



Contents

page 1: Protect your freedom by
managing your privacy

page 4: GHM 2022 in Turkey: A
personal reflection

page 6: Copyright assignment with
the FSF

page 8: Charting a course to a free
world

page 9: LibrePlanet 2023, March 18-19

page 10: Upcoming release of Trisquel
11, codenamed Aramo

page 11: FSF and GNU anniversaries

Protect your freedom by managing your privacy

By Zoë Kooyman

Executive Director

Privacy is just one of the reasons we advocate for free software at the Free Software Foundation (FSF). The philosophy of software freedom encourages a culture in which computer users are not exploited, but respected. Being able to run, modify, copy, and share (u.fsf.org/2cj) our software is a precondition to privacy. It allows us to trust the software we use, simply *because we have the freedom to check it*. These four freedoms protect us against the

software we run on our devices. Proprietary software and unethical network services continue to violate the trust of users by collecting, studying, and brokering their data.

Today, the abortion rights conversations that are happening mostly in the US, but also in many other countries, are compelling the general public to face these concerns head on, and what we are seeing as a result is a growing public concern about mass surveillance.

Beth Schlachter, director of advocacy with the International Planned Parenthood Federation (IPPF) warns that “the surveillance of pregnancy and abortions using technology and data is increasing.” Users who collect reproductive health data in popular proprietary period tracking apps can have their information used against



*Sneak “peak” at upcoming
Trisquel 11 release (see page 11).*

them. These apps not only collect and study data provided by users, but also send it to third parties, including data tech giants like Meta, where it gets linked to other data available about these users. Tech companies that store and broker your data can be ordered by a court to hand over user information, even if they don't want to.

The trail of data people in today's society shed is easy to follow when they are not being careful, and it's reckless at best to think that the challenges this poses to our safety is limited to the situation in the US, or even to the matter of abortion. In recent years, we have seen similar concerns for the safety of marginalized communities, protestors, and even students.

How can we protect our privacy in a world where mass surveillance has become the norm? It's hard; it is nearly impossible to go completely unnoticed. Common cases show messaging history, cell phone tower location data, and IP address search histories are handed over to authorities—sometimes even without warrants.

Our free software work is all about taking control over our digital lives and protecting our freedom. But the reality is that we cannot control every aspect of all technology in our lives. For example, many people have computers or phones from work or

school that they are required to use. Users do not have any freedoms in regard to the proprietary applications they use, either by choice or because of expectations. We also don't control the software on the other end of our communications with our friends, family, or colleagues. In the end, all software has bugs or can have unforeseen consequences; even free software. And to top all that, it will be a long way before mobile phones—the pocket-sized computers that track our movements, our interests, and our behaviors—are fully free. These are enough reasons for us to be extra-vigilant about how we handle our data. This lack of control chips away at our freedom, and, sometimes, it affects our safety.

That does not mean you are completely at the mercy of technology when it comes to your privacy. You can install a free software operating system on your computer or phone. The fully free Android-based mobile OS Replicant, and others, will give you more control over the type of software you run and decrease your dependence on omnipresent tracking corporations like Google. You can also download F-Droid, a repository of free software applications where you can download apps for almost any purpose, like the free software period tracking applications Periodical (u.fsf.org/3ux) or Drip (u.fsf.org/3uy), which were

reviewed by Consumer Reports as having superior privacy practices, and they “show that it is possible to build a period tracker that doesn’t sell out users’ information.”

You *can* do things to protect yourself, such as monitor your own data trail. You can browse the Web using a privacy-respecting browser such as Tor Browser. There are also different search engines you can select from your settings menu in most browsers. These measures can mask your activity and block any trackers, and they are available on your phone as well. You can also opt to mask your IP address using a VPN, but make sure it is one you trust.

Privacy is also a community effort, and your protection only gets stronger when everyone participates. Use the GNU Privacy Guard (GPG) (gnupg.org) for encryption. The FSF’s Email Self-Defense Guide (emailselfdefense.org) will help you and your friends implement this so that you can encrypt your emails and files as much as possible. We advise GPG because not all communication about encryption can be trusted. You can do some research, and insist on encryption software that is free software and studied by experts for security.

