FREE SOFTWARE FOUNDATION

Bulletin

Issue 34

Spring 2019

Contents

FSF teaches free software to public school youth

By Devin Ulibarri

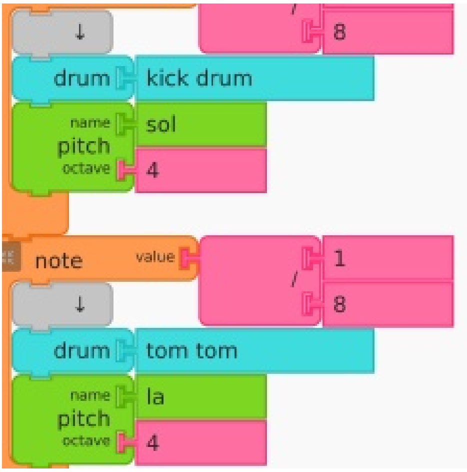
Free Software Activist &

Sugar Labs Oversight Board Member

Free software is a critical component to a free society, and nowhere is society's influence more apparent than in the public school system. Software that students use in schools greatly impacts the students' lives, and consequently impacts society as a whole. This spring, the FSF, with funding from the Staples Foundation, brought free software to two public middle schools and an after-school program in the Boston area.

We reached out to these schools to offer workshops that introduce children to the fundamentals of commands, scripts, and computation, followed by an introduction to the principles of software freedom. For the workshops, we chose Music Blocks (`musicblocks.sugarlabs.org`), a visual programming language for music, which is licensed under AGPLv3 or later by SugarLabs, a community-driven organization with the mission to construct free software tools for teaching and learning.

Music Blocks is a good tool for introducing children to musical and programming concepts for many



This is an example of code used to create music in Music Blocks, the program we used to teach public school children about free software reasons: first, music is fun, and it's fun to make sounds with a computer. Also, music is forgiving: mistakes present relatively little risk to a new learner, since the "mistakes" result in unexpected, often funny sounds.

Since Music Blocks is a programming language, it offers the opportunity for students to create their own scripts/programs beyond Music Blocks, which they can share with their friends. Remixing becomes a vehicle for exchanging both programming and musical ideas. As such, Music Blocks is a waypoint on the path to learning more about software and the four freedoms.

When we walked into the classroom, there was a palpable excitement in the room. We started each class with everyone in a circle, to do a musical activity together that would later be scripted in Music Blocks. Students learned a snippet of music by first listening and singing back the snippet. Then, we studied a graphical representation of the music, and even explored modifying the original musical snippet, to engage their creative thinking. The students then inputted the music into Music Blocks using laptops we prepared for them with all free software. The lessons were basic, but not simple (see handouts at `u.fsf.org/2uo`); they required a fair amount of problem-solving and collaboration to complete, which helped to give the students a frame of reference when we introduced the four freedoms of free software.

Devin, and FSF's Web developer Michael, working with kids in the classroom

After the lesson, we collected some feedback from the students in order to assess the project's efficacy. One student said, "I learned that music is in code," and another said they learned "how to music-code." A few students talked about software freedom specifically, saying they learned "the four freedoms" and "what freedom means."

For this project, we used computers that fully respected the students' freedom. FSF tech intern Valessio Brito put in many hours flashing 25 T400 laptops with Libreboot and installing Trisquel GNU/Linux. As part of the overarching project to bring free software to these schools, the FSF then donated ten of the fully freed T400s to the participating schools. The laptops are accompanied with handouts that explain the four freedoms, the importance of free software for education, an introduction to Trisquel, and bootable Trisquel USB sticks. This gives the schools access to freedom-respecting computers and software throughout the year.

This project is a small step to ensure that students, teachers, parents, and administrators understand the implications for free software in education. Education needs free software, but free software also needs education (see: `u.fsf.org/spb10`). The free software movement needs to find entry points into education so that youth can be introduced and empowered at an early stage. This project is the first in an anticipated series of projects by the FSF to bring free software philosophy and ethics to the educational system.

Free software: It's not just for privacy

By John Sullivan

Executive Director

Privacy violations by large technology companies continue to be big news in 2019. In April, we learned that Amazon employs thousands of people around the world to review and transcribe conversations overheard by its Alexa-enabled Echo devices (see: `u.fsf.org/spb1`). In May, we read that Snapchat employees used their access to spy repeatedly on users (see: `u.fsf.org/2ui`). And Facebook is facing a potential five billion dollar fine for its privacy failures (see: `u.fsf.org/2uj`).

