



Bulletin

Contents

Respecting freedom is better for business 1

Free software and climate change 3

On the road with RMS 5

Respects Your Freedom in 2017: The best year yet! . . . 5

High Priority Projects update 7

Six months of equipment upgrades at the FSF 8

Free software in government: Munich and LiMux 9

Weekend projects: Make your own email server 10

protect computer user freedom when the surrounding laws are failing to do so. Nevertheless, since every use of the GPL – whether motivated by concern for freedom or not – helps more users be free, it makes sense to talk about the potential monetary benefits of copyleft.

Because commercial use has to be allowed by a program’s licensing terms for it to be considered free software, GPLed software does make a lot of businesses and a lot of people a lot of money. While the specifics in this area depend on empirical research, there are structural reasons to believe that the GPL is even better for this than lax permissive licenses.

Respecting freedom is better for business

*By John Sullivan
Executive Director*

The GNU General Public License (GPL) does not exist to maximize a company’s ability to make money compared to proprietary mechanisms. Making money at the expense of people’s freedom is not an ethical way to make money, and the overriding purpose of the GPL is to



Before we start, let’s set expectations and state the obvious – people don’t have an inalienable right to make money in every given way just because it’s possible to do so. We

know this, but it is still forgotten when people start talking about how policy impacts profit, including whether the government should take action to protect certain business models. We have innumerable laws regulating sale of controlled substances, banning various kinds of money lending, requiring minimum wages for workers, and so on. Many of the activities now restricted were allowed at some point in history. Even if the GPL turned out to be worse than proprietary terms for selling software, this wouldn't mean rejecting the GPL.

But it turns out that the GPL is very well-suited for commercial usage, and even better than lax permissive licenses. The GPL ensures that you will benefit from improvements others make and distribute to your code. With lax licenses, others, including your business's competitors, can fix bugs in and add features to your code, then share the result under proprietary terms so that you can't make use of them. When using lax licenses, you are essentially doing unpaid work for your competitors. With the GPL, you are engaging in a mutually beneficial form of targeted cooperation with them, enabling you to focus your time and resources on what truly differentiates your business. You are also maximizing the common interest everyone distributing that code has in making it the best it can be.

Similarly, the GPL also enables constructive and profitable cooperation with individual users. People purchasing and using your product will have ideas about how to make it better. Providing them with the source code and permission to make modifications enables them to actually share

these improvements with you. Any free software license does this, but using the GPL sends a motivating signal to your users encouraging them to participate. It says you respect them, are committed to software that respects their freedom and protects them, and are not just dipping a toe in while keeping the option of pulling the rug out from under them. Users respond to this, and we at the FSF are happy to help you publicly highlight your commitment, to make sure our supporters know.

This is also why you should specifically use GPLv3, which prohibits locking down the devices shipping your software. When the device is locked down, users generally can't install or share their modified versions of the software without violating the Digital Millennium Copyright Act (or similar laws around the world) – a criminal offense. Users will not be able to test their innovations. This means no thriving community around your products.

Defending your business and your users against patent lawsuits is extraordinarily expensive, and the risk of such suits is a deterrent to user contributions. Protection against infringement claims is therefore a strong third financial reason to use the GPL – especially GPLv3, which strengthened patent guarantees over GPLv2. While software idea patents have been reduced by recent court decisions, they are still an enormous threat, especially to small companies and individual developers.

The advantages of copyleft become more evident over time. Lax licenses may seem appealing at first because they appear to be simpler; works un-

der them can be combined more easily with works under other licenses. But since they allow proprietary versions in the future, they are in the long-term an invitation to the extreme incompatibility, complexity, and compliance costs intrinsic to proprietary software licenses. There may seem to be no chance of someone making proprietary versions of your software now, but that option will always be there, and as soon as someone exercises it, your space will become fragmented and difficult. To work effectively with other companies in that world, you'll end up needing additional legal structure, like trade associations or bilateral contracts.