Private conversations are *not* for social media or microblogs. Fantastic apps exist for messaging, maps, storage, and specific needs like period

tracking, all while protecting your privacy as well as your freedom. You can select, and research, many of these applications in the Free Software Directory (u.fsf.org/ky). When using apps, it is good to remember some basic rules: do not allow apps permissions they do not need and remember to turn them off when they are not in use; look for ways to use free software applications as your daily tools instead of proprietary apps; and, in some cases, turning off or leaving your phone at home is a sensible thing to do.

The security of our data and control over our technology is constantly challenged, which is a difficult thing to process, because we would all like to believe we can live freely within the set boundaries of democracy. But what is accepted as free today may not be tomorrow, and in some cases it affects our safety. The FSF has been talking about these dangers for the past thirty-seven years, but right now, today, it is more important than ever to urge others to take measures to protect their safety and our own, as well as the safety of their communities. The more people implement free software, privacy-respecting applications and encryption into their day-to-day lives, the stronger our communities will be in withstanding oppressive mass surveillance. 🙋

GHM 2022 in Turkey: A personal reflection

By Luca Saiu

GNU Hacker

The following is my personal reflection of this year's GNU Hackers' Meeting (GHM) and how we were able to organize an enjoyable meeting against all odds. Enjoy!

According to its definition on `gnu.org`, GHMs are “a venue to discuss *technical* topics related to GNU and free software.”

While it is true that the (more or less) annual GHMs are structured as “conferences” with presentations and slides—very “technical” indeed!—if we look back at every year since their beginning in 2007, we find that GHMs are mostly for the fun of spending time with our GNU friends in a relaxed environment.

After many years in which most GHMs took place in Europe for no particular reason other than us regulars living in Europe, we opted to hold GHM 2022 in Turkey: `u.fsf.org/ghm22`.

There have been various volunteer organizers over the years. This year, I organized the talks, schedule, Web site, and most of the remote communication with participants. Since my wife Egeyar Saiu is not only a capable computer scientist but also a practical person who speaks Turkish and knows the region, she dealt

with the hotel and restaurants. Ege also proposed the dates and location: October, because the weather is pleasant that time of year; and Izmir, a large university city on the Mediterranean coast. Planning the GHM was tiring, but it all worked well in the end.

And then, of course, the main reason for a new venue: to reach new people. We hoped for locals to see the announcement about a GNU meeting nearby and attend out of curiosity, and those people may become friends and contribute to some common project.

Several people from Izmir reacted to my announcement with great excitement. However, in the end, only a few people in Turkey could attend in person, so we resorted to some remote and pre-recorded presentations. Thanks to our good ol' friend (and master hacker) José Marchesi, we were able to stream most of the event live.

Ege and I arrived in Izmir beforehand to test the equipment at the venue. The following day, we were joined by José and by some early attendees for the pre-event dinner. While there, we had a surprise: two of the participants, with whom we had exchanged only emails prior, turned out to be younger than we thought: thirteen and fourteen years old! Both were running GNU/Linux on their laptops and spoke good English;

they were not expert programmers (yet), but eager to learn. It turned out they already understood the free software philosophy very well.

The next two days were dedicated to the conference itself. Of the attendees, some were professionals: expert embedded programmers, for example. There was one older student, obviously very knowledgeable; a couple of newly-enrolled students about to begin their university careers—one a Lisper, who asked several deep questions; the other an enthusiastic, committed but shy fellow, who eventually found the courage to give a talk after my Jitter workshop. He did it and promised to improve his English in advance of the next hacker gathering. My take: “It was already awesome, friend.”

Sometimes we had to improvise in order to overcome a technical difficulty such as when our server connections dropped; our loudspeakers squeaked, we lost a cable, and we were unable to stream one presenter

who uses only the terminal console. ...Looking back at it now, I smile at that chaos, which has blurred into a cheerful memory.

