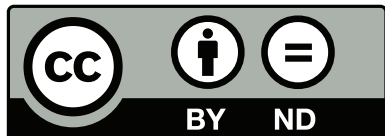


Copyright © 2009
Free Software Foundation
51 Franklin Street, 5th Floor
Boston, MA 02110-1301
(617)542-5942
info@fsf.org
http://www.fsf.org/

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.



Adopt a GNU today

← Aww, cute!

shop.fsf.org

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 15
November 2009

Contents

Welcome to the Bulletin	1
FSF in 2009	2
Life in the Licensing Compliance Lab	4
The importance of individual membership	5
Software for education, not babysitting	6
Campaigns Update	7
End Software Patents	7
The computer in my pocket	9

and some of the upcoming projects for us here at the FSF.

Our Web site, www.fsf.org, has also been updated to reflect our current fundraising drive, with video features from prominent members of the free software community, including Samba developer Jeremy Allison, hardware designer for the \$100 laptop, Mary Lou Jepsen, long-term GNU hacker and the present lead developer of Gnash, Rob Savoye, and copyright activist Cory Doctorow.

These videos represent the vanguard of what is possible on the Web now — with support for the free Ogg Theora and Vorbis multimedia formats built into many of the Web browsers being used, we can bring more interactive and engaging video media to our advocacy work. As an example of this, we've dug up a video of former campaigns manager and now operations manager, John Sullivan, from the archives. This video, shot in 2007, features John explaining the basics of free software and makes an excellent introduction to the subject for any newcomers, as well as providing a good example of how our various campaigns relate to each other. Work done by the pioneering Xiph.org hackers and advocacy from PlayOgg is now affecting real change in all of the FSF's campaign actions. ♡

Welcome to the Bulletin

by *Matt Lee*
Campaigns manager

Welcome to another issue of the Free Software Foundation *Bulletin*. Right now, we're in the middle of our busy fundraising season. As a supporter of the Free Software Foundation, we depend on your generosity — in the form of donations and associate memberships — to do the work we do. So, in a change of pace from our usual *Bulletin* format, I have asked the staff at the Free Software Foundation to give an update on how your financial support has been used recently,

FSF in 2009

by *Peter Brown*
Executive Director

The free software movement is one of the most successful social movements to emerge in the past 25 years, driven by a worldwide community of ethical programmers dedicated to freedom and sharing. Its impact on our future is growing every day. But the ultimate success of the free software movement depends on teaching our friends, neighbors and work colleagues to recognize the danger of not having software freedom — a freedom that they have lost, often without recognizing it, to proprietary software.

The FSF is currently working on three fronts to advance the cause of the free software movement. First, in software, we sponsor the GNU project and promote the adoption of fully free GNU/Linux distributions gnu.org/distros like gNewSense and Trisquel. We identify high priority free software projects at fsf.org/campaigns/priority.html that need developer focus and resources to advance the adoption of GNU/Linux systems, and we work to alert the community to threats to free software, such as the seduction by popular but patent-encumbered platforms, or misleading efforts that direct developers to create free software for proprietary platforms.

Second, in licensing, we publish the world's most popular free software licenses, including the GNU General Public License, and provide licensing help and guidance to the free software developer community through our Free Software Licensing and Compliance Lab. At the Lab we collect copyrights from thousands of developers working on the GNU operating sys-

tem, register those copyrights with the US Copyright Office, and use the copyright system to enforce the terms of our copyleft licenses to guarantee respect for the freedom that our licenses promise, to all recipients.

Third, we campaign to raise awareness of the ethical benefits of free software and against the use of proprietary software. Our campaign against Digital Restrictions Management (DRM) at defectivebydesign.org turned acceptance of anti-user technology measures into a public campaign that now makes DRM systems highly unpopular. Our successful campaign against adoption of Windows Vista, and our new campaign against Windows 7, have raised widespread concern about how proprietary software works against the interest of all citizens. Our campaigns to promote free formats such as OpenDocument and PlayOgg have achieved widespread support. And finally, our 20-year campaign against software patents will this year see our legal brief from our End Software Patents campaign gain attention from the US Supreme Court in a landmark ruling expected in May 2010.