Lax licensing is free (as long as you are also providing the source code) and so is better than proprietary terms both ethically and for getting some of the practical benefits above. But copyleft – in particular the GPLv3 – is the best choice in the vast majority of situations. Choosing it is not quite enough by itself. To be fully effective, people do have to believe that the GPL will be enforced when its terms aren't followed. Otherwise, the imperative to pass on the same freedoms one receives are just words on paper, and the GPL functions in practice as a lax license.

Because of this, the whole commercial sector benefits from nonprofits like the FSF with an agenda only of protecting user rights enforcing the GPL, preventing freeriding, and keeping the perceived strength of the license high. This is why, once you start making money from your GPLed software, you should start giving back to organizations like the FSF who do GPL enforcement in accordance with

the Principles of Community-Oriented GPL Enforcement, which we drafted together with Software Freedom Conservancy.¹ These principles prioritize the ethical goals of the license – including helping companies properly distribute free software – while holding legal action as a last resort. Support for enforcement work done in this way is an investment in your own success and future. But whether you donate or not, you can and should make use of the freedom and the commercial benefits the GPL provides for you and your business. 🍷

Free software and climate change

*By Georgia Young
Program Manager*

The Free Software Foundation is focused on threats to your freedom, both as computer users and as people living in a world where software touches nearly every aspect of our lives. I want to talk about the increased risk of environmental disasters relating to climate change, and look at several projects that have used free software's philosophy of transparency, sharing, and experimentation to create software that can easily adapt to a community's needs.

As I write, Puerto Rico is without electricity, and desperately in need of food and water, following Hurricane Maria. Forest fires ravaged the Canary Islands, as well as California and the Pacific Northwest here in the United States, and earthquakes in Mexico killed hundreds. Flooding has devastated India, Nepal, and Bangladesh, and Hurricane Harvey

¹u.fsf.org/1yq

dumped fifty inches of rain on Houston, Texas in one week. These are only a few environmental disasters we have faced in recent months, and with thousands of deaths and mass destruction of buildings and infrastructure, recovery from these events will likely take years.

There are quite a few free software projects that aim to help people study and respond to environmental threats. Here are three:

Apache Open Climate Workbench

Climate models let us study current systems and project future outcomes. The Workbench, at climate.apache.org, performs climate model evaluation using outputs from a variety of sources, including the Earth System Grid Federation, the Coordinated Regional Climate Downscaling Experiment, and temporal/spatial scales with remote sensing data from the National Aeronautics and Space Administration, the National Oceanic and Atmospheric Administration, and others. It is licensed under the Apache License 2.0, and welcomes more contributors.

Missing Maps

Missing Maps, at missingmaps.org, maps areas where humanitarian groups are working with vulnerable people to plan disaster response activities and other efforts, so they are prepared when disasters occur. Volunteers anywhere in the world can “trace satellite imagery” into OpenStreetMap, and then volunteers in the community being mapped can

fill in the details based on local knowledge. The Missing Maps blog details some uses of the project, including mapping a typhoon recovery area in the Philippines, and a huge swath of West Africa that was affected by Ebola in 2014 and 2015.

Public Lab

Public Lab, at publiclab.org, is “a community where you can learn how to investigate environmental concerns.” Public Lab creates affordable tools and accessible techniques for people to monitor their environment. Their goal is to help people participate in decisions being made about their communities, particularly when faced with environmental hazards.



A Public Lab team documenting a site

You can buy hardware from Public Lab at reasonable prices, or use their designs and source code to make your own, from a spectrometer for environmental monitoring, to an infrared camera to help monitor plant health, to balloon and kite kits for aerial mapping. They use a mix of free licenses, including Creative Commons Attribution-ShareAlike 3.0 Unported and the GNU General Public License v3.

