



### Contents

- page 1: Free software at the core of the right to repair
- page 3: Making of: How to create a hybrid conference
- page 5: Ground zero: Navigating in freedom
- page 7: Freedom and privacy on the web: Our interview with the JShelter team
- page 9: New DRM shirt!
- page 10: Trademarks, volunteering, and code-generating LLM

### Free software at the core of the right to repair

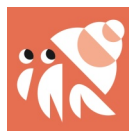
*By Zoë Kooyman*  
*Executive Director*

**I**n the beginning of 2021, we started our Fight to Repair campaign with our animated video of the same name ([u.fsf.org/repair](http://u.fsf.org/repair)). The right to repair movement is one that strongly aligns with our fight for free software. When we launched the campaign, we explained:

*You can't even begin to repair something if you can't open it up and look at it. You can't do the repair if you aren't allowed to move the parts around or add your own new parts.*

*When the “something” is software, this means you need to be able to look inside that software and at its source code, and you need to be able — and allowed — to change it. Even if you don't like to do repairs yourself, you need to be able to choose any repair person you trust to do them.*

These movements have always been strongly aligned, but two years ago, we felt the need to make the point of stating the fight should continue well beyond obtaining the rights to replace a screen, a battery, or any other mechanical part of our tools. Today, the two movements have become even more intertwined, as many hardware tools include software. For example, home appliances, cars, and tractors are among the growing number of hardware that run software on a so-called “license” model. This raises the question: who owns your machine?



## JShelter

*In this issue, we interview two JShelter developers, Libor Polčák and Giorgio Maone (see page 7).*

And as we have learned recently with the help of some clever hackers and persistent farmers, some of the software running on this hardware is actually free software. At LibrePlanet 2023: Charting the Course, we lined up some outstanding right to repair advocates to speak at the event. We welcomed Elizabeth Chamberlain from iFixit for our closing keynote. We also had Australian hacker Sick.Codes tell us all about how he hacked into John Deere tractors to discover that the machines seem to run mostly free software, but John Deere does not distribute the source code as per the licensing terms of that software. Afterward, they both joined a panel hosted by Free Software Award winner Paul Roberts, which also included engineer and farmer Kevin Kenney.

All three of these talks are well worth watching, especially if you want a better understanding of how the restrictions commonly discussed in the right to repair movement overlap with the fight for free software. The talks discuss Digital Restrictions Management (DRM) for example — but they also illustrate how the freedom to run, copy, distribute, study, change, and improve are purposefully trampled on in order to deny users their right to repair, and how free software licenses are violated by major corporations. As a result, users are subject to the control of these

corporations and lack true ownership.

These right to repair advocates are making sure the question of who is in control of our technology is being heard. All across the US, they are bringing the conversation into state governments; globally, they are forcefully backing manufacturers into corners. You can help in this endeavor. As Chamberlain said during her keynote at LibrePlanet: *“GNU General Public License (GPL) enforcement is one of the best tools we’ve got for getting manufacturers to free their software in a way that’s useful for repair.”*

However, GPL enforcement is only possible because of the copyright assignments that have been collected over the years. The current support we can give to the right to repair movement at this time is a strong argument for the FSF to continue this work. The free software movement will continue to benefit from a strong position in copyright enforcement. As is illustrated so beautifully by the right to repair, other movements are also served by our free software work. This mutually beneficial relationship helps protect our freedoms from being trampled on in more ways than one.

We join the right to repair in their call upon you to look for GPL’d software in your devices and, when the vendor is not distributing the source code as required, appeal to the manufacturers to obtain the source

code. You can do this by sending an email to FSF’s licensing team at [license-violation@gnu.org](mailto:license-violation@gnu.org). Help the free software movement by supporting the right to repair. Help the right to repair movement and, in

the specific case of John Deere, farmers everywhere by supporting free software. These are both movements that are protecting people’s four freedoms, and we stand stronger together. 🤖



*A photo of Rubén Rodríguez Pérez and Ian Kelling, swapping out video cards during LibrePlanet 2023.*

## Making of: How to create a hybrid conference

*By Ian Kelling*

*Senior Systems Administrator*

From 2020 to 2022, the LibrePlanet conference was fully online due to the COVID-19 pandemic. In 2023, we could finally have an in-person conference again, but we still wanted to offer speakers and audience members a way to participate remotely if they couldn’t make it to Boston or didn’t want to take an airplane for environmental reasons. If you’re wondering how we made that happen, here are some of the best

parts of that story.

