FREE DVD

FREE INSIDE! $39.90 VALUE!

ALL 191 ISSUES ON A SEARCHABLE DVD!

HAPPY BIRTHDAY!
Linux celebrates 25 years

Searchable Linux Archive: 17,000 Pages of Practical Know-How
16 YEARS OF LINUX! Issues 1-191
▶ All the Tricks
▶ All the Hacks
▶ All the Apps
ISSUE 192 NOV 2016
LINUX MAGAZINE
DVD ROM
THE COMPLETE LINUX MAGAZINE ARCHIVE

LINUX PRO MAGAZINE

NOVEMBER 2016

25 YEARS OF LINUX

The Linux universe celebrates 25 years of perpetual disruption

Exclusive Interview with Linus!

Linux creator tells why the year of the Linux desktop has finally arrived

Peculiar Distros

Undead Linux, Suicide Linux, and other strange creations

Maperitive

Create your own custom street maps with online mapping data

Better Privacy

Keep your GPG keyring ready and up to date

LXC

Easy interface for Linux containers

Sirius and Lucida

Intelligent personal assistants built from free components

Issue 192 November 2016
US$ 17.99 CAN$ 19.99

WELCOME TO THE PARTY

Dear Linux Pro Reader,

This issue celebrates 25 years of Linux, and 25 years is a remarkable run for an operating system that began as a college student's hobby, but the story of Linux is a truly remarkable story.

Linux creator Linus Torvalds (who we are privileged and excited to interview in this issue) had a lot of help along the way. The help started at the very beginning of the project, with curious developers around the world pitching in with ideas and code. The community even pitched in at one point with an online collection to help Linus pay off his Intel 386 computer.

Companies like Red Hat and SUSE joined up fairly early in Linux history, as did community distros such as Slackware and Debian. IBM invested $1 billion in Linux in 2001 – an incredibly insightful act for a massive company with much to lose in the IT industry – and since then, other companies have poured billions more into developing and maintaining Linux. But it wasn't just developers who helped make Linux what it is. Thousands of volunteers tested and documented Linux and its flotilla of accompanying applications. And, what might surprise you if you just tuned in (but not if you've been watching for a while), Linux also got plenty of help from lawyers, with trademarks, patent defenses, and licenses.

Speaking of licenses, the biggest boost of all for Linux was from the legendary GNU project and its iconic leader, Richard Stallman. From GNU, Linux inherited an able compiler, lots of utilities and applications, and, of course, a revolutionary license that keeps the Linux community open, accountable, and united.

I have to say, as someone who's watched this story for a while, I'm pleasantly surprised that the underlying legal systems of the world's governments have stood by Linux. Not that they shouldn't have, but, as we all know, governments don't always do what they're *supposed* to do. It hasn't been all perfect. In the area of software patents, for instance, things didn't always work out, and much work remains. But several years ago, when it seemed like some very powerful companies were trying very hard to make Linux go away, it seemed ill-advised to bet more than your lunch money on Linux surviving a daunting succession of legal challenges, and for the most part, the courts did the right thing. The plucky penguin survived its battles with SCO, the seemingly endless barrage of challenges from Microsoft, and lots of skirmishes with random programmers and entrepreneurs who thought they didn't have to take the GPL seriously.

How do you get to age 25? You survive your teenage years, then you wake up one day, and you notice that the world has moved over to make a place for you. Happy birthday Linux!

