HISTORY OF FREE AND OPEN-SOURCE SOFTWARE

Booklets Series

Collective Stewardship

RTC

CATALOG SHEET

Published by the Revolutionary Technical Committee
Author: Collective Stewardship of the Revolutionary Technical Committee
Work: History of Free and Open-Source Software
Pages: 17 whole, 15 work

First Secretary: Leah Pasiune
Booklets Series

CSWW
A-01-001
revteccom.tk/csww.html

LICENSING

ADDITIONAL INFORMATION

If you wish to join the Revolutionary Technical Committee, contact us.
E-Mail: rtc.contact@proton.me

If you desire to financially support our work, anonymous payment options are available at https://revteccom.tk/support.html

Introduction

We find it of extreme importance to catalog the overall history of the Free and Open-Source movement in one document, while, at the same time, pointing out the fundamental mistakes that led to its failure. The fact that our organization is a marxist one certainly does not prevent it from engaging in the same mistakes of previous organizations.

To see how self-denominated "communist" movements fell over to the flaws of those before them, someone just has to look at how the "Telekommunisten" collective engaged in the delusion that peer-to-peer models would save the proletariat in the internet age, and that they were diametrically opposed to the "centralized" interests of capital - only for history to later show that italian writer Amadeo Bordiga was right on one count:

"*The hell of capitalism is the firm, not the fact that the firm has a boss*"

We, the communists, have always fought a battle in the realm of information: to divulge the truth and spread class consciousness. Computer programs are also information, though materialized in a medium that allows for explicit rather than implicit control. The way we will fight the next battles in this realm is not by handing over the task of creating class consciousness to disorganized masses of small firms, that are embodied by the technicians' co-operatives. We must follow the example of the bolsheviks and speak with one voice, while utilizing the advantages that new technologies have brought us to prepare for every possibility. We must unite all those who create in this new era of information and have them create for the interests of the oppressed - for the interests of the class and those inside it who are marginalized.

We must also not fall for the fallacy that FOSS activists fell for: the fallacy of legalism, which has manifested itself as a liberal cancer in our space. The activists thought they would be able to secure a software utopia by making the capitalists "play fair" to the terms of extremely long-winded license agreements, and eventually, they were destroyed, because they did not know to separate their legal methods inside bourgeois law from their methods of action outside of it.

FOSS never even came close to "winning", in their own terms. It stopped posing a real threat when companies quickly realized that they could just have the "open-source" part, and that making development more accessible would, in fact, make them way more money. Once they could directly translate the accessibility of free compilers and SDKs into contributions and easy deployments of the software they use in their own infrastructure, the small loss they had was quickly dwarfed by the profits they made from shortening development time.

The elitist character of skilled workmanship in programming has blinded those in the field for a very long time. FOSS and the peer-to-peer obsessions were both “solutions” that had to contend with the mystical quality of the “individual programmer”, which via the careful honing of their own craft can produce a work that is functionally indistinguishable from magic to the average observer. Now, programmers are being proletarianized, and the newcomers to the field are, sometimes, placed in conditions analogous to slavery. Those new programmers do not suffer from the delusions of the last generation. They do not view their own work in such high individual regard specifically because everything outside their strict area of competency is now outsourced to other workers. Despite their below-average scientific education, they are no longer as alienated and know the high level of abstraction they are dealing with.

Those programmers don’t need a magical license agreement that will “save their code”, and they don’t need a small firm to exploit them either. They already know what both actually look like! What they need is collective bargaining power, what they need are worker’s organizations, what they need is to understand how much control they actually have, as a collective. Only then can they be an effective tool in building a class union front.

So, workers in computer science, contributors to bigger projects, and students: When you read this small historiography, understand that there is a possibility to bargain, not just by creating a small union that will get easily crushed, but by understanding how building up the power of your field strengthens the class in a revolutionary struggle.

Early Years

From the creation of the first true programs written by Alan Turing and Neumann János in the late 1940s, bourgeois intellectual property laws had not yet been enforced on software outside of rare cases, mostly because utilizing state and academic resources to quicken research was seen as advantageous by electronics corporations. As a result of this relationship, a great number of software projects were developed by academics in direct contact with the private sector, following a model where companies would commission new features to be added to existing programs.

