

Happy 25, GNU!

Contents

GNU is 25!	1
The Wikipedia Naming Controversy	2
The Last Mile is Always the Hardest	4
The Free Software Jobs Page	6
Volunteer Spotlight: Yavor Doganov	7
End Software Patents	9
Free Software Supporter	11

ries of short videos about GNU and free software. The first video, which will be airing on gnu.org to coincide with the anniversary, explains free software to a new generation of people, and encourages them to download and try gNewSense. These videos, coupled with a fresh new look for gnu.org and upcoming membership events in Portland and San Francisco, promise to make the 25th anniversary something really special.

You can help too! We are looking for any relics of GNU and FSF history that you may have lying around. If they're something you can email, feel free to send them to campaigns@fsf.org, and if they're something a little more tangible, drop us a line and if they're something of interest, we'd love to see them.

The most important thing of course is that GNU continues to grow, and continues to stand up for freedom in a potentially hostile world. Twenty-five years ago, the threats to our freedom came from proprietary operating systems. Thankfully these days, we have a few completely free distributions of the GNU operating system, but we must not become complacent. Proprietary software in the form of popular programs like Flash and Skype are constantly seducing many in our community to use proprietary software, and we must stand up to it,

GNU is 25!

by *Matt Lee*
Campaigns Manager

After some difficult developmental years, this September marks the 25th birthday of GNU, and we're going to be celebrating in style. We spent most of May working in close contact with a well-known celebrity, shooting the first of what we hope will be a se-

first by installing and using a distribution of GNU that values freedom, such as gNewSense, but also by encouraging others to do so, even if it is at the cost of some convenience.

Our past experience gives us every reason to believe that we will succeed, but it's going to take a lot of work.

Here's to another twenty-five, GNU!



The bi-annual free software related tales of Attitude Cat (and his friend Simon) by Matt Lee & Rob Myers



The Wikipedia Naming Controversy

by Joshua Gay
Campaigns Manager

On the English version of Wikipedia there is an article that discusses the naming controversy over whether one should call the operating system “GNU/Linux” or “Linux.” In that article, some contend that Linux is the more popular and common name for the system. But when writing an encyclopedia, neither popularity nor com-

monality are the paramount concerns. Calling the system “GNU/Linux” is more factually accurate, as the GNU project largely forms the base of all distributions of the operating system. For example, GNU packages accounts for 14.79% of the 16.5GB of source packages used to build the Main repository of the gNewSense GNU/Linux distribution (deltad). They also constitute 6.69% of the 27GBs of source packages from which the Universe repository is built. Linux weighs in at about 253MB and accounts for approximately 1.5% of the source code needed to build the Main repository. Furthermore, Linux itself is generally built using GNU libraries and GNU tools, and on many systems depends on them being there.

However, it's not just a matter of accuracy as it relates to lines of code. It's about the motivation and goals that got those lines written. By mentioning GNU, you are foregrounding the ethical commitment its hackers have to free software. The foregrounding of these principles is exactly the reason why some would prefer we elide GNU. While the code for the kernel Linux is distributed as free software under the GPL, the term “Linux” when applied to the whole operating system is often used as a branding tactic by companies to reduce the visibility of the ethical aspect of free software.

It should be understood we are not talking about a single operating system but rather a very large class of operating systems, all of which have at their core the Linux kernel and a suite of libraries, programs, and utilities from the GNU operating system. All distributions of this operating system contain software from outside the GNU project and the Linux kernel. Further-

Free Software Supporter

As many of you know we recently rolled out the Free Software Supporter, the Free Software Foundation's monthly news digest and action update. Over the past six months over 6,000 people have subscribed to the list (info-fsf@gnu.org) and are receiving the newsletter. You can subscribe and read back issues at www.fsf.org/free-software-supporter. Special thanks go to Karl Berry for writing the GNU Spotlight each month. 🐾



Visit shop.fsf.org.

This bulletin was produced using only free software: LaTeX, Emacs, Inkscape, GIMP, Imagemagick and Ghostview.

