Copyright © 2010
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.

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!

FREE SOFTWARE FOUNDATION Bulletin

Issue 16
May 2010

Contents

Working together for free software 1
LibrePlanet 2010 2
Freedom-friendly government policy 4
Encouraging nonprofits to work together for free software 5
End Software Patents 8
Interview: Nina Paley 9
DRM sticker contest 12
Systems update 12
Free software needs free tools 14
New releases by GNU Press 16
PayPal update 16
Assignments on the rise 17
What's in store 17
GNU network 18
Putting the "me" in membership 19
Mobile freedom 20
Much a dot about nothing 22
Gone in a flash 22

LibrePlanet shirts now available

Working together for free software

by Peter Brown
Executive Director

A few weeks ago, my six-year-old son Michael looked at my t-shirt from our LibrePlanet conference and started asking me to name each of the various characters and objects shown in the t-shirt design. These characters are the mascots of various well-known (ahem) free software projects. Shamefaced, my memory slipped on a few and I had to go look them up for him.

The symbolism of the t-shirt is reinforced by the tag line "Working Together for Free Software" and this is a theme that the Free Software Foundation is working to promote within the community—that we need to do a better job driving awareness and solidarity to the cause of software freedom.

Free software is strong because of its values and because there are many heads to the free software hydra. For every project that goes moribund another two (dozen it seems) projects rise to take its place. But all too often we see high-profile projects, that are often corporately controlled, acting in ways that hurt free software, often putting their narrow self-interest ahead of the wider adoption of free software platforms, or promoting ancillary proprietary software at the expense of other free software projects. The most common problem is the lack of effort to educate users to the values of the free software they distribute. Leaving a typical user valuing the software only because it can be acquired for little or no cost.

Our campaign for software freedom is not a campaign for freedom of choice. Free software isn't just an alternative to proprietary software. Free software is a social movement, a movement to rid the world of software that would otherwise be used to divide us and keep us powerless. The software we use is not a matter of utility or convenience, it is a matter of securing our freedom now and ever more so in a future where we become increasingly dependent on the integrity of the software we run.

In the US, we may have a Bill of Rights that prevents government from restricting free speech, free press or free assembly, but government can be ignored and these rights removed when proprietary software corporations have control over a citizen's computing.

We need to strengthen the free software movement for the long haul. The key to this is to impress software freedom values on our friends and all the people we introduce to free software. Our campaign asks free software supporters and projects to promote free software in ways that consistently emphasize everyone's right to freedom.

Working Together for Free Software means:

- Telling all users that they *deserve* to have freedom and that they should be in control of their computing.
- Promoting free software as a civil liberty, that protects citizens from government and undue influence in their lives.
- Prioritizing software development for free platforms, and to recognize that the aim is to eliminate proprietary software like any anti-social behavior.

Please join us in promoting our Working Together for Free Software campaign.

LibrePlanet 2010

by Matt Lee
Campaigns Manager

In March, the FSF hosted the second of its annual free software conferences, LibrePlanet 2010. The conference replaces the FSF's traditional associate member meetings, which ran from 2003 until 2008.

This year's conference was a huge success, surpassing the first conference on all levels. The conference was expanded into three days, with each day having two separate tracks of events. This, coupled with the increase in attendance, made the event into one of the greatest free software events of all time.

Friday kicked off with a fantastic introduction and tutorial into the GNU/Linux command line, led by former campaigns manager Joshua

Gone in a flash

by Osama Khalid
FSF campaigns intern, Spring 2010

Is Flash the only, or even the best way to share your favorite, funny video clip with your friends over Internet? Surely not. Thanks to HTML5 and pro-standard browsers for proving otherwise.

It is true that we already have a great Flash player, Gnash, but it does not move us far towards the software freedom world, because of the non-free Flash programs usually used by video-sharing websites to play videos.

But even with HTML5, it is not enough to be able to play videos without Flash if you, or your browser developer, still need a license to legally do that. It is very important for the freedom of the web to support patent-free formats to ensure that no authority controls such an essential technology.

Several proprietary browsers are pushing for the patented codecs. Internet Explorer, Safari and Chrome all do (or will soonish) support the harmful, patented H.264 codec.

The simple step we can do here is to use no Flash and no HTML5 with patent-encumbered video formats. Does that mean you can no more watch videos and share them with your friends? Of course not.

For your own watching, you should consider installing free customizing scripts such as "YouTube without Flash Auto" for Greasemonkey that can easily remove Flash and play videos using your free multimedia player within your favorite free browser. You can also download videos and watch them locally.

For sharing with friends, there is TinyOgg at `http://tinyogg.com` — a free service I created to convert YouTube videos and other Flash-based videos into Ogg Theora and Vorbis and host them. By using it, you can make sure that you do not recommend using Flash for your friends who may not have customizing scripts.

You can also help by teaching your friends and colleagues about this issue, spreading TinyOgg, joining its Python development project and participating in FSF's ongoing PlayOgg and End Software Patents campaigns.

Everyone is capable of voting for the free choice. We cannot lose this fight!

Interested in interning at the FSF? This is an opportunity to work for the organization that sponsors the GNU project, publishes the GNU General Public License (GPL), and fights for software freedom. See `fsf.org/volunteer/internships` for more information.

replacements. It's catch-up work, but our community is usually successful at such tasks. So, let's get coding on mobile!

Further reading on this topic is available at: `ebb.org/faifmobile`

Much a dot about nothing

by Peter Olson
Senior Systems Adminstrator

When a Web server has problems, we turn to the server logs to help figure out what is wrong. But a server log is a blizzard of detail and it is hard to discern the big picture, especially when low-level debugging is turned on. Recently I did this and obtained a log over 2 GB in size in less than a day.

Fortunately, the low level debugging produces messages all of which have a function name in an easy-to-recognize position, along with other explanatory text and numeric values. Since I knew nothing about how this part of the software works, I decided to make the log tell me how the program execution goes from one function to another, and what the typical path of execution is. I wrote a program to parse the debugging messages, creating a matrix showing the transitions in successive messages. I used this to generate an input file for the graphic visualization program *dot*, which among other things can diagram a state machine. I labeled the arcs with a count of how many times each transition was observed.