Such a state of affairs lasted until the late 1980s. In developer Keith Packard's talk, "A Political History of X", he describes how the development of the popular and still-utilized X Window System was handled by a corporate-academic consortium, as the internet simply wasn't developed enough to handle development between different corporate offices:

> "*They said: what we're going to do instead is concentrate the development with hired and paid developers at MIT, we're gonna build a consortium, hire a bunch of developers,* [...] *they gave up on the notion of doing development over the internet, because the Internet of this era was seen as fundamentally broken* [...]
>
> *It* [X] *was funded by consortium members,* [...] *the members who were paying, by the way, also voted at all the standards.*"

The high level of academic involvement in software development meant that, at the time, the principles of openness and cooperation prominent in theoretical academia were still the most widely-accepted doctrine in the software space. Programs were still widely distributed in source-code form due to incompatibilities between machines making usage impossible without slight modification. Early FORTRAN compilers are most likely an example of this, as the original FORTRAN compiler, written in the 1950s, still has archived copies of its source-code in the form of direct scans of paper cards.

The Bell Corporation (AT&T) was the first to close the door on early software sharing. Some companies had already started charging for the documentation and technical specifications that made software usable, while only providing programs in their binary form, but AT&T was the first to take advantage of a 1974 ruling in the United States that granted companies intellectual property rights on the programs themselves, and in 1979 began to commercialize their UNIX operating system with a restrictive licensing scheme.

Initially, AT&T granted free copies to academic institutions and to the United States government. In the beginning of the 1980s, it began to cut down on the benefits of their academic licensing program, settling on charging universities full price for copies of the software.

The situation of software as private property would become clearer in 1983, when *Apple v. Franklin*, a case over whether computer firmware was subject to copyright, would be ruled in favor of Apple, which claimed that the Franklin Computer Corporation had illegally copied part of the Apple II's BIOS. This resulted in both the source-code and compiled versions of programs being treated similarly to literary works, where source-code was legally similar to the version of a book etched in ink presses, while the compiled version was similar to a finished paper copy of the book. The case was initially ruled in favor of Franklin, however, the ruling was reverted only 3 days later, after an appeal by Apple made the courts rush to protect private property.

After the rulings, academics and hobbyists continued to share software via public listings stapled to information boards in companies and universities, magazine listings, and later via digital bulletin board systems. Some still continued to distribute proprietary software and reverse-engineer it, resulting in the formation of modern hacking culture. Born from this rebellious culture was the wish for an organized framework when it came to software sharing.

That framework would come soon, as in 1983, Richard Matthew Stallman created the GNU Project, which was intended to be a freely-shareable implementation of the UNIX operating system. The project exalted the values of collaboration and solidarity between programmers, and expressed the wish of having software be free for redistribution, creating the modern free software movement. However, the movement as it existed was solely concerned with keeping software free and shareable, and not with contributing to the external struggle for freedom of information and for breaking out knowledge about software from academic spheres. Thus, it was a petty-bourgeois movement by its very nature.

A couple years later, in 1985, the Free Software Foundation (FSF) was founded as a non-profit corporation, and received control of all copyrights over the GNU system. Work began to create a legal framework for the enforcement of free software principles, which would later dominate the extent of the free software movement's social action.

There was a radical component to free software at the time, as some small circles involved with the FSF went far enough as to support the full abolishment of copyright and intellectual property laws. However, those radical elements still agreed to the creation of a license, as it was seen as a provisional measure until copyright over software works could be abolished. In 1989, the first version of the GNU General Public License (GPL) was published, with the text most likely being written by Richard Stallman, who now held the position of president in the FSF. In 1991, the license was revised, updating the legal wording and adding an additional clause to prevent issues over patent litigation.