These projects offer heartening con-

firmation that the FSF and its members are doing things right. As we advocate for free software and its ideals, we enable free software to be used for very sensitive – even lifesaving – applications. If you know of other environment, climate, and public health-related free software projects that we should tell people about, please email us at campaigns@fsf.org.♥

On the road with RMS

*By Jeanne Rasata
Assistant to the President*

Forty thousand! That’s about the number of GNU and free software stickers FSF founder and president Richard Stallman has handed out at the twenty-three speeches he’s given since last June. He spoke in seventeen cities in five countries on three continents, to about six thousand attendees. A quarter of his speeches were in Spanish, a tenth in French, and the rest in English. He spoke at nine conferences, five of which he keynoted.

At the end of May and in June, in Brazil, he spoke in Belo Horizonte, as part of the Federal University of Minas Gerais’s 90th anniversary celebration. He then gave talks in Campinas, Curitiba, and Brasília.

He then visited Argentina, speaking in Santa Fe, Buenos Aires, Salta, and in San Salvador de Jujuy, where the National University of Jujuy awarded him an honorary doctorate, his fifteenth to date. He also met with legislators to discuss the importance of using only free software in state government and in education. Before leaving the city, he gave a speech to workers in the cyber-defense division of Argentina’s armed forces.

In July, in Europe, he gave a keynote speech at the Libre Software Meeting, in Saint-Étienne, France. Next came the International Day Against Digital Restrictions Management (DRM) in Genoa, Italy, where he delivered a speech titled “Nonfree software is designed to bite you!”

In August, he returned to the US to deliver a keynote speech at the Free and Open Source Software for Geospatial conference in Boston, and a talk in Chicago, in September.

Back in Europe, he gave a speech at the Hackathon for Rare Diseases, in Florence, Italy, and keynoted at Informatik 2017, in Chemnitz, Germany. In October, in Bucharest, he keynoted at the Coliberator conference, an annual event organized by Fundația Ceata, a Romanian foundation promoting free software and free culture.

See u.fsf.org/zi for a list of RMS’s confirmed engagements. Please write to rms-assist@gnu.org to extend him a speaking invitation. Please also send us any photographs you would like us to share on his blog, at fsf.org/blogs/rms, or recordings of his speeches for our audio-video archive at audio-video.gnu.org.♥

Respects Your Freedom in 2017: The best year yet!

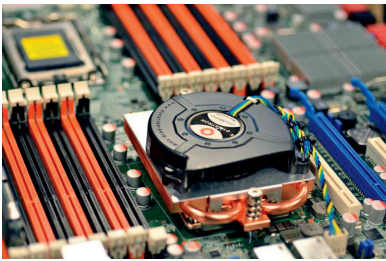
*By Donald Robertson
Licensing and Compliance
Manager*

Our Respects Your Freedom (RYF) certification program helps users to find hardware whose accompanying software can be trusted by users. It is not easy to make a device that

meets our standards. Each device goes through a rigorous review process, and not everyone who applies is able to gain certification in the end. The source code and documentation must all be distributed under a free license. This requires a deep understanding of all the ways that code can hide on a device, and the ability to modify things as needed.

Since our first certification in 2012, the RYF program has slowly gained steam. The program itself is still developing; we've worked to put in place best practices that applicants can follow. The good news is that the foundation laid out over the past five years paid off in a big way in 2017.

In March we saw the certification of three new devices from Vikings. The devices, the Vikings D16 Mainboard, the Vikings X200 libre-friendly laptop, and the Vikings USB Stereo Sound Adapter, expanded the program not just in terms of breadth, but also presented for the first time a mainboard that is certified to Respect Your Freedom.



Then in June, Technoethical (formerly Tehnoetic) launched fifteen new devices! The launch included laptops, peripherals, a mainboard, and more, bringing the total of currently available devices in the RYF program to nearly thirty. With the help of Vikings and Technoethical, we were able to

more than double the number of devices that users can trust in just one year. That is a massive leap forward.



2017 thus far has been a big year, but now that we know what is possible, it's time to aim our sights even higher: RYF has to grow if we are to achieve our goal of making freedom-respecting devices easily and widely available. If we want to scale RYF up to that task, we have to keep adapting and streamlining the work that we do. The RYF team is currently just a few staff members and some volunteers, so building the program to a global scale is no small task.

