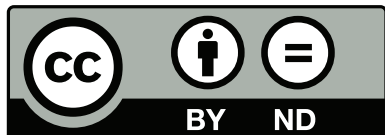


Copyright © 2011  
Free Software Foundation  
51 Franklin Street, 5th Floor  
Boston, MA 02110-1301  
(617) 542-5942  
info@fsf.org  
http://www.fsf.org/

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](mailto: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](http://member.fsf.org) or write to [membership@fsf.org](mailto:membership@fsf.org).

**Online:** Use your credit card or PayPal account to make a donation at [donate.fsf.org](http://donate.fsf.org) or contact [donate@fsf.org](mailto:donate@fsf.org) for more information on supporting the FSF.

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

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

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

**Free Software Supporter:** Make sure you're getting all the very latest news from the FSF at [fsf.org/fss](http://fsf.org/fss).

**LibrePlanet:** Find local groups in your area or start your own at [libreplanet.org](http://libreplanet.org)!



# Bulletin

Issue 19  
November 2011

## Contents

Born to run	1
The new Free Software Directory	3
Automating systems at the FSF	4
PlayFreedom campaign	5
Holiday Buying Guide	5
Growing trends in free software licensing	7
GNU contributors update	7
In defense of negativity	8
How I became aware of free software	10

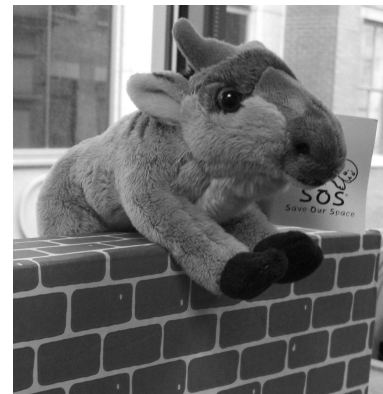
## Born to run

by John Sullivan  
Executive Director

A primary goal of the Free Software Foundation is a world where everyone can do what they need to do with their computers using only free software. Great progress has been made toward this goal. Free software has reached a point where it is now easily possible to buy an affordable computer with hardware — including Wi-Fi and accelerated graphics — that is fully supported by free software (an excellent resource for choosing appropriate hardware is [h-node.com](http://h-node.com)).

But just when our community has achieved this, we are faced with the possibility that most computers in the future could be made *by design* incompatible with free software. They will be born to run only proprietary operating systems.

Microsoft has recently announced that if computer makers wish to distribute machines with the Windows 8 compatibility logo, they will have to implement a measure called “Secure Boot.” However, it is currently up for grabs whether this technology will live up to its name, or will instead earn the



The new baby gnu from [shop.fsf.org](http://shop.fsf.org) (brick wall not included)

name Restricted Boot.<sup>1</sup>

When done correctly, “Secure Boot” is designed to protect against malware by preventing computers from loading unauthorized binary programs when booting. In effect, this means that computers implementing it won’t boot unauthorized operating systems.

This could be a feature deserving of the name, as long as the user is able to authorize the programs she wants to use, so she can run free software written and modified by herself or people she trusts, and not run software by people she doesn’t trust. This should also extend to the firmware (which performs much the same tasks as the BIOS on other computers); users must be able to replace their computers’ proprietary firmware with free software.

However, we are concerned that Microsoft and hardware manufacturers will collude to implement these boot restrictions in a way that will prevent users from changing the firmware or booting anything other than Windows. In this case, a better name for the technology would be Restricted Boot, since such a requirement would be a disastrous restriction on computer users and not a security feature at all.

The specter of Restricted Boot is especially worrisome because it is not alone. Some of the most successful mobile devices (Apple’s iPhone and iPad, and many Android phones) enforce restrictions aimed at preventing users from installing a different or modified operating system — despite the fact that they are general purpose computers. Apple seems to be moving toward a similar world for their desktop and laptop systems.

