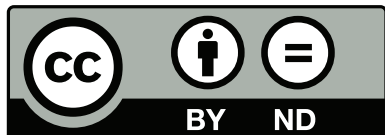


Copyright © 2010
Free Software Foundation
51 Franklin Street, 5th Floor
Boston, MA 02110-1301
(617)542-5942
info@fsf.org
http://www.fsf.org/

If you recently joined as a member, your mailing includes a USB flash drive membership card preloaded with a full live distribution of Trisquel 4.0. You can download source code for the distribution at <http://fsf.org/associate/source/>. If you would like us to send you the source code on two DVDs for 5 dollars, please write to info@fsf.org or at the address above.

This bulletin was produced using only free software.

The articles in this bulletin are individually licensed under the **Creative Commons Attribution No Derivative Works 3.0 United States License**. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nd/3.0/us/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.



How to Contribute

Associate Membership:

Become an associate member of the FSF. Members will receive a bootable USB card and e-mail forwarding. 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. See fsf.org/jobs for details.

United Way: As a 501(c)(3) tax-exempt organization, the FSF is eligible to receive United Way funds. See fsf.org/uw for more information.

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

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

Free Software Supporter: Make sure you're getting all the very latest news from the FSF at fsf.org/fss.

LibrePlanet: Find local groups in your area or start your own at libreplanet.org/

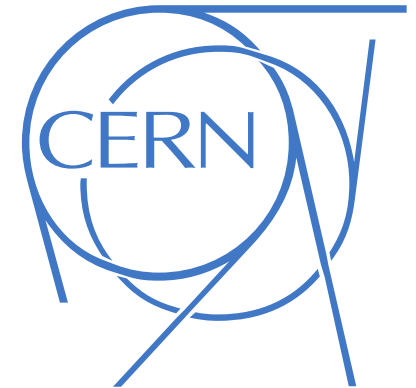


Bulletin

Issue 17
December 2010

Contents

FSF at 25	1
When Free Software Sucks	2
GNU Hackers Meeting: Gothenburg, Sweden	5
Hardware Endorsement Program	7
The MusicBrainz Project	8
Solving The Lending Problem	9
The Appleseed Project	11
GNU Contributors Around The World	12
Free Software, Free Society: Second Edition	13
GNU Parallel: A Design For Life	14
A View From The Server Room	14
The Free Game Lag	15
Free Software From An Artistic Perspective	17
The Saga of Sun RPC	18



Free software at CERN

our new free software adoption campaign, **Why aren't we using free software for that?**

We've learned that the key to helping institutions adopt free software is to educate and empower decision makers to question why free software isn't being considered or used. Our campaign will provide the inspiration to ask these critical questions and the materials to help make the case for free software.

"When you look at who's using free software and why, it begs the question, why aren't we using free software for that?"

To help demonstrate that free software is used even in the most mission-critical situations, we've published a new resource featuring executives from

FSF at 25

by Peter Brown
Executive Director

The FSF has turned 25! To help us celebrate you can join us in kick-starting the new year with support for

high-profile institutions talking about how and why they deploy free software. Some examples: Mark Schulz, the leader of the Grid Deployment group at the European Organization for Nuclear Research (CERN), tells us how the scientific community at CERN uses free software extensively and how the Large Hadron Collider depends on GNU/Linux; Daniel Risacher, of the US Department of Defense, shares how Free Software is critical to the armed forces' ability to adapt to new challenges; Jimmy Wales, asserting that "free knowledge requires free software," justifies his intransigence on Wikipedia's use of free (and only free) software; and Steve Rubinow, the CIO of the New York Stock Exchange, explains how free software's effectiveness in handling the demands of the industry and the fact that it is so well supported make it a logical choice for the entire electronic exchange-and-trading industry.

These profiles provide a fascinating picture of the success of free software and reinforce that it is the freedom of free software that inspires its use.¹

"Free Software is software you can study, modify, and share without restriction. Around the world, schools, governments, businesses, and leading technology and research institutions are adopting it. What's our plan for moving to free software?"

Free software is the better ethical choice, and publicizing success stories helps decision makers make a case for their ethical push for free software adoption. At www.fsf.org we will be highlighting an ongoing series of interviews with executives from a diverse range

¹Read them all at <http://www.fsf.org/working-together/whos-using-free-software>

of organizations that have chosen to adopt free software.

"The free software community is a worldwide movement of people dedicated to the goal of freedom in the use of technology. Anyone can be part of this community and we can share in its benefits."

In a world where media companies educate children not to share, the ideas presented by the free software community can seem foreign and threatening. What are these people's motivations for collaborating to produce and share valuable software?

Publicizing the motivations people have for participating in the free software community can go a long way to reinforce the trust people should put in its use. That's why we've also published a new resource: **Meet the Free Software Community**, a series of profiles to inform and inspire. On the list:²

- Matt Mullenweg, cofounder and lead developer of WordPress, highlights the benefit of freedom in helping build a business: *"As a businessman I love building on free software because you can be certain of your rights and freedoms with regards to the software."*
- Chris Blizzard, Firefox product and platform manager at Mozilla, talks about the expectations he has when working with free software: *"My first operating system relationship was built on free software. I grew up with radical transparency in my com-*

²You can see all the profiles at <http://www.fsf.org/working-together/profiles/meet-the-free-software-community>

The funny thing about this license is everyone knew it was Sun's intention to make it free software. The code is so old, it dates back to a time when the drafting of free software licenses wasn't well understood (old-schoolers will, for example, remember the annoying advertising clause in early BSD licenses). Thus, by our modern standards, the Sun RPC license does appear on its face as trivially non-Free, but in its historical context, the intent was actually clear, in my opinion.