We hope to end the year with more big announcements, so keep an eye out. Until then, you can see all the amazing RYF devices currently available at fsf.org/ryf. 🐧

High Priority Projects update

*By Dana Morgenstein
Outreach and Communications
Coordinator*

Established in 2005 and most recently revised in 2016, the High Priority Projects initiative draws attention to the projects deemed to be of greatest strategic importance to our goal of freedom for all computer users.

PROJECT: Free phone operating system

- Replicant 6.0, the latest version of the only fully-free Android software distribution, more than doubled the number of supported devices, making it accessible to way more users! It also has important security fixes and other much-appreciated improvements. Replicant is supported through our Working Together for Free Software Fund; please donate to support this project at u.fsf.org/1h1.
- Purism Librem 5 phones, now available for pre-order, will run a GNU/Linux-based operating system called PureOS by default, and will allow users to install a different GNU/Linux distribution if they choose, potentially making this the first phone on the market with fully libre userspace. Purism emphasizes privacy and security, with features that include encrypted text and email support, hardware kill switches, and more. They've already overshoot their fundraising goal, indicating that there is a serious audi-

ence for a fully free phone. Unfortunately, at the time of this writing, Purism has not committed to avoid nonfree blobs – please help us encourage them to do so.

PROJECT: Decentralization, federation, and self-hosting

- In our Spring issue, we interviewed an author of Mastodon, a federated social network platform compatible with GNU social, about its explosion of interest and users. It is now approaching 1 million users, with over 132 million posts!
- In September, Mastodon announced support for the federation specification ActivityPub. ActivityPub achieved World Wide Web Consortium Candidate Recommendation status in September, and will be used by the decentralized media sharing platform GNU MediaGoblin, among others. 🍷

Get 10% off!

Support the Free Software Foundation by purchasing GNU Press merchandise

Visit our shop.fsf.org with new stuff coming soon! Enter discount code FALL2017

Six months of equipment upgrades at the FSF

*By Ruben Rodriguez
Senior Systems Administrator*

Last equinox marked a milestone for infrastructural renovation, bringing us a new generation of freer, faster and more reliable servers.

In the last year, we have given some venerable machines a well-earned retirement. One office server we replaced was a sixteen-year-old Pentium III, our Nagios server called Klaxon. We also retired our cranky virtual machine server Pegasus, and our mystical router Valis – which had been on for over two thousand days! Those services now run on a Respects Your Freedom-certified server running an ASUS KGPE-D16 motherboard, with Libreboot and Trisquel.²

D16 motherboards are at the center of another larger infrastructure renovation project that took over a year of research and development from the whole tech team. Special thanks to our 2016 summer intern Samuel Cantero, who got the research rolling, and our 2017 summer intern Andrew Cabey, who helped wrap it up. The result is a High-Performance/High-Availability cluster of Ceph storage servers attached to host machines running Qemu-KVM/libvirt.

Stored on large, fast, solid-state replicated disks, and shared through a reliable 20Gb/s network, this new server stack will replace and extend our existing main infrastructure. It takes full advantage of redundancy for all components: anything from a network

cable to an entire machine could fail without services going down, and the duplicated components also boost performance and throughput. The combination of Ceph and libvirt also provides great features like live migration of services between different hosts for maintenance or load-balancing, and the ability to instantly clone a virtual machine into a development copy where new features can be tested, lowering the maintenance cost.

We made great progress on the software front as well. Our Web developer, Andrew, has upgraded and polished the services we rely on the most every day, including RT, our ticket tracker; Brains, our wiki; and our CiviCRM instance, which hosts our members, donors, events, and campaigns. Now that we have the room and the processing power to host new virtual machines, we're ready to begin the second phase of renovation. We'll be upgrading the software on the many public-facing services that we host, and making deep changes to the deployment system to automate background tasks including deployment, backup, and monitoring. This will simplify maintenance and make the service more robust.