Linux and Open-Source

After the GPL was published, the GNU C Compiler (later renamed to the GNU Compiler Collection) was released under the license. Previous versions had already been released under their own licenses for the past 2 years, but the introduction of the GPL, and the fact that the software was shared for free in bulletin boards and at a low cost by mail, along with being copied around in the hobbyist sphere, made it one of the most popular compilers of all time and an ideal alternative to proprietary C compilers, which were often sold by operating system developers at absurd prices. The release of GCC turned the FSF from a small collective of developers into a large organization, introducing many developers to free software and bolstering memberships and donations.

This quick growth attracted many qualified engineers, and after almost a decade, the GNU Operating System was nearly complete, with the only piece left being the kernel. This piece would be filled by Linus Torvalds, who released version 0.12 of the Linux kernel under the GPL in February 1992. A few days later, the release of MCC Interim Linux by Owen Le Blanc marked the first time a fully-functional free software operating system was ever distributed.

The quick rise of GNU/Linux from small projects into a fully functional and free operating system meant that, in the 1990s, free software began to be seen as commercially viable, and shrink-wrapped copies of FOSS projects started being sold in stores, with companies providing paid on-the-phone technical support to corporate users of free software products. The first FOSS products to be sold were, of course, Linux distributions. Most of the early well-known ones were sold in stores, and most of the time, they were sold along with guidebooks. However, one particular shrink-wrap release sticks out, as it had direct involvement from the Free Software Foundation: VALinux's release of Debian 2.1, which, by all accounts, was terrible to install. This release was part of a boom in physical releases of Debian, which made the distribution one of the most popular Linux distributions ever.

The FSF, which sponsored physical releases of Debian, would not take advantage of the popularity of the OS. Despite many Debian contributors also being members of the FSF, the organization didn't certify the distribution as a fully free software one, because its developers hosted an unofficial repository that contained proprietary software to make installing firmware and fonts easier. This showed one of the most glaring flaws of the FSF: caring about the minutiae of consistency where it simply did not matter, instead of taking advantage of the software's popularity to promote other pieces of free software.

FOSS’s brief period of general success attracted the eyes of some petty-bourgeois intellectuals, which began to study ways in which they could utilize the community aspect of the model while not having to comply with the requirement in the GNU GPL to keep all versions of the software FOSS forever. The man who would eventually find success in pitching a profitable FOSS-based model to companies was Eric Steven Raymond, a vocal homophobe, racist, and supporter of racialized police violence. In 1997, Raymond published the essay “The Cathedral and the Bazaar”, in which he pitched FOSS as a public and rapid development model that allowed for fast iteration. After the book was published, circles around Raymond sought to commercially rebrand free software as open-source software, casting away what remained of the FSF's social activism, which by that point was beginning to be almost exclusively focused on the GPL.

After the year 2000, open-source began to be adopted as corporate branding, and permissive FOSS licenses, which removed the requirement to share the source-code to publicly released versions of software, were quickly adopted by corporate developers. Eventually, most developers began to identify with open-source, and free software was overshadowed. The broad adoption of open-source, and the idea of absolute freedom of redistribution appealed to developers, which began to utilize permissive licenses more and more. Eventually, companies began to reap the benefits of the infrastructure that was created by the free labor generated by open-source projects, which they could use to develop their own proprietary projects upon.

An example of this parasitic relationship is the one between most corporate providers and OpenSSL, a secure communications library which powers millions of computers. The library, despite being used on millions of commercial devices around the world, faced critical underfunding, being mostly supported by individual donations before receiving small corporate grants after negligence resulted in Heartbleed, one of the biggest vulnerabilities to face the computing world.

In 2007, the FSF released Version 3 of the GPL, which made it impossible for any GPL software to be legally considered as Digital Restrictions Management software. It didn’t disallow GPLed DRM, but it made it so that if the protection was cracked, the copyright holder had no legal recourse to prosecute the violator. The new version of the license also forbade “tivoization”, which is the practice of creating a machine that runs free software but making it so the actual software running on the machine could not be replaced by other versions.