When I first did this, I got an enormously complex graph with lots of apparently random transitions having low transition counts. This suggested to me that simple chronological ordering was not giving a meaningful result, so I then collected transition information along with the thread identifier (present in the log message) and was pleased to see that the graph now resolved into a diagram with 17 states. This proves that the sequence of transitions is coherent within each thread independently (not surprising), something one *could* deduce by careful examination of the log itself but which is dramatically illustrated when put into graphic form.

The graph also has a property similar to an electronic circuit: for the most part the sum of the counts of the incoming arcs to a node less the counts of the outgoing arcs is zero. Occasionally you will find an arc with a higher count than expected, but this occurs because the node it is attached to is visited more than once in a typical pattern of transitions, and you can find two other paths whose counts sum to the larger one.

Some nodes have arcs which link to themselves, indicating a sequence of messages that occur within a single call to the function. I modified the program to create subgraphs for these cases, so I can see what are the typical paths occurring within each function. To do this I created node names out of the other information in the message, changing numbers to 0 and so on to discard the variable part of each message.

At press time, I have not yet found the silver bullet for this bug, but I am very pleased with the information I am getting out of the graphs.

The *dot* program is part of the package *graphviz*.[19]

[19] `http://www.graphviz.org/`

Gay and membership coordinator Deb Nicholson. This made for an excellent follow-up to the work that was done last year, to write a new manual during the conference — the book, *Introduction to the command line*, was created by Adam Hyde from FLOSS Manuals with volunteers from the community both online and in-person. This was followed by a session demonstrating some of the latest in free software Web development tools.

Following lunch, there were sessions on using the GIMP and Inkscape graphics tools, as well as hacking sessions for people interested in the development of free software for smartphones. All of this was paralleled by the first GNU Hackers Meeting in the US, with GNU veterans John Gilmore, Rob Savoye, David Sugar and Richard Stallman, as well as new maintainers from GNU Generation, GNU LibreDWG, and GNU social in attendance.

Saturday kicked off with John Gilmore's keynote "*We're done cloning Unix, what next?*", followed by insightful talks on diverse subjects such as the law, free standards and increasing participation from young people, from the likes of Karen Sandler, Louis Suarez-Potts, Max Shinn, and Steven DuBois.

No free software event would be complete without an opportunity for some keysigning — the exchange of public encryption key signatures — to allow people to communicate without surveillance over the Internet.

In the afternoon, we were pleased to have a late arrival to the schedule of none other than Eben Moglen, a familiar face to many in the free software world, and head of the Software Freedom Law Center, a law firm specializing in building understanding and awareness of free software licensing. Eben's talk gave a real sense of a job well-done and a mission complete — all the major parts of a free software system are now complete, with just a few loose ends and high-priority emerging threats as the next set of battles to be won.

This led nicely into the worldwide premiere of a new movie — *Patent Absurdity: How software patents broke the system.* Directed by Luca Lucarini and made possible by a grant from the Free Software Foundation's associate membership program, the documentary movie features interviews with Dan Ravicher, Eben Moglen, Richard Stallman, and others, illustrating some of the pitfalls that software patents have created, and daring to imagine a world in which software patents were removed from the field of play.

The movie, produced entirely with free software, also featured the animation work of Chicago's Chris Webber, with a pastiche of a familiar scene from the movie *WarGames* illustrating the problems that software patents have created for everyone.

The afternoon was rounded out with talks on two emerging graphical user interfaces. From Walter Bender, talking about his work on the Sugar graphical interface and Marina Zhurakhinskaya on her work on the next-generation interface for the GNU desktop, the GNOME shell.

The day was rounded out by Chris Hofstader, talking about his new project to increase the accessibility for the GNU Project. Accessibility is for everyone, but especially people using assistive technology, such as a screen reader, which reads Web pages, emails and other on-screen items to users who are blind or have low-vision. In the other hall, licensing stalwarts Brett Smith and Donald Robertson took a series of questions about all aspects of free software licensing in a talk dubbed “The Licensing Hoedown.”

Richard Stallman rounded out the day, with the presentation of his newest philosophy article, “Who does your server really serve?” — talking about the dangers posed by Software as a Service (SaaS).

Sunday brought us a day of talks and workshops on the issue of increasing the participation of women in free software. Currently, women comprise less than 2% of the free software community. This was led by an introduction from our own Deb Nicholson, followed by Leslie Hawthorn, discussing how best to handle free software mentoring. Later, Chris Ball, Hanna Wallach, Erinn Clark and Denise Paolucci gave us their insights into ways to recruit and retain women in free software projects. After lunch, Luis Villa gave us his look at network services from a client perspective, while lightning talks took place in Hall A. The day rounded out with talks from Chris Montgomery and Gregory Maxwell from the Theora project and a workshop on non-coding roles in free software from Selena Decklemann, followed by GNU LibreDWG and GNU Gnash giving us their own lightning talks on their respective projects.

The conference was a huge success, with over 230 people attending. Next year’s conference will have a lot to live up to.

Freedom-friendly government policy

by Brett Smith
Licensing compliance engineer

The FSF’s Compliance Lab has been involved in many different activities where free software development interacts with the law: license drafting and evaluation, crafting best practices for projects, working for compliance and enforcement, and more. In the past few months, we’ve started pushing out into a new area: advocating for free software users and developers when government sets policy. We’ve already had the opportunity to provide comments and feedback in a couple of different cases, and we’re on the lookout for more.

The first case was for the US Trade Representative’s (USTR) Special 301 Review. The Special 301 Review is a process that the USTR undertakes every year to evaluate the enactment and enforcement of copyright, patent, and trademark laws throughout the world. The office then produces a report placing countries on a Watch List—or even a Priority Watch List—if the USTR feels the laws and enforcement aren’t forceful enough.

Traditionally, the report is a huge favor to the big copyright industries from the US government. It encourages foreign countries to enact laws that are as outrageously bad for freedom as its own, including longer terms for copyright restriction, and Digital Millennium Copyright Act-like (DMCA) legislation. This year, for the first time, the USTR accepted comments from the public throughout the month of February. We sent a letter explaining how anti-circumvention

Google’s goals do not match that of the software freedom community, so in some cases, a given device will give the user more software freedom than the N900, but in many cases it will give much less.

The HTC Dream is the only such device I know of where a careful examination of the necessary proprietary components have been analyzed. There also are about twenty hardware interface libraries that do not have source code available in a public repository. However, when lined up against the N900 with Maemo, Android on the HTC Dream can be used as an operational mobile telephone and 3G Internet device using only three proprietary components: a proprietary GSM firmware, proprietary wifi firmware, and two audio interface libraries. Further proprietary components are needed if you want a working accelerometer, camera, and video codecs, as their hardware interface libraries are all proprietary.