The first machines that got moved into the new infrastructure were the Savannah servers hosting the GNU Project collaboration and code-sharing services, and they will be followed with a new stack of mail servers focused on better spam filtering, improved reliability, and a modernization of our Mailman, hosting over three thousand mailing lists for hundreds of free software projects.³⁴

²fsf.org/ryf

³savannah.gnu.org

⁴lists.gnu.org

All of this investment and effort could not have been done without the generous donations that sustain us. Big renovation projects require extensive research and development, and expensive hardware as well, but are necessary to keep our efforts going and growing. Thank you for your contribution to this work!❤️

Free software in government: Munich and LiMux

*By Molly de Blanc
Campaigns Manager*

Government adoption is an important step for the advancement of free software. When governments make the switch from proprietary technology, larger-scale change may follow: workers who use free technologies bring them home from the office, and students bring file formats, specialized software, and services like online homework submission systems home from school. Government offices also purchase software on massive scales, and their money can have large-scale impact on technology.

Government use of free software is also good for the governed. Using free file formats, for example, means that government-produced documents and studies can be accessed by any user – and digital evidence being levied against someone can be viewed by defense teams. Citizens aren't locked into particular software to do their taxes or submit a petition. Freeing government software democratizes government software.

Also, government software is paid for with tax dollars, and technology paid for by citizens should belong to

citizens. This understanding is what drove the initial development of the US Federal Source Code Policy, and is inspiring similar discussions within the European Union. Within the United States, the FSF has worked on this issue with state governments, including New York and Massachusetts, and we have worked with the US Department of Defense to formulate policies around creating and sharing software.⁵

Many governments and agencies have created custom free software modified for their specific needs, including Cuba, Turkey, Venezuela, autonomous regions in Spain, and, finally, the city of Munich.⁶ Munich is an example of the successes of government adoption of free software, but also the forces that can undermine its use.

In 2003, the city council of Munich voted to plan a migration from a Microsoft-based system to a GNU/Linux one. By 2004, bidding was open for companies interested in performing the migration, but was temporarily halted due to patent concerns. (This came to light at the same time as a 2002 HP memo stating that Microsoft was planning to launch a “patent-based legal offensive against [GNU/Linux] and other free software projects.”)⁷

Nevertheless, after an extended pilot study, migration started in 2006, when the mayor's office started running on Debian GNU/Linux. By 2013, more than 15,000 machines moved to LiMux, which was customized for Munich. Along with LibreOffice,

⁵u.fsf.org/2cr

⁶u.fsf.org/2cs

⁷u.fsf.org/2ct

LiMux includes WollMux, a LibreOffice extension, to handle the templates, forms, and letterheads used by the city.

Of course, every system has issues, and there were complaints, both during the switch and after. However, in 2012, the number of monthly complaints to the city’s IT support department dropped over 30%, in comparison to when they were using Windows. The free software-based systems have cost millions of Euros less than maintaining proprietary services. The migration did not cost significantly more than the alternative proposed at the time, which would have simply updated Windows and Microsoft Office. In addition to the respect the use of free software has shown employees and residents of Munich, these financial benefits made a compelling argument for those focused on the bottom line.

When Microsoft began expressing interest in moving their German headquarters to Munich in 2014, mayor Dieter Reiter proposed returning to Windows – a move which wouldn’t make sense in light of the success. From a practical standpoint, GNU/Linux was saving Munich money while making the correct choice from a rights standpoint. While Josef Schmid, Munich deputy mayor, described the original move to GNU/Linux as being “driven by ideology,” rather than “financial prudence,” Karl-Heinz Schneider, head of IT services in Munich, insists there is no technical reason to change.

As of October 2017, Munich has not made the decision to downgrade to Windows, but a vote is scheduled for November.⁸ We hope the city

council will maintain the status quo, rather than kowtowing to the misleading language of corporate interests. We would love to see more governments adopting free software policies, in procurement, standards, and development, and we encourage free software supporters to urge their politicians and institutions to usher in the changes we need.♥

Weekend projects: Make your own email server

*By Ted Teah
Copyright and Licensing
Associate*

Setting up your own email server is a nifty project for free software enthusiasts, which gives you complete freedom over how you read, organize, and store your email. It took me only one weekend, and wasn’t that hard for me, as a GNU/Linux user with intermediate knowledge.