Copyright © 2008
Free Software Foundation
51 Franklin Street, 5th Floor
Boston, MA 02110-1301
(617)542-5942
info@fsf.org

Verbatim copying and distribution of this entire bulletin is permitted in any medium, provided this notice is preserved.

How to Contribute

Associate Membership: Become a “card-carrying” associate member of the FSF. Benefits include a copy of Richard Stallman's book, *Free Software Free Society* or an FSF USB Key, plus a bootable membership 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.

Phone: You can also make a credit card contribution by calling us at +1-617-542-5942.

United Way: As a 501(c)(3) tax-exempt organization, the FSF is eligible to receive United Way funds. On the donor form, check the “Specific Requests” box and include the sentence, “Send my gift to the Free Software Foundation, 51 Franklin Street, 5th Floor, Boston, MA 02110”.

Buy GNU Gear: Order one of our T-shirts or manuals at shop.fsf.org. You can even get a copy of *Free Software Free Society* signed by Richard Stallman!

Volunteer: See fsf.org/volunteer for ideas.

Contact donate@fsf.org for more information on supporting the FSF.

abstract algorithm onto a computer is a loophole and not a real means of making an abstract concept physical—and was granted leave to present oral argument to the en banc hearing.

A joking title for this article could have read “Microsoft briefs court to End Software Patents (sort of).” Microsoft’s brief to the court said, “. . . a patent should not be granted under 103 if the inventor combined well known computer hardware with inventive but otherwise unpatentable software.” The brief is written in a confused manner, describing how one could possibly eliminate certain types of software patents, if found worthy of elimination, via Section 103 of the Patent Act. This confusion seems to mirror the business situation they find themselves in, where they want to use software patents to threaten free software developers but at the same time they are one of the largest targets for software patent litigation.

In reaction to the hearing, ESP director Ben Klemens said, “We can be relatively optimistic about the odds that the courts will actually eliminate patents on intangibles like software. There will be twelve judges hearing *Bilski*’s case, and only five of them are guilty of bringing us into this mess, and all twelve are aware that the Supreme Court will overturn a ruling that doesn’t do enough.”

To date, ESP has been focused on bringing about a judicial review, because we have believed that it is the most likely path to change. When we first analyzed the strategic course for the project, Klemens predicted that the courts were our only realistic hope for directly addressing what is patentable subject matter, and I’m happy to say that he was right

— though on a significantly shorter timetable than we had planned for. Now, in filing the ESP brief to the court and as we shift our strategic focus to oppose other legislative action that might impose software patents, Klemens has decided to step down as director of the project. Klemens writes:

I always thought of the options for reform of patent law along two threads: roughly, the legislative or judicial. The approaches are very different, involve very different campaigns and campaign activities, and require different skills for implementation. I primarily talk about the law and economics of the matter, but I really do think that these patents are unethical, and want to see them eliminated. Working with the FSF has been fun and I would of course feel great if I had a major hand in saving the day in the end. But it’s important to distinguish between what I enjoy doing and what will work best for a campaign aimed at the public and legislature.

Klemens will continue to advise the FSF as we await the Court’s ruling and move ahead with the campaign. 🙄

more, the name of this system is not written in stone — for any given distribution you can feel free to rename it and redistribute it under any name you choose. For example, I can call it the Josh kernel, the Josh Project, and distribute JoshOS. However, I don’t want to name the system after myself — I’d want people know that they are getting GNU.

When the name is GNU, you should hear, “This system exists because of people who care about freedom. Join us, value your freedom, and together we can preserve it.”¹ We will often refer to Linux in conjunction with GNU, because without it, the GNU operating system would be unable to run on thousands of different hardware platforms. However, the Linux kernel project itself has not made a full commitment to freedom. They have included proprietary software in their project, so distributions such as blag² and gNewSense make sure that there exist versions modified to remove the proprietary blobs.