The changes in the new version of the license were not appreciated by Linus Torvalds and the Linux Foundation, which at that point were both sustained solely by the good graces of gigantic technology companies. Torvalds refused to relicense Linux under the GPLv3, as Linux development relied too much on support from manufacturers of home media sets and IoT devices. This marked a definitive split between the Free Software Foundation and the Linux Foundation. A few years later, in 2012, Debian joined Raymond's Open Source Initiative. But free software's problems wouldn't be limited solely by inter-project politics.

Reactionarism in Open-Source

Being a phenomena born mostly from petty-bourgeois intelligentsia who were placed in a further position of privilege by being white and male, the free software movement's reactionary content did not have a chance to show itself until a contradiction appeared. Unanimous in their various different forms of prejudice, there was just no point where this movement, mostly disconnected from other social issues, got to show just how structurally inept it was to guarantee the basic safety of its minority members.

In 2013, the internet saw the rise of a reactionary movement that would go beyond the mere cycle of outrage, and instead would have a direct influence in which figures would lead reactionary movements in the US, in a conflict that would be invisible to many. While that movement is often given unifying names such as "GamerGate", the sparks that set it off are not of particular importance to this text, nor its "main lore", because it is the kind of historical phenomena that would have been set off by anything else in that specific moment.

In summary, it was a movement that attempted to preserve the traditional environment that made up tech culture in the internet era: an environment dominated by men, mostly white, mostly straight and mostly cisgender - who saw the communities around games as an outlet for their frustrations, where they could pretend that the plea of marginalized communities didn't exist, and optionally make fun of them. That is exactly the same demographic that would dominate the software development space, mostly because it was the same demographic prominent in academia which had guided the discourse around technology, including the discourse around the perceived "social issues" around it.

Programming was beginning to become more accessible to people that didn't have any previous references in the field and that didn't have any contact with lengthy manuals and academic works, and this had a downstream effect. The small contingent of software developers who didn't fit the archetype was becoming noticeable - no longer could the white man code and have his debates on free software in peace without the presence of "aberrations" and the inferior sex, no longer could the white man openly condemn a whole repertoire of "degeneracy" without being questioned.

Groups which had been ignored for so long had appealed to the legalistic instinct of the free software movement, and asked for the establishment of codes of conduct that would guarantee their basic safety and comfort in the absence of critical thinking capabilities from project moderators. Of course, the communities of those projects saw this as a significant intrusion orchestrated by the kinds of people that would have previously been harassed away.

The first battle that would occur in the preface to this new wave of reactionarism was the “civility discussion” in the Linux Kernel Mailing List. The discussion would not be centered on the status of minorities in free software, but would evoke the first strong wave of right-wing harassment against a specific target right before GamerGate happened. It all started after Linus Torvalds encouraged maintainer Greg Kroah-Hartman to yell at contributors, as he was being flooded with patch requests to the kernel:

“*Clearly at least some people say 'I know this patch isn't important enough to send to Linus, but I know Greg will silently accept it after the fact, so I'll just wait and mark it for stable'. You may need to learn to shout at people.* ”

Knowing how Linus treated fellow contributors at that time, by “shout”, he meant “verbally abuse”. For example, this is how he reacted when he discovered the existence of some badly-authored code:

“*Who the f*ck* [sic] *does idiotic things like that? How did they noty* [sic] *die as babies, considering that they were likely too stupid to find a tit to suck on?*"

After Linus’ comment towards Greg, Sage Sharp replied condemning Linus’ behavior, and dared others to have a serious discussion about it in person at that year’s Kernel Summit. One thing led to another, the discussion developed, and it all caught the eye of the press. Eventually, the discussion made very little difference, as Linus would not change his ways until immense pressure forced Linux to adopt a code of conduct in 2018.

What drew attention, however, was how the free software community replied to the incident with intense misogyny. At that time, Sage presented as a woman, and so, they drew the ire of emotionally stunted men around the world. Previously, the backlash wouldn’t have been so pronounced, but now, there were numerous comments about the “female proclivity to stop people from accomplishing things” and about “women having their feelings hurt”. Such things, which were now being shared way more publicly, were previously reserved to sparsely-populated IRCs, with the popular approach being to just attack projects made for and by women instead.

