



Contents

- page 1: New energies for a remote FSF
- page 3: Tracing the FSF's footsteps
- page 5: What abandonware teaches us about the importance of software freedom
- page 7: FSF SysOps cleaning up the internet
- page 10: Society will never be free under nonfree software

New energies for a remote FSF

By *Zoë Kooyman*
Executive Director

The Free Software Foundation (FSF) has pushed for the creation and adoption of free software since 1985; the solution to the rise of proprietary software, and its protecting strategy, copyleft, were the main focus of the early FSF. As the FSF grew in size and scope, it added campaigning, education, protesting, advocacy, compliance, supporting other community projects with tech support or fiscal sponsorship, and more. On page three in this *Bulletin*, Jason gives us a detailed overview of all the

locations the FSF occupied over the years, because for most of the FSF's existence, the FSF had a physical home. On August 31, 2024, the FSF moved out of our long-serving office.

The move entailed days of sorting through and moving materials, including: a range of items with more or less historical value, like awards, original artworks, and different versions of the baby GNU; Respects Your Freedom (RYF) certified devices; organizational documents; endless amounts of tech materials; copyright assignments; GNU Press shop merchandise; trash; recycling; and one inconveniently large conference table.



The former FSF tech room at 51 Franklin St., all cleared out.

A year ago, I couldn't imagine that the fifth floor of 51 Franklin St. would no longer sport the red Free Software Foundation logo on its green wall behind the entrance desk, or that the preamble of the GPL would be scratched off the glass windows of the tech offices. The Boston office saw many adventures. It hosted people of many different talents and backgrounds, and facilitated the birth of many of our regular events that made the office full of life. It was where we protested with the community on International Day Against DRM (IDAD), teamed up with volunteers for biannual fundraiser mailing events, and socialized around the annual LibrePlanet conferences. It was also the place where we, the current staff members of the FSF, came into work every single day to unite against the proprietary software corporations of this world.



The FSF front desk after everything has been removed, except the logo.

Now, a little more than a year before our fortieth birthday, the FSF became a remote organization, with

many of our staff not even based in Boston anymore. This didn't happen overnight, of course. Before the COVID-19 pandemic, aside from a few remote staff, most of the FSF staff were following Franklin St. in-person office hours with only two days of remote working allowed per two weeks! Since then, like many other organizations, our staff and policies changed and moved, and we became accustomed to working remotely. This shift moved our center of gravity, leaving Franklin St. quiet.

As a 501(c)(3) focused on putting software users front and center, we always keep in mind that sticking to our mission of protecting computer user rights is more important than any financial consideration. With the majority of our income coming from individuals, if the economy suffers, our supporters suffer, and so do we. Pushing for change is not a profitable business — there are too many people and corporations that have something to lose. But we handle the money entrusted to us by the community with the utmost care. And so, on the eve of a twenty-year anniversary at Boston's Downtown Crossing, we decided to let go of an unreasonable expense in an already difficult economy and put the funds we have to work for more activism instead of holding on to a physical location.

We will be highlighting connections across the world and

focusing on new community-building initiatives, both virtual and in-person, and we can't wait to see you there. And, we will be using our fortieth anniversary year and our newfound decentralized presence to our advantage to continue advocating for free software far and wide. Leaving an era and a home behind can be difficult, but we have also found it to be a fresh and freeing start. Without the costs of the office we will be able to recover from our financial challenges faster, put a larger proportion of donations towards executing the work, and find motivation and energy in new places.



Tracing the FSF's Footsteps

By Jason Self

Free software advocate

The FSF's seemingly mundane physical location history offers a glimpse into the organization's growth and tells the story of an important part of the free software movement. Let's follow the FSF's footsteps through the years.

The FSF emerged on October 4, 1985 during the burgeoning tech revolution. The FSF's mission was clear: grant users the freedom to run, study, share, and modify software. In its early days, the FSF primarily focused on hiring talented programmers to contribute to the GNU Project, an ambitious endeavor to create a completely free operating

system. Its mission of free software for all was welcome in the thriving hacker culture at MIT. The FSF's initial headquarters were nestled within the walls of Lisp Machines, Inc. (LMI), a company with deep ties to the local hacker community. LMI generously provided the fledgling FSF with office space, computing resources, and a mailing address, and MIT even supplied desks. While LMI was the first home of the FSF, it would not be the forever home. Once a promising innovation, Lisp Machines faced harsh market realities and went out of business in 1987, leading the FSF to begin the search for a new base.