A few good reasons to have your own email server:

- 1) You have far more control over the data, and thus more privacy.
- 2) You control the policies of the server, so there are no arbitrary restrictions on receiving or sending mail.
- 3) You decide what features go in, and how they’re configured.

First, make sure that you have access to the specific TCP ports that you’ll need. These ports are often blocked by your residential Internet Service Provider, so you usually can’t use your basic home Internet connection. Even when available, these messages are more likely marked as spam by email recipients, so one option is to

⁸u.fsf.org/2cu

set up your own mail server for receiving mail, but use an external Simple Mail Transfer Protocol (SMTP) server for sending mail.

Another solution to the port problem is to find a server that does not have any ports blocked on its network connections – I used a Virtual Private Server (VPS). In addition to the server, you will need a domain name, and have its mail exchanger record (MX record) point to your server.

Next, consider whether you want to use a turnkey solution, or put your server together from scratch. Newer users might consider a simpler approach like Inboxen, Mailcow, or Citadel. Or, if you're going to cobble together the main components on your own, you need a mail transfer agent (MTA), mail delivery agent (MDA), and an Internet Message Access Protocol/Post Office Protocol server (IMAP/POP3).

I picked Postfix, a popular piece of free software, as the MTA to route and deliver the electronic mail through SMTP. Postfix is nice in that it supports two storage formats: mbox, a system where all messages are concatenated and stored as plain text in a single file, and Maildir, a system where each message is kept in a separate file with a unique name, and each folder is a directory. For the MDA and IMAP/POP server, I went with Dovecot. These two programs are mainstays in the field, and there are a lot of different channels of support.

You can also employ additional components to combat spam and viruses: I recommend SpamAssassin for spam, and ClamAV is basically the standard in malware protection. You can even choose a webmail pro-

gram to view, send, and manage your email: there are ample choices, including Cypht, Mailpile, RainLoop, and more.

Once you have all of your components, try finding an online guide to walk you through the process (which sadly, we don't have space to outline here). For my distribution, Parabola, I followed the guide at HowtoForge.com – see u.fsf.org/2cg. I had to do a little outside research here and there, but it was pretty straightforward. For Advanced Package Tool based systems there are many guides to follow, but there is one that looks like a good starting point at scaron.info – see u.fsf.org/2ch.

At the end of your weekend, you'll hopefully have a secure, personalized email system that you control fully. As one of our sysadmins pointed out, most of us think of our email as an extension of ourselves, so it's worth the effort for the increased peace of mind! 🍷

CREDITS

- Page 1: Doodle by Dana Morgenstein, photo by Ruben Rodriguez, Creative Commons Attribution-ShareAlike 2.0 Generic License.
- Page 4: Photo by Stevie at Public Lab, Creative Commons Attribution-ShareAlike 3.0 Unported License, from <https://i.publiclab.org/notes/stevie/03-22-2017/how-to-visually-document-a-site>, 2017.
- Page 6: Photos by Ruben Rodriguez, Creative Commons Attribution-ShareAlike 2.0 Generic License.



Donate to the FSF with Bitcoin
15K53iB8oEqXB42ebyzEY4SMoLdVQqghLA

Copyright ©2017
Free Software Foundation, Inc.

The articles in this Bulletin are individually licensed under the Creative Commons Attribution No Derivative Works 4.0. <https://creativecommons.org/licenses/by-nd/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 LyX, L^AT_EX, and GIMP.

Thank you to Alex Jordan for helping us to conceptualize and edit this issue of the Bulletin.



How to Contribute

Associate Membership:

Become an associate member of the FSF. Members will receive a bootable USB card, email forwarding, and an account on the FSF's Jabber/XMPP server. To sign up or get more information, visit member.fsf.org or write to membership@fsf.org.

Online: Use your credit card or PayPal account to 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!

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