As an FSF supporter, I don’t need to tell you how frightening and unacceptable expansion of this trend to other computers would be. Users would have to go through complicated and risky measures to circumvent the restrictions; the popular trend of reviving old hardware with GNU/Linux would come to an end, causing more hardware to be tossed in landfills; and proprietary operating system companies would gain a giant advantage over the free software movement, because of their connections with manufacturers.

This summer, we launched a local initiative we’re calling Free Software Fridays, where FSF staff and volunteers went out into the busy Boston downtown area at lunch time, and handed out copies of Trisquel GNU/Linux. In addition to the GNU/Linux operating system, we have future plans to distribute copies of other useful cross-platform free software like LibreOffice, so that even people still stuck on proprietary operating systems can start giving free software a try. This is the easiest, most basic form of free software activism — sharing software with your community. And it would be made impossible if Restricted Boot were pushed on users under the guise of “security.”

We will be monitoring developments in this area closely, and actively campaigning to make sure this important freedom is protected. Our first step is to demonstrate that people value this freedom, and will not purchase or recommend computers that attempt to restrict it. If you aren’t one of the 20,000 people who have already signed our statement affirming that, I hope you will take a minute to

to be superior compared to closed source software because anyone could take part in the development process, fix bugs, and innovate. But there were plenty of instances where I felt the closed sourced software was clearly better than its open source counterpart. For example, it was much easier to create high-quality presentations in Microsoft PowerPoint 2007 compared to OpenOffice.org at the time. Why was it that companies such as Apple and Microsoft were regularly able to create software that was technically superior to the open source movement’s if open source was supposed to be better?

While searching for satisfying answers to my questions, I noticed something interesting in discussions about open source: a few people often used the terms “FOSS” or “FLOSS.” I found they stood for “free/open source software” and “free/libre/open source software” respectively. I wondered why anyone would attach “free” next to “open source,” and I found “libre” just confusing. So I looked up free software and found that it was backed by the Free Software Foundation (FSF).

Intrigued by what this foundation stood for, I explored their Web site and found that they were going to release new editions of their books explaining free software and its motivations soon. A few months ago, the books became available for purchase, and I bought them almost immediately.

Everything started to click in my mind when I read the books. The fundamental principle of free software is that users should have basic freedoms over their software. This includes the freedom to study how a software works, to modify it, and to share the software and any modifications with oth-

ers. This idea strongly resonated with me because of the incident with my laptop. Free software would ensure that users are not beholden to companies to use their own computers.

Free software even addresses the issues I had with open source. It doesn’t matter if proprietary software is technically superior; it fails the most basic test of providing freedom to users. We just have to develop alternatives that do. The idea of free software sits very well with me. There truly is something magical about being able to study how any of the software I’m using works and make changes.

I’m taking steps to contribute to free software and advance it. Countless like-minded people have done so despite facing tremendous challenges and discouragement. In the process, they have made society a better and freer place. I don’t see why I can’t join them and do the same. 🍷

---

<sup>1</sup>[fsf.org/campaigns/secure-boot-vs-restricted-boot](http://fsf.org/campaigns/secure-boot-vs-restricted-boot)



Working together for free software by otheree on Flickr

## How I became aware of free software

by *Ali Razeen*  
*Free Software Activist*

I entered college four years ago, and like most college students, I relied heavily on my laptop, which ran Windows XP. One day, when I turned it on to get some work done, I was greeted with an error message; something failed during the boot process and Windows couldn't load, not even in Safe Mode. I tried to fix the issue by using the repair option on the Windows XP setup disc, but that didn't work either.

I was frustrated because everything was fine the last time I used my laptop. The usual recovery methods weren't working. Reinstalling Windows wasn't an option because doing so would have erased my school work. My assignment was due very soon and I really needed to work on it. Racking my mind desperately for solutions, I remembered discussions on Slashdot about how people used GNU/Linux to fix their Windows problems.

I found a GNU/Linux distribution and used it to boot my laptop. It started up correctly and detected my hard disk. Feeling optimistic, I tried copying my files onto a thumb drive, and it worked! I was extremely relieved. This meant I didn't have to redo my assignment from scratch after all. As I stared at the pop-up showing me the copy's progress, it suddenly struck me that all along, Microsoft was holding me hostage to my own data!