The FSF and other free software activists have highlighted these tragic stories to illustrate the urgent importance of free software to the general public. The ever-expanding collection of mistakes and abuses committed by proprietary software and unethical network services should make a strong case for changing software culture from one that inherently mistreats and exploits users to one that respects us.

Richard Stallman says part of his inspiration for starting the free software movement was the frustration he encountered when he was refused the source code necessary to fix a bug in a printer. Over the years, this story has been persuasive to others as well. Similarly, stories about privacy leaks and other bad behavior by proprietary software and unethical network services can be signposts that lead new people to join the movement. But right now, those signs are pointing in the wrong direction.

The mainstream response to stories of major privacy violations has *not* been to demand user control over the software. Instead, the calls have been almost exclusively for policy measures like fines and antitrust actions. In a recent *New York Times* special section on Privacy and Technology, neither free software nor open source were even mentioned (`u.fsf.org/spb2`).

We've also seen the narrative around privacy actually boost proprietary companies. This year, Apple launched an advertising campaign focused entirely on claims that it protects user privacy (u.fsf.org/spb3). Other companies, especially Facebook, have attempted to flip the script by offering more user-configurable privacy settings.

Policy measures might be good, but we know that free software is a precondition for true privacy. Without the source code, claims like Apple's can never be fully verifiable. Without a switch to decentralized free software, Facebook's settings can just be "placebo buttons" – like phony office thermostats that let you think you're controlling the temperature. And without control over the software, users will never have privacy from the company that claims to own it.

Because free software is a prerequisite, this movement's work advocating and developing it is clearly helping to protect user privacy. The *New York Times* and regulators need to hear us on that. But we also must be careful about what we promise, and what we emphasize.

All software has bugs; even free software can fail at privacy. Like any other practical benefit of free software – better stability, speed, compatibility with older hardware – privacy claims are also vulnerable to exploitation by proprietary software pushers – as we see Apple doing now. The most important benefit of free software, and the only justification it needs, is freedom.

We should keep pointing to abuses committed by the Facebooks, Amazons, Snapchats, and Apples of the world. We should then point at free software, not as a suite of perfect software solutions, but as a set of cultural values where we can empower ourselves to solve these and other problems instead of being kept helpless.

The FSF has shared many of the inspiring stories we've heard from people energized to help their communities, refusing to be pushed around by unethical software. We published several in a video at u.fsf.org/1mb. We need to do more of that, in order to seize these opportunities to make free software a salient issue for everyone. I encourage you to share your story publicly, and let us know about it at campaigns@fsf.org!

Sparking change: What free software can learn from social justice movements

By MK Fain

Software Engineer and Activist

In his celebrated 1963 *Letter From Birmingham Jail*, Dr. Martin Luther King, Jr. contrasted "a negative peace which is the absence of tension to a positive peace which is the presence of justice." Negative peace, he claimed, was the greatest force against civil progress. King's description of negative and positive peace is something that still rings true across social movements today; from movements for racial equity, to ending sexual violence, to animal rights, or to economic justice. Those who take direct action are often chided, even by those who support their end goal, for being too radical, too confrontational, or too emotional – yet these actions are vital for social change.

To apply this concept to the free software movement, the general public remains largely unaware of the consequences of the software they use. Meanwhile, under the surface, it is violating their privacy and stomping on their freedom by giving corporations control over their computing and thus, by extension, what they read, think, and know. Despite the fact that proprietary software poses unprecedented risks to individual liberty, democracy, and societal freedoms, Google, Facebook, Microsoft, and Apple act with impunity. Software is deeply intertwined with all means of production, communication, and travel. We are quickly approaching a society in which every action is determined, predicted, or prevented by an algorithm. This lack of tension, while injustice remains the status quo, is what negative peace looks like.

Throughout history hundreds of methods of protest, resistance, direct action, and subversion have been used to topple oppressive regimes and systems. The Albert Einstein Institution recognized 198 Methods of Nonviolent Action (see: `u.fsf.org/spb4`). Although there have been notable exceptions by organizations such as the FSF and its partners, the bulk of the free software movement on the ground has primarily focused on one method: #72, non-consumption of boycotted goods.