Near the entrance, we covered a table with free software stickers for the attendees and their name badges, each card prepared with my pathetically old printer, and then cut and inserted (manually by me) into a plastic holder. And even if our hackers’ table was nowhere as perfect as the hotel’s flawlessly arranged snack spreads, one must at least concede that our hackish organization was *sincere*. That sincerity is worth something.

The Web page now contains most video recordings and presentations slides. But recordings do not tell the whole story. Anybody interested in participating (or assist with organizing) is welcome to subscribe to the ghm-discuss mailing list at u.fsf.org/ghmlist and join us next year! 🐧



Final group photo of GHM 2022 attendees.

Copyright assignment with the FSF

By Craig Topham

*Copyright & Licensing
Associate*

Since 1985, the FSF has promoted computer user freedom through a myriad of campaigns including the steadfast and demonstrated promise to support the GNU Project. To this end, the FSF holds the copyright on the family of GNU General Public Licenses (GPLs), and a large portion of the GNU operating system through copyright assignment. We have been doing this work for thirty-seven years. In order to give us the best position to uphold our commitment to the GNU Project, we need contributors to not only contribute their code, but to do so in a way that keeps the GNU Project legally secure. This is accomplished through the FSF's copyright assignment program (u.fsf.org/3fu).

Because the copyright assignment program is so vital, we feel it is an important topic to revisit from time to time and describe what happens when a contributor assigns copyright to the FSF for a GNU package. First and foremost, it is well worth noting that in this process, the contributor gives an altruistic yet practical gift to the free software community in the form of free software. Writing code that protects its users' freedoms is

truly an act of kindness which benefits all of society, and assigning copyright to the FSF entrusts us to defend and protect such freedoms. Taking the legal steps to keep such freedoms intact is paramount because software freedom is under constant attack. If the opponents of free software had their way, all software would be proprietary and under the control of a select few. It is fair to say that copylefted, free software has made it difficult for those seeking control over users.

Simply put, copyright assignment is an agreement between the developer and the FSF. By entering into this assignment agreement, the developer's rights as author of a work, which are protected by law, are transferred and the FSF becomes the new copyright holder. The primary driving force behind this approach is that the FSF can defend the copyright in court, if necessary. This is because only the copyright holder is in a legal position to do so. While this approach keeps the FSF in the best position to uphold the GPL, there are a lot of misconceptions around the process and outcome.

Some developers worry that assigning copyright will strip them of all their rights to the code they've created. To address this, the FSF includes a "license grantback" to the developer in the agreement contract. For the developer, a license grantback

means they can continue to modify and share their code, and technically, they could even distribute their software under a different license. In other words, by assigning copyright to the FSF, the developer does not give up any of these sorts of rights.

Another common concern among developers pertains to what exactly is being transferred. The agreement contract makes clear that the author's copyright for changes and/or enhancements to a specific GNU package is what is being transferred, and nothing else. However, the ultimate decision to submit code is firmly left in the hands of the contributor. When the contributor submits code to the GNU Project, this fulfills the requirement to report the changes and/or enhancements covered by the agreement with the FSF. If the contributor decides not to submit their changes and/or enhancements to the project, then the agreement does not apply to that code. What the copyright assignment does not do is force the developer to contribute all code they write. Although it is very rare, the contributor may also send a physical letter enumerating the assigned files to the FSF if they wish to be more formal or specific.

Free software is under constant attack, which is a warning worth repeating. One very real, and very concerning, vector for such an attack

is when an employer claims the outside work of their employee, or when a university does the same with a student. The last thing anybody in the free software community needs is an army of well-paid lawyers from a large corporation or university pulling apart code and making copyright claims. In order to prevent this, the FSF requires employer disclaimers from contributors employed to program, and university disclaimers from students enrolled in a university. These help remove the danger of proprietary claims on a developer's contributions to any GNU packages. Although assigning copyright for a specific GNU Project package is a one-time process, a contributing developer may have several jobs throughout their career; therefore, it is important for the developer to get an employer disclaimer in place when starting any new programming job.