Of course, we do all this work in collaboration with free software users and developers like you, who volunteer their time to help a campaign, or who join a GNU project to hack on code, or who become associate members to show their support and fund our efforts. Thank you! 🍷

Life in the Licensing Compliance Lab

by *Brett Smith*
Licensing Engineer

laptops. Ubuntu's decision to ship a new binary driver remains more controversial than the fact that the vast majority of the world's computer-using population knows nothing other than phone-based computers that remain almost unthinkably unfree and which remain almost entirely unfreeable when compared to personal computers. For most of the world's computer users, there is no option of, and essentially no hope for, freedom on their current devices.

It shocks me that anyone, especially free software advocates, would happily put up with such nonfree computers. I think part of the reason lies in the fact that most users of mobile phones, and even most phone users that care about software freedom and technological autonomy, don't think of their phones as computers. Thinking that our phones as computers will not solve any of the problems I've alluded to. But doing so remains an essential first step toward any solution. Although we must still work to build viable, widely accessible, and compelling free phones, we must first convince both users and developers that this is an important goal. Reminding people that our phones, both free and nonfree, are powerful general-purpose computers remains an important and still largely unfulfilled part of this process.

We must find ways to remind ourselves and others of the fact that modern phones are powerful computers with powerful interfaces that are useful for an unimaginable variety of arbitrary applications. We must focus on the fact that these computers have microphones, sensors, and other sensors and that we trust them with our closest secrets and most sensitive data.

We must not forget that, in almost all cases, these computers remain controlled, completely and ultimately, by companies that very few of us trust at all.

I'm not sure how we will accomplish this task. But more of us need to think long, hard, and creatively about this problem. I'll be calling my phone "my computer" as a first, very personal, step. I have done this over the last week and it has led to some conversations with slightly confused acquaintances. Of course, this doesn't make my phone any less free. But it does mean I'm talking more about the non-freeness most of us have put up with too silently.

At this stage, that seems like progress. 🍷

Microsoft in 2007 began claiming that the kernel violates 235 of its patents — although the patents have never been specified. Neither could be precise, but they give us ballpark figures.

The kernel is one component, and because the human-written source code is online, we can see it is approximately 4,000,000 lines long. Given that a distribution of the GNU/Linux operating system, complete with applications, can contain software with more than 225 million lines of source code, when we extrapolate from the kernel numbers we arrive at the possibility of 13,160 or 15,848 patent infringements per complete distribution. All of this in something that can be distributed once or a thousand times, usually at no cost, sometimes by large corporations, sometimes by individuals.

This is a degree of uncertainty that can't be fixed by changes in evaluation standards.

There was a time when if you wrote something, you owned it. You could distribute it, you could use it as a starting point for collaboration. Whether the ownership is a good or bad thing for society depends on what freedoms you grant the recipients, but at least those who did the right thing had legal certainty. Now, ownership of a piece of software is hopeful speculation. There is no reliable way to have a settled expectation regarding the boundaries or the extent to which you own a piece of software. The Supreme Court now has the chance to rid us of this uncertainty and this unfair regulation, by giving the United States Patent and Trademark Office a reliable tool for excluding software ideas from patentable subject matter. You can support our efforts and follow ongoing news as the

Bilski case unfolds at news.swpat.org. 

The computer in my pocket

by Benjamin Mako Hill
Director

If we've kept up with projections, by the end of this year, the world will be home to three billion mobile phones. That's nearly one phone for every other living human being. Although these phones open up a world of important new opportunities in communication, creativity, and cooperation — and it's important not to understate this fact — they also represent a step toward a sort of technological dystopia not unlike Stallman's *Right To Read*.⁴ Phones represent one of the most locked-down, proprietary, and generally unfree technologies in wide distribution. The implications for software freedom and technological empowerment are dire.

But despite the fact that mobile phones represent what may be the greatest threat to software freedom today, the free software community has — with a number of notable exceptions that I want to both thank and draw increased attention to — been mostly silent on the issue.