I’m part of the FSF tech team, and a big focus of our job during the conference is to make the audio and video work — and to make it work with free software. The conference lasted two days, and there were three rooms where talks were happening simultaneously. One of the rooms, for example, was dedicated to remote presenters. How this worked was that the speakers joined a video conference with a computer in the conference room at the venue. We call that computer a “stream station.” The stream station projected the

presenter's webcam video feed and slide deck onto a projector screen in the conference room and played their voice through loudspeakers. It also streamed this all out to a video stream embedded in the conference website, [libreplanet.org](http://libreplanet.org). Attendees could ask questions of the remote presenters by speaking into a microphone, just as they could of in-person presenters. As we've done for many years, remote attendees could ask questions via IRC, which would be read by a volunteer in the conference room. It may sound simple in theory, but, in practice, getting the audio to go to all the right places and sounding good was quite complicated and had a few hiccups.

In LibrePlanet 2022, we had two or three crashes in our stream stations due to instability in Nouveau, a GPU driver in the kernel Linux. This year, we hoped that these issues would be resolved by newer versions of the software. The computers are very stable for our day-to-day work, running the newest version of the Nouveau software, and they also worked perfectly fine during our pre-conference testing of video and streaming. That stability, however, didn't last, and Murphy's law came to mind after the first mid-conference crash. However, we had a backup plan: AMD GPUs. Trisquel added support for most AMD GPUs in the recent version II release. Between day one and two, we did a swap of Nvidia to

AMD for the remote room and were prepared to do it for the other two rooms if they had a crash. As it turned out, we needed to do a swap, and in the photo on page 3 you can see Rubén Rodríguez Pérez and me quickly swapping out a video card mid-conference.

The GPUs hadn't been a problem in past in-person conferences because we used X200 laptops, which have reliable Intel graphics capabilities built into their CPUs. However, those computers are not quite fast enough to do video conferencing and streaming at the same time, so this time we used ASUS KGPE-D16 motherboards with AMD CPUs. As for software, we used the Trisquel distribution of GNU/Linux along with BIOSes that are an entirely free software build of either Coreboot or Libreboot. You can read more about the software we used on our LibrePlanet wiki: [u.fsf.org/3zm](http://u.fsf.org/3zm)

This year's LibrePlanet was also at a new venue, and this presented additional challenges. Unlike in previous years, we had to provide our own loudspeakers, amplifiers, projectors, projector screens, projector stands, and all the cables for that equipment. The venue also required that cables not be routed in the path of any attendees, which required a lot of careful planning on our part.

In addition to all of this, for one of

the talks, we chose to have a panel discussion of both remote and in-person speakers. This was a bit of an experiment for us — but, in the end, it worked! I expect we will be able to accommodate this format again at next year’s conference.

We are proud to show that it’s

possible to mount a hybrid conference using our own computers and servers, using free software from the BIOS on up. We learned a lot from LibrePlanet 2023, and we look forward to using that knowledge to put on an amazing LibrePlanet 2024. 🐧



*Photo of a printed map taken within the command center of AKUT, Turkey’s search and rescue association.*

## Ground zero: Navigating in freedom

*By Alper Atmaca  
Hacker with Free Software  
Association (Özgür Yazılım  
Derneği) in Turkey*

**I**n the early morning of February 6, 2023, the southeastern part of Turkey was shaken with unprecedented tremors. As a personal rescue volunteer, I was ill prepared for my duties in the aftermath of this earthquake. However, my country was even less prepared for this moment.

Major disasters bring a lot of challenges to anyone who tries to operate in the affected zone. Modern infrastructure, while having immense capabilities, can also be very fragile. Buildings collapsed during Turkey’s recent earthquakes. The electric grid also went down, taking with it almost every utility, which we all-too-often take for granted, including the most fragile modern utility, the Internet.

Every first response has this conundrum. Being self-sufficient is of paramount importance, and there are

many utilities and services that are necessary in order to achieve a successful response. Navigation services, for example, are necessary because an emergency team first has to triage the damage, and this requires accurate location and mapping. Only after that is it possible to divert resources and reallocate them to the correct locations. How would you do that if all you have used and know is proprietary mapping systems that have a hardcoded dependency on a continuous, fast broadband connection?