A healthy community-oriented phone operating system project will ultimately be an essential component to software freedom on these devices — on this point, I must also mention the Neo FreeRunner device and the OpenMoko project. This was a noble experiment: a freely specified hardware platform running 100% free software. I used an OpenMoko FreeRunner myself, hoping that it would be the mobile phone our community could rally around. I do think the device and its (various) software stack(s) have a future as an experimental, hobbyist device. But, just as GNU/Linux needed to focus on x86 hardware to succeed, so must software freedom efforts in mobile systems focus on mass-market, widely used, and widely available hardware.

Fear of an FCC crack down when mobile users have software freedom is beyond the scope of this article. However, what Atheros has done with their Wifi devices shows that software freedom and FCC compliance can coexist. Furthermore, the central piece of FCC’s concern — the GSM chipset and firmware — runs on a separate processor in modern mobile devices. This is a software freedom battle for another day, but it shows that the FCC can be pacified in the meantime by keeping the GSM device a black box to the free software running on the primary processor of the device.

Seeking software freedom on mobile devices will remain a complicated endeavor for some time. Our community should utilize the free software releases from companies, but should not forget that, until viable community forks exist, software freedom on these devices exists at the whim of these companies. A traditional “get some volunteers together and write some code” approach can achieve great advancement toward community-oriented free software systems on mobile devices. Developers interested in applications should initially focus on applications for the existing mostly free platforms of MeeGo and Android/Linux. Meanwhile, the challenging and more urgent work is to replace lower-level proprietary components on these systems with free software alternatives, but admittedly needs special programming skills that aren’t easy to find.

We should be hopefully optimistic about the mobile space. There are challenges for software freedom, but they are challenges our community knows well how to face: we need to identify the proprietary software that is important, and write free software terials you need to do it right!

If you're just getting the spark, we want to help you fan the flames. As you read the stories and ideas, in our blogs, and on the LibrePlanet wiki, we hope you'll be inspired. Please contact us if you need feedback on your ideas to promote free software.

We want you to succeed so we can feature your stories next!

Mobile freedom

by Bradley Kuhn
FSF board member

The mobile telephone market has never functioned like the traditional computer market. Historically, the mobile user made arrangements with some network carrier through a long-term contract. That carrier "gave" the user a phone or discounted it as a loss-leader. Under that system, few people take their phone hardware choice all that seriously. Perhaps users pay a bit more for a slightly better phone, but generally they nearly always pick among the limited choices provided by the given carrier.

Meanwhile, Research in Motion was the first to provide corporate-slave-oriented email-enabled devices. Indeed, with the very recent focus on public-oriented devices like the iPhone, most users forget that Apple is by far not the preferred fruit for the smart phone user. Today, most people using a "smart phone" are using one given to them by their employer to chain them to their office email 24/7.

Apple, excellent at manipulating users into paying more for a product merely because it is shiny, also convinced everyone that now a phone should be paid for separately, and contracts should go even longer. The "race to mediocrity" of the phone market has ended. Phones need real features to stand out. Phones, in fact, aren't phones anymore. They are small mobile computers that can also make phone calls.

The current state of mobile software freedom

For its part, Nokia likely benefited greatly from the traditional carrier system. Most of their phones were provided relatively cheaply with contracts. Nokia sold new hardware every time a phone contract was renewed, and the carrier paid the difference between the loss-leader price and Nokia's wholesale cost. The software on the devices was simple and mostly internally developed.

In parallel, Nokia chased another market: the tablet PC. GNU/Linux remains the ideal system for these devices, and Nokia saw that. Nokia built the Debian-based Maemo system as a tablet system, with no phone. This eventually became the tablet/phone hybrid: the N900. This is among only a few available phones that make any strides toward a fully free software phone platform. Yet, the list of proprietary components required for operation remains quite long. The common joke is that you can't even charge the battery on your N900 without proprietary software.

Android/Linux is a nearly fully free non-copylefted phone operating system platform where Linux is the only GPL-licensed component essential to Android's operation. Ideally, Google wants to see it adopted broadly in both free software and mixed free/proprietary deployments.

laws like the DMCA hurt free software developers and technology users generally, and asked the USTR to stop advocating similar legislation in the Special 301 Review.

Then, in March, the newly created executive "Intellectual Property Enforcement Coordinator" heard comments on how the US government could go about stronger enforcement of copyrights, patents, and trademarks. We wrote in to say that the government would be better off focusing on adoption of free software for its own functioning. It would be the ethical choice, allowing the government to share software with its citizens and help them more actively participate in society. It would also let the government provide an unprecedented amount of transparency about its inner workings, and reallocate resources currently wasted on compliance and enforcement for proprietary software licenses.

The Special 301 Report was released in mid-May. We were disappointed but unsurprised to discover it still advocates for the same bad legislative changes as before. We won't be surprised if we hear similarly negative news from the "Intellectual Property Enforcement Coordinator," either. But Rome wasn't built in a day. We're putting policymakers on notice: these are not one-sided issues. We represent constituents who want to see policies that help free software, rather than hurt it. And that's clearly a huge group of people; our submissions rank highly in web searches about the issues. By bringing the force of that group to these public discussions, and their coverage in the media, we let the government know that their choices have negative consequences that they'll have to deal with sooner or later.

Overall, we like the results we've seen from these efforts thus far, and we plan to participate in other processes like this. Follow the Compliance Lab blog at `http://fsf.org/blogs/licensing` for more information about these activities, along with all the other work we do.

High-priority projects include replacing Flash and Skype.

Encouraging nonprofits to work together for free software

by John Sullivan
Operations Manager

It's obvious that the Free Software Foundation must use free software for its daily operations, because promoting and protecting free software is our mission. Using Microsoft Office at the FSF would be like the Nature Conservancy draining some wetlands to build its new headquarters.

But other socially oriented nonprofits, whether or not their missions relate to technology, should also be using free software. All such charities depend for their existence and effectiveness toward their respective goals on a functioning civil society, where individuals can freely communicate and associate. The more they use technology to do that communication and organize that association, the more they must be concerned with the nature of that technology. A nonprofit attempting to communicate with potential supporters and the public undermines itself when it hands veto power over those communications to a company with interests opposed to the organization's freedom or autonomy.

