and this means nothing else than to eradicate the detected scarcity. There is no such thing as a particular finite number of bread knives in the world, more can be manufactured. Indeed, one has to do something for it, but nothing simply is “in short supply”.⁶ However, in order to manufacture something one has to have access to the means of production which, again, are privately owned. And in this regard – again – it does not matter whether one ‘really’ needs them or whether they are currently in use.

Yet, there is indeed a difference between software and bread knives: the contemporary means of production for software meanwhile are cheap mass products that most people have at home anyways. One can write a lot of state-of-the-art software with a five year old computer from a car boot sale.⁷

Thus, the production of software ‘only’ requires an investment of education and labour time, while when it comes to, e.g., bread knives one is excluded from the means of production at the level of the state-of-the-art. In order to be able to produce bread knives one would indeed need the corresponding factory, and this wants to be bought first.

4) The means of production are not simply “in short supply” either but can also be produced, by and large. One is excluded from the means of production as their purpose for the owner is access to the wealth of society in the form of money. The owner knows she has to come to agreements with others in order to get their products. Hence, she uses her factory – as well as people who do not have one, i.e. workers – to manufacture something that she can sell. With the proceeds she then can either buy goods for herself or she can reinvest in workers and means of production so that another round of fun may commence. In a society based on the division of labour one is dependent on others and their products, be it intangible or material goods. Because in this society this trivial fact does not lead to a self-conscious interaction of producers but rather the regime of property prevails, one is excluded from the products of others and therefore is required to exploit their needs to one’s own advantage. This absurdity can be put differently: it is precisely because one is dependent on others that one insists on the exclusion of others from what one owns. If everyone gives only if given an equivalent in return, then certainly it makes sense to deploy what one has as means of access to the stuff under the control of others by matching their exclusion with one’s own.

Property is characterised by exclusion whether it concerns material or immaterial goods. The free-software movement disagrees though – and it shares this fallacy with the majority of people. In other words: the political wing of the free-software movement insists on drawing a strict distinction between digital and material goods in order to criticise the regime of property regarding digital goods. Yet, it is exactly their line of argument that reaffirms the exclusion from the things people need: the regime of property. Some radical activists want to use free software as

a tool for the abolition of private property, for example the slogan “free software today, free carrots tomorrow” can be read this way. This is futile as the the *reference* to the free-software movement’s ‘criticism of property’ takes up the false idea of free software proponents that carrots can never be free and for all instead of critiquing it.

Copyleft licences – critique of property law by legal means

Access to open-source software is defined and regulated in legal terms. First of all, copyright law applies regardless of what the author chooses to do. This law forms the general basis and is applied by the state to anything it considers to have a creator. But moreover, an open-source licence determines what anyone else is allowed and not allowed to do with, say, a piece of software by means of the law – no difference from other areas of bourgeois society. Usually open-source licences allow to read, modify and further distribute the source code.⁸

The various licences differ considerably in terms of their precise provisions. Roughly, there are two versions of openness. The above mentioned GPL determines that any program using software parts licensed under the GPL has to entirely be licensed under the GPL or a compatible licence as well. This means that the licence is ‘virulent’ and components mutually affect each other. It is, for instance, not allowed to simply take the Linux kernel (i.e. the operating system’s core) modify it here and there and then distribute the result without also releasing the source code of the

modifications. In contrast, the BSD-family of licences is less strict.⁹ BSD programs are part of Microsoft Windows, for example, and there is no obligation to publish any source code. The licence mainly stipulates what must happen *if* source code is distributed, namely that copyright holders must be named. Secondly, it provides that no one may sue the authors in case something goes wrong. An exclusion of liability: the software is provided “as is”. Both camps – GPL vs. BSD – do not get tired arguing these differences. The GPL camp holds that liberty is to be protected by force whereas the BSD camp is convinced this way liberty is lost.¹⁰ Who is right, whether this question even can be settled or not or whether it cannot be conclusively answered because this type of freedom includes its opposite – domination – is perhaps better saved for another text. Here, we may conclude, though, that this kind of practical criticism of property necessarily presupposes a title to ownership in a software product. This is the reason why Richard Stallman calls the GPL a “legal hack”, i.e. a trick on legal grounds¹¹: one insists on one’s property by way of claiming the terms of a licence in order to guarantee free access.¹²

But, “you can’t hack the law”¹³. The legal system – guaranteed by the state’s authority – cannot be tricked: licences (no matter what kind) are legally binding contracts following the logic of the law that, if in doubt, always can be enforced in case one of the contracting parties claims its right.

¹⁴ The result of this is that, e.g. scientists who make their research-software available to others have to deal with a maze of different incompatible licence versions. Hence, questions such as the following arise: am I legally allowed to combine another scientist’s open-source software with my own?¹⁵ A creative use of and tricking the law – Stallman & Co. creatively use the law – turns into principal submission to the law – the law dictates Stallman & Co. its terms – that is how the law works.