Enter OpenStreetMap (OSM).

At ground zero, OSM provided us with the mapping that we very much needed. When I was finally able to arrive in the area and was first deployed on the field, nobody was yet able to navigate properly. Along with much of our critical equipment, our dedicated GPS devices were sent somewhere else because the location of the earthquake's epicenter was misjudged, and almost all the personnel were trying to use their data connection to download the same maps over and over again in order to respond to every direction given by the command center. It was in this setting that I introduced a powerful free software tool called "OsmAnd~" (OSM for Android), along with pre-downloaded maps that I had on my handheld device, which I was using to flawlessly navigate through

the city.

When others saw this, it was received by the group as "magic." Not only could I navigate offline, but I was able to process meaningful Universal Transverse Mercator (WGS 84 UTM) coordinates, which are critically useful in outdoor maps, where referencing needs to be done by hand. This makes coordinating teams over single-channel, low-bandwidth radio systems extremely easy. These coordinates are hidden from the user in all proprietary mapping systems and with *no way to bring them to the fore — not even in a crisis!*

Another challenge we faced was working with limited electrical resources. Batteries only last for so long, and charging takes so much time and power. Therefore, we needed a way to print accurate maps to paper, because paper maps require no batteries or electricity. OSM's maps are ready to print, without restriction, even in these harshest of conditions.

Internet access and electrical resources were not the only problems solved with OSM. Navigational needs, for example, are not just limited to finding your destination. Functionality to help you find a *way* to get there is also a requirement. And there isn't a worse place on earth one might navigate than a major urban area recently inflicted by a major earthquake. While all other sources cared not about the issue, the world's



OSM community took it upon themselves to harness the free (as in freedom) nature of OSM and map the undermapped areas in Turkey and Syria — and they did so incredibly quickly! Thousands of volunteers around the world mapped millions of buildings and their damage in mere days while the Turkish community were providing readymade layers of the data via VPN over satellite link, whose direct connection to OSM servers was being censored counterproductively by the Turkish

## Freedom and privacy on the web: Our interview with the JShelter team

*By Greg Farough  
Campaigns Manager*

*Our full, unabridged interview is available at: [u.fsf.org/3zo](http://u.fsf.org/3zo)*

In order to catch up with development on the browser extension JShelter, which limits the degree to which JavaScript can control your browser, Greg from the FSF campaigns team (GF) conducted an interview with lead developers Libor Polčák (LP) and Giorgio Maone (GM).

**GF:** *How would you describe JShelter?*

**GM:** To me, JShelter is an excellent exercise in trying to enumerate, analyze, and mitigate the almost countless ways modern web APIs

government! The outcome was a detailed map of the damaged terrain, marked without draining any of the already scarce human resources from the field.

Through this experience, we felt how the power of volunteers in free software communities, such as those working on OSM development and mapping, can exceed what governments and corporations merely pretend to provide.

Stay safe and keep mapping. 🙋

available to JavaScript can be abused to collect fine-grained data about users' hardware and software configuration. This data, in turn, is collected to reliably fingerprint, identify, and track users across websites. NoScript takes the blunt approach of disabling JavaScript and other active content by default, and allows users to enable specific, trusted scripts and sites one-by-one. This approach reduces the attack surface — not just for privacy, but for security in particular (e.g. to defend against zero-day remote code execution attacks). JShelter, on the other hand, assumes all JavaScript is enabled by default, but focuses on counteracting scripts' ability to spy on users by creating a “fake” execution environment, different each session, where user/device-specific information is randomized, quantized, or obfuscated, therefore preventing

global privacy-violating correlations.

**GF:** *How did the project start?*

**LP:** The project started in 2017 at Brno University of Technology, when Zbyněk Červinka asked me to work on a diploma thesis that would aim at creating a privacy-improving WebExtension. At that time, we did not know what exactly we would like to create.

In 2020, we had a WebExtension, a couple of users, and a lot of ideas about what would need to be done before JShelter could be used daily. We applied for a grant from the NGIO PET Fund, a fund established by NLnet. Interestingly, the FSF had very similar ideas and applied for the same grant. NLnet introduced us, and we have worked together since. What started as “JavaScript Restrictor” has transformed into JShelter, now with a new UI, nice web pages, improved reliability, and improved protections.