Things would become worse, as in 2014, new waves of harassment would take place. It’s difficult to pinpoint major events, as not only did they become more numerous, but they also spread around different communities. However, it was really the year where the reactionary gamer crowd that happened to have an interest in software went on the offensive. That year, there were multiple incidents at conferences where techbros would attempt to "debunk feminism" to vaguely progressive speakers.

However, the most publicized incident related to gaming/techbro reactionarism that would involve the FOSS community happened over Randi Harper's "ggautoblock" perl script, which was made with the sole purpose of automatically blocking people on Twitter engaging in harassment campaigns against women and minorities online. Obviously, those people didn't like the fact that people now had a choice to not listen to their preaching, and so they began to direct their harassment at Harper. Imagine how they reacted when they discovered Harper was a contributor at the FreeBSD project!

Before the new right-wing wave hit the internet, Harper was very much a collaborator to the misogynistic culture in FOSS, accusing a woman of "wanting special treatment" when she pointed out a prejudiced remark on a FreeBSD manual page. Harper's "awakening" only happened after they were personally affected by the misogyny to the point where they became annoyed. However, the way FreeBSD handled the harassment against Harper after the incident was nothing short of negligent, with nothing being done about another contributor incessantly asking that she be removed from the project, leaking conversations in private chats, and encouraging others to impersonate Harper to get access to other private IRCs.

This particular incident would lead many smaller projects to adopt codes of conduct, including FreeBSD (albeit a few years later). However, in an incredible show of regression, a later vote depoliticized the CoC, removing anything referring to prejudice, instead settling on the broader concept of "respect".

The years of 2012 up until 2015 elicited major change within the FOSS community - however, despite all the effort for "inclusion" mostly done within a liberal framework, those notable events elicited very little demographic change. Comparing the 2017 Open Source Survey with the 2002 FLOSS Project Survey shows a nearly 200% increase in women developers! Which is much less impressive when you consider the fact that the total amount rose from 1.1% to 3%.

A recent private survey with a sample size of around 100, carried out by the Revolutionary Technical Committee, focusing specifically on the Minecraft modding community, showed a better, but still dire situation - only 16% of developers were women, with 90% of developers being white!

The success of the struggle of minorities in the FOSS space has been greatly exaggerated - especially in already-established projects. Projects can even put up a trans flag banner once in a while, but that doesn't mean they accept trans people, just like they can put up a women's liberation banner while project management doesn't actually care about what women have to say.

Acceptance is better than it was in the early 2000s and mid-2010s, but that isn't saying much, especially considering that minorities have been mostly assimilated into the culture of FOSS in general.

And the keyword here is "assimilation", and not "liberation". The very nature of FOSS as being a model where labor is made available at the lowest cost possible, as we described in the previous chapter, means that losing out on that 3% edge of women means losing on 3% of all contributions to open-source projects. Going back to the FreeBSD example, they specifically chose to dial back the tone on their code of conduct when the community already had a lot of marginalized people contributing inside it.

Either way, the social struggles within FOSS wouldn't end, they would merely evolve, quickened by an explosion set off by a familiar figure.

The Hellhole of Free Software

Richard Stallman would not maintain his position as an unsung hero of technology for too long. In 2019, he published a defense of right-wing politician and pedophile Jeffrey Epstein, and was met with significant backlash. This led to his resignation from MIT and from the Free Software Foundation. Later, it was brought to light that he had published a statement where he likened having a child with down syndrome with having a pet.

"*If you'd like to love and care for a pet that doesn't have normal human mental capacity, don't create a handicapped human being to be your pet. Get a dog or a parrot. It will appreciate your love, and it will never feel bad for being less capable than normal humans.*"

Truly a compelling argument for mothers around the world! It didn't end there, however, as his other statements absolving multiple forms of pedophilia were also revealed. In his infinite wisdom, he had thought about, written, AND published the following phrase:

"*There is little evidence to justify the widespread assumption that willing participation in pedophilia hurts children.*"

He has also defended the possession of child pornography, arguing:

"*Making such* [pornographic] *photos* [of children] *should be a crime, and is a crime, but that is no reason to prohibit possessing copies of the photos.*"

The insensitive ranting of the world's most disliked toenail-eater, however, did not end on what would, proportionately, become a high note for him, after it became known that he published the following comment on the rape of a 14-year-old by an adult woman:

"*I wish an attractive woman had 'abused' me that way when I was 14.*"

Stallman, in his constant showcase of neuron activations that can certainly be called "thoughts", exemplifies the stereotype of men in STEM: Scientific calculators, which can measure almost anything, but at the end of the day, are just total tools.