Nevertheless, by 2002, we knew how to look at licenses objectively and critically, and it was clear to many people that the license had problems. Competing legal theories existed, but the concerns of Debian were enough to get everyone moving toward a solution.

For my part, I checked in regularly from 2002 to 2004 with Danese Cooper (who was, effectively, Simon Phipps' predecessor at Sun) until I was practically begging her to pay attention to the issue. While I could frequently get verbal assurances from Danese and other Sun officials that it was their clear intention that glibc be permitted to include the code under the LGPL, I could never get something in writing. I had a hundred other things to worry about and, eventually, I stopped worrying about it. I remember thinking at the time, "Well, I've notes on all these calls and discussions I've had with Sun people about the license. Worst case scenario: I'll have to testify to this when Sun sues some free software project, and there will be a good estoppel defense".

Meanwhile, around early 2004, my friend and colleague at the FSF, David "Novalis" Turner took up the cause in earnest. I think he spent a year or two as I did: desperately trying to get oth-

ers to pay attention and solve the problem. Eventually, he left the FSF for other work, and others, including Brett Smith (who took over Novalis' FSF job), took up the cause—by the time Brett came on, Spot was also paying attention to this. Both Brett and Spot worked hard to get Simon Phipps' attention on it, which finally happened. But around then began that long waiting period while Oracle was preparing to buy Sun. It stopped almost anything anyone wanted to get done with Sun, so everyone just waited (again). It was around that time that I decided I was pretty sure I never wanted to hear the phrase "Sun RPC license" again in my life.

Meanwhile, Richard Fontana had gone to work for Red Hat, and his self-proclaimed pathological obsession with free software (which can only be rivaled by my own) led him to begin discussing the Sun RPC issue again. He and Spot were also doing their best negotiating with Oracle to get it fixed. They took us the last miles of this marathon, and now the job is done.

Thanks to everyone in this great cast of characters who made this ultimately beneficial production of licensing theater possible. I am honored to have shared the stage in the first few acts, and am sorry that I hid backstage for the last few. It was right to keep working on it until the job was done. As Fontana said, "Estoppel may be relevant but never enough; software freedom principle[s] should matter as much as legal risk. [the] standard for FaiF can't simply be good defense to copyright infringement likely." Thanks to everyone; I'm so glad I no longer have to wait in fear of a subpoena from Oracle in a lawsuit claiming infringement of their Sun RPC copyrights. ♡

to use for inspiration (or to be amused by). There's another group consisting of an artist, critic and collector who make up a simulation of precisely how the art world doesn't work called The Cybernetic Artworld. They don't pretend to be anything other than bots, so it's fun when people encounter them and still attribute personality to their output or suggest ways they could be improved.

I've also written bots to generate and post random recipes or random design project ideas for commercial projects. Having them as a stream in a social network makes people consider them differently. It can be a good way of introducing ambient information or entertainment into your social network feed. Because I like Lisp, the bots are written in Common Lisp, but people have said that the code is very readable even if you don't know Lisp so it's still worth looking at. And everyone should learn Lisp. I've never written a spectator bot for The Cybernetic Artworld, so maybe people could write their own using the microblog-bot library.

ML: To any artists who are still using proprietary tools, what advice would you give in their switch to free software?

RM: There are tutorials and books that can help you learn free software replacements for proprietary tools, but the best thing to do is to get involved in the community. If you can find someone who knows the software you want to use, then their advice can be invaluable. Free software tools can seem very different to proprietary ones, particularly if you've been using proprietary software for years or even for decades. But the differences are usually just in how the interface is organized, and the major difference is a

positive one: you have the freedom to study and extend the software to better be able to achieve what you want.

The one thing that you will need to be patient about is CMYK support. Some tools support CMYK, most don't, but you can create CMYK print-ready art in GNU/Linux. For me the common thread to all this is the fact that despite the cliched images of the solitary hacker in their cubicle or the solitary artist in their studio, free software and art are both social activities. I'd love to see them come together more. Artists need the freedom to pursue their ideas, and they'll use free software in ways that will lead to interesting new possibilities. ♡

The Saga of Sun RPC

by Bradley M. Kuhn
Director, Free Software Foundation

I first became aware of the Sun RPC license in mid-2001, but my email archives indicate the issue had been one of consideration since 1994. Recently, it was finally resolved. It now ranks as the longest-standing free software licensing problem of my career. A cast of dozens deserve credit for getting it resolved:

Tom "spot" Callaway does a good job summarizing the recent occurrences on this issue (and by recent, I mean since 2005—it's been going long enough that five years ago is "recent") and its final resolution, and Simon Phipps, who worked on this issue during his time as the Chief Open Source Officer of Sun, also wrote about his work on the issue. For my part, I'll try to cover the middle part of the story from 2001 to 2005.

puting life—that's hard to grow away from."

- Marina Zhurakhinskaya, Senior Software Engineer working on GNOME desktop at Red Hat, speaks about how our community is able to adapt to deliver freedom for all users: "*Free software is an important social movement and a welcoming community. Working on Free Software is varied and rewarding and is impacting usability, accessibility, and access to computers in poor communities.*"

Improving access to advocacy materials

Our campaign also aims to make sure that free software advocacy gets a higher priority in the community.

It's an unfortunate fact that the most popular distributions of GNU/Linux provide little in the way of free software advocacy materials for new users. I've spoken with the community representatives of Fedora, Ubuntu, and OpenSUSE, and they all have expressed a desire to see FSF-developed advocacy materials find a home in their distributions. But it isn't just the major distributions that have a duty to educate their users—all free software projects should consider distributing advocacy materials as part and parcel of their work, and we aim to help them make that a reality.