**GF:** *Can you give some examples of what malicious JavaScript on the web can do? How does JShelter circumvent it?*

**GM:** We tend to forget that every web page we visit is potentially a full-fledged application written in JavaScript, and, nowadays, the capabilities available to these are comparable to native applications. The main difference between the two is that we typically choose to install

a native application with intention, and we can usually vet its usefulness and trustworthiness in advance, while, in the case of web apps, we may just “stumble upon” malicious web pages at random.

Now, very nefarious things can happen if an evil web page exploits a browser vulnerability to escape and access your system, trying, for example, to install a rootkit or to exfiltrate local data. These attacks are considered to be relatively rare and targeted. More commonly, “regular” web APIs are abused, without need for specific unpatched vulnerabilities, either to “parasite” your computational resources (e.g. by cryptominers) or to spy on your online behavior, possibly to sell your profile to data brokers. The latter is the specific threat which JShelter addresses as its priority.

**GF:** *In your opinion, what does the “perfect world” for the web look like? For example, would you put specific limits on what JavaScript can or can’t access?*

**GM:** The web was originally envisioned as a mere collection of hypertexts: just static interconnected text documents. But today’s web is a chaotic execution environment, almost an OS of its own, with very loosely defined boundaries across applications and little or no user awareness of its power and potential for abuse.



Of course there's no coming back from the web as an application platform, which also has benefits, such as its openness and democratic access. But we need to provide users with more control on the trusted entities allowed to run code on their systems (for instance by creating and enforcing a code-signing mechanism) and on the permissions granted to those applications, which is something we (browser vendors and standardization bodies) started to do only recently with the most powerful and privacy-sensitive APIs, e.g. geolocation or microphone/camera access.

*GF: How can someone get involved with the JShelter project?*

*LP: Using JShelter and reporting any problems in our issue tracker is a huge*

help. If someone wants to contribute to the project itself, the best way is to post the ideas on the issue tracker or mail them to us. For simple changes, creating a pull request might be enough. We are also looking for a research partner.

*GF: Thank you both for the interview!* 🍷

**Get 10% off!**

**Support the FSF by  
purchasing FSF merchandise!**

**Visit [shop.fsf.org](http://shop.fsf.org) and enter  
discount code **SPRING2023****

**Valid: 6/1/23 - 8/15/23**

## Same message. New design.

Start a conversation wherever you go with our new DRM shirt.



- Ethically manufactured in the US
- Supports the FSF and GNU
- Conversation starter



Get yours at [shop.fsf.org](http://shop.fsf.org)  
We'll start shipping the shirts in June.

*Colors may vary from final product. Logo placement and size are approximate.*

## Trademarks, volunteering, and code-generating LLM

*By Craig Topham, Copyright &  
Licensing Associate*

This year's LibrePlanet saw a large swath of fantastic talks on a wide variety of topics related to free software, including a track specifically dedicated to licensing. This track featured three talks: one about the free licensing of trademarks, another hosted by the FSF's Licensing and Compliance Lab, and a post-LibrePlanet presentation on the ethics of code-generating large language model (LLM) systems.

On the first day of LibrePlanet talks, Julian Daich from LINDS, a 501(c)(3) nonprofit that is developing biomedical technology released under free licenses, presented an interesting approach to applying free licensing to trademarks. As you may already know, trademarks are used to identify services or products as authentic to a specific organization. Daich is working toward a system of trademark licensing that enables the identification of free works by a free trademark. This system includes, at the same time, the licensing for the commercial use of the trademarked services or products. The hope is to offer a method for uncompromisingly protecting the freedoms of work created by a concerned community, while offering a system for that same

community to generate revenue via commercial (but still free as in freedom) use and distribution under different branding.

Closer to the end of the day, the licensing volunteers of the FSF's Lab gave a talk about their history with free software, as well as the experiences and benefits of working as licensing volunteers.

Yoni Rabkin, who facilitated the talk and has been a volunteer since 2006, expressed how he identifies with the ideals of the free software movement, and how the combination of his passion for free software and legal work made being a licensing volunteer a perfect fit.