That veto power comes in many forms, including proprietary software copyright licenses and End User Licensing Agreements (EULAs). Such licenses and agreements often specify that any use of the software is subject to the permission and inspection of the software's developers. For an organization to give up control over technology in this way is to lose not just control over medium; it's also to lose control over message.

While it may seem like just legalese fine print, the threat posed to social change by this power continues to become more tangible. Some nonprofits have been appealing to people using iPhones and iPads, but Apple has been shamelessly dictating what applications can and cannot be run on these computers. Apple claims that it is illegal for users to install applications from anywhere other than the official Application Store — and they arbitrarily and without explanation reject and remove applications from that Store. The applications affected have often directly related to matters of expression, including political cartoons and columnists.

Companies like Amazon have provided further illustration of the problems that can arise, when they remotely deleted books from their customers' Kindle Swindle ebook readers. They claim that the power to do this stems from the EULA people agree to when they use their Kindles.

It is a short step for any other company selling proprietary software or "software as a service" to commit crimes against civil society similar to Apple's and Amazon's. Nonprofits of any kind owe it to themselves, their supporters, and civil society in general, to resist this control. They should not hand information about their supporters over to software as a service groups,[1] where that information can be more easily subpoenaed or compromised, and they should not require their supporters or themselves to agree to software licenses that give consent to searches while prohibiting them from helping themselves by installing and improving their own software.

Dependency on proprietary software also manifests in substantive ways for nonprofits, such as upgrade costs, migration pains, and lock-in. But most importantly, nonprofits need free software — software which can be studied, modified, and shared — to accomplish their missions. They need the freedom to make decisions based on the need of their missions. Free software is the only way to guarantee this freedom, and even if its cost were actually higher than the cost of proprietary software, it would be worth it.

As people knowledgeable about free software, we can all help encourage other organizations to recognize these facts and the impact of this technology. You can help by writing to other nonprofits that you donate to, and ask-

[1] gnu.org/philosophy/who-does-that-server-really-serve.html

age; peers that contribute to the network are rewarded with better service.

GNU social — `gnu.org/s/social`

GNU social development has begun — this summer, Ian Denhart and Sean Corbett from Clark University in Worcester, Massachusetts, together with volunteers from the wider free software community will show off the first public version of GNU social. GNU social represents a dramatic change in the way most social networks have worked until now — decentralized, secure communication amongst public and private servers running the GNU social stack.

Putting the "me" in membership

by Deborah Nicholson
Membership coordinator

In today's world, we use computers for correspondence, for newsgathering, and for enjoying media. Even more importantly, we have the capability to use our computers for social criticism, anonymous communication and dissent. As an FSF member, you're someone who understands that free software is more than an interest. It is a belief that empowering all people to be in full control of their computing will ultimately help to build a better world.

You're more than a name on our list of supporters, you're our proxy in your local community.

For some members that means being a free software advocate in the workplace or a usergroup organizer, for others it means being the neighborhood software expert or your household's resident geek.

Whether you're bridging the gap by designing new user interfaces or quietly converting your whole apartment building into free software users, we want to tell your story — free software is all about helping your neighbor.

In addition to sharing the software, the FSF wants to facilitate neighborly sharing of ideas and tactics for promoting free software locally.

Starting this fall, we'll be regularly featuring free software activists on our Web site. Have you recently hosted an inspiring local free software event? We'd love to publish a blog about it. Were you instrumental in getting free software adopted at your school, library or office? We'd love to interview you for `fsf.org`. Maybe this isn't you, but you know someone who's been doing great work to promote software freedom — please tell us about them!

Got something to say about bringing free software to schools or some pointers on approaching your local library? Please add it to our activist guide![18]

Have you been doing a great job of bringing women to your Python group or diversifying your school's CS program? Please tell us how you did it!

Got a precocious free software advocate in your house? We bet they'd enjoy connecting with other young free software users in GNU Generation.

Want to be a free software ambassador at an event in your area? We've reached out to folks at all kinds of events, everything from Earth Day extravaganzas to film festivals. Let us know and we can send you all the ma-

[18] http://groups.fsf.org/wiki/ActivismGuide

designs — "Live The Dream," "GNU Head," "Happy Hacking," and "LibrePlanet." The women's sizes of "Live The Dream" are available in royal blue, and "GNU Head" in lovely pink.

The sticker pack is always one of the most popular GNU gear items for showing your support of GNU, the FSF and our current campaigns against DRM and Windows 7, as well as stickers for GPLv3 and more. Each package comes with 50 stickers. To thank you for your support of the FSF, we are happy to customize a sticker pack for you — if you have a favorite sticker of ours and would like to receive more samples of it than of the others, you can send in an email to `sales@fsf.org` after you have placed your order through the online shop.

Please visit `shop.fsf.org`, the only place that you can get the unique shirts for yourself or for someone you know! Wear the special shirts around to support free software, and check out the store's new stickers, books and manuals.

We welcome any new product ideas, so please send in your suggestions for GNU Press to `sales@fsf.org`. Your support and ideas are, as always, valuable assets to us.

GNU network

by Matt Lee
Maintainer, GNU network

GNU network is a new initiative within GNU to encourage the development of free network services and web applications that are free software that can replace existing, proprietary services and create new ways for people to communicate in freedom.

Currently, three projects are part of GNU network: GNU FM, GNUnet and GNU social. Here's a rundown on each of these projects, and how it's being used. If you'd like to suggest a new project for GNU network, please write to `network-new@gnu.org` — we also have a mailing list for discussion around the creation of network services in GNU.

GNU FM — `gnu.org/s/fm`

GNU FM is a project to create both a server and user-facing component for the reporting of music listening habits. Implementing the Audioscrobbler API created by Richard Jones in 2002 and most commonly associated with Last.fm, GNU FM allows anyone to set up and run their own similar system. GNU FM is most commonly associated with the `Libre.fm` website, which is currently the largest GNU FM installation, with over 30,000 users and over 23 million recorded song listens.

GNUnet — `gnu.org/s/gnunet`

GNUnet is a framework for secure peer-to-peer networking that does not use any centralized or otherwise trusted services. A first service implemented on top of the networking layer allows anonymous censorship-resistant file-sharing. Anonymity is provided by making messages originating from a peer indistinguishable from messages that the peer is routing. All peers act as routers and use link-encrypted connections with stable bandwidth utilization to communicate with each other. GNUnet uses a simple, excess-based economic model to allocate resources. Peers in GNUnet monitor each other's behavior with respect to resource us-

ing them to publicly support free software, and to adopt an internal technology policy that commits them to using more and more free software over time.