An operating system stores files on a hard disk in some structured format. In Windows XP and later versions of the operating system, the format used is usually NTFS. As the NTFS specifications have not been released by Microsoft, it turns out I depended on Microsoft to be able to access and modify my own data. I was able to use GNU/Linux to get back my files only because talented hackers spent countless hours figuring out how NTFS works. I wouldn't be as lucky next time if Microsoft used a format that was tougher to reverse engineer in the future.

This realization didn't sit well with me, and a couple of things happened. First, when I reinstalled Windows on my laptop, I made a separate partition using the Extended file system, a format whose specifications are freely available, and stored all my data there. That ensured I didn't need Microsoft's help to access my own work should Windows fail again. Second, I paid close attention to the numerous discussions held on sites such as Slashdot. Very quickly, I became a fan of open source and extolled its benefits to my friends.

But before long, I started having issues with open source. The way I understood it, open source was supposed

add your name.<sup>2</sup>

It has been very encouraging to see so much support behind this statement. This support has not only come in the form of signatures — we also welcomed 80 new members and many new donors to the FSF in the month after publishing the statement. Such financial support empowers us to amplify your voice and take action on critical threats like this. If you aren't already a member, please join us at [fsf.org/jfb](http://fsf.org/jfb). If you are a member, thank you! I hope you can double your contribution this year by talking to a friend or colleague about why you think free software is such an important cause, and why now would be a great time for them to join you in supporting it.

We have reason to be optimistic that Restricted Boot will not become a reality. But this is too important to sit and wait for companies to do the right thing. They need to hear from you. We will do as much as we can on your behalf, but we need your support to get it done. 🍷

## The new Free Software Directory

by *Josh Gay*  
*Campaigns Manager*

In September we relaunched the Free Software Directory. We have written a lot about our new plans for the Directory and how you can help us improve and build on more than 6,500 packages

<sup>2</sup>[fsf.org/campaigns/secure-boot-vs-restricted-boot/statement](http://fsf.org/campaigns/secure-boot-vs-restricted-boot/statement)

already listed there. What we haven't written about is how you can begin to help improve the functionality, design, and user experience of the Directory — just by editing pages on the Web site!

One of the coolest things about the new Directory is the functionality provided by MediaWiki and over a dozen extensions, including Semantic MediaWiki and Semantic Forms. Together, they offer a plethora of development tools for the wiki:

- template macros
- an HTML form system to populate templates
- typed variables
- SQL-style queries to find and list pages
- built-in formatting functions for HTML, RSS, JSON, and more
- dozens of array and string functions

If you look at the source of any page on the wiki, odds are you'll find extensive use of nested templates, conditional statements, complex queries, and other powerful semantic features.

If you are interested in trying out some of these tools, a great place to start is our quick hacking tutorial.<sup>3</sup> This will teach you the basics of how to edit your user page, use and create templates, query the directory, create and manipulate arrays, and much more.

We hope that you will try out our tutorial and help us improve the look, feel, and functionality of the Free Software Directory! 🍷

<sup>3</sup>[directory.fsf.org/hack](http://directory.fsf.org/hack)

# Automating systems at the FSF

by *Ward Vandewege*  
*Chief Technology Officer*

The sysadmins at the FSF have put great effort into consolidating our server infrastructure over the past 12 months. We have retired more than a dozen servers in the process, and now host most of our infrastructure on three potent machines with many CPU cores, ample RAM, and plenty of disk space. As you may know, we use Xen to virtualize our servers. Virtualization allows us to securely partition our servers into many virtual machines, each dedicated to a limited number of tasks.

We have also embarked on a journey to automate our systems configuration as much as possible. We have selected Puppet, a systems configuration management and automation tool, to help in that task.