Joe

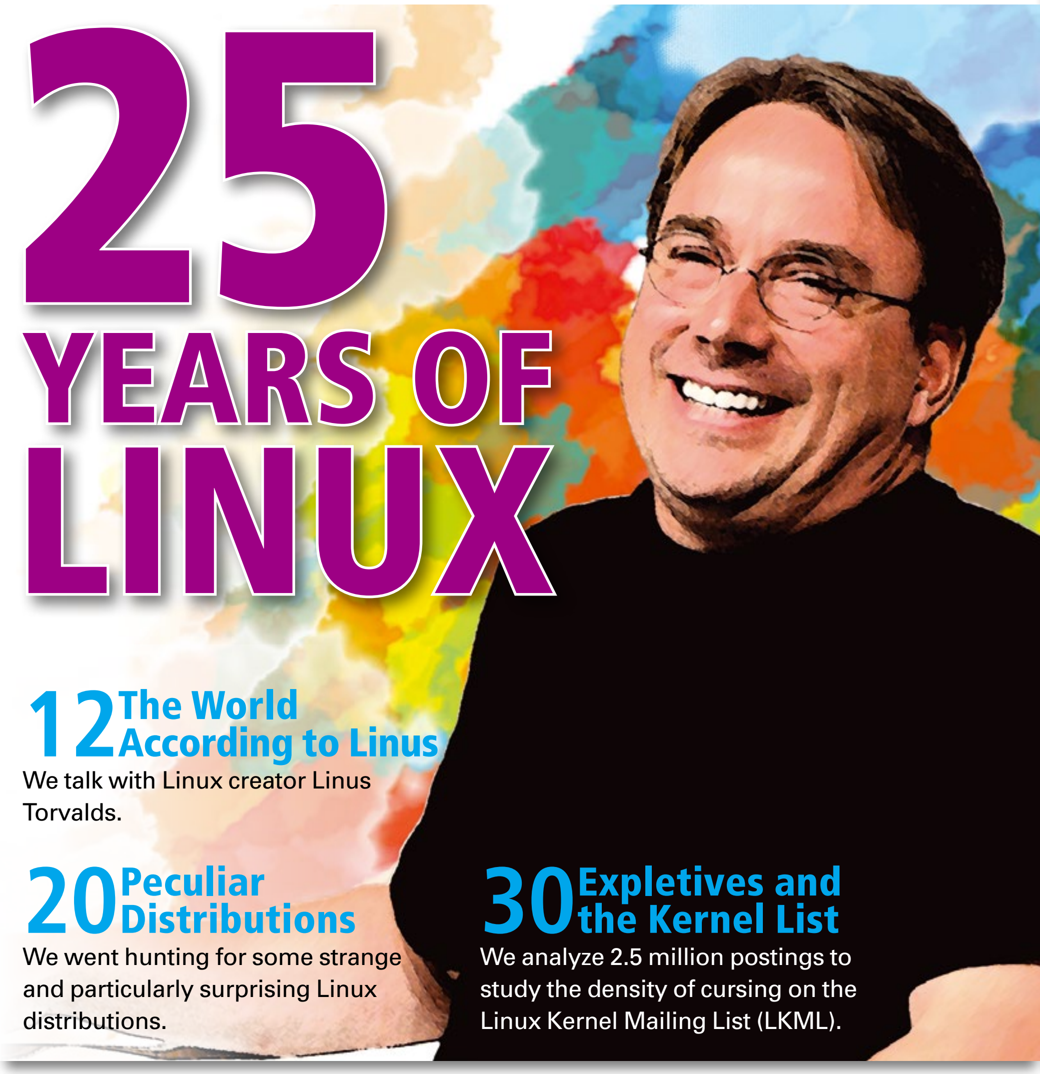
Joe Casad,
Editor in Chief

©Julianna Funk, 123RF

LINUX PRO MAGAZINE

FREE DVD
FREE INSIDE! $39.90 VALUE!
191 ISSUES ON A SINGLE SEARCHABLE DISC!
KEEP THIS DVD AS A HANDY LINUX REFERENCE
HAPPY BIRTHDAY! Linux celebrates 25 years
LINUX PRO MAGAZINE
NOVEMBER 2016
25 YEARS OF LINUX
The Linux universe celebrates 25 years of perpetual disruption
Exclusive Interview with Linus!
Linux creator tells why the year of the Linux desktop has finally arrived
Peculiar Distros
Undead Linux, Suicide Linux, and other strange creations
Maperitive
Create your own custom street maps with online mapping data
Better Privacy
Keep your GPG keyring ready and up to date
LXC
Easy interface for Linux containers
Sirius and Lucida
Intelligent personal assistants built from free components
WWW.LINUXPROMAGAZINE.COM

news



25 YEARS OF LINUX

SERVICE

LINUX PRO MAGAZINE (ISSN 1752-9050) is published monthly by Linux New Media USA, LLC, 616 Kentucky St., Lawrence, KS, 66044, USA. Periodicals Postage paid at Lawrence, KS and additional mailing offices. Ride-Along Enclosed. POSTMASTER: Please send address changes to Linux Pro Magazine, 616 Kentucky St., Lawrence, KS 66044, USA.