However, even if Linux were to ship without blobs, GNU/Linux distributions should still mention GNU. The fact is, there exist distributions of GNU/Linux that even contain full proprietary applications. The extent to which they can do this is severely tempered by the significance of the GNU name. The name is inseparable from the ethical motivations behind free software development, so anyone trying to sell you proprietary software is going to do their best to keep that quiet.

Sun Microsystems, a company that

¹This quote was taken from the essay *Linux, GNU, and Freedom*, <http://www.gnu.org/philosophy/linux-gnu-freedom.html>

²www.blagblagblag.org

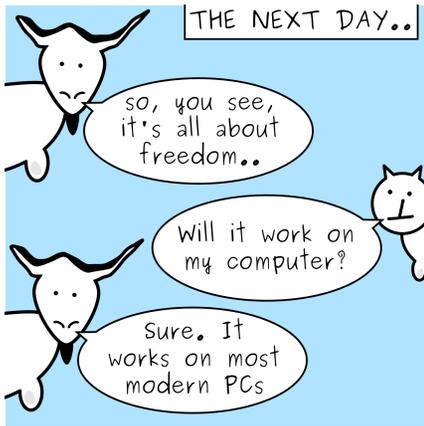
has made an increasing commitment to free software over the years will regularly make statements such as, “Sun’s GNU/Linux Offerings,” or “Sun brings a comprehensive systems approach to GNU/Linux-based operating systems.” However, many of Sun’s partners that distribute “leading, branded GNU/Linux operating systems,” suppress the GNU when “branding” their distributions.

Those companies that suppress the GNU name from their distributions are some of the worst offenders in not only distributing proprietary software, but also openly developing, promoting, and encouraging its proliferation. In many ways, these companies are hijacking the free software movement for their own gain, and their suppression of GNU is just one way of distracting people from the fact that they are unwilling to make an outright commitment to free software.

The marketing tactics of such companies often results in people adopting the same language habits, unaware of all that is at play. Even worse, some who know better will actually use the marketing language as a justification for suppressing GNU because it is “common.” I hope that the editors of Wikipedia currently engaged in this debate will stick to Wikipedia’s principles and refrain from engaging in such marketing tactics, and will refer to GNU when talking about the class of operating systems that are built with the GNU system.

I’d like to encourage all of our supporters and readers out there to work hard to combat such tactics by mentioning GNU when you see others avoiding or suppressing it. Let them know that every GNU project is guaranteed to carry freedom to the user

— freedom to run it for any purpose, share it with neighbors, improve it for your own purposes, and modify and redistribute your modifications for the benefit of the whole community. As we reach the 25th anniversary of the GNU project, I'd like to thank GNU and the thousands of free software developers and supporters, past and present. And I'd like to encourage everyone else to show their support too, by giving credit where credit is due and saying GNU! ♡



The Last Mile is Always the Hardest

by John Sullivan
Operations Manager

Software licenses and patents aren't the only ways proprietary software vendors can restrict the freedoms of computer users. A key component of our overall campaign for software freedom is making sure that the hardware we buy is not an obstacle to the exercise of our freedoms. We've made it

almost all the way to easily available fully free systems, but it's still going to take some work to finish this last mile.

When purchasing a computer, free software supporters don't want to buy a proprietary operating system license, like Windows Vista or Mac OS X. We can always reject the license, removing the proprietary operating system and installing a free one — but this is a nuisance and puts money in the pockets of people who work against our freedom. Sometimes a refund can (and should!) be obtained for the cost of the proprietary license — but this is another nuisance and doesn't communicate a clear signal to the vendor, who remains under the impression that they sold a copy of Windows or Mac OS X.

Because computers are expected to be sold and used with proprietary operating systems in this way, free software users can end up with computers that aren't well-supported by the free software they want to run. In particular they can end up with network and video cards that require proprietary drivers. This is a problem even with companies like Dell and Lenovo that are now selling systems preinstalled with GNU/Linux. Instead of choosing hardware that does not require proprietary drivers, they have chosen to just provide the proprietary drivers.

