

Issue 021: Open Source

Adrian Kosmaczewski

June 1st, 2020



Welcome to the twenty-first issue of *De Programmatica Ipsum*, dedicated to the subject of *Open Source*. In this edition:

- Graham explores the implications behind the distinction¹ between “free” and “open source” software.
- Adrian argues that the common discussions about free and open source software are a distraction at best².
- In the Library section³, Graham celebrates the unnamed authors⁴ of free and open source documentation.

Enjoy this issue! Please subscribe to our free newsletter⁵ to stay updated about new releases, or contribute⁶ if you would like to support our work.

Cover photo by Markus Spiske⁷ on Unsplash⁸.

¹<https://deprogrammaticaipsum.com/once-the-rockets-go-up-who-cares-where-they-come-down/>

²<https://deprogrammaticaipsum.com/open-always-wins/>

³<https://deprogrammaticaipsum.com/category/library/>

⁴<https://deprogrammaticaipsum.com/the-community/>

⁵<https://deprogrammaticaipsum.com/newsletter/>

⁶<https://deprogrammaticaipsum.com/contribute/>

⁷https://unsplash.com/@markusspiske?utm_source=unsplash&utm_medium=referral&utm_content=creditCopyText

⁸https://unsplash.com/s/photos/open-source?utm_source=unsplash&utm_medium=referral&utm_content=creditCopyText

Once The Rockets Go Up, Who Cares Where They Come Down?

Graham Lee

June 1st, 2020



Somehow, I have yet to come down with COVID-19 (as an asthmatic I'm not expecting it to end well). One of my last big gatherings before the travel bans and lockdowns were enacted

was at FOSDEM¹ in Brussels, a chance for nearly 10,000 people in the supra-European software community to get sick. I'm in my element at FOSDEM, as I'm one of those irritating people who makes a distinction between "Open Source" and "Free Software", and uses the terms in different contexts.

But is there really a difference? Obviously the term Free Software came first, and is summarised in the Free Software Definition² as the Four Freedoms (themselves often summarised in Free Software Foundation of Europe literature as "Use, Study, Share, Improve"). The Debian Free Software Guidelines³ are part of the Debian Social Contract, and largely explains the expectations the Debian project has on how software that forms part of their project can be used. In this way, it constrains the license limitations that can appear in Debian project components, or constrains the Debian project's use of software to that with compatible license terms.

The Open Source Definition⁴ is, as stated in the linked page, itself derived from the Debian Free Software Guidelines. They are almost the same document. A quick diff reveals that where the DFSG says licenses must not be specific "to Debian", the OSD says they must not be specific "to a particular product", then one clause is specific to each document. Bruce Perens, one-time Debian Project Leader, wrote both documents.

The DFSG has a clause listing example licenses. The OSD does not, but the OSI's site lists lots of Open Source-compatible licenses and they frequently have discussions on their mailing lists or at conferences on what licenses are or aren't open source. Note that the OSI do not have any proprietorial control over the phrase "open source" so can't enforce any use or proscription of the term, but are seen as influential in the field. The OSD has a clause "The license must be technology-neutral", which the DFSG lacks.

It seems that there isn't any significant, material difference between the definition of Free Software and the definition of Open Source, so why be pedantic about the distinction? The GNU project offers one suggestion, saying that Open Source misses the point⁵: Free Software is a social justice movement offering freedom to users of software. Open Source minimises this social angle, instead focusing on the practical aspect of the availability of source code.

I think that the "open source misses the point" essay was probably correct at time of writing. Open Source was certainly created as an unloaded term, avoiding the connotations of "freedom" in a justice sense and "free" in the sense of zero-cost. Netscape wanted a new business model, and it wouldn't have been compelling for Eric S. Raymond to start his pitch with "there's no money in this, but why don't you..."