Community Notebook

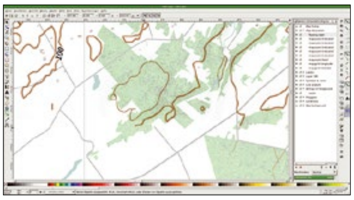
HIGHLIGHTS

FEATURES

REVIEW

LINUXUSER

Monkeysign (scan)
File
Edit
Identity
Video device
Hide QR code
Frank Hofmann (privat) <fh@efho.de> (5F962E33)
Frank Hofmann (Hofmann EDV) <frank.hofmann@efho.de> (D431AC07)
Frank Hofmann (LinuxBus) <linuxbus@efho.de> (222694C4)
No video device detected.



Searchable Linux Archive: 17,000 Pages of Practical Know-How
16 YEARS OF LINUX! Issues 1-191
All the Tricks
All the Hacks
All the Apps
ISSUE 192 NOV 2016
LINUX
THE COMPLETE LINUX MAGAZINE ARCHIVE
Be an expert! Answer all your Linux and Open Source questions with this ultimate knowledge archive!

On the DVD

Searchable Linux Archive:
17,000 Pages of Practical Know-How
16 YEARS OF LINUX! Issues 1-191
▶ All the Tricks
▶ All the Hacks
▶ All the Apps
ISSUE 192
NOV 2016
LINUX MAGAZINE
DVD ROM
THE COMPLETE LINUX MAGAZINE ARCHIVE

Linux Pro Magazine Archive DVD

This month's DVD includes the complete Linux Magazine archive – every previous issue of Linux Magazine on a single, searchable disc. Browse the pages of every article we've ever published, and experience our special brand of technical yet accessible how-to insights.

If you want to learn about a common tool in the open source space, search this disc – we've probably written about it more than once over the last 16 years. Catch all the articles you missed – on scripting, system administration, security, and more. Discover desktop applications that will save you time and simplify your life, and relive important moments in the history of Linux: the SCO case, the Novell/Microsoft pact, the birth of Git, the mobile revolution, ...

All the issues on this disc would cost you about $2,000 in our shop – if they all still existed, but many older issues are out of print: The only way to experience them is through the Linux Magazine DVD!

Searchable Linux Archive:
17,000 Pages of Practical Know-How
16 YEARS OF LINUX! Issues 1-191
THE COMPLETE LINUX MAGAZINE ARCHIVE
LINUX MAGAZINE
Be an expert! Answer all your Linux and Open Source questions with this ultimate knowledge archive!

BONUS! FREE Slimcase Inlay
www.linuxpromagazine.com/Issues/2016/192

Defective discs will be replaced. Please send an email to cs@linuxpromagazine.com.

AMD
FIREPRO

AMD FIREPRO™ S9300 x2
AMD

NEWS

Updates on technologies, trends, and tools

THIS MONTH'S NEWS

Linux Turns 25

On August 25, 1991, Linus Torvalds made an announcement about a project he was working on. He wrote on the Minix mailing list that he was working on a free operating system for 386/486 AT clones, which he stated was just a hobby and wouldn't be big and professional like GNU.

It's been 25 years since that announcement, and today, Linux is a dominating presence in the IT world. Linux powers a huge chunk of the Internet, as well as data centers, mission-critical operations like stock exchanges, supercomputers, mobile phones (Android), consumer desktops (Chrome OS), embedded devices, and much more.