One of the advantages of Puppet is that one can start small. Once a system is under Puppet's control, it is easy to expand the Puppet configuration over time, and thus automate more and more of it. We started out with a very limited configuration that defined some settings that are common to all our systems — for instance, making sure the `sshd` configuration is secure. Puppet makes it easy to differentiate rules based on “facts” about a machine, like the version of the operating system it runs, or whether the system is a physical server or a virtual machine. Each system under Puppet's control gets its own configura-

tion stanza, so it is also possible to do things that are specific to one machine.

Here are some of the things that we now do with Puppet, rather than by hand:

- install appropriate software packages, and standard GNU/FSF configurations for them
- create and manage user accounts
- distribute SSH public keys and SSL certificates
- update xen-tools configurations on virtualization host systems

We create new virtual machines with xen-tools, which pulls in the Puppet packages. We then add the new virtual machine to our Puppet configuration, which pulls in the default GNU/FSF configuration without any additional work on our part. We save a lot of time setting up new machines, and we get peace of mind: all our systems under Puppet's control are guaranteed to have our standard configuration.

We are currently working towards the goal of generating our automated systems monitoring configuration from our Puppet configuration. This requires us to migrate more service configurations to Puppet. With enough of that done, it should be possible for Puppet to know that, for instance, `www.gnu.org` runs a web server on port 80. With that knowledge, Puppet can instruct our monitoring hosts to check for the availability of that service, all without manual intervention from the sysadmins.

I would like to conclude this article with a brief word of thanks to Bernie Innocenti, who left the FSF for another

software, software patents, DRM, centralized network services, and the firms behind these practices. I've done so because I believe that if one is taking an ethical position, it is justified, and often necessary, to not only speak about the benefits of freedom but against acts of dispossession and disenfranchisement.

In some of the most effective social movements, unambiguously negative messages have been central. Should a campaign for abolishing child labor talk only about how valuable adult workers are to their employers, or how happy kids are when they don't work? Should a campaign trying to abolish land mines talk only about the benefits of bomb-free fields and intact lower limbs? Should a free speech organization only speak out about the social welfare brought by a free press and never against acts of censorship? These may seem like outlandish comparisons but you can find people writing, only a couple centuries ago, about how slavery should be abolished by arguing in favor of the benefits of paid labor. Even if the economic arguments in favor of paid work are strong, these arguments seem irrelevant and offensive today. Whether slavery is more or less efficient is a moot point. Society has rejected it because it is wrong.

We have made important strides toward eliminating injustices like child labor and slavery because activists waged decidedly negative campaigns against them and convinced others to join in opposition. In doing so, activists declared the status quo unconscionable and created an ethical responsibility to find alternatives and to redefine what was “realistic.” While I will not suggest that the movement for software freedom is comparable in

ethical weight to these other causes, I know that the free software mission is similar in kind.

Of course, if one does not think that user control over technology is an ethical issue but is instead merely a matter of choice, one will probably oppose negative campaigns. It is also possible that a particular negative campaign is tactically unwise in that it is unlikely to reach a large audience, unlikely to change people's minds, or difficult to carry out successfully. But such campaigns are a bad idea because they are ineffective, not because they are negative. Additionally, a movement that is purely negative and offers no reasonable alternative to the stated ill may also be unlikely to succeed. This is why, for example, I believe it is good that the FSF uses the large majority of its resources in the “positive” role of supporting free software.

For those that do treat technological empowerment as an ethical ideal, it is both justified and essential to condemn the systematic disempowerment of others through nonfree software just as we celebrate the benefits of software freedom. “Negative” campaigns against proprietary software, software patents, and DRM in music have already led our community to important — if incomplete — victories. The desire to right wrongs has been a critical part of our movement's success and of many others'. We would be wise not to give it up. ♡

FSF has for many years taken copyright assignments on many GNU packages. Thousands have contributed to FSF-assigned packages like the GNU Compiler Collection (GCC) and GNU Emacs, but the assignment process has always created a slight barrier for new contributors. That is why we were so happy in October when we announced two major changes to how we handle copyright assignments at the FSF.

Our lawyers at the Software Freedom Law Center have given us the go-ahead to begin accepting scanned copies of assignments from contributors located in the United States. We are in the process of researching if we can accept scanned documents from contributors based in other countries. As we hear back from lawyers from around the world, we will be updating our list of countries where scanned documents can be accepted.