During and after the relocation to 675 Massachusetts Ave, the FSF remained steadfast in its mission, mailing out tapes of free software and raising funds to support the development of GNU. This period was marked by significant achievements, including the release of GNU General Public License (GPL) versions 1 and 2 in 1989 and 1991, respectively. These licenses became the bedrock of the free software movement, ensuring that software released under their terms would remain free for users to use, study, modify, and share. A year later, in 1992, Linus Torvalds, the creator of the Linux kernel, decided to relicense his program under the GPL instead of the original nonfree license. This seemingly small act had a profound impact, effectively

completing the GNU operating system. The GNU Project, which had been diligently developing a free operating system since 1983, finally had all the necessary components to realize its vision. The dream of using a computer with complete freedom was now a reality with GNU/Linux.

By 1995, the foundation's activities had expanded beyond just coding. They were now actively promoting and advocating for free software, educating the public, and fighting legal battles. The small office at 675 Massachusetts Avenue was bursting at the seams, unable to accommodate the growing team and the increasing demands of their mission. In response, the FSF moved to 59 Temple Place, Suite 330, Boston, MA, in 1995. With more space, the foundation could dedicate more resources to non-programming activities, such as public outreach and legal advocacy. Key developments during the FSF's time at 59 Temple Place include:

- Continued rapid progress on the GNU Project, including the development of GNOME (then called the GNU Network Object Model Environment);
- Expansion of the GNU Press in publishing more books and manuals explaining free software's philosophy and practicalities;
- The FSF began campaigning against software patents,

recognizing them as a significant threat to free software and advocated for legal reforms to protect software developers' freedom to create and share code;

- The FSF also became a vocal critic of Digital Rights Management (DRM), which they termed Digital Restrictions Management to reframe the issue as restricting user freedoms.



Almost twenty years of achievement at 51 Franklin St.

Driven by the need for a more modern and spacious office, the FSF relocated to 51 Franklin Street in 2005. This location provided the FSF with ample room for an expanding team. The foundation thrived at 51 Franklin Street for nearly two decades, achieving several milestones and further solidifying its role as a champion of software freedom. Some notable events and accomplishments during this period include:

- Publishing the third version of the GNU General Public License (GPL), a significant update to the cornerstone license of the free software movement. GPLv3 addressed contemporary legal challenges and further strengthened the protections for user freedoms;
- Cisco Case: The FSF took a stand against GPL violations by filing a lawsuit concerning Cisco's Linksys routers. This legal action underscored the FSF's commitment to enforcing the GPL and ensuring that companies comply with its terms, safeguarding user freedoms;
- OpenWRT Project: The release of the source code from the Linksys lawsuit resulted in the formation of the OpenWRT project; the first commit into the version control system is that source code;
- Respects Your Freedom (RYF) Certification: The FSF launched the Respects Your Freedom hardware endorsement program, providing certification for hardware products that meet their rigorous standards for user freedom, privacy, and control. The RYF certification empowers individuals to make informed choices and supports companies aligning with the FSF's values.

The FSF's tenure at 51 Franklin St. is

a testament to their enduring commitment to software freedom and the ability to adapt and evolve in a rapidly changing technological landscape. In 2024, the FSF further adapted to the needs of an international community and transitioned to an entirely virtual organization, allowing it to more effectively connect with a global community of free software advocates and contributors. From its early days at LMI to its current virtual existence, the FSF has consistently championed the cause of software freedom, empowering users and shaping the digital landscape for the better. 🍷

What abandonware teaches us about the importance of software freedom
By Krzysztof Siewicz
Licensing and Compliance Manager

There are quite a few older computer programs that hobbyists, retro game enthusiasts, and everyday users alike still want to use. This is not easy and sometimes even impossible with proprietary programs whose vendors have abandoned them. When a program becomes abandonware, users who had not previously secured a copy of the program cannot obtain a copy without risking civil or even criminal liability due to copyright restrictions.

To further hinder users, even if they had obtained legal copies of the abandonware, they cannot freely adapt it to their needs and redistribute it without restrictions. This is an effective death sentence for a large proportion of abandonware since it cannot be maintained or shared. For some abandonware, it is not the end of the story thanks to numerous dedicated users fighting to resurrect these programs. An interesting example is proprietary games developed for discontinued hardware or operating systems. For a majority of nostalgic games, the only technical problem is the unavailability of the platform, but the games themselves are still perfectly playable. A common method of abandonware resurrection involves developing emulators, which are programs that simulate the platform capable of reading the old game's binary.