In a movement that is so vital to the freedom of not just software, but to the toppling of all oppression, we can't afford to just use one tool in our box. Boycotting proprietary software is crucial, but it is not enough: while free software advocates have also engaged in development, written letters and called government representatives, and provided educational resources and legal tools, there's so much more we can do. As software freedom activists, we also need to focus on bringing visibility to this social crisis, so that it can no longer be ignored.

In past social movements, ending the state of negative peace through nonviolent direct action has been a necessary part of achieving change. Taking an example from the suffragettes in England, real change was only achieved by those activists with a disruptive and confrontational approach to the cause. For example, in 1906, Emmeline and Christabel Pankhurst started disrupting political events and got themselves arrested on purpose. They used their time in jail to garner more public attention and sympathy through hunger strikes. This breaking of the negative peace, and the creation of controversy, meant that the men in power could no longer ignore the issue.

The free software movement can learn from such successes. We can start by building grassroots networks of connected and empowered activists through conferences (such as LibrePlanet and SeaGL), as well as local organizations. Connect with activists in other movements, and learn from them. Plan protests and marches together, coordinate sit-ins and disruptions of events, organize walk-outs, and raise the tension (see: `u.fsf.org/spb5`). Attend these events when others plan them, even if you've never protested before. (See `u.fsf.org/spb6` for tips for your first protest!)

If you can't take direct action yourself, support the activists who do. Become comfortable with nonviolent confrontation in your daily life. Correct language normalizing proprietary software (see: `u.fsf.org/spb7`). Challenge its use even when you know you won't win. Refuse to remain silent.

The threat of proprietary software is too great for us to allow the state of negative peace to remain. Breaking it is our clear path forward, and this will be how we spark change.

What "Respects Your Freedom" is for, and what it isn't

By Donald Robertson, III

Licensing and Compliance Manager

Retailers who meet the FSF's criteria for protecting the rights of users – verified by a rigorous review process – can gain certification under our Respects Your Freedom (RYF) program for particular devices they sell. Once they gain certification, they are able to use the RYF certification mark in association with those certified devices.

You can think of the certification mark as being like a trademark, in that it is a name and logo that helps inform users about the products that bear it. But rather than referring to either everything a company sells, or a product no matter where it is sold, the mark refers to specific products as sold by a specific retailer. The RYF mark on a device means that the user knows that the device comes with freedom inside, and that the point of purchase is also freedom-respecting.

Sometimes people get confused as to what RYF means exactly. One point of confusion is around it being a "hardware" certification program. RYF only refers to devices that are sold by the particular retailer who gained certification for that device. You can find a list of certifications at `fsf.org/ryf`. On that list, you will find that there are several companies that offer the same hardware, and those devices will even come with basically the same software. But each retailer had to go through certification for their version of the device. That is because we don't certify hardware, we certify a retailer's process of delivering that device in a way that respects the user.

If a different company comes along and offers the same hardware, then that is not automatically an RYF-certified device, because that retailer has not received certification. Their Web site may require proprietary JavaScript in order to make a purchase or review support information. They could add other nonfree software to the device, or steer users toward nonfree software in connection with the product. In order to make sure that users get the freedom they deserve, we need to review how the particular retailer actually offers the device. So even though a different retailer might offer exactly the same hardware, it is not guaranteed to respect your rights as a user.

In the same vein, you could Libreboot your own device, and load it up with a fully free GNU/Linux distribution like Trisquel. But even though your machine would have all the freedom that you need, it still wouldn't be an RYF-certified device.

The retailers in the RYF certification program work very hard to find and root out freedom-related issues for their devices. And in order to gain certification, they each promise to continue that hard work, and to fix any problems that may arise. It is more than just "hardware that works with free software" (for that, you should check out `h-node.org`): it is dedication on the part of the retailer to protect their users. As the program continues to grow, and more retailers start working for users instead of against them, you can support that work by supporting those retailers who are up to the task of gaining RYF certification for the products they sell. And keep an eye out for RYF's new and improved Web site, coming soon!

The free software movement needs you: How to become a volunteer

By Andrew Engelbrecht

Senior Systems Administrator

When I first got into free software, I had a lot of fun learning how to administer my machine when I ran into small issues. One day I saw a posting by the FSF looking for help on a free social network, so I went in to volunteer. After a while, I was given the opportunity to learn about system administration, and after an internship, was later hired by the FSF as a Web developer, and then as a senior sysadmin.