However, whether or not Open Source was intended as a "neutral" term – and whether there can be such a thing – the phrase now stands in opposition to the Free Software movement and its mission that it initially intended to support through tacit acceptance. In modern usage, Free Software is the creation and curation of an intellectual commons defined by software that enables the four freedoms. There isn't so much focus on justice; the Free Software Foundation has long-running campaigns against bulk surveillance⁶ and digital restrictions

¹<https://fosdem.org/2020/>

²<https://www.gnu.org/philosophy/free-sw.html>

³https://www.debian.org/social_contract#guidelines

⁴<https://opensource.org/osd>

⁵<https://www.gnu.org/philosophy/open-source-misses-the-point.html>

⁶<https://www.fsf.org/campaigns/surveillance>

management⁷, but much of the discussion at a conference like FOSDEM will focus rather on the availability of free software for particular use cases, the enforcement or suitability of particular licences, and ways to advocate for free software in various contexts.

If Free Software is the maintenance of an intellectual commons, then open source is the enclosure of that commons by and for corporate interests. This is definitely not the intention of the Open Source Initiative, who claim to be defending the “our” in “source”. In practice, companies use the phrase “open source” when they want to extend their existing business into the free software commons. We see certain practices as typical examples of this enclosure.

One is the formation of not-for-profit “Foundations” designed to give an impression of charity to what is still a for-profit business pursuit. Many of these are business interest associations⁸, non-profits certainly but bound by law to advance the interests of their corporate sponsors. The Linux Foundation, supposedly a “neutral home for collaborative development”, is thus actually a corporate interest group. Projects developed under the aegis of the Linux Foundation (including the OpenJS Foundation, Jenkins, the RISC-V Foundation, Let’s Encrypt, and HospitalRun) are not neutral, but are run in the interests of sponsors like Google, Facebook, AT&T, Cisco, Samsung, and company.

Another practice is the corporate Contribution License Agreement. The Free Software Foundation instituted copyright assignment on the GNU project to improve their position when enforcing copyleft⁹ in US-based disputes. Inspired by this precedent, corporations use Contributor License Agreements (CLAs) to gain proprietorial control over contributions from third parties, ensure a lack of competing ownership interest, and enable future relicensing including commercial licensing of the work.

The use of open source contributions in hiring decisions is another way in which companies have enclosed the free software commons. In addition to hobbyist and line-of-work contributions, employees are now expected to do additional open source work as a form of resume building. This work must be visible on Github so that it is easy for hiring corporations to discover and use the work for free, and it must be on popular projects (those used by a lot of other corporate open source users) to be deemed valuable and raise the profile of the contributor. Large companies freely admit that they use their open source projects as recruitment exercises, relying on the free labour contributed by the non-employee pool as information on who to groom for their notoriously low-value interview processes¹⁰.

You will also see corporations downplay their interests in a project that they have commercialised, pretending that the software is entirely community-supported to both externalise their costs onto individual community members and minimise the impression that the project has been commercially enclosed. In a recent announcement of a new version of the GCC compiler¹¹, an employee at Red Hat—a \$34Bn division of the \$109Bn IBM corporation—asked readers to “consider a donation to the GNU Toolchain Fund to support the continued development of GCC”. Of course, GCC will continue development as long as IBM needs it to support their Red Hat, Power and Blue Gene products, and will continue developing in the direction they and the other corporate interests (Intel, Oracle, ARM, others) dictate due to their stuffing of the ballot boxes, but sure, let’s pretend that

⁷<https://www.defectivebydesign.org/>

⁸<http://ebb.org/bkuhn/blog/2015/02/10/node-foundation.html>

⁹<https://www.gnu.org/licenses/why-assign.html>

¹⁰<https://deprogrammaticaipsum.com/tales-of-the-interview/>

¹¹<https://gcc.gnu.org/pipermail/gcc-announce/2020/000163.html>

chipping in a few dollars keeps this a community-interest endeavour. (Meanwhile, De Programmatica Ipsum is the work of two people who are unpaid, and would welcome your contributions¹².)