Linux has become so dominant in the enterprise space that even Microsoft is now investing in Linux; they have developed a Linux-based modular operating system for data center networking. The company is putting Linux and Windows on an even playing field by bringing Linux development tools to Windows and Windows development tools (e.g., PowerShell) to Linux.

Jim Zemlin, the executive director of The Linux Foundation, claims that Linux is the biggest shared technology, and he has stats to back it up. According to the latest Linux kernel development report published by The Linux Foundation, "since 2015 more than 14,000 individual developers from over 1,300 different companies have contributed to the kernel."

The top companies that contribute to the development of the Linux kernel include Intel, Red Hat, Linaro, Samsung, SUSE, IBM, Renesas, Google, AMD, Texas Instruments, and ARM.

The size of the kernel has increased over the past 25 years. The first release of the kernel had more than 10,000 lines of code in 1991. Now the Linux kernel has more than 22 million lines of code.

Image © Laurent Davoust, 123RF.com

LinuxCon North America Convenes in Toronto, Canada

One of the largest Linux and open source events, LinuxCon NA, was held in Toronto, Canada, August 22–24. Jim Zemlin kicked off the event reflecting on the 25 years of Linux. Zemlin said that Linux has proven you can better yourself by bettering others at the same time.

The event featured keynotes by luminaries from the Linux and open source world, including Jim Whitehurst (CEO of Red Hat), Brian Behlendorf (Executive Director of the Hyperledger Project), Wim Coekaerts (Corporate Vice President of Enterprise Open Source at Microsoft), and Karen Sandler (Executive Director of the Software Freedom Conservancy).

Popular science fiction author and activist Cory Doctorow talked about the dangers of DRM in his keynote. Dr. Ainissa Ramirez, science evangelist and author, talked about the effect of technology on humans. Linus Torvalds, the creator of Linux, Git, and Subsurface, sat down with Dirk Hohndel (VP, Chief Open Source Officer at VMware) and talked about the kernel and open source. To support gender diversity, this year's event included a networking luncheon for women, which allowed women to network and interact with each other.

On-site childcare was provided so working parents could participate in the event. The Linux Foundation partnered with MakerKids and Kids on Computers to host an entire day of workshops for school-aged kids to learn about computer programming. Throughout the event, there were tons of break-out sessions and lectures where developers talked about open source technologies ranging from the Linux kernel to Linux containers.

Image courtesy of Swapnil Bhartiya

Adobe Gives New Life to NPAPI Plugin for Linux

Adobe is breathing new life into its Flash Player plugin for Linux. The company has released a new version of its NPAPI plugin for Linux, bumping the version number from NPAPI 11.2 to 23, bringing the Linux version of the NPAPI plugin in sync with the current version in the latest branch.

Linux has two versions of Flash Player: NPAPI and PPAPI. PPAPI is used by Chrome and Chromium web browsers and is fully maintained. Although Adobe stopped working on the NPAPI plugin for Linux, with end of life scheduled for 2017, with the new release, NPAPI gets a life extension.

Adobe said in its blog post that this change in plans is a security initiative for those Linux users who still need the plugin, but it will not get many of the features found in the PPAPI plugin.

MORE ONLINE

Linux Pro Magazine

www.linuxpromagazine.com

Off the Beat • Bruce Byfield

OpenOffice: A Project in Search of an Exit Strategy

In theory, I should be all over the story about Apache OpenOffice's struggle for survival. Over the years, I have written dozens of articles about OpenOffice.org, OpenOffice, and LibreOffice. Yet now as OpenOffice tries to revive itself, my main thought is: Could someone please put OpenOffice out of its misery?

LibreOffice's Legacy Debt

LibreOffice has had so many changes of name that its age is hidden. However, go back through Oracle OpenOffice and OpenOffice.org, StarOffice, StarDivision and StarWriter, and the word processor is over 30 years old. Probably, very little of the code written in 1985 remains in use, but many of the features do – and that means that LibreOffice is carrying a legacy code debt that is becoming increasingly irrelevant.