Fortunately, we've made progress in this area. Los Alamos Computers³ has been working in cooperation with the FSF to offer systems that come preinstalled with a free operating system like gNewSense⁴ and fully working hardware. This is exciting by itself, but they are also donating a portion of

³<http://laclinux.com/gnu>

⁴<http://gnewsense.org>

it has helped me a lot in my decision to entirely reject non-free software some years ago. ♡



Stream Ogg Vorbis on WBUR.org at www.wbur.org/listen/feed/ogg.m3u. Read the blog post and letter to WBUR at www.fsf.org/blogs/wbur-playogg

End Software Patents

by Peter Brown
Executive Director

On Thursday May 8, 2008, the United States Court of Appeals for the Federal Circuit (CAFC), in an en banc hearing listened to oral arguments in *In Re Bilski*. The Bilski case is where the legal battle over software patents in the U.S. is currently being fought, and is particularly important because the court has specifically requested guidance on the scope of patentable subject matter. This “en banc” hearing is a very rare occurrence; it means all twelve judges of the court participate in the hearing and will consent or dissent with the majority ruling.

The case itself is about the validity of a patent for a business method for managing risk costs of commodity selling. The patent is held by Bernie Bilski. Business method patents have the same essential flaw that software

patents have—they are little more than algorithms that have long been held not to be patentable subject matter.

The en banc hearing was prompted by a series of rulings from the Supreme Court that have overturned decisions made by the lower CAFC court, calling into question CAFC's competence. The problem has always been that the CAFC was composed of patent lawyers with a vested interest in expanding the scope of patent law. Now the question is, can they fix the mess they've made?

The FSF-backed End Software Patents (ESP) project filed an amicus curiae brief to the court.¹⁰ The ESP brief recommends re-establishing the Supreme Court's rule that information should not be patentable, even when claimed in tandem with a physical afterthought. The brief explains that it is widely recognized in U.S. law that pure information is not patentable. Further, the Supreme Court has ruled three times that pure information does not necessarily become patentable when recited in combination with a physical object, such as information written to paper or loaded into a computer's memory. The brief notes that the appeals court of the Federal Circuit ignored the Supreme Court's repeated rulings, and began allowing patents on information plus any physical component: a formula, if saved to a computer's hard drive; a price list, if money is eventually moved; not a correlation, but the act of correlating.

Most interestingly, the financial services industry lead by Bank Of America briefed the court in line with our arguments—that just loading an

¹⁰<http://endsoftpatents.org/bilski>

ion, in the other camp. Translating programs is more or less a monotone activity, and people usually burn out after 3-5 years. Translating documentation is substantially harder. Translating the GNU philosophy is the hardest thing ever for a translator, because it is a difficult activity that requires a lot of thought, understanding and responsibility. Awareness about the ethical issues that the free software movement set out to solve is still very low, which leads to the lower number of translators contributing to `gnu.org` compared to, say, GNOME.

Matt: Have you looked at HTML 5 at all?

Yavor: No, I dislike anything related to HTML and read only what is necessary to complete the tasks I'm doing. Perhaps even less. I find it ironic that W3C develops standard after standard, each determined to solve entirely all problems. And we are doomed to maintain all standards forever, since no one will use a browser that cannot render a page in HTML 2.0. The only thing I know about HTML 5 is that the committee rejected Ogg Vorbis/Theora which (provided it's true) doesn't make me feel excited and filled with hope.

Matt: If someone wanted to get involved with translating `gnu.org`, what is the work like? What kind of skills do they need?

Yavor: Well, the work is enormous given the fact of how many articles are available to translate. Even if we count only the essays, and only the most important essays, it is still a titanic effort for a new language. But don't despair, GNU was not built in one day! If the amount of work had scared away the countless number of GNU maintainers and contributors, we wouldn't have our

beloved free system today.

A translator should read and understand English well and have good knowledge about our philosophy and the various issues raised and discussed in the articles. It is not a requirement to be a fluent speaker — in fact, many translators cannot speak English well but they understand it perfectly when reading. That's the important part. Of course, time is always necessary. A relatively short essay might look like a job for an hour or two, but it often turns out to take days or weeks. Sometimes it is necessary to do some research, or discuss problematic phrases, and this takes time.