To find the needed expertise, you can suggest to nonprofits that they advertise on the FSF Jobs Board.[2] You can offer yourself as a consultant, or point to the FSF Service Directory,[3] to further help nonprofits make the transition.

Don't be afraid to start small — getting your local neighborhood association to use free software is a victory in itself, and can lead to more change, as volunteers and staff move back and forth to other organizations. Small organizations are often very eager to accept volunteer help with their infrastructure — offering to set up a free software contact database for a local group might be a simple and effective way to introduce them to free software. You can amplify your efforts further by sharing the text of any emails or letters you write on the LibrePlanet wiki, so others can use them and improve them.[4]

When talking to nonprofits, it helps to be familiar with some free options for the kinds of software they typically need. Knowing what software the FSF uses might be a good place to start. This is not an exhaustive list; there are many more free software options in each of these categories. More can be found in the Free Software Directory.[5]

- To start with, all of our desktops and servers run the GNU/Linux operating system. For distributions to recommend, see our list of fully free distributions.[6]
- CiviCRM can take the place of proprietary fundraising programs like Raiser's Edge and software as a service products like Convio. It handles donation records, email and postal mailings, and contact management.[7]
- We do graphic design, image editing, and typesetting with Inkscape, the GNU Image Manipulation Program (GIMP), and LaTeX.[8]
- For double-entry accounting, we use SQL Ledger.[9]
- To maintain job documentation and other organizational knowledge, and to collaboratively edit text for publication, we run a couple of internal wikis, using Ikiwiki and Mediawiki.[10]
- Our online store selling t-shirts, books, and other merchandise, uses Satchmo, an e-commerce product built on the Django web framework.[11]
- We use free software Web site revision systems for our sites, including Drupal and Plone. Many other organizations use Wordpress, or Movable Type.[12]

In addition to using free software themselves, there are some basic best

[2] `fsf.org/jobs`
[3] `fsf.org/resources/service`
[4] `groups.fsf.org`
[5] `directory.fsf.org`
[6] `gnu.org/distros`
[7] `civicrm.org`
[8] See `inkscape.org`, `gimp.org`, and `latex-project.org`.
[9] `sql-ledger.org/`
[10] `ikiwiki.info` and `mediawiki.org`
[11] See `www.satchmoproject.com` and `www.djangoproject.com`.
[12] `drupal.org`, `plone.org`, `wordpress.org`, and `movabletype.org/download.html`

practices that they should follow in order to be compatible with the use of free software by their supporters:

- Web sites should be friendly to free software browsers like Icecat and Firefox, and should not require or encourage Adobe Flash or Microsoft Silverlight.
- Audio, video, and text should be distributed in free formats, like Ogg Vorbis for audio; Ogg Theora for video; and OpenDocument, PDF, HTML or plain text for documents.
- Any applications the organization might produce or distribute should be free software and run on free operating systems. This means, for example, resisting the trend of releasing iPhone and iPad-only applications, since Apple has banned free software for those platforms.

Of course nonprofits sometimes need to appeal to the technologies that people already have, in order to get their message out, but they also have a leadership role, and need to consider what kind future they are supporting.

When they do make appeals in problematic ways, they can take a minute to encourage people in those places to follow them using better methods. For example, they can provide their audio recordings in both the patent-restricted MP3 patent-free Vorbis formats, but highlight the Vorbis version. No matter what, the problematic channels and formats should not be requirements for people wanting to keep up with an organization's work.

Free software is more than capable of meeting all of a nonprofit's organizational and multimedia needs.

Encouraging everyone to see free software as fundamental to a free society, in which nonprofits pursuing a variety of charitable missions can operate with freedom, is a key part of **working together for free software**.

End Software Patents

by Ciaran O'Riordan
Director, End Software Patents

As you read this, the US Supreme Court will probably have published the *Bilski* decision and the software patents debate will be roaring, and not just in the USA. There is also software patent legislation on the table or being written in New Zealand, Australia, Israel, and other countries. How ready are we? I mean the broad "we" — we around the world who are against software patents.

We have important landmark victories. Getting the Software Patents Directive rejected in the European Union in 2005 was a big victory after seven years of work. We also have important stepping stone victories, like the recent recommendation by New Zealand's government that "computer programs [be included] among inventions that may not be patented." One common factor between these two victories is that we took par.t In many other activities, taking part seems obvious, but when it comes to politics, software enthusiasts often stop short of taking part.

A contrast is the Australian situation. A government-commissioned study concluded in 2009 that "in new areas of patenting such as software and business methods, there is strong evidence that existing [. . .] arrangements are hampering innovation." Software

ing their local bank accounts and debit/ATM cards, rather than potentially costly credit-card transactions.

The FSF recently discovered that PayPal had added a proprietary software license to its User Agreement.

Of course, the FSF couldn't agree to those terms, so as soon as we learned about them, we contacted PayPal to see if we could make other arrangements. The company listened to our concerns, and specifically excepted us from these conditions.

But not only that: next year, PayPal is also updating its user agreement to ensure that the free software community can continue to receive and make payments without having to accept a proprietary software license.

Assignments on the rise

by Donald Robertson
Copyright Administrator

As part of its support for the GNU Project, the FSF accepts copyright assignment on numerous GNU packages.

We do this in order to be in the best position to protect and promote these packages and ensure that they remain free.

Whenever a new hacker decides to contribute to one of the hundreds of packages held by the FSF, or a veteran hacker decides to try her hand at a different package, I get to help them with the process of assigning that code to the FSF. As the copyright administrator here at the FSF, I essentially get to take the pulse of the GNU Project as I watch the assignments come rolling in. I am pleased to report that the GNU Project is quite healthy, and is attracting more talented coders than ever before.

The number of contributors seeking assignment has more than doubled in the past few months, with close to 150 new assignment requests since April. There is usually an up-tick in contributions at this time of year, as students look to use their summer wisely by contributing to free software projects, but it has never been this large before. New projects like GNU social are generating a lot of interest and bringing in plenty of new faces to GNU. Even well-established packages are gaining more recruits these days, particularly younger coders, many of whom have not previously contributed.