No matter if a nonfree game is played on its original hardware or an emulator, it is still nonfree.

While many of the emulators developed and used to play these games are free software, they don't remove the copyright restrictions

from abandoned proprietary programs, and the program's source code remains unavailable. The abandonware is still entrapped by the same restrictions it was when released by its proprietary vendor, and no effort invested by users will increase the freedom of the program. While it's understandable why necromancers of abandonware seek to use these programs again, their efforts could be put towards building a fully free version of their once beloved program, which would not perpetuate the abandonware trap.

At the FSF, we've been fighting for a world where you don't have to depend on proprietary vendors to release and maintain beloved games, or any other software for that matter, since 1985. It's one of the reasons why we support the GNU System, an initiative to provide a completely free operating system and applications.

Free software comes with the freedom to run, copy, distribute, study, change, and improve the program. With a free software game, there isn't a looming threat of copyright restrictions making a game legally unplayable and erasing the creative work of the many people who contributed to the game the moment a vendor goes out of business or drops the program. It can continue to be used and reused in freedom indefinitely to your heart's content. You and the community of people who

enjoy the program are free to do whatever you want with it and are not left at the whims of a proprietary vendor.

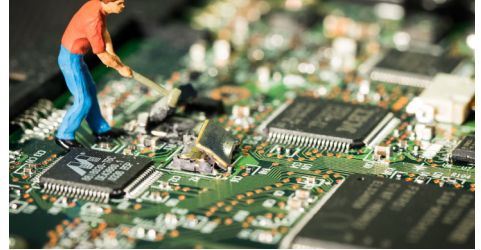
Programs such as free software games are community-developed, and you don't even need to be a programming expert to contribute. Developers of free software games actually are in need of a wide variety of skills including drawing, music composition, writing, singing, acting, and pretty much any other artistic skill you can think of. If you want to get involved in working on free software games, take a look around the Free Software Directory's collection of games, join a project or start your own, and save yourself from the abandonware trap! 🙄

FSF SysOps cleaning up the Internet

*By Michael McMahon
GNU/Linux Systems
Administrator*

The SysOps team, consisting of only two full-time FSF staff members and a handful of volunteers, spends a considerable amount of time preventing attacks from taking down our services in addition to our usual operations. We self-host over sixty-three different services, platforms, and websites for the FSF staff, the GNU Project, community projects, and the wider free software community on our

own twelve-year-old hardware which uses free BIOSes. Like for many other entities on the Internet, attacks often come in the form of a Distributed Denial of Service attack (DDoS), which can use thousands of devices to fulfill its goal of flooding its target's servers, and ultimately bring it down.



The SysOps team uses many tools to prevent and fight server attacks.

Some of our most serious incidents happened in 2016 with a third-party DDoS mitigation service, and even involved the FBI. I am not going to write about the specifics of our recent or ongoing incidents because I do not want to help anyone that is actively attacking us at this moment. What I can mention however, is that one of the recent attacks from the last few months, required blocking more than 40,000 IP addresses from a DDoS attack. And I can also share some of the tools that we use weekly to investigate attacks, which are somewhat simplified for this article but hopefully helpful for other sysadmins:

Monitoring

With tools such as Prometheus and

Uptime Kuma, we can be alerted when a service goes down or when the response time increases. A pattern of alerts may indicate that an investigation needs to take place.

Analyzing logs

The logs of the service affected usually tell a story, but it is hard to read the logs by going line by line. Logs can be analyzed using variations of classic utilities such as Bash, cat, zcat, grep, egrep, awk, sed, sort, uniq, and tail. I typically find the top ten IP addresses communicating with a service and then look for the top three user agents used by that address and a sampling of what they are accessing.

Finding patterns

You will probably notice odd traffic looking at log files from any publicly available website. If a normal crawler is accessing 1,000 pages, something is likely off when another address has 30,000+ lines in the log. Real examples of when something is off include: continually looking for WordPress specific pages and the site is not using WordPress; claiming to be Googlebot and their host does not resolve as Google; not following the robots.txt file; if their user agent is a standard desktop browser and they are generating thousands of lines of log;

if a user agent string is empty; or if they request pages several times a second.