I know passionate advocates of software freedom who work tirelessly to rid themselves and the world of a handful of binary blobs in the kernel Linux — important work that we all benefit from. And yet, even some of these “hardliners” don't seem to hold their phones to their same standards as their

⁴gnu.org/philosophy/right-to-read.html



Watch Jeremy Allison's video at fsf.org

Initially, the Free Software Foundation staff wrote software for the GNU operating system. Since the early 1990s, much of this work has been done by non-FSF staff, thanks to the increasing popularity of GNU. Some of our work has then shifted to writing and maintaining the GNU General Public License (GPL), a copyright license which protects free software by allowing people to run, modify and copy software, but on the condition that anyone else receiving the software have the same rights. The FSF's Free Software Licensing and Compliance Lab works to protect these rights, and to help update the GNU licenses when needed to deal with the ever-changing free software landscape.

The past few months have been exciting ones for us in the Lab. Just reviewing the news from the past few months demonstrates how we work on many different fronts to protect software freedom for everyone.

Shortly after we published our last *Bulletin*, we settled the lawsuit we had brought against Cisco over their violations of various free software licenses. As part of that agreement, Cisco appointed a Free Software Director for its Linksys subsidiary to oversee the company's compliance work. It also took various steps to notify its previous cus-

tomers that they had the right to share and change some of the software they received. That finally put an end to a case that had been open for more than five years.

Adoption of GPLv3 and the Lesser GNU General Public License version 3 (LGPLv3) continues apace—we've even handled a couple of compliance cases that involved GPLv3-covered software. Most GNU projects have had multiple releases under the latest licenses now. The few that haven't generally need new exception text, and we've been working on getting those updated, slowly but surely. We released new exception text for Autoconf in August, and we're still drafting more.

In September, we filed an *amicus curiae* brief in *The Authors Guild, Inc., et al. v. Google Inc.*—more colloquially known as the “Google Book Search case.” This case began when a group of authors sued Google for scanning books for their Google Book Search product, alleging that such use infringed their copyrights. As the case progressed, the parties proposed a wide-reaching class action settlement that would generally grant Google permission to display and sell all books—including out-of-print works and orphan works whose copyright holders can't be found—under the condition that they pay royalties to the authors, who can opt out of the program if they wish. One consequence of the settlement that was proposed was that it would grant Google permission to publish works released under the GNU Free Documentation License, and other copyleft licenses, without following those licenses' terms. They would not have to provide the work in a form that people could mod-

ify; they could distribute the work wrapped in a Digital Restrictions Management (DRM) format if they wished; and more. All they would need to do is pay royalties to the authors. We suggested to the Court that the settlement would do better to require license compliance when the work is already free, rather than royalty payments. Since then, the parties in the case have announced that they are working on a revised settlement, which hasn't been released yet. We're still following this case, and hopeful that the negotiating parties will take our suggestions into account.

And of course, there's still all the work that goes on behind the scenes. In order to pursue violations, we need to stay on top of our copyright assignments and registrations. Donald Robertson, our copyright administrator, has been working to beef up our registration process to make sure that we're always on completely solid legal ground when we handle these cases. We continue to resolve most violations cooperatively, and work to raise awareness about the licenses' requirements.

As the stewards for some of the most popular licenses in the free software community, and legal guardian for the GNU Project, we have unique opportunities to educate people about free software and make sure that others respect the terms of our licenses. Your support makes it possible for us to do that work.

You can find out more about the FSF Free Software Licensing and Compliance Lab at its website, fsf.org/licensing and questions about our licenses can be sent to licensing@fsf.org. Brett, Donald and the licensing volunteers also have a blog, fsf.org/blogs/licensing and are

always on the lookout for interns interested in licensing. ♡

The importance of individual membership

*by Deborah Nicholson
Membership Coordinator*

I'm sure you'll hear the assertion that "membership is important" more than once this season — in the current economy every organization could certainly use money, but why exactly is membership so important?