When the benefits of sharing and cooperation are understood and valued by society, free software is the natural choice. ♡

When Free Software Sucks

by Benjamin Mako Hill
Director, Free Software Foundation

The Open Source Initiative's mission statement reads, "Open source is a development method for software that harnesses the power of distributed peer review and transparency of process. The promise of open source is better quality, higher reliability, more flexibility, lower cost, and an end to predatory vendor lock-in."

For more than a decade now, the Free Software Foundation has argued against this "open source" characterization of the free software movement. Free software advocates have primarily argued against this framing because "open source" is an explicit effort to deemphasize our core message of freedom and obscure our movement's role in the success of the software we have built. We have argued that "open source" is bad, fundamentally, because it attempts to keep people from talking about software freedom. But there is another reason we should be wary of the open source framing. The fundamental open source argument, as quoted in the mission statement above, is often incorrect.

Although the Open Source Initiative suggests "the promise of open source is better quality, higher reliability, more flexibility," this promise is not always realized. Although we do not often advertise the fact, any user of an early-stage free software project can explain that free software is not always as convenient, in purely practical terms, as its proprietary competitors. Free software is sometimes low quality. It is sometimes unreliable. It is sometimes inflexible. If people take

the arguments in favor of open source seriously, they must explain why open source has not lived up to its “promise” and conclude that proprietary tools would be a better choice. There is no reason we should have to do either.

Richard Stallman speaks to this in his article on “Why Open Source Misses the Point³” when he explains, “The idea of open source is that allowing users to change and redistribute the software will make it more powerful and reliable. But this is not guaranteed. Developers of proprietary software are not necessarily incompetent. Sometimes they produce a program that is powerful and reliable, even though it does not respect the users’ freedom.”

For open source, poor-quality software is a problem to be explained away or a reason to eschew the software altogether. For free software, it is a problem to be worked through. For free software advocates, glitches and missing features are never a source of shame. Any piece of free software that respects users’ freedom has a strong inherent advantage over a proprietary competitor that does not. Even if it has other issues, free software always has freedom.

Of course, every piece of free software must start somewhere. A brand-new piece of software, for example, is unlikely to be more featureful than an established proprietary tool. Projects begin with many bugs and improve over time. While open source advocates might argue that a project will grow into usefulness over time and with luck, free software projects represent important contributions on day one to

a free software advocate. Every piece of software that gives users control over their technology is a step forward. Improved quality as a project matures is the icing on the cake.

A second, perhaps even more damning, fact is that the collaborative, distributed, peer-review development process at the heart of the definition of open source bears little resemblance to the practice of software development in the vast majority of projects under free (or “open source”) licenses.

Several academic studies of free software hosting sites SourceForge⁴ and Savannah⁵ have shown what many free software developers who have put a codebase online already know firsthand. The vast majority of free software projects are not particularly collaborative. The median number of contributors to a free software project on SourceForge? One. A lone developer. SourceForge projects at the ninety-fifth percentile by participant size have only five contributors. More than half of these free software projects—and even most projects that have made several successful releases and been downloaded frequently, are the work of a single developer with little outside help.

By emphasizing the power of collaborative development and “distributed peer review,” open source approaches seem to have very little to say about why one should use, or contribute to, the vast majority of free software projects. Because the purported benefits of collaboration cannot be realized when there is no collaboration, the vast majority of free development projects are at no technical ad-

releasing as free software can only help them engage a wider community and gain contributions from anyone interested in improving their software. Still, many games will not require a subscription, and there are plenty of ways for those to fund their development as well.

The possibilities don’t end there, and hopefully with these examples it becomes clearer how free gaming can advance with enough interest. Free gaming will never look like the world of proprietary games today. They won’t use DRM to prevent you from sharing them, and they won’t limit your freedom otherwise. We can look forward to games which are not crippled by anti-features and are able to build upon each other to develop faster than they would have otherwise. In the meantime, we should keep supporting free games and have confidence in them. We should in fact take it as a great sign when critical questions that were once raised against free software as whole are now just pinned on one subset of software. Now, next time anyone asks about free software gaming, we should have a good answer for them. ♡

Free Software From An Artistic Perspective

by Matt Lee
Campaigns Manager

On my travels north, I meet with Rob Myers, former GNU chief webmaster and one of the developers on GNU social. Rob is an artist who uses free software exclusively for his work. Previously, he had an exhibition of art created using free software in Zagreb, Croatia. Today, we meet in the

less exotic location of Peterborough.

ML: GNU social is one of the latest round of free software social networks. What opportunities do you see for these networks in the artistic community?

RM: When I spoke about GNU FM at the Make Art conference in Poitiers, France, last year, I mentioned GNU social briefly at the end of my presentation, and everyone really wanted to hear more about it. Artists thrive on the ideas, critique, and publicity that Internet technologies can give access to, but are often concerned about the control that proprietary social networks give hidden actors over them. Free software social networking systems return the power to create your own virtual social space to the artistic individuals and institutions that participate in them. It’s reminiscent of the email listserv era, which was very important for several different art movements in the 1990s and early 2000s. MySpace and Facebook really haven’t replaced that — they lack the focus and sense of community.

Artists can take control of their online social presence much more effectively with free software — galleries, museums, artists groups, individual artists, can all have their own nodes in the distributed social network. Some of these will almost certainly be declared artworks in themselves by an enterprising conceptual artist.

ML: You previously made an art project out of bots that communicate with each other on networks like StatusNet and Identi.ca. What work is happening with these, and how can other people use them for their own purposes, artistic or otherwise?