While I have ended up in a technical position at the FSF, much of my volunteer work was nontechnical, and many organizations and free software projects benefit from volunteers who help in ways other than software programming. These groups need help with Web site and application design, bug reporting and triage, fundraising, writing, outreach through social media, answering questions online, and other creative work. By helping the free software movement in these ways, you have the opportunity to personally grow as you contribute.

One of the biggest volunteering opportunities offered by the FSF is the annual LibrePlanet conference. We don't have enough staff to run the entire conference ourselves, so we depend upon the help of volunteers. At our last conference, over 50 volunteers assisted with tasks like registration, GNU Press shop sales, room moderation, answering questions, and streaming video live over the Internet. Our streaming software was created by a previous FSF tech team volunteer intern, David Testé, who flies in every year from Paris to help us with the event.

Another job that requires a lot of hands is our twice-annual *Bulletin* mailing: we stuff over 10,000 envelopes, with the help of about 35 volunteers. It's a fun time that involves many passionate activists filling our conference room, working together, connecting, and talking about our shared interests.

The free software movement also relies upon activists to help spread the message of software freedom. We're indebted to the people who tell their friends and loved ones about the value of free software, and the technical issues that matter to us all. Many years ago, I was fortunate to meet a few people who inspired me to use GNU/Linux and to learn more about software freedom, and to them I am thankful.

If you are a programmer and want to contribute to free software while building your programming skills, ask a project maintainer for a list of entry-level software issues to start with. If you do your best to create quality patches, you will grow your coding skills as you contribute.

If you're new to coding, or would like to contribute your non-coding skills, there are many free software projects that could use your help. If you care a lot about a program that you use, find its project Web site for more information. Nontechnical contributions are also valued immensely, and volunteering presents an opportunity to develop your skills further, whether you design Web sites, update documentation, or spread the word.

I encourage you to volunteer your time and skills to whatever project or organization that may need your help. You have the opportunity to meet great people and to grow in the process. It also can be a lot of fun. The FSF, and other free software organizations, benefit from your contributions, and we hope to work with more wonderful people like you! For a list of ways to volunteer, see `fsf.org/volunteer`.

On the road with RMS

By Jeanne Rasata

Assistant to the President

FSF founder and president Richard Stallman (RMS) continued traveling the world this past semester, to raise awareness of free software. He visited twenty-two cities, in eight countries, across three continents, taking part in two panels and giving a total of twenty-eight speeches, ten of them part of larger events.

Starting out in Europe, in November, he spoke twice (in Burgos and in Madrid, both in Spain) at the III Foro de la Cultura, a forum for examining "the ethical and moral issues around technology" and the need for human beings to "defend [their] values in the face of the increasing power of the machine." Then, in Amsterdam, he took part in the year's edition of the Stichting Nationaal Informatica Congres (SNiC), SingularIT, at which the speakers shared "how their field [would] be affected by technology that [was] smarter than humanity." RMS delivered a speech on whether we should "have more surveillance than the USSR." Back in Spain, he gave a couple of freestanding speeches, in Burjassot and in Valladolid, and spoke at LibreCon and at ConsulCon, in Bilbao and Madrid, respectively.

In January, RMS went to India, where he gave speeches in New Delhi, Kozhikode, Thiruvanamthapuram, and Bengaluru. He also took part in the Kerala Literature Festival (KLF), in Kozhikode, and in the Kochi-Muziris Biennale, an art exhibition and festival that is part of the KLF. Finally, in Mandya, he took part in the inauguration of this year's Free Software Movement Karnataka Camps (FSMK Camps), a five-day training program for students to learn how to adopt free software and help their schools transition to using it.

RMS at the LibrePlanet 2019 conference in Boston

Back in North America, in February, he spoke at York University, in Toronto, Canada; in the United States, at Montana State University, in Bozeman; and, back in Boston, at LibrePlanet 2019, he both spoke and presented the Free Software Awards, as is custom.

In March and April, RMS was back in Europe for a series of talks in France and Switzerland. He started out in Rennes, at inOut, an event that "explores new mobility solutions," and at which he had spoken last year as well. He took part in a panel focused on anonymous transportation. He then gave freestanding speeches at the University of Rennes and in Anthy-sur-Léman, before giving several speeches in Switzerland, in Geneva, Grandvaux, Bern, Gland, and Lausanne.

He finished up stateside, in April, with an event at Cold Spring Harbor Laboratory, on Long Island, in New York.