For the FSF it provides funding to employ a small staff of twelve so we can support the efforts of thousands of volunteers all over the world, as well as providing the kind of autonomy that an organization subject to grant-making trends and corporate whims can't enjoy. Most importantly, as members **you** are the face of the FSF in your community.

We get a huge amount of work done for such a small staff — we maintain the GNU General Public License (GPL), house the GNU Project, run the Defective by Design campaign and defend free software (and free software users) from patent abuse, secret formats and hardware that requires non-free drivers. I find that I sometimes surprise our supporters when I tell them there are only twelve of us. The frequent visitors to our office are given the tour and I get the feeling that they don't quite believe us. Perhaps they suspect that we're not showing them the secret floors where the wizards and their minions work?

Membership is individual, personal and if you wish, it can also be private. Your decision to support free software

anything to gain for the patent holder (although the patent will still be enforced to sink the piece of software so that computer users are pushed toward a program which will pay royalties to the patent holder).

In software, rather than supporting innovators, patents protect the old against the new.

This issue is further exasperated by a problem which applies to all types of software developer: in no other domain are modern standards as crucial as they are in software. If you want to cure rubber, there are many ways to do it. When patents block the developer of physical products from using one method, there's the possibility of useful innovation when that developer looks for an alternative method. In software, being blocked from using an email, image, or document format equates to being prohibited from writing a functional email reader, image viewer or word processor. An innovative word processor that can't read any existing documents is simply useless, and any innovations therein are thus wasted effort.

For video, the standards problem is very real. The MPEG-LA group claims to represent more than twenty patent holders which each have one or more essential patents for implementing the commonly used mpeg video format. There's no license available for freely redistributable software, and even royalty payers have to agree to MPEG-LA's terms. The committee developing the next standard for Web pages, HTML5, spent months searching and debating which video format they could recommend in the standard, and the final answer was that, due to software patents, there is today no format they can recommend.

Now, it's important to look at the output of the mentioned user communities. If like, say, hobbyist watch-makers, they just catered for themselves and a few friends, or a small enough clientele that didn't attract the attention of patent holders, then this wouldn't be a big problem. The system would still be unjust, but if the injustice never manifested itself, then it would be theoretical issue.

However (as supporters of the FSF know), freely redistributable software and the work that was begun by idealists and hobbyists has now lead to the world's most used Web server, the world's second most used Web browser, and the GNU/Linux operating system. Indeed, the "users" are nowadays often employees, and their collaborative development models have emerged as the primary competitors in many software domains. Blocking collaboration turns out not only to be a restriction on useful individual activities, but it also stifles competition and the mass production of useful software. Lists and lists of research suggests that patents reduce software innovation.

Although large firms now contribute to these projects, many of the developers are still individuals and people who don't directly profit. The terms of distribution for this software are the same now as they always have been. It's a proven formula, and a key clause is that you can't distribute if patent royalties will be required.

The kernel of the GNU/Linux operating system was examined by patent attorney Dan Ravicher, who announced on August 2, 2004, that he had found no court-validated patents to be infringed but 283 *issued patents* existed which could potentially be used to support patent claims. Thereafter,

The *Bilski* case is the first review of patentable subject matter since 1981. This decision could make the rules for decades to come and the justices' comments at the November 9th hearing indicate that they do see problems with the patenting of software. This hands us our biggest opportunity, and a heavy responsibility.

Some legal experts have speculated that the ruling will be handed down in early Spring 2010. Others have suggested that whatever the result, legislative change will be proposed. The ongoing Supreme Court case makes it easier for us to raise media interest in this topic, and if it comes to legislation, we're going to need broad public support for abolition of software patents. In the next three months, we have a unique chance and a need to build a groundswell for abolition of software patents in the USA. That's where we need you.

The Supreme Court isn't obliged to rule on the patentability of software ideas. *Bilski's* patent is a business method patent, not a software patent. So why might the court make a broad ruling which would cover software?

The low cost of entry to software development means the number of small companies is particularly large, but we'll leave that aside to look at a bigger difference. In most patentable fields, the array of big and small companies describes how products are made. If this were true for software, then the decision of patentability would only have economic implications, and patents would only raise economic problems. But in software, this is only half the story.