Finally, enclosing open source companies try to apply anti-competitive clauses in license terms: the Commons clause¹³ for example, the Server-Side Public License¹⁴ and Defold License¹⁵ are attempts to make particular business models feasible while still sharing the source code to the products on which the businesses are based. These practices are quickly found to be incompatible with both the Free Software and Open Source definition, and cause ructions in the community. There may well be businesses that have made these clauses stick, but often the reaction from “the open source community” causes these companies to backtrack.

The reason that these last attempts to subvert free software fail is that anti-competitive clauses are so clearly not within the Four Freedoms. Freedom Zero is “the freedom to use the software, for any purpose”, which is expressed in the DFSG and the OSD as “No Discrimination Against Persons or Groups” and “No Discrimination Against Fields of Endeavor”. As a common wealth, the corpus of Free Software must be made available to all who can make use of it, regardless of whether we agree with that use. Denying a license to someone who will compete with our business is an understandable commercial decision, but not the basis of a free commons.

This aspect of free software – freedom to all – is therefore itself under attack by corporate interests who wish to undermine the movement in order to enclose and exploit the commons. Under the banner of supposedly reasonable moral concerns, they suggest that free software’s “failure” is that it doesn’t impose any ethical constraints, and propose that it would be better to limit open source software to “good” uses. This movement started, perhaps, with Douglas Crockford’s satirical JSMIn license¹⁶:

The Software shall be used for Good, not Evil.

It has since been taken seriously, with the Hippocratic License¹⁷ among others. For contentious people who like to think that they abide by a consistent ethical code, this idea can seem alluring. Why *not* restrict use of open source software to ethical applications? The answers, though, are manifold.

First is the question of moral authority. Who is going to deem a specific practice acceptable or unacceptable? Do they apply a Benthamite utilitarian analysis to identify “moral” or “immoral” software, or do they assume a Kantian categorical imperative? Or is there a religious moral code that underpins the morality of software, and does it apply to software creators and users who practise a different religion, or no religion? Or do we follow Nietzsche, and accept that there is no “one-size-fits-all” moral code, indeed that it is immoral to demand that some people follow the same moral code as others?

The Hippocratic license indirectly chooses an answer this question by appealing to authority, but only in the case of disagreement between author and licensee over fitting their use case into the framework of the United Nations convention on human rights. Free software licenses are already difficult to enforce, without requiring arbitration in the Hague, or every contributor to interpret international convention and argue their cases with myriad users.

¹²<https://deprogrammaticaipsum.com/contribute/>

¹³<https://commonsclause.com/>

¹⁴<https://www.mongodb.com/licensing/server-side-public-license/faq>

¹⁵<https://defold.com/license/>

¹⁶<https://github.com/douglascrockford/JSMIn/blob/master/jsmin.c#L16>

¹⁷<https://firstdonoharm.dev/version/2/1/license.html>

Whether or not moral decisions are appropriate when choosing who gets to use software, they are certainly expensive to produce, reproduce, and enforce: a situation that of course tips the balance in favour of the large corporation. Indeed proprietary software licences are already supported by an infrastructure that chooses which customers to acquire and retain, so the Hippocratic license looks more like a particular source-available choice for commercial software than an adaptation of free software.

Also at issue is the appropriateness of the question. If free software users all have to conform to the same moral position in order to enjoy their freedoms, are they truly free? Bradley M. Kuhn, policy fellow at the Software Freedom Conservancy, draws a parallel with the defence of free speech¹⁸:

Ultimately, fighting for software freedom is a social justice cause similar to that of fighting for free speech and other causes that require equal rights for all. We will always find groups exploiting those freedoms for ill rather than good. We, as software freedom activists, will have to sometimes grit our teeth and defend the rights to modify and improve software for those we otherwise oppose. Indeed, they may even utilize that software for those objectionable activities. It's particularly annoying to do that for companies that otherwise produce proprietary software: after all, in another realm, they are actively working against our cause. Nevertheless, either we believe the Four Software Freedoms are universal, or we don't. If we do, even our active political opponents deserve them, too.