Panos Alevropoulos, former FSF intern and most recent addition to the team of licensing volunteers, described the importance of informing the public about software licensing, invoking the concept of "legal awareness." Legal awareness is when everyday people are empowered with basic literacy in common law. When legal awareness is achieved, the public can apply this knowledge to better participate in a democratic society, becoming informed and active citizens. In today's technology-pervasive world, it is important to create awareness about free software and the laws surrounding it. Panos also commented on the Free Software Directory as a robust resource for assistance in free software licensing

and cataloging of free programs.

Paulius Gaulubickas, another licensing volunteer, touched on the need for an international body to facilitate arbitration for enforcing free software licenses on a global scale, especially for those without access to legal assistance.

Yuchen Pei and Ineiev were unable to attend LibrePlanet 2023, but I'm honored to mention them here for their great contributions as volunteers to the Lab.

And as for me, I was able to describe the time when I first discovered the free software movement, which has provided me a vehicle to express myself in order to make the world a better place through free software. To watch a video of the complete talk, please go to: [u.fsf.org/3zn](https://u.fsf.org/3zn)

A few weeks after the in-person conference, Harm de Vries, a staff research scientist at ServiceNow, and Leandro von Werra, a machine learning engineer at HuggingFace, represented the BigCode Project and gave an online presentation on one of the hottest topics of today, the ethical implications of code-generating LLMs. The BigCode project is a scientific collaboration (with over 350 participants) working on the responsible and ethical development of code-generating LLM systems. In their talk, they discussed how they navigate the legal, ethical, and

governance-related aspects around the development of these models, including how to develop a permissively licensed code dataset. They also feel it is essential to give developers the option to remove their code from the training data, redact personally identifiable information (PII), and attribute generated programs to the original code snippet. This workshop has not been published yet, but stay tuned to the LibrePlanet 2023 talks page for its eventual release.

On Sunday, the Lab hosted a licensing table at LibrePlanet. The intention was to provide a space for an open discussion about free software licensing, where conference attendees could drop in, either virtually or in person, and ask questions. I enjoyed the conversation with attendees and was delighted to see their interest in free software licensing. We hope to expand the licensing table in the years to come.

LibrePlanet continues to be a magnet for innovations in ethical licensing of which I am proud of being a part, and I already look forward to LibrePlanet 2024. I hope that you will watch the recorded talks, and I look forward to seeing you at a future conference, whether online or in person. 🤖



Donate to the FSF with Bitcoin:  
1JZYAhVmMPCqZdkkj  
LUj9SFfGQaLCHFpGk

Copyright ©2023  
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 twice yearly by the Free  
Software Foundation, 51 Franklin  
Street, 5th Floor, Boston, MA 02110-  
1335, (617) 542-5942 [info@fsf.org](mailto:info@fsf.org)

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

## **IMAGE CREDITS**

Page 1: Logo design by Manufactura  
Independente\*  
Page 3: Photo by Red Shade Studios\*  
Page 5: Photo by Alper Atmaca. Copyright ©  
2023. CC BY-SA 4.0.  
Page 9: Image by Miriam Bastian\*

\*Images Copyright © 2023. Free Software  
Foundation, Inc., licensed under a Creative  
Commons Attribution ShareAlike 4.0  
International license.

## **How to Contribute**

### **Associate Membership:**

Become an associate member of  
the FSF. 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](http://forum.members.fsf.org)! To  
sign up or get more information,  
visit [member.fsf.org](http://member.fsf.org) or write  
to [membership@fsf.org](mailto:membership@fsf.org)

**Online:** Make a donation at  
[donate.fsf.org](http://donate.fsf.org), or contact  
[donate@fsf.org](mailto:donate@fsf.org) for more  
information on supporting the  
FSF.

**Jobs:** List your job offers on our  
jobs page: [fsf.org/jobs](http://fsf.org/jobs)

**Free Software Directory:**  
Browse and download from  
thousands of different free  
software projects:  
[directory.fsf.org](http://directory.fsf.org)

**Volunteer:** To learn more, visit  
[fsf.org/volunteer](http://fsf.org/volunteer)

**LibrePlanet:** Find local groups in  
your area or start your own at  
[libreplanet.org](http://libreplanet.org)! And join us  
in person or online for the yearly  
LibrePlanet conference next  
spring.

**Free Software Supporter:** Receive  
our monthly email newsletter:  
[fsf.org/fss](http://fsf.org/fss)