Productivity Sauce • Dmitri Popov

Extract Pages from a PDF File with a GUI Bash Script

Every now and then I need to extract individual pages from PDF files. Recently, I stumbled upon a handy Bash script that generates a simple graphical interface for extracting pages from a PDF file.

ADMIN HPC

http://hpc.admin-magazine.com/

Let the Editor Wars Begin! • Jeff Layton

Editors, particularly command-line editors, are an important tool for high-performance computing systems administrators. We point out some editor options with little to no bloodshed.

ADMIN Online

http://www.admin-magazine.com/

Swagger and OpenAPI Specification for documents • Tim Schürmann

A REST API is especially useful for a developer if the API provider extensively documents the methods used. The Swagger tools not only semiautomatically generate an API reference, but a matching client SDK for many programming languages, as well.

The Practical Benefits of Network Namespaces • Paul Thompson

With network namespaces, you can create very sophisticated and resource-saving setups using the tools inside a running Linux system – without the use of containers.

Users who are looking for GPU 3D acceleration and playback of DRM'd videos must use the PPAPI plugin. Adobe said in a blog post, "If you require this functionality, we recommend that you use the PPAPI version of Flash Player. That said, we believe that the new NPAPI build represents a significant step forward in functionality, stability, and security and look forward to hearing your feedback."

Despite this lease on life, Adobe Flash is on its way out. Major browsers have already started blocking Flash content by default.

End of OpenOffice?

Apache OpenOffice Vice President Dennis Hamilton sent an email on September 1, 2016, inquiring about the fate of Apache OpenOffice. Hamilton wrote that "the Apache OpenOffice project has limited capacity for sustaining the project in an energetic manner."

The dearth of developers and interest in OpenOffice came about because a majority of OpenOffice developers moved to LibreOffice, a fork of OpenOffice under the governance of The Document Foundation.

Retiring the project is now a possibility. Hamilton wrote in the email, "I cannot make a prediction how this will all work out. It is remiss of me not to point out that retirement of the project is a serious possibility."

Opera's Password Sync Service Compromised

Opera Software has reported a security breach that compromised Opera Sync Service, the password manager for the Opera web browser. Opera has more than 350 million users and approximate 1.7 million people are using Opera Sync Service.

To its credit, Opera Software acted swiftly and notified its users of the breach by email. The company said in the blog post, "although we only store encrypted (for synchronized passwords) or hashed and salted (for authentication) passwords in this system, we have reset all the Opera Sync account passwords as a precaution."

Opera Sync users are also advised to change the passwords of third-party services, such as email, that they used on the browser, because this information might have been stored on the Opera cloud and therefore could have been compromised.

Password managers are used by web browsers to make it easy for users to store usernames, account information, and passwords on the cloud. All major browsers, including Google Chrome, Firefox, and Apple Safari offer such features. However, unlike Chrome and Safari, Opera Sync doesn't offer the additional security of two-factor authentication.

Is Google Working on a New Operating System?

Google is reportedly working on a new operating system that is not based on Linux. The company hasn't made an official announcement yet; however, the source code is already available on GitHub.

The project is named Fuchsia OS. According to the GitHub page, "Magenta is the core platform that powers the Fuchsia OS." Magenta comprises a microkernel and a small set of userspace services, drivers, and libraries required by the system.

Magenta is targeting modern phones as well as desktop PCs. The kernel used in it is capable of running in really small systems such as embedded devices.

The GitHub pages say that LK (Little Kernel) "is a kernel designed for small systems typically used in embedded applications. It is good alternative to commercial offerings like FreeRTOS or ThreadX. Such systems often have a very limited amount of ram, a fixed set of peripherals and a bounded set of tasks."

Google works on many projects, which may or may not translate into products. Fuchsia OS could just be another experiment or it might replace Android or Chrome OS as a single OS to run across devices, including phones, PCs, and embedded systems like Onehub and Google Home.

We talk with Linux creator Linus Torvalds

Colonel of the Kernel

By Swapnil Bhartiya

The 25th birthday of Linux is an important milestone for everyone in the Linux community. Who better to help us ring in the occasion than the man who started it all: Linux creator Linus Torvalds. Linus was a 21-year-old Finnish university student when he sent a note to the Minux user group stating that he was working on a new operating system that was "… just a hobby."