Translation teams play an important role by establishing the most appropriate terminology for the specific language and of course by presenting the philosophy of the free software movement to readers who do not understand English (and very often, to those who understand English but would not bother to read an essay in English).

The most valuable "feature" a translator could have is her dedication, and her determination to keep going; her firm belief that translating these essays is a job that is useful for the society. Everything else, like small technical skills, the usual translator's "sense", etc., will get settled by itself, one way or another.

Matt: And finally, what's your favorite piece of free software?

Yavor: GNU Emacs, undoubtedly. Emacs is special in so many ways that it is impossible to explain its nature to someone who has not seen/used it. It is most definitely one of the Wonders of the World but unlike them it keeps surprising you every day. I use Emacs for absolutely everything; I also believe

those sales to the FSF to further support free software. We will be working with more vendors to follow their example.

Wireless networking has in the past been a real headache for people who want to have a fully free system. There are a number of cards that are supported under GNU/Linux and they often have drivers released under the GPL or another free software license — but those drivers depend on chunks of proprietary binary code. To avoid these binary blobs, users have thus far been limited to a handful of chipsets, primarily those manufactured by Ralink using the `rt2500` driver.

We now have an additional option in the `ath5k` driver, which is descended from `Madwifi`, `OpenHAL`, and `OpenBSD's ar5k`. This driver supports several Atheros wireless cards without requiring any binary blobs, and is included in Linux as of version 2.6.25. The Software Freedom Law Center reviewed the driver⁶ in September 2007 and verified that it is free.

In the world of video drivers, 3D acceleration has long been a sore spot for anyone not using Intel hardware. But late last year ATI announced that it would be releasing code and specifications to assist the community in development of fully capable free software drivers for all of its newer Radeon chipsets. Based on steps they have already taken, it does appear that they intend to follow through on this commitment. VIA very recently made a similar announcement, but has yet to take action. Widespread free software drivers supporting 3D acceleration will undoubtedly help free software games

⁵<http://madwifi.org/wiki/About/ath5k>

⁶<http://www.softwarefreedom.org/resources/2007/ath5k-code-analysis.html>

and graphics applications, which have been notable weak spots in the past.

Drivers aren't the only remaining concern — other than the One Laptop Per Child XO (which unfortunately has a proprietary wireless driver), there are still no laptops or desktops readily available with a free BIOS. With the help of a grant from the Mozilla Foundation, the FSF has continued working to help promote and support projects like `coreboot`⁷, which provide a free software BIOS replacement. Thanks to the work of FSF sysadmin and `coreboot` contributor Ward Vandewege, we now have free BIOS desktops alongside the free BIOS servers running in the FSF office. Silicon Mechanics has also started selling a server pre-flashed with `coreboot`, and we are hopeful that other vendors will follow their lead. You can help this campaign by sending positive feedback to companies making such moves, and by being critical of companies like Intel that continue obstructing progress.⁸

There is a lot to keep up with, but the FSF hardware database⁹ has been expanding as a resource to consult before purchasing a system or accessories. You can find valuable information there about which exact chipsets and models are known to work with fully free GNU/Linux systems. The information comes from the testing we do at the FSF and from people around the world who send us their working hardware configurations. We can always use more volunteers to help us process the information we receive. If you have some time and would like to help by maintaining a sec-

⁷<http://coreboot.org>

⁸<http://www.fsf.org/campaigns/free-bios.html>

⁹<http://fsf.org/resources/hw>

tion of the database, please write to us at hardware@fsf.org. Getting commonly available hardware to be fully compatible with free software is a critical component of the FSF's mission, and it's a great way you can make a difference. We're almost there — we just need to get that last mile. 🐾



DefectiveByDesign.org action alert! Tell libraries to kick DRM out. Sign the letter at defectivebydesign.org/LetterToLibraries