Taking the principled stand that software freedoms are rights in themselves requires ensuring those rights are defended for everyone. Defending those rights for everyone makes it clear that you are a person of principle, which lends greater weight to your argument when you speak out against immoral behaviour.

It is correct that software creators, particularly professionals, should practice their craft ethically, should refuse to engage in immoral practices, and should use their skills to combat unethical behaviour. We explored that topic in issue 5¹⁹. But we should not deny the basic freedoms of our field while claiming that to do so is ethically just.

Photo by Ronan Furuta²⁰ on Unsplash²¹.

¹⁸<http://ebb.org/bkuhn/blog/2018/08/30/on-social-justice-software-licensing.html>

¹⁹<https://deprogrammaticaipsum.com/issue-5-ethics/>

²⁰https://unsplash.com/@ronan18?utm_source=unsplash&utm_medium=referral&utm_content=creditCopyText

²¹https://unsplash.com/s/photos/missile?utm_source=unsplash&utm_medium=referral&utm_content=creditCopyText

Open Always Wins

Adrian Kosmaczewski

June 1st, 2020



Few debates in the computer industry are as passionate as those around Free and Open Source Software (otherwise referred to as “FOSS”) and how it “benefits society” or promotes “ethics” or some other optimist outcome for the future of mankind. The problem is that this ongoing debate for the past 20 years has only served the purpose of making people look in the wrong direction.

During my first days in University, around 1994, a friend told me about a hobby operating system he installed in his computer, something called “Linux.” He had downloaded it from that venerable anonymous FTP site called “ftp.funet.fi”¹. He told me the long hours he had to spend to (I quote) “recompile the Kernel” and that he was happy because he was running “free software.”

I barely knew what a “kernel” was, even less why it would need to be recompiled, and I obviously thought that free meant “without cost.” When you are a cash strapped university student, free is always good; it does not really matter if it is a free lunch, a free beer, or a free kernel. Free is good™®, or so they say.

I started my career writing software on Windows 95, and running it in Windows NT 4.0 boxes running SQL Server 6.5 and IIS; a rather closed source, “proprietary” galaxy. I kept on hearing stories of how Linux on Apache was better (read: faster, snappier, cheaper, etc.) than IIS; it might have been, but the business did not care. As closed as the whole Microsoft galaxy could be, the company I worked for could get a working website for a fraction of the infamous Total Cost of Ownership (aka TCO) of what it would have cost using Linux.

¹<http://ftp.funet.fi/>

The tradeoff between both galaxies was simple: either buy sanctioned Wintel hardware with sanctioned Wintel software, pay a monthly fee for a MSDN subscription and get regular support and updates. Or get a Red Hat Linux CD-ROM² from Walnut Creek³, deal with non-existing Linux device drivers, and search for solutions in AltaVista⁴, evolt⁵, or Slashdot⁶.

Everything changed in 1998. That is when Christine Peterson coined the phrase⁷ “Open Source,” deliberately masking the business-unfriendly moniker of “Free Software” which Wall Street abhorred so much. More or less at the same time, Netscape decided to rewrite its own product. Such wise decision⁸, widely celebrated by the “community” helped Microsoft win the infamous browser wars; this situation ultimately begat Internet Explorer 6 and its horrible crush on web development for the next decade.

The dot-com boom contributed to the success of FOSS; or did it? Well for a time it seemed so. Instead of using non-existent venture capital to bootstrap your company, just use FOSS. This is how AWS won over datacenters. This is how Ruby on Rails won over Java. This is how WordPress won over Movable Type⁹. This is why 2005 is still remembered as the year of Linux on the Desktop. Well, not really, that’s actually 2020¹⁰. FOSS became an interesting alternative mostly because proprietary options were quite bad by all standards; the whole Windows Vista scandal convinced most users that... Macs were worth considering again, which made Apple win a lot of money.