Since then, Linux has found its way onto millions of systems around the world, and Linus has become an iconic figure for a vast community of Linux users and developers. Our news editor, Swapnil Bhartiya, sat down with Linus for an interesting exclusive interview on the kernel, the Linux desktop, and the quest for balance in work and life.

Linux Magazine: It's the 25th anniversary of Linux. You sent out that humble email some 25 years ago…

Linus Torvalds: It's not even clear what the real anniversary date is. There are really two of them. August 25th is when I made the public announcement, when I emailed that I would be announcing it. The actual release, 0.01, was two weeks later, but that one never got announced publicly at all.

LM: Can we consider Linux's beginning when you started working on it? When was the very first line of code of the Linux kernel, as we know today, written?

LT: I don't even remember exactly when the coding started, but it must have been roughly March or April of 1991. I got my machine in January; that was the first Linux machine. But at that point I had no idea that I needed to write my own operating system. Then it took a few months to get Minux set up and from there it took me some time to realize that I needed to do my own OS. I think I started coding maybe in April. It is clearly more than 25 years now. It's a long time.

LM: Linux has come a long way from that humble beginning. It's running on almost everything. Do you ever reflect back on 25 years to see what all it has achieved?

Images Courtesy of Swapnil Bhartiya

LT: Not really. Because of the whole 25 year thing, I get that question a lot from journalists over email. One thing that I react to is that so much else has changed in my life and in many ways Linux has stayed the same.

I mean the details of Linux have changed completely – the kind of machines it runs on, the number of people involved, the number of companies involved. All of that has changed dramatically. But at the same time what hasn't really changed for me is the fact that, even 25 years ago, the reason I did Linux was the interaction with the hardware, to explore it, and the technical challenges that come with it.

While the details have changed, I don't write any code anymore; it's more about directing people. The basics, the really core basics are still the same. The thing I'm interested in tends to be the close interaction with the hardware, the technical challenges.

In a very real sense, Linux has changed much less than everything else around it. The fact that, in the last 25 years, I now have three almost grown-up children, and I moved from Finland to the US. I went from being a university student to working at a startup to working at The Linux Foundation. Everything else has changed so much, and, at the same time, the reason I'm doing Linux has not changed at all.

LM: I understand that back in those days you had new hardware and you wanted to do something. Do you have the same itch to scratch now, as most hardware these days works with Linux or comes with some form of Linux preinstalled?

LT: No. That has changed. It used to be, but not for very long actually. I wrote Linux because it was something I needed, but that stopped very early on. At that point, it was doing what I expected it to do. Since I made it available, there were people coming in and saying "I need this" or "I think it should do this." If that was not the case, I would probably had been done by the end of 1991. I will give you an example. Early on my machine had 4MB of RAM. That's ridiculously little, but it was plenty for what I was doing, and GCC, at that point, didn't actually need that much. But somebody had a machine with 2MB of RAM, and he could not compile the kernel under Linux because that was not enough for GCC. That was around Christmas 1991. I started working on paging to disk so that this guy with two megs of RAM could actually build his system on it. So literally in late 1991, I was implementing major features that I actually didn't need, but because somebody else was asking for it.

From very early on, the motivation for doing Linux went from this is something I need, to other people are using it. It changed, but it changed so early that I think of this as, this is how I do Linux. That the things people ask for really come from outside. That's not new to me anymore.

LM: I recall my previous interview with you where you said that it's not spreading the code, but also the vision. Different people come to Linux with their own vision, instead of you creating a vision for everyone. That's quite remarkable. I have never seen that with any other project.

LT: And that's what made it [Linux] much more interesting. Some other projects start with a vision of where they want to go and Linux never really had that. Linux didn't even start as an operating system. That meant that from the very first day after I made the announcement, when people started making comments, I think I was more open than a lot of other projects to just taking input from other people.