I hope this article broadens your understanding of the FSF's copyright assignment program. With this knowledge, I also encourage you to join us and the thousands of hackers around the world who have placed their work's copyright and their trust with the FSF. With the community's contributions securely under the FSF's protection, the FSF can keep a vital promise which is also contained within the agreement: we will *always* keep the software free. 🐧

Charting a course to a free world

By *Greg Farough*
Campaigns Manager

Over here at the FSF, we're often accused of being a "utopian" organization. That is, when we're not being called out for presenting so many aspects of digital life as a *dystopia*. But, for a moment, let's forget about what Apple, Microsoft, and the corporation formerly known as Facebook are up to and consider this "utopian" vision. If we're going to make any progress on the FSF's (and assumedly your!) mission to bring freedom to all computer users, envisioning just *where* we want to go can help us keep the goal in mind.

At least as evidenced in our most recent talks with American high schoolers, it's rare to encounter people who really grasp just what the word "program" means, let alone why you ought to be unrestricted in how you run one, and free to share it if you wish. What we need (and what free software activists are working for) is a world where users are able to do just that: copy programs from a friend, study them, and change them to suit their needs before offering it to others. Ideally, and as GNU's education pages (gnu.org/education) helpfully point out, sharing programs would begin in the classroom.

While it doesn't promote the idea of software freedom directly, Alan Kay's original essay on the "Dyna-book," (u.fsf.org/dyna) the so-called "personal computer for children of all ages" outlines some of the best parts of what a free software world might look like. If you're unfamiliar with it, it's well worth a read. Despite its being written fifty years ago, we should still think of it as pre-figuring the world to come: where, just like in the paper, children are editing programs on the fly in the classroom, transparently accessing the world's knowledge without onerous burdens on their freedom, and learning as much about outer space as they are about the basic elements of computing.

Free software developers arguably are the people who benefit the most from computing freedom, given that they're the members of our community currently able to make the most use of that freedom. From where we stand today, there's an increasingly widening gap between developers and the everyday users of software. It's true that software development requires certain special skills, but it's not these skills alone that have caused such a wide disparity between those who can and can't get their computer to "compute" things in the way that they want by talking to it in a programming language. After all, imagine how hectic things would get if that

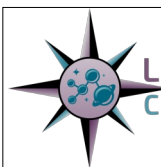
same gap existed between people who can and can't drive a car! Or, perhaps even worse, dependent on someone to drive you somewhere who does not let you decide how you get there.

It's my opinion that this disparity has been a calculated move on the part of large proprietary software development firms which, from there, has trickled down into the way we all use computer software. Users are often left in the dark on what the software they use actually does, and are at the mercy of whatever these corporations say is necessary for the program to function. *I refuse to believe that an email address or Wi-Fi connection is necessary for a vacuum cleaner to function.*

If free software had been the norm from the beginning, that gap between mere users and *developers* of software wouldn't have become so profound. In that world, where free software was the norm, the average computer user would look a lot more like the "power user" in the GNU Emacs (gnu.org/s/emacs) or general GNU/Linux communities. That is, someone who knows how to customize their environment to their liking and who's written a few commands or aliases to make life a little easier.

Such an individual might not have a burning desire to learn about assembly languages and other lower

levels of operation, but they're able to *extend* the program to fit their needs. If it were common for users to write as many pieces of code extending a program as the GNU Emacs (and GNU Guix) communities do, we'd not only have better programs; we'd also have a healthier relationship to technology. In such a world, it'd be harder for programs to actively abuse users. It also would be one where an actually beneficial relationship between humans and machines takes hold: not a fake, buzzword driven "metaverse" where a single corporation dictates how users can express themselves, but one where every user is equipped to *extend* the programs that they use however they please. Keep this in mind when considering the free software movement, and please think about the world you're working toward when it comes to your own free software advocacy. As a computer user, it's your movement! 🐼



LIBREPLANET 2023
CHARTING THE COURSE

Registration for LibrePlanet 2023: Charting the Course, happening both online and in person March 18-19, 2023, is now open!