But even for contributors in countries where scanned assignments cannot be accepted, we have implemented a change that significantly speeds up the assignment process. Instead of mailing forms via the postal service to each contributor, we now simply email a PDF of the assignment that the contributor can print themselves. Previously, we would only accept forms that were printed on our letterhead, in order to ensure that the form was not changed. As the number of contributors from around the world grows, relying on the postal service is increasingly inefficient. Hence, we decided to start sending forms electronically, reducing transit times and ensuring more consistent delivery.

Accepting scanned documents will greatly cut down delays in the assignment process, and using electronic forms wherever possible will help avoid

the vagaries of the international postal system.

Together, these changes will make it easier than ever to contribute, and hopefully encourage more hackers to help the GNU Project. 🍷

## In defense of negativity

*by Mako Hill*  
*Director, Free Software Foundation*

I often hear criticism of “negative campaigning” in the free software movement. For example, in reply to a blog post I once wrote about an FSF campaign, several people argued against “negative campaigning of any sort, in any realm.” Drawing an analogy to political smear campaigns, some members of the free software community have taken the position that negative campaigning in general is not useful and that negativity has no place in our advocacy.

First, it is important to be clear on what we mean by a negative campaign. I believe that there is a fundamental difference between speaking out against policies or actions and smear campaigns that employ untrue claims, ad hominem attacks, and that attempt to avoid a real conversation about issues. I will categorically condemn the latter form of smear campaigning in campaigns for software freedom or for anything else.

That said, negativity directed at negativity has had a positive effect in many social movements. I have supported and participated in “negative” campaigns against proprietary

job in September. We are extremely grateful for his contributions as an FSF sysadmin; he was instrumental in the server consolidation and Puppet setup efforts described here. 🍷

## PlayFreedom campaign

*by Matt Lee*  
*Campaigns Manager*

PlayFreedom is a new campaign from the Free Software Foundation. You can think of it as the successor to our PlayOgg campaign; our efforts there will be rolled into PlayFreedom.

PlayFreedom aims to promote all kinds of free media formats: for now, that means Google’s new WebM format and the entire Ogg family from Xiph, including Theora, Vorbis, Speex, and FLAC. The Web site will have three main sections: Tools, Showcase and Activism.

“Tools” provides information about free software you can use to produce, edit, and work with free media formats. Whether you’re simply looking for a program to play video, or the tools you need to edit and publish a movie with free formats, you’ll find what you need here.

“Showcase” is an area of the site where we’ll be showing you some of the best video and audio around, all in free formats. Part of this will include an exhaustive list of all known free media sites, including radio stations, music download services, bands, filmmakers, and community sites.

“Activism” helps us all push free media formats to success in new areas.

Here, PlayFreedom supporters will find links to blog posts, action items, and suggestions on how to get involved and promote free media formats.

While we put the finishing touches on this campaign, you can let us know about any sites you’d like to see featured on the Web site, or any tools you think we might not have heard of by emailing us at [campaigns@fsf.org](mailto:campaigns@fsf.org). To stay up to date on our progress, join the mailing list at [playfreedom.org](http://playfreedom.org). 🍷

## Holiday Buying Guide

*by Josh Gay*  
*Campaigns Manager*

With the holidays fast approaching, we thought a holiday buyer’s guide was appropriate. People often buy expensive electronic gadgets as gifts, but many of these gadgets are bad for your freedom.

Here’s our guide to the best and worst gifts this season:

**Video games** — With all of the big game console manufacturers supporting and using more Digital Restrictions Management (DRM) in their products, there aren’t good options here. But in particular, avoid Sony, since they have been aggressively prosecuting users ([defectivebydesign.org/sony](http://defectivebydesign.org/sony)), and Nintendo, since they claim the right to remotely “brick” some devices ([defectivebydesign.org/nintendo](http://defectivebydesign.org/nintendo)).