RM: They’ve been running for over a year now. Some post short, random, colour or shape descriptions for people

³<http://www.gnu.org/philosophy/open-source-misses-the-point.html>

⁴<http://sf.net>

⁵<http://sv.gnu.org>

The Free Game Lag

by Sarvodaya

Campaigns Intern, Summer 2010

There is one category of software that many see as being unsustainable as free software: Free video games have lagged behind other areas of free software, and the reasons behind this are fairly simple. Still, even many free software proponents may fail to provide an answer to those who are skeptical about the viability of free gaming. While it is true that software should be ethical, video games need not suffer for it. The business models for production simply need to change, and just like they have for other software, they will for gaming as well. When people ask you how gaming as we know it can exist in a free software world, you should open with your response with, “It can’t, but it can be better.”

There is a natural tendency for free software to take on more essential aspects of computing first. While subjective, it is clear that gaming is not a top-priority and, as such, has not advanced as rapidly as say, web browsers or word processors. That isn’t to say that no progress has been made. Indeed, free gaming has certainly been catching up, but it will take a while to surpass the quality of proprietary games. This should not be surprising or alarming. We will get there in good time. As free software continues to spread, interest will build for free games, and a lack of understanding how such games could support themselves should not need to be a reason to make video games an exception to free software.

The state of non-free gaming has gotten so bad, that an effort called The Humble Indie Bundle launched

to sell games that did not force you to use a particular platform, and did not use DRM. Through a simple pay-what-you-want model, contributors put down a total of over \$1.2M because so many people are desperate to escape the norms that have evolved out of the proprietary software world. Even though it wasn’t promised, after being so wildly successful, most of the games in the bundle were released as free software. There is clear interest in what free software gaming offers, and gradually there are more and more efforts to produce these free games.

It’s always funny to face the same arguments that have been presented to the free software movement and completely disproven in practice (e.g. Why would anybody produce free software?). The possible incentives for creating free games are as numerous as the motivations for producing other free software. Perhaps a graphics hardware company wants to fund the development of a game to show of the capabilities of their hardware. Perhaps a hospital wants to fund an enjoyable way for surgeons to improve their dexterity. Perhaps a school wants to fund a suite of educational games for students. Perhaps a competitive gaming league wants to fund their own game for tournaments. There are already a few notable examples of free games that are proving business models can be built around free games.

Through a partnership with the Free Software Foundation, Winch Gate Properties Ltd released Ryzom, the massively multiplayer online role-playing game, as free software under the AGPL, and its artwork as free cultural works under the CC-BY-SA license. As an online game, they fund development through subscriptions, so

vantage with respect to a proprietary competitor.

For free software advocates, these same projects are each seen as important successes. Because every piece of free software respects its users’ freedom, advocates of software freedom argue that each piece of free software begins with an inherent ethical advantage over proprietary competitors — even a more featureful one. By emphasizing freedom over practical advantages, free software’s advocacy is rooted in a technical reality in a way that open source is often not. When free software is better, we can celebrate this fact. When it is not, we need not treat it as a damning critique of free software advocacy or even as a compelling argument against the use of the software in question.

Open source advocates must defend their thesis that freely developed software should, or will with time, be better than proprietary software. Free software supporters can instead ask, “How can we make free software better?” In a free software framing, high quality software exists as a means to an end rather than an end itself. Free software developers should strive to create functional, flexible software that serves its users well. But doing so is not the only way to make steps toward solving what is both an easier and a much more profoundly important goal: respecting and protecting their freedom.

Of course, we do not need to reject arguments that collaboration can play an important role in creating high-quality software. In many of the most successful free software projects, it clearly has done exactly that. The benefits of collaboration become something to understand, support, and work towards, rather than something

to take for granted in the face of evidence that refuses to conform to ideology. ☹

GNU Hackers Meeting: Gothenburg, Sweden

by Matt Lee

Campaigns Manager

In Gothenburg, Sweden, I attended an informal bar meetup with several GNU hackers. Among them were Brian Gough, who sits on the GNU Advisory Committee, Jos Marchesi, a GNU PDF developer, Michael Foetsch, from gNewSense, Ralf Wildenhues, who hacks on the GNU Autotools (autoconf, automake, libtool), Simon Josefsson from GNU TLS, and Alfred Szmidt, a veteran of many projects, including GCC, GDB, Hurd and, more recently, the GNU networking utilities, inetutils.

ML: Do you think awareness of GNU is increasing?

BG: That’s hard to measure but the community is certainly getting stronger. We had about 40 people at the most recent GNU Hackers Meeting in the Hague, and that number has been increasing every meeting over the past years. At FOSDEM in February 2011 we will have a dev room for 100 people.

GS: In these last years I have noticed a wider awareness of what the Free Software and the GNU Project are, and the understanding of the importance to have a Free operating system. On the other hand, I have got the impression that the number of real contributors hasn’t increased at the same rate. Communities are a great thing but it is also important to get hands

“dirty,” there are new problems to face every day and there is definitely need of new contributors. Everyone can contribute and support actively the development of the GNU system, there are many ways to do it without be a programmer.

AS: Over the last 5 years I’ve seen a younger generation getting involved, including things like the GNU Hackers Meetings that are popping up all over the world. There also seems to be a wider awareness of what the GNU Project has done; people don’t seem as suprised if you say you work on the GNU system of the GNU Project. So yes, I would think so.

ML: Of all the projects represented here, gNewSense and GNU PDF are perhaps the most likely to be used by a typical non-developer user. What are the goals of these projects?

JM: The goal of the GNU PDF project is to provide a free (GPLv3+) and complete implementation of the PDF format and associated technologies. It is not yet ready for end-user usage, but we are working on it.