In software, unlike in other patentable fields, there are two additional categories of developers. The first is

the software developers that sit in the IT departments of every medium-sized company. They're the folks that keep the emails flowing, who write internal software, extend software bought by the company, and who run the Web site. The second group is individuals, informal groups and communities who program for their own benefit or for social reasons such as providing alternatives to software seen as overly restrictive.

The existence of these two categories changes everything because it's impractical to require them to work within the slow patent system and bear the legal and financial risks involved. Obviously, patent incentives are not necessary to motivate IT departments to fix problems. Further, when a company manager reports a Web site problem, they don't expect the IT department to reply about first seeking legal advice for a patent search, and they don't expect to later have a bill or a cease-and-desist letter from a patent holder because of the way in which the IT department happened to fix the problem.

A second issue is that applying industrial regulations to activities people do for fun or for the benefit of their community is unjust. For user communities programming to suit their own needs, the veto power that the patent holder gains over distribution of the software is far too powerful. If the software is written for the purpose of having a freely redistributable program, then this third-party veto can spoil the developer's efforts at any moment. There will be no direct profits from which to offer royalty pay-

ments, so the result is a lose-lose situation where the developer's work is destroyed, and there was never even

is not subject to the desires of a group of shareholders. It is not a decision which is made with the eyes of the grant-making community on you. It is not even made with your employer's approval or disapproval in mind. Unless you are one of those happy folk employed at a free software company, where FSF membership demonstrates your commitment to shared goals, your coworkers may not even know that you're one of us.

I left the best bit for last — the thing that we cherish the most about our members, is our members themselves. You get updates from us every week or so and so you know about the challenges to free software. Maybe you pass the news along to your friends and colleagues and spread the message of free software that way. Maybe you've helped a relative install Inkscape or GIMP and explained to them a little bit about free software as you worked. Nobody else in the free software movement has the easy opportunities that you do to have conversations with your cousin or your neighbor about user freedom, which is precisely why you are so important. You are already right there, in your community — a card-carrying member of the Free Software Foundation — you are the movement.



Rob Savoye, GNU hacker – 'I support free software and the FSF'

Software for education, not babysitting

by John Sullivan
Operations Manager

Education is one of the most important fronts for the advocacy of free software. Free software is better philosophically for education, because it acknowledges the freedom of students to do what they are supposed to be doing — learning and applying what they've learned. Whereas proprietary software sets technical and legal restrictions which limit how far kids can go in their learning, even calling them criminals if they try too learn too much or get too creative, free software encourages them to go as far as they want to go. Educational environments are also critical because school is an important time of life for acclimating people to software; it's one of the main places where people are indoctrinated with proprietary software and the idea that to use a computer is to use Microsoft Windows.

Despite the perfect fit between free software and education, it's proved one of the hardest areas for our message to be heard. While free software can be found in university computer science departments, it is rarely found in non-specialist areas of education. Proprietary software companies offer both students and schools discounts, or even give it away at no charge. Many universities and even some high schools are distributing laptops or other mobile computers to their students now — loaded with proprietary software. This means that we won't succeed in promoting free software with weaker "open source" arguments like price or convenience. We need to stress the free

software values of freedom, autonomy, and pedagogy.

We'll have to make arguments like Walter Bender of Sugar Labs made in his inspirational talk at this year's Software Freedom Day event in Boston. He stressed that education is action, and that kids learn best by doing. Only free software enables this in the end — proprietary software can let kids drive different vehicles, but it won't ever let them look under the hood. He described the culture of free software as a culture of sharing and critique, and proprietary software as a culture of babysitting.

Walter also announced the first success of a project that the FSF had been discussing with Sugar Labs — a fully free version of “Sugar on a Stick.” This is a bootable USB stick running the FSF-endorsed Trisquel GNU/Linux distribution, which loads the Sugar learning environment.¹ Sugar has been used in the One Laptop Per Child project, but using this USB stick, it can run on other laptops and desktops as well. Because it doesn't require installing any software on the computer, it's a great way for people to demonstrate and introduce free software in schools. I encourage you to give it a try — the way it provides guided creative activities while also exposing the workings of the activities for people who want to tinker is an amazing model.