Have a look at u.fsf.org/lp23, and sign up to the announcements list on top of the page for updates. 🐼

Upcoming release of Trisquel 11, codenamed Aramo

By Andrew Engelbrecht, Senior Systems Administrator; contributions by Rubén Rodríguez, Trisquel Hacker



At the FSF, we make a great effort to do our work with free software. As a systems administrator, an ethical distribution such as Trisquel is a relief, because it means that everything, from kernel components to the browser plugins, has been vetted. Recently, I had the pleasure to learn what's up and coming in the soon-to-be-released version of Trisquel. Here's what I found out.

At the time of writing, the developers working on Trisquel, a fully free distribution of GNU/Linux (u.fsf.org/1td), have completed the main packaging work for the upcoming Trisquel 11 installable/bootable disk images. They're planning to release to the wider world by year's end, and with all new support for 64-bit Arm, POWER9, and Power10 CPU architectures, the latter of which are supported by modern Respects Your

Freedom-certified hardware like the Talos II Mainboard (ryf.fsf.org). The FSF runs Trisquel on most of our servers and office computers, and we host colocation space and support for one of Trisquel's development machines.

This release reverts a few troublesome changes in newer versions of Ubuntu, its upstream distribution. For example, the latest Ubuntu releases (22.04 and newer) aren't distributing Firefox as a .deb (Debian) package but instead via the "snap" repository, which unfortunately contains proprietary software and is exclusively hosted by Ubuntu servers.

Rubén Rodríguez, the main Trisquel developer, tells us that contributing to the project is now much easier than before. Developer meetings are held on Wednesdays at 12:00 EST (17:00 UTC), so if you're interested in helping out, please make sure to swing by and say "hello!" Details can be found on the Trisquel Web site: trisquel.info.

Get 10% off!

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

**Visit shop.fsf.org and
enter discount code FALL2022
11/1/22 - 1/15/22**

Trisquel has always had a great default theme, and it ships with updated artwork that I personally look forward to seeing each time there's a major new version. This year's theme, chosen, as always, by the Trisquel community, is Aramo. Aramo is a Celtic deity, protector of paths and crossroads, and "El Aramo" is a mountain range in northern Spain. The artwork is based on Rubén's own photos of these mountains (featured on page 10).

If all goes according to plan, then soon after reading this article, you'll be able to download, run, and install the beta release of Trisquel 11 from their Web site. Happy upgrading! 🐧

FSF and GNU anniversaries



This fall, both the FSF and GNU had their thirty-seventh and thirty-ninth anniversaries, respectively.

The free software community celebrated in many different ways, and free software advocate Jason Self made the artwork featured above in honor of the festivities. The artwork has Freedo (the fully-free Linux-libre mascot) and a GNU enjoying a colorful cake together. Read more about the history of FSF and GNU at fsf.org/history. 🐧

All Gnu Baby. All new.



- ✓ Ethically made in Germany
- ✓ Supports the FSF and GNU
- ✓ Stuffed with kindness

Get yours at
shop.fsf.org



Donate to the FSF with Bitcoin:

1NuAuEc68LPLKQKms
s68CPiJFtGht548NW

Copyright ©2022

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/>

Published twice yearly by the Free Software Foundation, 51 Franklin Street, 5th Floor, Boston, MA 02110-1335, (617) 542-5942 info@fsf.org

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

IMAGE CREDITS

Page 1: Image by The Trisquel Project © 2022, GFDL.

Page 5: Image by GNU Hacker © 2022, CC BY-ND.

Page 9: Image by Zoë Kooyman.*

Page 10: Photo by Rubén Rodríguez, CC BY-SA.

Page 11: Image by Jason Self © 2022, GFDL-v1.3-or-later.

Page 11: Image by Davis Rimmel.*

*Images Copyright © 2022. 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! 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! 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.