All of this is good news for the GNU Project. It means that we are having success at recruiting the next generation of hackers to a project that has been integral to the goal of creating a free computing environment. It is a testament to the strength of GNU that so many who were born after its inception are now joining its ranks and contributing code.

What's in store

by Jasimin Huang
Operations Assistant

This spring, the GNU Press introduced the newly designed LibrePlanet t-shirt. We have seen a dramatic increase of member and nonmembers orders of this new shirt, which is now one of the most popular items in the shop. The cream-colored text with the popular free software icons are a perfect match to the brown shirt background. In case you haven't noticed, we have also introduced shirts in women's sizes in several

Further thoughts on this topic can be found in "On the savannah, where the gnu roam" at fsf.org/blogs/community/savannah

New releases by GNU Press

by Jeanne Rasata
Assistant to the president

We are pleased to announce the upcoming release of the second edition of *Free Software, Free Society: Selected Essays of Richard M. Stallman.*

This new edition features updated versions of the essays on the GNU Project and free software and, in lieu of some of Stallman's speech transcripts, incorporates many of the essays he has written since the first edition was published in 2002. A concerted effort was made to draw a distinction between the often conflated fields of copyright and patents, and new essays relating to the latter will, we hope, define the issues and help clear up the confusion.

To make the licenses more palatable and accessible to the uninitiated, we've included an introduction — which provides some historical background and context and explains the significance of the documents — and an essay on why projects should upgrade to version 3 of the GNU General Public License. A more prominent place has been given to the importance of using the correct nomenclature; a part of the book focuses on the issues of vocabulary and the unfortunate consequences of not referring to something by its right name.

Finally, the last two parts inform us on some of the types of traps we as users face — showing the insidious ways in which these traps can erode our freedoms — and assess the situation and invite us to be more proactive and to look beyond convenience to civic values, keeping the ideals of freedom, community, and neighborliness in the foreground.

In tandem with this second edition, GNU Press will be releasing a second edition of *Free as in Freedom,* Sam Williams' biography of Richard Stallman. Proceeds from the sale of the books will help fund our campaigns to promote and defend computer users' rights. We hope that you will download or buy the books to help spread the word and support free software.

Richard Stallman continues to further the cause of free software, speaking to audiences worldwide: in June, he will give speeches in the Balkans, Lebanon, Germany, and Spain; in July, he will be in Venezuela and the US; the following month, he will visit China, India, and New Zealand; and, so far, trips to Australia, Armenia, and Switzerland are planned for the fall.

Please help spread the word about his speeches by keeping an eye on upcoming speeches at `fsf.org/events/` as well as sharing it with your friends and colleagues.

Please let us know if you would like Richard to visit your city or area.

PayPal update

The FSF now accepts associate membership payment via monthly PayPal payments!

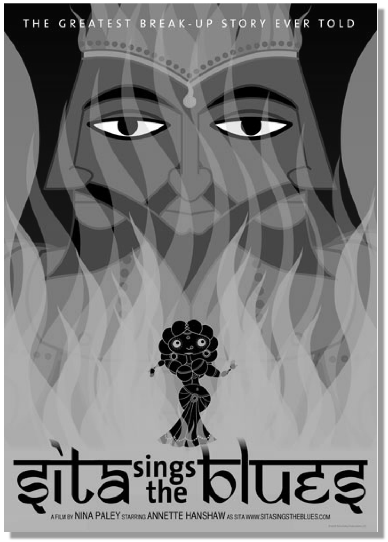
This payment method is ideal for non-US members, who can pay us-

is currently very patentable in Australia, so this is a clear criticism of software patents. However, when the government then held a public consultation on what should be patentable, the only software organization that replied was Microsoft. "We" didn't take part, and the government's legislative proposal based on this consultation will now likely be an uphill battle for us.

The participation in New Zealand was that a half-dozen letters were sent in response to a public consultation, and some of those letter writers also turned up to oral hearings to explain their letters. It's a surprisingly manageable amount of work for such a solid victory. The EU victory involved massive work, including protests on the streets, but it was also based largely on writing letters and meeting the politicians. There's a reason that interacting with politicians on this issue works: we have studies and other evidence to back up our claims.

One of End Software Patents' main projects is the `en.swpat.org` wiki, where you can find lists of studies showing that software patents harm the economy and innovation, and you can find lists of examples where software patents were used to destroy software projects, or to block software projects from having features which users require.

I've interacted with campaigns against software patents from more than twenty countries, and the free software community has played a key role in every one of them. For post-*Bilski*, my advice is simply to participate. Write to politicians, quoting studies and real world examples, and follow up with a phone call. This is something we've proven we can do well, and it's a powerful tool, but it can slip through our fingers if we each leave it to someone else. You'll find most or all of what you need on `en.swpat.org`. That wiki is publicly editable and your contributions will in turn help others.


THE GREATEST BREAK-UP STORY EVER TOLD
sita sings the blues
A FILM BY NINA PALEY STARRING ANNETTE HANSHAW AS SITA WWW.SITASINGSTHEBLUES.COM

Sita Sings The Blues by Nina Paley

Interview: Nina Paley

by Adrin Yanes Martnez
DRM Elimination Crew

Nina Paley is an American cartoonist, animator and free culture activist. She directed the animated feature film *Sita Sings the Blues.* She was the artist and often the writer of comic strips *Nina's Adventures* and *Fluff*, but most of her recent work has been in animation. Her early short films include *Fetch!*, *The Stork*, and *The Wit & Wisdom of Cancer.*

Interview questions were asked by Adrin Yanes Martnez.

What is the motivation to spread your works?

Cultural works have more value the more they are shared. Consider languages: the more people speak English, the more want to learn it, and the more people need to have it. That's cultural value. I need as many people as possible to see my work for it to have cultural value. This value becomes evident when people quote the work, or share it, or build on it, or talk about it. Parodies are a great sign of cultural value.

What kind of freedom is the film industry providing to authors? Are the authors' choices respected?

It depends what you mean by the "film industry." Hollywood has no place for someone like me; big studios are attached to increasingly obsolete business models. I don't consider that oppressive; I just stay out of that system. Today the tools of production are so affordable, artists can make films on their own, without needing a big studio. The less dependent the artist is on studios, investors, and institutions, the more freedom they have.

Digital Restrictions Management (DRM): necessity or paranoia?