Moreover, such a “hack” develops its very own dynamic in a society of law appreciating citizens. The field in which licences are applied in this manner has meanwhile massively grown. The Creative Commons movement¹⁶ recommends scientists, creative artists as well as hobby

photographers uploading their holiday snapshots to the Internet to claim ownership of their respective products of information. They are encouraged to exclude third parties more or less from using such products by choosing from a toolbox of legal restrictions. Contrary to Richard Stallman, the Creative Commons initiative by Lawrence Lessig does not problematise the really existing copyright regime. Hence, the initiative quite correctly notes: “Creative Commons licenses are copyright licenses — plain and simple. CC-licenses are legal tools that creators can use to offer certain usage rights to the public, while reserving other rights. Without copyright, these tools don’t work.”¹⁷ Meanwhile, even things that a few years back no one would have expected to be ruled by copyright law, such as the above mentioned holiday snapshots, are now subsumed under its regime.¹⁸

How deeply ingrained the formalism of the law is in these peoples’ minds is aptly expressed by the controversy around the DevNations 2.0 Licence and its subsequent withdrawal.¹⁹ The DevNations 2.0 Licence stipulated that people from ‘developing countries’ were allowed to use products under the licence free of cost whereas people from capitalist centres were not entitled to this. Hence, it was a licence that at least acknowledged real material differences.²⁰ The licence was withdrawn because of its discrimination against people living in rich countries. Hence, it violated the equality before the law; but this equality, i.e. non-discrimination, is a requirement for any licence hoping to be verified as an open-source licence by the Open Source Initiative. If the open-source movement is said to have started off with a criticism of property – even if restricted to intangible goods — or that it was bothered by people being excluded from the digital wealth of societies, then it is safe to say it achieved the opposite: you cannot hack the law. What remains is to (practically) critique it.

Software commons for profits

The open-source movement succeeds because it gets along well with an IT industry whose prosperity is otherwise based on every known principle of private exploitation. In the following we give some short examples to illustrate how business and open source work hand in hand, i.e. to unpack the apparent contradiction of making money from something that is made available for free.

The Mozilla Foundation – best known for its web browser Firefox – receives a good deal of its income from Google Inc., as Google Inc. pays so that the browser’s default search engine is Google. Apple’s operating system OS X is built upon an open-source foundation: Darwin. Apple now and then even collaborates in open-source projects using the results of this collaboration to sell hardware, software packages, films and music – lately rather successfully we hear. Furthermore, according to a study only 7.7% of the development of the kernel of the Linux operating system was explicitly non-paid volunteer work.²¹ Red Hat Linux, IBM and Novell are the biggest companies directing their employees to collaborate on this operating system, each one of them a global player on the international IT-market. They co-develop Linux in order to do profitable business with it. For example, they sell applications that run on Linux or provide support contracts to companies: you buy our product, we make sure everything runs smoothly. Companies pay for this service even though it would be possible to compile the result by means of open-source projects themselves – to save the hassle. Google distributes its operating system Android and its web browser under an open-source licence, especially so that users of smart-

phones use Google's products by which Google directly or indirectly makes money by means of advertising. Many companies contribute to developing the GCC-Compiler because it is a central piece of infrastructure for every software company.²² Co-development is cheaper than to independently create alternatives. Meanwhile even Microsoft published some products under open-source licences.

Modern politicians concerned with the economic success of their respective nation-states have understood the power of open source – by all means, they promote and encourage the blossoming and expansion of this infrastructure which is available to all. Firstly, this is to strengthen the economy of their nation-state, secondly, it simply is cheaper for their own administrative bodies to use open-source products. By the way, long before the C64²³, bourgeois states provided fundamental research and knowledge for the benefit of the national economic growth by means of its university system. It is hence fitting that the two most popular open-source licences (GPL and BSD) were developed at American top-tier universities (MIT and Berkeley).

The bourgeois state also realised that its patent law not only enables the private exploitation of innovations but also serves as a barrier – and in this regard it does appreciate the worries of open-source/free-software activists. For if existing innovations cannot be used for the development of new ones that means bad prospects for economic growth. So the bourgeois state implemented a patent law that grants patents for a certain period of time only. Regarding the exploitation and perpetuation of technology it provides a mediating form for the competing interests of individual capitalists – in the interest of total social capital. On the one hand, individual capitalists want to massively exploit their patented inventions by excluding every non-payer from the use of those patents. On the other hand, they want to use others' patents as basis and means for their own success.

Within the cultural sector, where CC-licences are widely used, things are the same. Incidentally, this also applies to those that choose a non-commercial CC-licence for their products which allows the use on a non-commercial basis only and serves the purpose to exclude others from monetarily profiting from ones own output. This right is reserved to the person uploading a holiday snapshot or producing a music track. The whole concept has nothing to do with the critique of a society that is based on the principles of reciprocal exclusion from useful things and in which every individual necessarily relies on her own property or labour-power. There is no critique of the social conditions in which we live to be found in insisting on the right of the creator – this is the owner's competitive position vis-a-vis the competition.