Looking up ASNs

We use IPtoASN to provide updated autonomous system number (ASN) tables at no cost. It is the only one we have found that involves downloading a database locally which can be queried. ASN is an identification number representing a collection of addresses that belong to a network operator. Once we have a local ASN table, we query the table with the addresses found to be questionable to gain a better understanding on a macro scale. Adding this information, as well as the behavior to firewall rules, helps us look for patterns over time.

Blocking

We use a variety of firewalls to prevent addresses from reaching our servers before restarting the affected service. We may need to block individual addresses, CIDR addresses, VPS providers, or even entire ASNs. When blocking an address or a block of addresses is not enough, rules can be written to look for patterns of behavior with tools such as fail2ban. The default example rule when fail2ban is installed is to temporarily block an IP address after three failed SSH attempts. Fail2ban rules can be

written to match patterns in any log file. When writing fail2ban rules, be aware that it can be easy to write rules that are too aggressive, which can cause a denial of service. While the fail2ban defaults are sensible, you could watch the fail2ban logs carefully and adjust the settings to avoid blocking valid traffic. It is important to find a good balance with fail2ban rules.

Abuse reports

Sometimes blocking isn't enough to prevent or stop persistent abuse. This abuse can be reported to hosting companies and Internet Service Providers (ISPs). Sometimes ISPs provide a page to report abuse, but that page may use nonfree JavaScript. We can usually get around using nonfree JavaScript in cases like these by sending an email with a description of the abuse, a snippet of the log, and expected behavior to `abuse@theirdomain.com`. Many abuse reports go unanswered, but when they do answer, it can be worthwhile.

We see three kinds of abuse activity most often on our web servers. Several organizations dedicate themselves to vulnerability scanning, an abuse that actively scans the Internet all of the time, most of which can be blocked with firewalls at the router level. Another kind of

scanner that can be persistent are programs written to search through entire websites, or website crawlers. Most crawler programs are gentle on the servers that they scan, but the same cannot be said for ones that ignore the `robots.txt` files, scan too fast, and take down sites, especially those written by large language model companies. We also see some continuous integration automation, which constantly scans websites for changes, such as those to our licensing pages.

Different technology

For various reasons, Internet traffic coming in and out of some locations, such as the People's Republic of China, behaves different than anywhere else in the world. This could have to do with the Great Firewall of China, carrier-grade NAT, and routing. Many different behaviors can be witnessed simultaneously coming from the same IP addresses that link to Chinese ASNs. Because there is good traffic mixed with bad traffic coming from the same addresses, traditional tools such as fail2ban and firewalls break functionality. If I create a fail2ban rule to block nefarious behavior seen from a Chinese address, I will also be blocking the good traffic that is trying to use the same IP address. I have recently started using a web application

firewall called Modsecurity, which allows traffic sculpting by user agent instead of just by IP address. Currently, we are just rate-limiting some user agents, but we plan to add more configurations in the future.

What can you do to help?

Shepherding our minimalist resources, conserving our computing, and stretching our financial resources take a lot of time and effort. Even with all of the tools we use to investigate and block attacks, we are always looking for more help, and we list many ways to volunteer at [fsf.org/volunteer](https://www.fsf.org/volunteer). The best way to signal long-term support with the FSF SysOps team, and the FSF as a whole, is to become an FSF associate member. 🙌

Society will never be free under nonfree software

By Eko K. A. Owen

Outreach & Communications Coordinator

Advocating for free software isn't just a matter of personal values (although it would be reason enough alone): it is an essential building block of any free society. In 2024, most people interact more or less with software that is controlled by a tech industry that jealously guards its secrets. When the choice to

understand and alter the technology that runs our world is locked away, we the people are forced to live in a nonfree society. We don't have to accept an existence reminiscent of dystopian novels. While free software doesn't guarantee a free society, it does provide many kinds of frameworks and tools needed for building and sustaining free societies.

At the individual user level, free software is an important component of a free society because it allows users to study source code and alter it to suit their needs instead of the desires of proprietary vendors. With freedom in mind, there is no good reason why a tech giant should be able to tell you how, on which device, and for what purpose you can run a program. There are only bad reasons to deprive you of the freedom to study, modify, and share the program. Everyone should have the choice to control their own computing.