It's certainly not a necessity. You can charge money for copies without putting DRM on them. You can charge money for a streaming service without DRM. DRM has absolutely nothing to do with whether you can charge money for something. DRM-free products are more useful and more valuable than DRM-encumbered ones, so logically you could charge more for them. But the mainstream entertainment industry doesn't offer such products, so their only competition is "pirates."[13] Lots of people would be happy to pay for non-DRMed products. They're getting pretty sick of paying for authorized products that are inferior to illegal versions.

What is the best choice for the artist that wants to publish without DRM?

There's still a vast and free Internet that can distribute non-DRMed works. But the more popular outlets insist on DRM, the more difficult it becomes for artists. I really want *Sita* in as many mainstream media channels as possible. I'd love the film to be on Netflix's video-on-demand system, but they currently offer no non-DRM option. There are some streaming services that don't use DRM, but of course they're smaller, because the big studios won't license their content on them. I worry that more devices will be designed to only play compatible DRMed content, making it impossible for free works to play on them. DVD is kind of like that; it's a crappy codec, certainly "defective by design," but for years it's been the most widespread standard of content delivery. I want people to see *Sita*, so I offer DVDs, but it pains me when much better video codecs are available.

Because *Sita* is available in multiple formats at `archive.org`, people who are willing to spend the time downloading high quality versions may do so. My hope is that enough people care about quality to develop easy-to-use delivery channels without DRM.

What are your words for DRM advocates?

No DRM for me, thanks.

[13] See `gnu.org/philosophy/words-to-avoid.html`

ware development, but this commitment does not extend to their own software that runs the development platforms. The source code to each of these systems remains private and unmodifiable by the developers using the services.

These nonfree development tools present a dilemma for many free software developers. The goal of many of these tools is, through more efficient free software development, more free software and more freedom. CollabNet, Google and GitHub each claim to want free software to succeed and claim they want to help it. For a series of reasons though these companies choose to support software freedom through means that are less in line with free software ethics than the the ones they seek to create. The result is developers who are disempowered. The software freedom of the code these hackers produce is contingent on unacceptable exclusivity.

First, the use of nonfree tools sends an unacceptable message to users of the free software produced. "Software freedom is important for you as users," developers seem to say, "but not for us." — such behavior undermines the basic effectiveness of the strong ethical commitment at the heart of the free software movement. As those that are already committed to free software, we should demonstrate that we can succeed — and thrive — using free software. We should support free alternatives to proprietary systems such as Savane which can replace SourceForge or Google Code and runs GNU Savannah, or Gitorious which can replace GitHub — by using them and by improving them in the areas where they fall short.

Secondly, we should realize that, going forward, the software we produce is only as free as the software it depends on for its continued use, distribution, and evolution.

The GNU GPL license and source code mean little to a user attempting to modify a program without free access to the software required to make that modification. Is is not only developers' freedom at stake but, eventually, their users and all future "downstream" developers as well. Those choosing to use nonfree tools put everyone at the whim of the groups and individuals who produce the tools they depend on.

While proprietary development tools may help free software developers create more free software in the short term, it is at an unacceptable cost. In the controversial area of private software and network services, free software developers should err on the side of "too much" freedom. To compromise our principles in attempts to achieve more freedom is self-defeating, unstable, and ultimately unfair, to our users and to the larger free software development community.

Just as the early GNU maintainers first focused on creating free tools for creating free software, we should ensure that we can produce software freely and using unambiguously free tools. Our failure to do so will result in software that is, indirectly, less free. We should resist using tools that do not allow us the freedoms we are trying to provide our users in the development of *their* software and we should apply pressure on the producers of our development tools. Free software has not achieved success by compromising our principles. We will not be well served, technically, pragmatically, or ethically, by compromising on freedom of the tools we use to build a free world.

Systems administration automation

We have started rolling out Puppet[17] to manage our ever-growing list of (virtual) servers. Puppet is a tool to keep machine configuration consistent. It allows systems administrators to manage more machines with less effort, because a lot of the work can be automated with Puppet "recipes." Puppet also guarantees that machine configs stay in line with the configured recipes, which provides peace of mind for overworked sysadmins.

Puppet can also drive auto generation of configuration files for our monitoring software, which is something we plan on doing.

Free software needs free tools

by Benjamin Mako Hill
FSF board member

Over the last decade, free software developers have been repeatedly tempted by development tools that offer the ability to build free software more efficiently or powerfully.

The only cost, we are told, is that the tools themselves are nonfree or run as network services with code we cannot see, copy, or run ourselves. In their decisions to use these tools and services — services such as BitKeeper, SourceForge, Google Code and GitHub — free software developers have made "ends-justify-the-means" decisions that trade away the freedom of both their developer communities and their users. These decisions to embrace nonfree and private development tools undermine our credibility in advocating for software freedom and compromise our freedom, and that of our users, in ways that we should reject.

In 2002, Linus Torvalds announced that the kernel Linux would move to the "BitKeeper" distributed version control system (DVCS). While the decision generated much alarm and debate, BitKeeper allowed kernel developers to work in a distributed fashion in a way that, at the time, was unsupported by free software tools — some Linux developers decided that benefits were worth the trade-off in developers' freedom. Three years later the skeptics were vindicated when BitKeeper's owner, Larry McVoy, revoked several core kernel developers' gratis licenses to BitKeeper after Andrew Tridgell attempted to write a free replacement for BitKeeper. Kernel developers were forced to write their own free software replacement: the project now known as Git.

Of course, free software's relationships to nonfree development tools is much larger than BitKeeper. The source to the free software development support service SourceForge was once available to its users but its authors have returned to a completely closed model. While SourceForge is built using free software, SourceForge users interact with the software over the web. Because users never have any copy of the SourceForge software, they can never demand source. Similar projects like CollabNet's Tigris.org, Google Code's "Open Source Project Hosting" services, and GitHub, each served similar purposes and have kept their code similarly out of reach. Their services are often provided without charge and promoted for free soft-

[17] `puppetlabs.com`

Difficult question: What is the direction of the industry with respect to DRM? Is the industry listening to their clients?

Remember I'm not part of the mainstream entertainment industry, so when you ask these questions about "the industry" I can only answer as a relative outsider.

Hollywood seems pretty wedded to DRM. They won't have any "clients" that don't apply DRM. So they'll continue to build a their own all-DRM world, and maybe sue fans who obtain higher-quality media illegally. They'll keep pushing for more draconian "Intellectual Property" laws.[14]

DRM wouldn't really be a problem if it weren't for the Digital Millennium Copyright Act (DMCA). Without bad laws, DRM would never survive "the discipline of the market." That's the real problem: bad laws. All DRM can be broken, but if it's illegal to break it, or even to help develop software that breaks it, then you get a nation of criminals. It's scary.