Apple; who was by then smart enough to open up key parts¹¹ of their software, but not more than necessary. Of course, always taking whatever they needed to get things done. Hence Konqueror begat WebKit and much joy was had, and Apple’s share prices went up.

The pressure of the mob in the streets of Redmond right after the crash of 2000 was strong enough for Microsoft to do two things, more or less at the same time. First, Ballmer treated all FOSS as communism¹² and later as cancer¹³. I do not blame him; after all, even Robert Metcalfe said it¹⁴ and he knows a thing or two about technology:

The Open Sores Movement asks us to ignore three decades of innovation. It’s just a notch above Luddism. At least they’re not bombing Redmond. Not yet anyway.

The second thing Ballmer did was to release parts of the .NET Framework under a “Shared Source¹⁵” license. To this day nobody understood exactly why, but thanks, why not.

Almost 15 years later, Ballmer resigned and shortly afterwards Satya Nadella bought

²https://archive.org/details/Linux_Red_Hat_5.0_CD-ROM_Walnut_Creek_February_1998

³https://en.wikipedia.org/wiki/Walnut_Creek_CDROM

⁴<https://en.wikipedia.org/wiki/AltaVista>

⁵<https://evolt.org/>

⁶<https://slashdot.org/>

⁷<https://opensource.com/article/18/2/coining-term-open-source-software>

⁸<https://www.joelonsoftware.com/2000/04/06/things-you-should-never-do-part-i/>

⁹<https://movabletype.com/>

¹⁰<https://www.theverge.com/2020/5/19/21263377/microsoft-windows-10-linux-gui-apps-gpu-acceleration-wsl-features>

¹¹<https://opensource.apple.com/>

¹²https://www.theregister.co.uk/2000/07/31/ms_ballmer_linux_is_communism/

¹³https://www.theregister.co.uk/2001/06/02/ballmer_linux_is_a_cancer/

¹⁴<https://web.archive.org/web/20070316025237/http://www.infoworld.com/articles/op/xml/99/06/21/906210pmetcalfe.html>

¹⁵https://github.com/SSCLI/sscli_20021101

GitHub¹⁶ for 7.5 billion USD. Then they decided to copy¹⁷ the open source AppGet¹⁸ package manager and release it as another open source project¹⁹ called WinGet, touted as a better²⁰ option than the MS Store. Sherlocking²¹ is not just for Apple, after all; even better in the case of open source, you can actually read the code while building a competitor. Of course, in both occasions, Microsoft's share price went up accordingly.

Microsoft was late to the game; Oracle had bought Sun Microsystems²² in 2009, paying more or less the same amount of money. Among all of the assets that Oracle got through that acquisition, there was MySQL; the "community" (whatever that is) reacted with anger, distrust, disgust, and dismay, and promptly "forked" the project, henceforth called MariaDB²³.

The Times They Are a-Changin' and FOSS became big business.

It is such a big business, that IBM pulled a record 34 billion USD out of the pocket and bought Red Hat²⁴. That is, almost the double of the record 19 billion that Facebook paid for WhatsApp²⁵ a few years earlier. FOSS is big business, because the truth is that people will use good tools, whether they are proprietary or not; the purpose of computing is to solve problems, and it turns out that people will happily pay actual money to get help from somebody.

The core of the issue is this one: if a tool is good at solving a particular problem, then others will pay for support contracts around them. That's true whether they are "closed" or "open" solutions. This fact does not matter; they are solutions, and all considerations of goodwill and positive externalities of FOSS for mankind are simply discarded, because ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. I'm not screaming, the BSD license²⁶ is.

Google is another great advocate of FOSS; hence WebKit begat Chromium, and much more joy was had. Chrome later crushed Firefox, and Microsoft ditched Edge's own rendering engine to use Chromium instead²⁷, and Google's stock price continued to rise. And Chromium is the new IE6²⁸, and websites now advertise themselves as "better viewed with..." just like in 1998.