**Blu-Ray** — Avoid Blu-Ray discs. They cannot be used with free software, and buying them is supporting

Old Media's latest system for controlling every step of what users can do with media they buy.

**Music players** — For music this season, don't look to Apple for anything; all their portable music players support DRM and many devices refuse to work on GNU/Linux without serious tinkering or jailbreaking.

**Cars** — If you're thinking of buying a new car, be sure to avoid cars with the MyFord Touch and MyLincoln Touch software. Glitches in this proprietary software have led Ford to issue an update. This means the car can be infected with malicious software and is best avoided.

**Smartphones** — Replicant, the project to produce a fully free distribution of Android recently added more phones to its list of supported models. Here at the FSF, we scored a bounty of HTC Dream phones, and many of us use them as our day-to-day cell phones. These phones are still problematic, because they include a proprietary firmware that could monitor all activity on the phone, but the ability to have a fully free application environment is a great step forward.

**Ebook readers** — All the ebook readers sold through Amazon and Barnes & Noble use DRM and proprietary software. Even if a user manages to only use these devices to read DRM-free materials, purchase of the device includes licenses for DRM and proprietary software.

**Laptops and Desktops** — There are a couple of companies who will sell you a computer with a fully-free GNU/Linux distribution preinstalled. Some of these companies even donate a small amount of money to the FSF. One such company is Los Alamos Computers, [laclinux.com/gnu](http://laclinux.com/gnu).

Instead of buying gifts that reward companies who restrict our freedom, consider making a donation to a charity that fights for your freedom — other than ourselves, we're especially fond of the work done by the Electronic Frontier Foundation, Creative Commons, and Public Knowledge. ♡

**BEFORE YOU  
BUY AN iPad,  
KINDLE, CAR,  
PHONE or A  
NEW LAPTOP  
BE SURE TO  
READ THE FSF  
2011 HOLIDAY  
BUYING GUIDE**

<http://defectivebydesign.org/holiday>

## Growing trends in free software licensing

by *Brett Smith*  
*License Compliance Engineer*

Lately I've noticed an uptick in the number of pundits who claim that free software developers have begun to prefer using lax free software licenses that don't have copyleft (like the Apache License) over ones that do (like the GPL) for their projects. They back up this claim by pointing to surveys that show increased adoption of lax licenses in free software projects, or high-profile projects that have recently adopted such licenses. That evidence tells a different story, however, when you better understand its background.

Typically, surveys show upticks in lax license adoption when they start counting projects on sites that are more specialized than general-purpose hosting sites like Savannah and SourceForge. For example, one such survey saw a huge one-month increase in the use of Microsoft's free software licenses simply because they started counting projects on the Microsoft-backed CodePlex hosting site. These projects deserve to be counted just like any other, but their dramatic one-time effect on a survey doesn't represent the kind of licensing sea change the accompanying headlines would suggest. I have not seen a survey rigorous enough to demonstrate that new projects choose copyleft licenses in fewer numbers than before.

Similarly, news about high-profile projects adopting lax licenses often fails to tell the whole story. The most recent example I'm aware of is

Ruby, which recently started using a 2-clause BSD license where they used to use GPLv2. However, much like Perl before it, Ruby has always been dual licensed, and the second license has never provided a strong copyleft. The developers didn't make this switch to move away from copyleft, but to make Ruby code compatible with more licenses: they even specifically mentioned GPLv3 compatibility in their rationale.

It seems undeniable that lax licenses like the Apache License are used by more projects today than before, but that growth doesn't necessarily come at copyleft's expense. Instead, I see this growth coming from two places. First, as more new free software projects are constantly released, of course some of them will be under lax licenses — that's always been true. Second, more developers understand the importance of license compatibility, and are switching from unpopular so-called "vanity licenses" to standard licenses that are widely recognized and respected. These migrations help users to better understand their rights, and developers to know how they can combine the code with other projects.

Increased use of lax licenses doesn't mean that copyleft is losing. It means that free software is gaining ground. ♡

## GNU contributors update

by *Donald Robertson*  
*Copyright Administrator*

As part of our mission to protect and promote free software, the