GS: These projects share the same final goal: give users the full control of their computers and data. GNU IceCat is the GNU version of the famous Mozilla Firefox browser; differently from Firefox, IceCat suggests to use only free addons and free plugins. GNU PDF project aims to develop a free library to manage the PDF file format. gNewSense is a fully free GNU/Linux distribution without any nonfree component. I would like to remind another very important GNU project, GNU Gnash, that provides a free Flash movie player. All of these projects (as any component of a GNU/Linux system) are very impor-

tant. Unfortunately still there is much work to do. Some features are missing but GNU/Linux and gNewSense are a reality — you can already get a taste of a Free operating system.

ML: What makes the GNU networking utilities different from those found in other operating systems and distributions?

SJ: Most distributions are using a variety of tools from a variety of sources: NetKit is widespread for telnet, ftp, and tftp. Debian GNU/Linux uses Marco d’Itri’s whois client while other programs, such as traceroute are separate packages too. Several of these packages are poorly maintained with no releases in many years — the last date in the changelog files for the ‘telnet’ and ‘ftp’ packages is 2000! Given this mixed picture, having a single source for network utilities, one that even makes regular releases and show some code activity, would be much better. At least that is why I’m helping the inetutils effort.

AS: The GNU Network Utilities are portable across different platforms, something that the BSD versions aren’t. We also support IPv6, Kerberos authentication, and TLS encryption, something that most other GNU and BSD systems lack in their standard version.

ML: Brian, what is the GNU Advisory Committee, and how is it changing the way GNU acts as a community?

BG: The GNU Advisory Committee was created about a year ago to improve coordination within the GNU Project. It provides an initial point of contact for questions from maintainers, FSF, and others. Members of the committee are appointed by Richard Stallman and meet by phone each month

ter done with GNU Parallel. ♡

A View From The Server Room

*by Bernie Innocenti
Senior Systems Administrator*

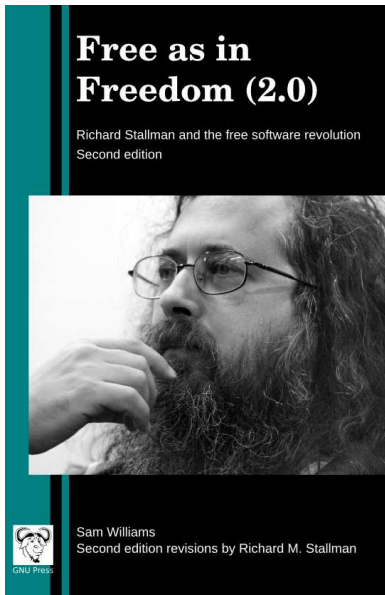
I’m the latest member to join the systems team at the Free Software Foundation. Prior to joining the FSF, I lived in Italy where I designed firmware and embedded systems for several high-tech companies in the Florence area. In the beginning my job there unfortunately required working on Windows with proprietary compilers to produce proprietary software. The situation improved significantly, though, over the years. My consulting firm was an early adopter of GCC, GNU/Linux, Python and other emerging technologies. It expanded rapidly as the local industry started to embrace free software at all levels: server infrastructure, process control, development workstations and, of course, embedded in devices. Demand for our expertise dramatically rose; we were being called to design new lines of products based on GNU/Linux and then to work side by side with the internal engineering team to transfer full ownership of the project. For fear of becoming redundant, traditional consultants were reluctant to share source code and best practices with their customers; as a result, for fear of losing control, customers were reluctant to outsource core projects. When free software entered the scene, this closed business model was easily swept aside. We were still able to maintain a technical edge, but not by means of keeping our tricks secret. Free licenses al-

lowed us to share code and best practices across the entire industry.

Witnessing the gradual liberation of the embedded scene was a tremendous experience for me—not only did it make my job easier and more productive, it also enabled me to share improvements with my peers around the world. These days, when many of us have the luxury of carrying powerful gadgets loaded with free software in their pockets, I wonder if some of that magic feeling is not lost.

In 2007, I became interested in the One Laptop Per Child project and moved to Boston to join the OLPC’s engineering team. OLPC’s XO-1 laptop was unique in many ways: along with the Asus EEE PC, it was one of the first mass-produced laptops preloaded with GNU/Linux and the first to ship with a free BIOS. While the laptop was plagued with a proprietary wireless chipset, it came preloaded with a revolutionary educational environment called Sugar, which embodied the same spirit of exploration and creativity that I had grown up with, in a format suitable for young hackers. These past three years, I travelled around the world for Sugar and OLPC.

I’ve been in Boston since October, working for the FSF. Its office in downtown Boston is one of the very first places I visited when I first arrived in the US, four years ago. It is a common place of pilgrimage for any respectable GNUdist. There’s a certain sense of wonder for me in being in the place where the free software movement started. And, finally, I’m working in a place where people care about freedom as much as I do. ♡



GNU Press is online at shop.fsf.org

Parallel — a shell tool for executing jobs in parallel, locally or using remote machines.

ML: Ole, GNU Parallel is a specialist tool for running multiple jobs at the command line at the same time. Why did you develop it, and what are your uses for it?

OT: I often get in the situation that I need to run a script on each line of a bunch of lines, so back in 2001 I made a wrapper script for `make -j` to run command lines in parallel. This was the first basic version of parallel that later became GNU Parallel. The full history of GNU Parallel is at <http://www.gnu.org/s/parallel/history.html>

Today I use GNU Parallel even for tasks that do not really need to be run in parallel, simply because of its ease of replacing arguments on the com-

mand line. Like emptying all tables in a database:

```
sql -n mysql:/// 'show tables'
| parallel sql mysql:/// DELETE
FROM {};
```

To me it has become a bit of a sport to see if the tasks I do can be done more efficiently using GNU Parallel. When you have gotten used to it, a lot of the once-off scripts can often be written on a single line using GNU Parallel — and they are often even easier to read. As an example, if you wanted to convert all *.mp3 to *.ogg running one process per CPU core on local computer and server2 you could simply do:

```
parallel --trc {}.ogg -j+0 -S
server2,,: 'mpg321 -w - {} | oggenc
-q0 - -o {}.ogg' ::: *.mp3
```

I encourage my users to share their smartest command lines on the email list parallel@gnu.org, so new uses can be found.