In addition to collaborations like this with other organizations, the FSF has been building its own efforts in the area of education. During the last year, we launched our new internship program, which has already connected the FSF with several students at both the graduate and undergraduate lev-

els.² While we've had interns helping out in the past, we've now formalized the process and started to build ongoing relationships with universities. These interns not only do great work at the FSF, but also return to their schools better equipped to teach other people about free software.

One of our interns, Max Shinn, started the GNU Generation program, a project to involve pre-university and high school students in free software.³ GNU Generation has attracted a great deal of interest and participation, and its members have already made valuable contributions from the local to the global, including starting free software groups at their own schools and translating the Windows 7 Sins campaign site. Another intern, Sarah Adelaide McIntire, helped build the LibrePlanet wiki at libreplanet.org with over 2,000 edits containing information about free software groups around the world. Niko Kern and Bernie Innocenti have provided very valuable assistance to the FSF systems administrators, supporting and improving the infrastructure the FSF community and the GNU Project depend on.

It is incredibly encouraging to see such efforts for promoting free software with young people and students gaining momentum and having success. The FSF will be putting even more energy into them in the coming year, as a critical step toward our goal of a creative, free society. You can help by applying to our internship program to spend a few months working closely with our staff and community on exciting free software projects, or by telling students you know about

²fsf.org/volunteer/internships

³fsf.org/gnugeneration

¹trisquel.info/en/trisquel-sugar

the opportunity. You can promote free software in your school or in your community's schools, and share any letters or texts you may write as part of that effort on LibrePlanet, for others to reuse. You can renew your membership, or convince a friend to join, to expand the resources we have to take on this work. Finally, you can let us know at campaigns@fsf.org about the threats and impediments to free software in your university or school, so we can take action. Together we can make computers tools for learning, not babysitting. ☞

Campaigns Update

*by Matt Lee and Holmes Wilson
Campaigns Managers*

The campaigns team has been busy since the last *Bulletin*. Here are some of the latest things we've been working on:

Windows 7 Sins — Microsoft Windows has hundreds of millions of people locked into proprietary software. It exemplifies some of proprietary software's most serious problems, like security and privacy. Windows 7 Sins (windows7sins.org) makes the case against Microsoft and proprietary software, in response to the latest release of Microsoft Windows. The launch of the campaign, coinciding with the launch of Windows 7, featured the construction of a giant trashcan in the Boston Common, into which participants threw mock boxes of popular proprietary software. Visit fsf.org for video of the campaign in the free Ogg Theora format.

Defective by Design — Digital Restrictions Management (DRM) robs us of control over the technology we

use and the culture we live in. DRM and the DMCA can make it illegal to share an article, backup your kids' favorite DVD, or move your music from one player to another. The Amazon Kindle ebook reader (we call it the “Swindle”) has been a target of several Defective by Design actions. So when Amazon stepped into a public relations nightmare this summer by remotely deleting hundreds of copies of George Orwell's *1984*, we responded with a petition demanding an end to Amazon's ebook DRM. It received over 4,000 signatures from readers, academics, and librarians, and the press coverage helped draw public attention to the fundamental problems of DRM. As the holiday shopping season approaches, we're planning other actions against the Swindle, as well as the Barnes and Noble Nook (better known at the “Crock”).

Software Freedom Day — At Software Freedom Day 2009 in Boston, FSF volunteer Dana Moser roved the event asking participants the following questions, while making videos of the responses: “Why do you like free software?” and “Why is software freedom important to you?” The responses came out great, but that's just the beginning. We've put out the call for FSF supporters to submit their own videos, and invite friends to do the same.

Keep up with FSF campaigns at fsf.org/campaigns and subscribe to the monthly update on all things FSF, the Free Software Supporter, fsf.org/fss. ☞

End Software Patents

*by Ciaran Riordian
End Software Patents*