In 2009, a Google blog article²⁹ stated the phrase "Open will win" together with a long list of naive predictions. One year after, Andy Rubin famously defined³⁰ what open means:

¹⁶<https://news.microsoft.com/2018/06/04/microsoft-to-acquire-github-for-7-5-billion/>

¹⁷<https://www.theverge.com/2020/5/28/21272964/microsoft-winget-windows-package-manager-appget-copied>

¹⁸<https://appget.net/>

¹⁹<https://github.com/microsoft/winget-cli>

²⁰<https://www.theverge.com/2020/5/20/21264739/microsoft-windows-package-manager-preview-download>

²¹<https://www.howtogeek.com/297651/what-does-it-mean-when-a-company-sherlocks-an-app/>

²²<https://www.oracle.com/corporate/pressrelease/oracle-buys-sun-042009.html>

²³<https://blog.mariadb.org/mariadb-foundation-to-safeguard-leading-open-source-database/>

²⁴<https://www.cnn.com/2019/07/09/ibm-closes-its-34-billion-acquisition-of-red-hat.html>

²⁵<https://money.cnn.com/2014/02/19/technology/social/facebook-whatsapp/index.html>

²⁶<https://opensource.org/licenses/BSD-3-Clause>

²⁷<https://www.theverge.com/2018/12/6/18128648/microsoft-edge-chrome-chromium-browser-changes>

²⁸<https://www.theverge.com/2018/1/4/16805216/google-chrome-only-sites-internet-explorer-6-web-standards>

²⁹<https://googleblog.blogspot.com/2009/12/meaning-of-open.html>

³⁰<https://twitter.com/arubin/status/27808662429>

the definition of open: “mkdir android ; cd android ; repo init -u
git://android.git.kernel.org/platform/manifest.git ; repo sync ; make”

Much³¹ fun³² was³³ had³⁴ by Apple pundits in the following decade about those perceptions. While Google might allow anyone to “compile the kernel,” handset manufacturers must abide³⁵ to quite strict Original Equipment Manufacturer (OEM) contracts. Do what I say, not what I do.

What neither the Google blog nor Andy Rubin explained is: what is won when “Open wins?” And by whom?

Michael Tiemann was right³⁶. Open always wins... money.

FOSS is an economic model of production based on unpaid work provided willingly by a certain amount of (sometimes starving) enthusiast developers. All impatient to add a line to their CV saying that they contributed to this or that project so that they can brag about it in Slashdot Reddit. Some of them, thanks to those pull requests, might occasionally land a job at Red Hat IBM, GitHub Microsoft, Google, or some other FOSS company, because meritocracy and things like that. Better yet, they pat each other on the shoulder, happy to have “made the world a better place” thanks to “open tools” that they can “fork” if some “nasty corporation” messes up with them. Those who don’t get recognition rant about it on Medium, so that everybody gets angry online, corporations keep on being nasty, and the cycle repeats again.

As a production model, FOSS would have surprised them all: Adam Smith, David Ricardo, Karl Marx. Nobody could see this one coming, not even the most optimist of trillionaires³⁷.

Cover photo by Chris Barbalis³⁸ on Unsplash³⁹.

³¹<https://daringfireball.net/linked/2009/12/22/open>

³²<https://twitter.com/JonyIveParody/status/1041625556259930112>

³³<https://twitter.com/asymco/status/1130234790374854656>

³⁴<https://twitter.com/counternotions/status/1146425869411315712>

³⁵<https://arstechnica.com/gadgets/2018/07/googles-iron-grip-on-android-controlling-open-source-by-any-means-necessary/>

³⁶<https://opensource.com/resources/ebook/open-always-wins>

³⁷<https://eu.usatoday.com/story/tech/2020/05/14/jeff-bezos-worlds-first-trillionaire-sparks-heated-debate/5189161002/>

³⁸https://unsplash.com/@cbarbalis?utm_source=unsplash&utm_medium=referral&utm_content=creditCopyText

³⁹https://unsplash.com/s/photos/open?utm_source=unsplash&utm_medium=referral&utm_content=creditCopyText

The Community

Graham Lee

June 1st, 2020



It would of course be easy to single out authors who have made important contributions to the world of Free, Libre and Open Source Software for this month's Library article. I'm sure we'll address their work in later issues. One of the most important reasons for the success of Free Software is its collaborative nature so this month we'll acknowledge the community effort to document open source software.