At a governmental level, a free society is dependent on accountable governance that is influenced only by its people, not by outside entities. When a government uses proprietary software, power is ripped away from the people and sold to the highest-paying corporation. Residents of a country have significantly less influence over their government when it's controlled by private money, and as a consequence, much less control

over the laws in place, the choices they have, and who gets to watch the lives they lead. There may even be a strong suggestion or requirement to use proprietary software for government business. In addition to all of these freedom violations resulting from a government using nonfree software, the government (and basically anyone), is unable to shield itself from foreign influences accessing a possibly unlimited amount of information. A government run on proprietary software must be held accountable by its people for not using free software.

For a society to be free, everyone must possess freedom of speech and the right to express dissent through protest. Free software allows activists to check and confirm privacy on their devices and channels before organizing and discussing issues. It allows movements and journalists, no matter how autocratic the society, to coordinate protests, express opinions, and publish facts without fear of prosecution or even violence. Free software makes it possible for people at every level to work for a free society.

A society is not free when it is rife with injustice, and free software is a vital weapon against injustice. Free software helps remove barriers to seeking any kind of information that a person may need or want, from life-saving reproductive and gender-

affirming healthcare to how to build a computer from scratch and program troubleshooting. It allows scientists of all levels and backgrounds to contribute to disaster response programs and equity-informed technology solutions that make human existence more sustainable. Free software also makes it possible for us to have control over the medical technology that helps keep us healthy. If you look at an aspect of a society that is currently not free and just, there is probably a way that free software can contribute to it being more free.



Free software is an essential tool for building a free society.

We don't have to accept the "predetermined" path of non-freedom. You and I, and everyone else who lives on this planet, deserve to make our own choices when it comes to software. We shouldn't be forced to allow tech giants to control our destinies. We can decide who gets to have a say over our technology and our societies. 🙌

Gift an FSF membership!
Visit: my.fsf.org/gift-a-membership



Donate to the FSF with Bitcoin:
1MiL7aKG3YAy8rKqW
HJaoE8w7ZWfFSLmjU

Copyright ©2024
Free Software Foundation, Inc.
The articles in this *Bulletin* are
individually licensed under the
Creative Commons Attribution-
ShareAlike 4.0 International license.
[https://
creativecommons.org/
licenses/by-sa/4.0/](https://creativecommons.org/licenses/by-sa/4.0/)

Published biannually by the Free
Software Foundation, 31 Milk Street,
960789, Boston, MA 02196,
(617) 542-5942 info@fsf.org

To opt out of physical mailings of
the FSF *Bulletin*: [https://
u.fsf.org/3zh](https://u.fsf.org/3zh).

This *Bulletin* was produced using all
free software, including GNU Emacs,
Inkscape, Scribus, and GIMP.

IMAGE CREDITS

All photographs © 2024 Free Software
Foundation, Inc. This image is licensed under
a CC BY-SA 4.0 International license.

Page 6: *Turbo Chameleon 64 cartridge in
use on C64 Games System (RetroMadrid
2017)*. © 2017 by AUIC Oficial. This image is
licensed under a CC BY-SA 2.0 Generic
license.

Page 7: *Hacker*. © 2014 by The Preiser
Project. This image is licensed under a CC
BY-2.0 Generic license.

Page 11: *Berlin Friedrichstraße Utopia 2048
small file*. © 2023 by Ærroscape & Lino
Zeddies. This image is licensed under a CC
BY-SA 4.0 International license.

How to contribute

Associate membership: Become an
associate member of the FSF.

Associate members will receive a
bootable 16GB USB card, email
forwarding, and an account on
the FSF's Jabber/XMPP and Jitsi
servers. Plus: access to our
members forum at

[forum.members.fsf.org!](https://forum.members.fsf.org/)

To sign up or get more
information, visit

member.fsf.org or write to
membership@fsf.org

Online: Make a donation at
donate.fsf.org, or contact
donate@fsf.org for more
information on supporting the
FSF.

Jobs: List your job offers on our
jobs page: fsf.org/jobs

Free Software Directory: Browse
and download from thousands of
different free software projects:
directory.fsf.org

Volunteer: To learn more, visit
fsf.org/volunteer

LibrePlanet: Find local groups in
your area or start your own at
[libreplanet.org!](https://libreplanet.org/) You can also use
our materials to teach free
software at a school near you:
u.fsf.org/42i

Free Software Supporter: Receive
our monthly email newsletter:
fsf.org/fss