What is your opinion on companies who use DRM to say what you can install on your computer and how you can install it?

People are forgetting that computers are machines built to serve US. You buy a computer, it should work for you. Instead, it works for Sony, Disney, Warner, Viacom, and other corporations. We're letting these corporations spy on us and control our machines. Many people surrender their autonomy and property far too easily. They think that the price of entertainment is not just the money they pay to see it, but their privacy and freedom too. That's sad. Maybe they don't feel worthy to actually own their computers. But I paid for mine; I own it, and I'm not OK with a handful of corporations trying to take away my authority over my own property.

In this way I am a "propertarian" — it's just that culture isn't property. Scarce goods can be property. Non-scarce goods, like language, culture, and information, cannot. And before some dork says, "Oh, then your bank account number isn't property," let me say indeed numbers (non-scarce) are not property, but the money in my bank account (scarce) is; my bank account number is not property, but it is PRIVATE. That's why I don't publish it. That's why I don't even charge tickets to see it, or sell it on a pay-per-view channel with DRM. People conflate copying non-scarce goods with fraud, which are quite different things. In fact I'm sure my bank account number shows up in equations, computations, maybe even in textbooks; should I look for all instances of that number and sue? No, because without a claim of identity associated with me and my bank account, it's just a number. Copying the number 449-36-2971 (which I just made up, but it could be a Social Security number) is not a crime, or immoral, or harmful; using it to LIE is.

But my computer, which I paid for? Hell yes! That is my property. And no one should be able to decide what I do with it, but me. It has "natural" limitations of course; that it can't make a live unicorn is not due to anyone taking away my rights. But considering all the wonderful things computers can do, intentionally crippling them, designing defects into them that serve nothing but an obsolete business model, that seems immoral. Tak-

[14] See `gnu.org/philosophy/not-ipr.html`.

ing something good and making it bad, something healthy and making it sick, something functional and making it dysfunctional, something beautiful and making it ugly — that crosses me on a deep level. It's one reason I avoid freelance work, because so often I'd produce something beautiful and then be paid to make it worse and worse.

I guess that answers your earlier question about what an artist is. An artist works in the service of beauty, quality, functionality. An artist chooses these over clients, or money, when there's such a choice to be made.

Finally: What do you say to people who are buying works under DRM because "they have no alternatives"?

I don't hold it against anyone. It sucks that most alternatives are illegal. The market is really broken because of the DMCA and information monopolies. If the market were free and functioning, there would be legal alternatives to inferior DRMed copies, and DRM would go away.

I don't think fans should be held responsible for DRM. Fans are forgiving, and generous with their attention; they're willing to go to great lengths to enjoy works they love, including putting up with DRM. I would focus on artists and authors.

Artists, I urge you to respect your fans. Make your work available so your fans can enjoy high quality without breaking the law.

That's why I paid through the nose to clear those godawful song licenses in *Sita Sings the Blues*. I respected my fans enough to not ask them to break the law to enjoy my work. Fans are the lifeblood of cultural works, and the main support of artists. Being good to fans means releasing work without DRM.

DRM sticker contest

by Matt Lee
DRM Elimination Crew

With the release of the iBad, Apple's latest restriction, and the recent furor over their new developer licensing agreement, it occurred to me that our anti-DRM sticker needed an update.

So, back in April, we quietly announced a contest for the design of a new one. I'm pleased to announce that Jeremy Todaro is the winner of our Defective by Design sticker contest for his accurate portrail of Steve Jobs as Big Brother. Jeremy is a freelance artist from Wentzville, Missouri who specializes in using free software tools for his work. Well done Jeremy!

Runner-up prizes will also go to Andreas Marschke, Valessio S. Brito, Diego Trujillo and William Demchick for their contributions.

Systems update

by Ward Vandewege
Senior Systems Adminstrator

As you may know, the FSF's Web site runs on Zope and Plone. Last year, we split our Zope/Plone instance into two separate instances — one for the public Web site, and one for the membership area. The membership area runs on top of a membership management application that we have developed internally. We have recently decided to start migrating away from this application, and are in the middle of planning a migration to Drupal and CiviCRM. [15]

You'll see why this year WILL be like "1984"
DefectiveByDesign.org

Jeremy Todaro's winning sticker depicts Apple CEO Steve Jobs talking to a crowd of drones using iBads, mimicking Apple's infamous "1984" commercial.

We currently have no plans to move the FSF Web site away from Zope and Plone.

GNU mail update

The sysadmins have also spent some time on the gnu.org e-mail infrastructure. The way gnu.org e-mail is routed between our various mailservers is a bit exotic, to put it kindly.

In February, we introduced a new primary mail gateway, which will over time replace the ageing monty-python.gnu.org. The new gateway is called 'eggs.gnu.org', and it lives at our colocation facility. Monty-python and lists are still at the FSF offices — behind a high-speed T1 internet connection — which is something we will address in the next few months. The plan is to retire monty-python altogether, and move lists to new hardware, at our co-location facility just outside of Boston.

This will improve mail processing times for the hundreds of mailing lists we host.

coreboot update

Coreboot is a free software project aimed at replacing the proprietary BIOS (firmware) you can find in most of today's computers. In many cases the BIOS is the only thing standing in the way of a person running their system using exclusively free software.

The FSF sysadmins have not had much time to contribute to the coreboot project lately, but exciting things are happening.

Another laptop is now supported — the Getac P470 — thanks to the hard work of coresystems GmbH.[16]

This work done under contract by the German equivalent of the Department of Defense. Clearly, the German government understands the security risks of running proprietary BIOS software.

In other news, AMD contributed support for a few new chipsets (AMD RS780 / SB700) which means that once again, there are quite a few desktop motherboards for sale that could easily be ported to coreboot. Coreboot will have 3 or 4 Google summer of code students this year, and a mass-porting effort is planned for a number of motherboards based on the AMD RS780/SB700 chipsets. Also as part of GsoC, a USB 3.0 software stack will be added to coreboot.

[15] `fsf.org/news/nonprofit-fundraising-civicrm`

[16] `coreboot.org/pipermail/coreboot-announce/2010-May/000007.html`