It should come as no surprise that the Free Software Foundation itself is known, even by those who have dabbled with it in passing, as a place that does not enforce its codes of conduct until people speak out and break their secrecy. Let us take, for example, the case of the transgender woman who was rejected by the FSF in the early 2000s because whoever handled their application said that she would be too ugly to look at.

Leah Rowe, who leaked the information about the case, had made a strong suggestion, via their actions, that the man who did the handling in question is John Sullivan. If you've ever taken a look at his ears, you'll see that he's not only a transphobe, but also the classic pot who calls the kettle black. On a better note, even when approached with suspicion, Sullivan seems to have changed into a better person - should probably pay restitution for time wasted, though.

After Stallman made his "statements", resignations started to happen, as most left-wing cadres of the Free Software Foundation stopped supporting the organization, and many vaguely progressive members had left. As a form of damage control, Stallman was made to resign from the position of President of the Free Software Foundation. Stallman's position remained vacant for 11 months, until an ex-treasurer of the FSF, Geoffrey Knauth, was elected.

This interim moment, however, saw a lot of change - not inside the FSF, but outside of it. Particularly, it resulted in belief in FOSS crashing down even harder than it already had, with a new legalistic text showing up every other day to try and remedy the previous legalistic failings of the organization. Out of those, the most notable was the "Anti-Capitalist Software License" (don't laugh!), which allowed users to, in the writers' own words, state the "moral center" of their code! The small manifesto that was published along with the license also seems to go through a brief moment of sobriety, by claiming that "the availability of source code is less important than the organization of software labor", of course, it talks about how source-availability opens code to "theft", and is generally a regressive document.

While the FSF was coughing up blood, another organization was formed: the Organization for Ethical Source. Its founder is Coraline Ada Ehmke, who was prominent in previous efforts to have projects adopt codes of conduct. The OES had no goals of liberating the working class, and did not attempt to provide means of true organization and confrontation for software developers - instead, it merely sought to create "morally correct" code which used definitions set by the United Nations as a precedent. Another incredibly funny component of the OES was that its main promoted license, the Hippocratic License (we're not joking), wanted committees to be held outside of court to vote on if the terms of the license were violated in case of legal action being taken.

The OES and the Hippocratic License faced a lot of ridicule, some of it deserved, most of it coming from regressive proponents of FOSS. The organization and the license, however, must be criticized from the angle that they are liberal and moralistic. The OES' only recourse is a legal one, however, their license is unenforceable. They can say whatever they want about "making a statement", but that doesn't change the fact that their "statement" isn't actually about the rights of all, is limited to a license agreement, and that it amounts to no tangible action.

Back in the FSF, Knauth did not take his chance to be seen in history as the silent but sensible successor to the insufferably annoying Stallman. In 2021, he oversaw Stallman's reintroduction as a board member in the FSF. Stallman's rejoining was met with significant backlash, and was announced a few days into the organization's Libreplanet conference, making it impossible for speakers to re-plan.

Currently, there is a real possibility that Stallman will once again become president of the FSF once more people forget about the scandal, as Knauth has reinforced that his stay as president is an interim one until the situation is stabilized. Either way, the future of the organization is unknown, however, it is expected to decay further. Free software in general will also decay as well, and nowadays has become even more infested by right-wing elements, as the most fervent FOSS proponents also tend to be fervent fascists.

Now, one task stands in front of proletarianized programmers: to organize, and to gather the power to dictate their own terms. To act in unison with workers from all other fields, to refuse to create weapons of death and surveillance for the bourgeois state. That goal will certainly be reached.