ML: Like many people, I'm sure, I often forget I'm using GNU Parallel. Has this ubiquity hurt your development efforts at all?

OT: A good tool is a tool that does not get in your way, but tries to support your work by providing reasonably defaults while remaining configurable for you own needs. GNU Parallel strives to accomplish this. This often also means that you do not really think about GNU Parallel as the tool is simply a step to accomplish your task.

The role that GNU Parallel plays will never be more than a supporting role and thus the best GNU Parallel can hope for is to be an integral part of every UNIX user's toolbox, so I would love to see people mentioning GNU Parallel when someone uses `xargs` or while-read loops for tasks that was bet-

to discuss current issues. Stallman, the founder of GNU and the president of the FSF, remains the Chief GNUisance with overall responsibility for and authority over the GNU Project. The Advisory Committee is trying to encourage more project-wide activities and get more people involved in GNU. For example, it helped to organise the first US GNU Hackers meeting in Boston at the FSF's LibrePlanet in March 2010. Prior to that there had not been a meeting where GNU contributors in the US could get together for many years. In the past, developers have mainly worked within their own projects — we want to encourage more communication and sharing of ideas among the project as a whole. ☺

Hardware Endorsement Program

by *Brett Smith*
Licensing Engineer

Buying hardware that plays well with free software has long been tricky business, even for us here at the FSF — for a while now, we've done our best to collect information about devices that require only free software drivers in our Hardware Directory. Unfortunately, these recommendations are often hard to follow and go out of date quickly.

At the same time, whenever we've talked to hardware manufacturers or sellers about this issue, they've often told us that they're interested in offering more hardware that meets our needs. The main reason they hold back is because they're not sure they can justify the expense involved.

In October, we publicly announced a hardware-endorsement program to help bridge this gap. Hardware that's endorsed by the FSF will do as much as possible to respect your freedom and give you control over the device. And the program demonstrates to sellers that there are already plenty of potential customers who take freedom seriously.

The program is designed to cover all kinds of hardware, such as full computers and parts, portable and embedded devices, and individual peripherals. Highlights of the criteria include:

- The device cannot require or recommend that you use any non-free software to use it fully. Any drivers, firmware loaded at runtime, or dedicated utilities to interact with the device must all be free.
- Almost all software running on the device must be free. (There's one narrow exception for software running on auxiliary processors where no free software is available, and the software is not meant to be upgraded by the seller or the user after shipment.) You must be able to modify and/or replace all of that software, too. And you'll be able to do it all without legal worry, because any patents the seller owns or has licensed must be licensed to you.
- You don't have to worry about the device spying on you. The device cannot share your personal information with others unless you give permission, or unless sharing that information is fundamental to the device's

operation. If the device supports formats encumbered with Digital Restrictions Management (DRM), the DRM must be implemented in free software (and thus breakable by users).

Products that meet all of the criteria will be fully endorsed by the FSF. We'll provide them with a seal to put on their boxes, a joint press release, listing on our web site, and other marketing help. Because of this, the criteria do also require that the product's marketing be consistent with our own public messaging, to avoid any conflicts between what we say and what the seller says.

As I write this article, we're still working on refining the criteria based on public feedback we've received so far. In the meantime, we've also received a lot of contacts from hardware sellers who are interested in obtaining our endorsement. Nothing's final yet, but we're very hopeful that we'll be able to announce the first few endorsed products soon. ♡

The MusicBrainz Project

by Matt Lee
Campaigns Manager

During November 2010, I travelled around Europe, meeting various free software developers on my way to FSCONS, the annual free software conference, in Gothenburg, Sweden.

In Amsterdam I met up with Kuno Woudt, better known to many of us as warp, developer for MusicBrainz. He is currently building the next generation of the MusicBrainz website, which supplies metadata on music and its prove-

nance to music player software, companies, and organizations like the BBC.

ML: How long have you been involved with MusicBrainz?

KW: I've been a user for about 5 years — mostly entering music, tagging music, etc. — I've been a developer for about a year, but I officially started working in February 2010. Before I could get started, I had to prove my worth by trying to get some code checked in as part of the interview process.

ML: Tell us a little about the existing site, and what free software you're using to build the new site?

KW: I don't know too much about the existing site, other than it's written in Perl... The site was started about 10 years ago and web development, especially free software, has moved on hugely since then. The new site is written with Catalyst, which is a web framework for Perl, similar to other frameworks like Ruby on Rails or CakePHP. We also use PostgreSQL for our database — as much as we'd like to, we're unable to support other databases because we have some modules that are written especially for PostgreSQL, mostly for collating Unicode data — we have albums and music in languages like Japanese and Korean, and presenting those to an international audience is tricky, so we use "libuca" (Unicode collation algorithm), which is free software.

ML: If people wants to get involved with MB, what are things that they could do?

KW: We have developers who are being paid to work on the site, but we can always use more community help. One of the main uses for MusicBrainz is people tagging their music

this month: the second edition of *Free Software, Free Society: Selected Essays of Richard M. Stallman and Free as in Freedom 2.0: Richard Stallman and the Free Software Revolution*, both in paperback.