The Free Software Jobs Page

by *Deborah Nicholson*
Directory Maintainer

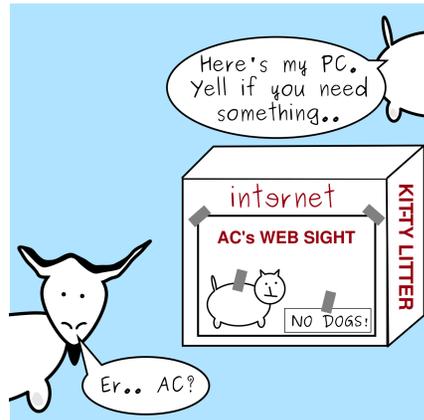
As a free software supporter, wouldn't you like to get paid for contributing to our community? Maybe you already do but you're sick of all that California sunshine or maybe you work somewhere that "pays the bills" but will never be able to buy your respect. That's why we've put together the free software jobs page — to help you find your dream job, whether it's right around the corner or halfway around the world.

Our community is stronger when free software companies employ the

best and most motivated people. We're putting employers in touch with dedicated job-seekers who are part of our community, people who are dedicated to seeing free software succeed, people like you who want to get paid for doing what they love.

The FSF is a 23-year-old nonprofit with thousands of members and supporters like you who regularly frequent our web site to keep in touch with the free software world. Our web sites get over 100,000 page views a day and our jobs page is always on the first search page for software jobs. Plus, you can trust us not to list just any jobs — we're only posting free software jobs.

Our members and supporters have already been spreading the word, by emailing their HR departments and friends who work at free software companies. So list a job, find yourself a job or share the news with someone else who's looking! Visit <http://fsf.org/jobs>. 🐾



Volunteer Spotlight: Yavor Doganov

with *Matt Lee*
Campaigns Manager

Yavor Doganov is the translations coordinator for gnu.org and the author of GNUUnited Nations, a new project which hopes to make management of translations significantly easier. He lives in Bulgaria.

Questions were asked by Matt Lee, FSF campaigns manager.

Matt: Yavor, you've been working on a new system for handling translations on our website. What's it called, and how does it work?

Yavor: Yes, it is called GNUUnited Nations (or GNUN for short). The name is, as you may guess, a pun of United Nations (UN).

GNUN works much like GNUgettext works for programs or like the GNOME Documentation Utilities (gnome-doc-utils) work for translations of GNOME manuals. For years, translators had to duplicate the HTML markup and examine changes as diffs between revisions of the original article(s). This is extremely tedious and error prone, which is why the GNOME Project has developed its own package to handle translations of the manuals (whose source is DocBook).

The source of almost all gnu.org articles is XHTML, so GNUN converts each article to a PO template, and each translation has its own PO file. The benefit is tremendous, because if the original article changes, each of its article.LANG.PO files is updated automatically, and subsequently all of the HTML translations are rebuilt. The translator can easily identify the changes using a PO editor and

updating many articles is possible with little effort.

The overall benefits for gnu.org are also worth mentioning. Since translations automatically follow the markup of the original articles, it would be possible to update them at once when we move to a new standard. If a team becomes understaffed and undermaintained, its translations will not rot but will get rebuilt automatically following the changes in the originals. Of course, this means mixed native/English language but we will turn this slight annoyance to be our ally: we will identify teams that need attention and will try to attract new volunteers.

Matt: What are the biggest challenges with the translation of gnu.org?

Yavor: I think GNUUnited Nations will solve most of the technical aspects of the problem. We will enhance the system to make it work for people who are not comfortable working on the GNU/Linux console, and will implement some sort of web-based automatic statistics.

But this is only part of the job, the easier part. The biggest challenge in gnu.org translation is the lack of volunteers, or more precisely, the constant lack of devoted contributors. This should not be surprising to anyone, though. In the Free World, people usually work on what they like and they switch to something else once they lose interest. The most appropriate example for a job that is always exciting and rewarding is being a programmer. Most programmers work on what they enjoy, and they switch projects as time goes by (well, sometimes). Programming is facing challenges every day, and finding (clever) ways to solve problems.

Translators are, in my humble opin-