See `fsf.org/events` for a list of RMS's confirmed engagements. If you would like to extend him a speaking invitation, write him at `rms-assist@gnu.org`. Please 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`.

News from the tech team

By Andrew Engelbrecht, Ian Kelling, Michael McMahon, and Ruben Rodriguez

In addition to our role in the educational project you read about on page one, for the last six months, the FSF tech team continued our daily work of maintaining and improving infrastructure for the FSF and GNU, as well as supporting the mission of growing the free software movement. We welcomed Web developer Michael McMahon as a new addition to the team.

At LibrePlanet 2019, in April, we took charge of streaming all the talks live, and publishing the videos in our GNU MediaGoblin instance at `media.libreplanet.org`. All systems were automatically deployed and managed using Ansible provisioning software, including the streaming and recording tool HUBAngl, written by former FSF intern David Testé. Our spring intern, Valessio Brito, created an information screen system showing schedule information and announcements in the lobby and in talk rooms. All of the tools used to run LibrePlanet are 100% free software, from registration to video streaming.

The tech team herds over a hundred virtual machines running on nine servers across four locations. At the main data center, a new donor, Hurricane Electric, agreed to provide our Internet service. Migrating to the new IP range took many hours of carefully changing and testing configurations, which was accomplished without causing any significant downtime.

We continued to migrate virtual machines into our new cluster, also upgrading them to newer software based on Trisquel

8. Senior systems administrator Ian Kelling is upgrading and fixing our email server infrastructure, which will make it follow best practices and be more reliable. As we write this, he has just finished upgrading GNU Mailman on our internal instance, and is tackling our main instance at `lists.gnu.org`. We have also been upgrading other systems, including our CiviCRM instance, a freedom-respecting relationship manager.

Last year, we deployed a Ceph redundant storage system to host our critical infrastructure (see: `u.fsf.org/spb8`). One server eventually developed hard-to-diagnose hardware issues that required intervention, resulting in a lengthy research and testing process, ending with replacing the disk controllers on all the servers. The incident was resolved with little downtime, and a performance upgrade.

In the last year, we have also given a big push to the GNU LibreJS project (see: `u.fsf.org/librejs`). This is an extension for Mozilla-based browsers that makes sure that you don't run unintended proprietary JavaScript. It also detects free software licenses, and it allows you to blacklist or whitelist individual scripts or full domains. LibreJS comes preinstalled with the fully-free, privacy-respecting browser GNU IceCat. You can read more about it in our previous *Bulletin* (see: `u.fsf.org/spb9`).

In 2017, GNU volunteer Nate Nichols started a reimplementation of the LibreJS code to make it compatible with the WebExtensions API, which was continued with research by FSF intern Ethan Dorta, and code by FSF chief technology officer Ruben Rodriguez. Since then, it has been extensively improved by Giorgio Maone, the author of the NoScript extension, as an FSF contractor funded by Handshake. His work made LibreJS fast, robust, and compatible with mobile Mozilla-derived browsers. He also added automated testing for developers, revamped the user interface, implemented whitelist and blacklist management, and fixed many quirks and corner cases.

The next step for LibreJS will be to extend and perfect the methods it implements for tagging and recognizing licenses on scripts and Web pages. With that, we will be launching new, attractive materials to provide technical reference for Web masters on how to properly state the licenses used on their sites, and information on how to integrate licensing data with other common practices for JavaScript deployment, documentation, and maintenance.

All our work would not be possible without the continuous support of our members and donors, and the work of thousands of free software contributors. You can help by joining the GNU developer community, or by applying for an internship at the Free Software Foundation (see: `u.fsf.org/gj`).

Donate to the FSF with Bitcoin: 13appuhi1CK54BCx73RbBSjesFMYVEUejt

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 Scribus and GIMP.

Photo credits:
Page 1: Screenshot of Music Blocks Copyright ©2019 Devin Ulibarri. This image is licensed under a Creative Commons Attribution ShareAlike 4.0 International license.
Page 2: Photo Copyright ©2019 Free Software Foundation, Inc., by Valessio Brito. This image is licensed under a Creative Commons Attribution ShareAlike 4.0 International license.
Page 9: Photo Copyright ©2019 Free Software Foundation, Inc., by Madi Muhlberg. This image is 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 USB card, email forwarding, and an account on the FSF's Jabber/XMPP server. Plus: participate in 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`*.*

Donate: *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 for the yearly LibrePlanet conference next spring.*

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