The new edition of *Free Software, Free Society* features a number of changes: updated versions of the essays on the GNU Project and free software and a selection of the essays Stallman has written since 2002, when the first edition was published. It draws a starker distinction between areas of copyright and those of patents in a way we hope will put an end to the confusion of the issues; an introduction, written by Brett Smith, the FSF's compliance engineer, now provides historical context for the drafting of the licenses and makes plain the importance of the documents; a part of the book is entirely devoted to the issue of naming and its significance in the survival of free software; another discusses the traps that erode our freedoms, and another urges us to choose civic values and community over convenience; and, finally, there is now an index, to maximize the usefulness of the book.

Also now in print is GNU Press' second edition of *Free as in Freedom*, Sam Williams' biography of Richard Stallman, revised and annotated by Richard Stallman. Williams had put the first edition of the book under the GNU Free Documentation License, thus allowing others to modify and improve upon the work. With *Free as in Freedom (2.0)* the reader can enjoy the manifold advantages presented by this unique format in which the subject's corrections, rebuttals, and opinions are laid beside the biographer's account and interpretations. Williams' interviews, outside perspective, and his

exposition of the facts are corrected, refined, and informed by Stallman's technical expertise and his first-hand knowledge of both the events and his own motivations. See what Stallman has to say about his own life and work and the free software movement.

Proceeds from the sale of these books will help fund our campaigns to promote and defend computer users' rights. Any way you could help spread the word would be useful to the movement: share the books (as stocking stuffers perhaps this holiday season!) with your friends and family, ask your local libraries and bookstores to stock a copy, ask your professors to consider ordering them for their classes, give a copy to the decision makers in your organization, or put a link to the online version on your homepage. Get these books — enjoy a 10 percent discount when you buy both Stallman's biography and his collection of essays from the FSF store — for yourself or for future free software supporters, and help get the word out on this most current and critical social issue.

*If you are not yet a member, please do consider joining. You will have the option of receiving *Free Software, Free Society as your welcome gift and enjoy a 20 percent discount on all items in the FSF store. Thanks for supporting free software.* ♡*

GNU Parallel: A Design For Life

by Matt Lee
Campaigns Manager

On my last night in Gothenburg, at FSCONS, I took a moment to briefly chat with Ole Tange, from GNU

press attention, and projects like Applesseed and GNU social also actively working on the problem. How are these projects working together, and do you think there is room for multiple projects in this area?

MC: We're all tackling the same problem, but this is undiscovered territory, so we're all doing it in our own way, which is very healthy at this early point. GNU social seems to be doing a lot of good work with OStatus, something the Applesseed project has watched closely for the future. And Diaspora has helped popularize the possibility of decentralized, open networks as an alternative to walled gardens like Facebook. I don't doubt we're all interested in interconnecting, and once the various software projects stabilize, it won't be until protocols are finalized, and adopted across all the free software social networking projects. ☺

GNU Contributors Around The World

*by Donald Robertson III
Copyright Administrator*

It's my job to handle the copyright-assignment process for the Free Software Foundation, and because of that I am well aware of the international scope of the GNU Project.

Every day I process and send out assignment contracts to people all across the world. It is part of the copyright registration process to keep a record of the country of citizenship of each contributor. By looking at the citizenship counts in our records, we can get a basic idea of what parts of the planet are involved in GNU devel-

opment. This count paints a rough but interesting picture of our community.

I have noted in my duties that while many GNU contributors are US citizens, the majority of them are hackers in other parts of the globe. Recently, I had an opportunity to compile some rough statistics on where they are all from: while US citizens represent the largest single chunk of contributors, they comprise less than a third of total GNU contributors; Germany and France are the second and third best-represented countries in our records, accounting for 14 and 7 percent respectively (Europe in general tends to be well represented, with the Netherlands in particular having a very high rate of contribution); and while the number of contributors from China and India are currently relatively small (even when combined they represent less than 5 percent of total contributors), the proportions are rapidly shifting, and both are likely to represent a larger share in the future.

In all, we have contributors from sixty-six different countries,⁷ and from every continent except Antarctica—which is pretty good considering one cannot be a citizen of Antarctica.

Sixty-six countries down, 137 to go! ☺

Free Software, Free Society: Second Edition

*by Jeanne Rasata
Assistant to the President*

We are very pleased and excited to announce the addition of two important books to the GNU Press collec-

⁷<http://www.fsf.org/blogs/licensing/gnu-stats-2010-11>

using Picard (written in Python) — Picard is entirely written by volunteers in the community, and we can always use more help there. If you know a lot about music, or you have a lot of records and CDs, you can help us think more about how to model the world of music in our database... for example: a song featuring multiple artists may be released in several ways — figuring out how those releases are labelled on the various CDs and records is a huge help. Classical music is especially difficult because this is not just **songs'** and **albums**," but a much larger body of work with a different structure, because much of the music is in the public domain — there is not just one release. We can use more help capturing that and storing it on our database.

All the software written by MusicBrainz is free software, and free software is used to run and develop the site. Download MusicBrainz software at their website, <http://musicbrainz.org/>. ☺

Solving The Lending Problem

*by John Sullivan
Operations Manager*

Lending and borrowing are not desirable activities. They are things we do when we have to, when there isn't enough of something to go around. Not to say that lending something like a book to a friend is without benefits beyond access to the material; it can create a shared experience that makes for good conversation, or provide an excuse to see each other.

But really, it's something we'd rather not have to do. Even the

best-intentioned friends forget to return things, and sometimes during the lending period we wish we had the book back to look up a favorite quote. A good friend of mine gets upset whenever even the corner of one of her books gets bent — not an uncommon fear among book lovers — and this is a heavy responsibility to bear when borrowing.