¹ The open-source/free-software scene partly acrimoniously fights over the question whether it is “open source” or “free software” that they develop. The former is a particular mode of developing software the latter a comprehensive approach to software in general; it is a demand, sometimes even called “philosophy”, for what one shall be able to do with software. In our text we often use the term “open source” simply because it is better known. To be entirely correct, we would have to almost always write “free software”, though, as our criticism is directed towards the comprehensive claim of this movement, as opposed to the simple endeavour of making software development more effective.

² With regard to the production of software it is common (and quite sensible) to put frequently used features into separate packages which then are used in various products. Those packages of features are aptly called libraries.

3 Bill Gates' letter to the Homebrew Computer Club is an interesting historical document highlighting the necessity to justify privatisation in the beginning of this new development: <http://www.digibarn.com/collections/newsletters/homebrew/V2_01/gatesletter.html> (last access 14. August 2013).

4 cited after <https://wiki.piratenpartei.de/Parteiprogramm> (last access November 2012), our translation, emphasis added.

5 <https://www.gnu.org/philosophy/why-free.html> (last access November 2012), emphasis added.

6 Hence, it is ridiculous that economists, for example, constantly present beach houses and famous paintings to illustrate their theories. They choose examples that indeed have the feature of being in short supply in order to say something about things such as bread, flats, cars and clothing. In other words, they use things as examples whose quantity cannot easily be increased by production in order to explain the economy, i.e. the sphere where things are produced.

7 This is currently changing so that this statement may no longer be true in a couple of years. If software runs on large networks of computers that together calculate something then a ten year old computer may not be the adequate means of production any longer.

8 Source code means the software program in a certain language that humans are more or less able to read ...well, except Perl.

9 BSD stands for Berkeley Software Distribution

10 Which licence to choose sometimes simply may have economic reasons. Most of the open-source software in the field of applied mathematics is licensed under a BSD-style licence as companies within this sector often do not intend to sell but use the software themselves. They also only collaborate on the terms that they may do so quite unrestrictedly. On the contrary, most of the open-source software in pure mathematics is licensed under the GPL: the only companies interested in these software packages are those making money from selling such software. That way the (often academic) authors protect themselves from being sold their own software as part of such commercial software.

11 It does not come as a surprise that he attempts to creatively apply the law. After all, he does not have a problem with the fact that daily needs cost money, i.e. that someone insists on his "every right" to get paid: "Many people believe that the spirit of the GNU Project is that you should not charge money for distributing copies of software, or that you should charge as little as possible – just enough to cover the cost. This is a misunderstanding. Actually, we encourage people who redistribute free software to charge as much as they wish or can." – <http://www.gnu.org/philosophy/selling.html> (last access November 2012).

12 By the way: in no way does an open-source licence mean that one gives up ownership. The licence terms always apply to others, i.e. the users, only, whereas the owner is of course free to do whatever she wants with her property. This is the base of a business model by which one makes available a (restricted) version of a product as open-source software and at the same time a(n optimised) version is sold as usual.

13 Cindy Cohn, Legal Director for the Electronic Frontier Foundation. It should be noted, though, that her meaning of hacking the law is rather different, if not contrary, to ours. See <http://s.shr.lc/10xUcQo>.

14 In the leading capitalist countries, the GPL “trick” meanwhile has been accepted as legally binding. This means that it is possible to sue someone in case of violations against the General Public Licence. If such a lawsuit is successful a party can be forced to release all source code of its product incorporating GPL code.

15

It is possible that the answer to this question is “no”, an example from the area of mathematical software highlights this:

<http://gmp.lib.org/list-archives/gmp-discuss/2008-May/003180.html>
(last access November 2012)

16 The Creative Commons (CC) movement emerged in response to branches of industry where direct producers such as musicians usually sign over considerable rights to record corporations – i.e. lose the ownership in their own products. That is somewhat similar to a factory worker who also does not own one single product he manufactured. In contrast, CC-licences first of all mean the claim of ownership of one’s own product.

17 <http://creativecommons.org/weblog/entry/22643> (last access November 2012)

18 On Flickr – a not as popular as it used to be photo sharing website – one is bothered with the question which licence ought to be applied to one’s photos, a rather absurd thought in the first instance.

19 See <http://creativecommons.org/licenses/devnations/2.0/> and <http://creativecommons.org/retiredlicenses> (last access November 2012)

20 Our elaborations on property earlier indicate that poverty cannot be abolished by means of such licences.

21 In case of 25% of the work it remains unclear if anyone or anything was paid. See <http://lwn.net/Articles/222773/> (last access November 2012)

22 GCC stands for the GNU Compiler Collection, a collection of compilers by the GNU Project. A compiler translates programs from the source code into a format which then can be executed on the respective computer. Free software does not make much sense without a free and reasonable compiler. If the compiler is not openly available it is in fact possible to change software in its source code, but the changes cannot be applied – unless you buy a licence for a compiler. If it is a poor compiler open-source programs are disadvantageous to the proprietary competition.

23 The Commodore 64 was a popular personal computer in the 1980s.