We start by enumerating the successful commercial efforts that were kickstarted by the work of the community. When Tim O'Reilly diversified his publishing house, O'Reilly and Associates, from technical writing services to publishing, a significant part of their catalogue came from printed copies of manuals for UNIX and GNU tools. Even their bespoke titles, like *Programming with GNU Software*¹, advise people to inspect the `man` and `info` documentation for the tools.

GNU Texinfo², by the way, is a much-overlooked source of fantastic information. As users of Unix-like systems of any level of experience will realise, the manual pages in `man` tend to be brief synopses of the behaviour of each option to the tools they cover, but aren't presented in a way that helps unfamiliar users adopt the tools. Texinfo documents can be turned into websites, books, or online manuals, are hyperlinked, and usually contain much richer information than the `man` page for a command. The `man` page for a C compiler might be eight

¹<http://shop.oreilly.com/product/9781565921122.do>

²<https://www.gnu.org/software/texinfo/>

(printed) pages of command-line options, while the `info` manual for `gcc` runs to over a thousand printed pages.

Anyway, as O'Reilly filled a gap for high-quality printed documentation for Unix systems (printing those 1,000 GNU pages on an Epson FX-80 takes time, drives you mad with the noise, and you end up with low-quality output), others were providing the community-written documentation as *the* documentation for their systems. Flip to the back of your NeXTSTEP developer manuals—a cubic foot of paper almost exactly the dimensions of the workstation—and you'll see documentation for `gcc`, `gdb`, and `make`. This documentation wasn't written by the team at NeXT: it's the GNU project's work. Similarly SUN's manuals for Solaris—a significantly larger and heavier set of binders that took up an entire shelf in my first office—were hardcopies of the manuals from AT&T, Berkeley, and other contributors.

When Linux became popular among hobbyists in the 1990s, the need to go beyond reference documentation to how-to guides and goal-oriented tutorials became obvious. Once again, the community stepped up. The Linux Documentation Project³ offered guidance on everything from switching from Windows⁴ to brewing coffee with Linux⁵. With TLDP, the documentation went a long way toward meeting the user: no longer a reference of what the functions in the software did, these HOWTO guides told the community how to achieve *their goals* using free software.

There has always been a need for quality documentation alongside free software projects, even when the most vehement of zealots don't recognise it. Whenever an ignorant programmer derides someone's question by demanding that they "RTFM", they are of course relying on the existence of The Fucking Manual and the (often voluntary, community) labour that has gone into producing, translating, and sharing it. Whenever someone gives up on the official guides and asks StackOverflow how to solve their problem, they're contributing to a community-supported project to make some startup dudes rich placing advertising in a collaborative documentation project. Indeed the current leader in the Stack Overflow popularity contest⁶ got his internet points by documenting the open source .NET⁷ and Java⁸ developer tools.

In our way, we're contributing to the free information available on software, too. Please chip in⁹ to support our work!

Photo by Museums Victoria¹⁰ on Unsplash¹¹.

³<http://tldp.org/>

⁴<http://tldp.org/HOWTO/DOS-Win-to-Linux-HOWTO.html>

⁵<http://tldp.org/HOWTO/Coffee.html>

⁶<https://stackoverflow.com/users/22656/jon-skeet>

⁷<https://github.com/dotnet/>

⁸<https://openjdk.java.net/>

⁹<https://deprogrammaticaipsum.com/contribute/>

¹⁰https://unsplash.com/@museums victoria?utm_source=unsplash&utm_medium=referral&utm_content=creditCopyText

¹¹https://unsplash.com/s/photos/library-amsterdam?utm_source=unsplash&utm_medium=referral&utm_content=creditCopyText