We like lending libraries because, like public schools, they give more people access to more information and culture regardless of their wealth. They allow us to explore and research topics we are interested in, in a single location and without the difficult task of actually acquiring dozens of physical books. For many of us, libraries are wonderful places; it can be inspirational to be holed up at a desk somewhere deep in the stacks.

But really, we'd rather not have to use a lending library. All too often, the books we want to borrow aren't there, because someone else has already borrowed them. We've all turned excitedly to a promising page from a book's index, only to find the page has been glued with chewing gum to the facing page, or its key words obscured by coffee stains. The borrowing part of the library is not the good part; the fact that only one person can have a particular book out at a time — and that we all have to repeatedly use the same copies — runs counter to the most important reasons we support the existence of libraries.

If we could have the benefits of lending to friends and borrowing from libraries without the inconveniences, we would. If we could just make a copy of that good book and give it to our friend at no noticeable cost, we would choose to do that, every time. If both

you and I could get the same book from the library at the same time, neither of us would deny the other that.

Fortunately, with electronic books, the inconveniences of lending and borrowing are solved problems. Books can now be infinitely read and shared. We can now all have access to all of them, all of the time. And this is why it sounds so strange to hear digital book companies like Amazon and Barnes & Noble bragging about how their ebook-reader devices “support lending.”

When they advertise this “feature,” what they mean is: “We have managed to take a digital book, and make it not work anymore!” They have removed one of the primary advances the digital book represents for civilization, and replaced it, by design, with a defective version.

They have managed to recreate, in the palm of a reader’s hand, the thrill of tracking down a call number deep in the library stacks only to find its spot occupied by empty space. With a clever arrangement of bytes, they have enabled users to experience the equivalent of being without their books while their friends’ dogs chew on them. Maybe if we’re lucky, next they’ll implement the feature that allows two electronic pages to be stuck together as if by gum, or that translates coffee spilled on the screen into equivalent damage to the digital pages.

It’s clear from these basic observations that these companies are doing us and our books no favors. They have taken a technology which solves the lending problem and twisted it to make lending even more of a problem. But when we consider more closely the details of *how* this electronically simulated lending works, it makes corporate excitement around the antifeature

even more baffling.

When Barnes & Noble announced that its ebook reader would support lending, we raised an eyebrow. Since we knew that it was going to be restricted by DRM, we were curious to know how the device would act out an equivalent temporary transfer of a book by one person to another and back. When Amazon made their “us too!” announcement, we were really intrigued.

It turns out that “lending” to them means a user can — if the publisher permits it — give a book to another person enrolled in the same ebook system (Kindle users can’t lend to Nook users, or vice versa), for a limited and preset time. During that time, the original user may not access her own copy — even though it still exists on her device. This charade can be executed one time *ever* per book. In order to receive the book, the lendee has to register with the authorities at whichever company the book was purchased from, divulging various bits of personal information and allowing the company to track her reading list (which is then also available to subpoena by law enforcement).

This setup bears only the most vague and insulting resemblance to lending. But even if they had managed to replicate the act perfectly, we wouldn’t want it. We lend and borrow because we have to; because physical books *are* physical property. If someone takes one from us, we don’t have it anymore. This is not true with electronic books, and it’s a mistake to try and force them to work the same old way. To do so is to distort the necessary limits of a physical object into unnecessary power relationships, where companies use software to dictate the

terms of our access to knowledge and culture.

Companies attempt to justify this power relationship as necessary to protect authors; but this is a false choice. Authors can be supported without robbing readers and scholars of freedoms from which everyone, including authors, benefit — especially given that there is no evidence such restrictions translate into more support for authors. Plenty of writers out there want their books read and shared, and don’t agree with digital restrictions as a means to secure a living. Companies pushing restrictions are working to increase their own power, not to protect authors — in fact, Apple and Sony have refused to publish ebooks without DRM *even when the authors requested they do so*.

Some publishers are recognizing that the restrictions are an affront — Springer Verlag for example, has been offering their electronic titles to libraries without DRM.⁶ We should support these authors and publishers, and reject the restrictions imposed by companies like Amazon, Sony, Apple, and Barnes & Noble.

The freedom we are protecting here is not the freedom to lend. Lending is the exercise of other kinds of freedom, unfortunately limited by connection with physical objects. The ultimate goal of lending is sharing. The more fundamental freedom we are protecting is the freedom to use our technology as a means to accomplish this same goal of sharing, to the benefit of ourselves and our communities, without agreeing to be subjugated, restricted, or divided from each other, by the companies who “own” the mediat-

⁶<http://ur1.ca/2glwb>

ing software.

You can help us stand for *these* freedoms by following our anti-DRM campaign at DefectiveByDesign.org. Please lend us your support, as we work to make lending a thing of the past. ☹

The Appleseed Project

by Matt Lee
Campaigns Manager

On my last night in Gothenburg, I briefly chatted with Michael Chisari, of the Appleseed project, which is creating a free software social network.

ML: Michael, Appleseed is getting a lot of attention as a free software social network. How can users of Appleseed help the project and what can developer-minded developers do besides running their own Appleseed nodes?

MC: Other than bug testing/fixing and documentation, which are always needed, the best thing developers can start by doing is familiarizing themselves with the framework. Appleseed is a very large project, and it’s a lot easier to plug people in to building features in the roadmap if they have a sense of how it works. There’s some documentation, and a heavily commented example component as a starting point, and the best bet for learning is to build a component from scratch, even if it doesn’t do much. From there, it should be pretty easy to start working on the core components and functionality, and helping out with building new features.

ML: Now is a good time for free software social networks, with Diaspora getting lots of