If you looked at in the time frame 25 years ago, the other big operating system project was obviously the BSDs. They had literally 20 years of history behind them. They had people who knew how things were supposed to be done. Linux didn't have that, because I didn't have that. So Linux was a project that was much more open to just saying: Okay, we'll do that.

It was much easier.

Initially individuals, and later companies, just wanted to push Linux in their own direction. Because I didn't have any particular goals in mind, I was perfectly happy with that. The only goal I had, and that's still the case, [was] that you can do whatever you want with Linux, but let's just make sure that the technicals are good.

LM: By design, you created a project that allowed people to take it any direction they wanted, so what are the compromises that you are willing to make, and, at the same time, what are the compromises you will never make with Linux?

LT: The thing I tend to care about is if you're getting a driver, let's just get it working. A driver does not impact the big picture development; it doesn't impact maintainability in the long run. One driver may be really ugly, but I'm very happy to get code like that merged into the kernel. Usually it comes through Greg's [Kroah-Hartman] staging tree, and some of those drivers are really ugly; they are doing a lot of bad things. But that's fine. If they don't impact another driver, we can fix them up.

But when it comes to really core, very core kernel stuff, I do want to make sure that there's a clean design. That there's a point that it's maintainable in the long run. Partly because the code is well written, but partly also because you have a notion of where hardware is going and where usage is going. So you have to pick something where the interfaces work for different hardware, where the interfaces work well for different loads, because different people will use Linux for so many different things.

So that's where I don't have strong opinions on what you do with Linux, but I do have very strong opinions on the technical side of making sure that really core stuff is well set up so that we don't paint ourselves into a corner.

LM: Let's change gears for a while. Let's talk about Microsoft. They have changed so much in recent times. They have a Linux-based OS for Azure; they are bringing PowerShell for Linux. What do you think about it all? People often quote this, did you ever say that if Microsoft ever wrote applications for Linux, you won?

LT: I do think I did say something like that. But you have to realize that back in the late '90s the situation was very, very different. We, in the Unix camp, used to make jokes about Windows crashing. It's gotten much better frankly, not that I've used Windows for a long, long time.

I think the situation just has changed. It used to be that it was a very antagonistic situation, partly [because] Microsoft used to be so dominant. They were the target for a lot of antagonism. I used to make jokes about Microsoft, and I actually stopped doing that sometime maybe 15 years ago. But it used to be very common. It used to be very much Microsoft versus Linux and Microsoft versus Unix. One of the reasons I stopped making Microsoft jokes was that it always generated all this unnecessary attention.

There were stories written about Linux versus Microsoft but that was never actually the case with me. It was more like making a cheap joke about Microsoft, but at the same time I didn't care. That was not the reason why I did Linux. Microsoft was mostly, fairly neutral on Linux except for those few outbursts from Microsoft that were bad. They do seem to have changed in the last couple of years.

LM: What do you think about this transformation of Microsoft?

LT: I really don't follow it that much. I don't think it's even so much about Microsoft. I think the computing market, and the operating system market in general, has matured to some degree. It's such a different situation than it was 20 or 25 years ago.

What I am trying to say is that it's actually across the whole board. Tech companies used to be much more hostile to each other. You had the Sun versus Microsoft wars. I don't think it was a very nice marketplace. There's still hostilities, and you still have the Google/Oracle thing going on and things like that, but I think the market has kind of calmed down, and people are not quite so antagonistic anymore.

LM: That's true. But I have come across desktop users who are still antagonistic and don't like the idea of running Microsoft software on Linux. So from your perspective, is it okay to use Microsoft apps on a Linux desktop?

LT: I remember back in those days before we had good applications for doing presentations, very early on I was doing presentations about Linux and I would actually use PowerPoint on Wabi. Before Wine, Wabi was Sun Microsystem's Windows emulator. So I would actually run PowerPoint on Linux back in 1995 or something like that.

LM: (Chuckles) Is that the reason you don't do presentations anymore?

LT: (Laughs) No. I'm happy that it was only for a short while I actually used PowerPoint. If you look at the whole "me versus Richard Stallman," I always felt that to me the whole open source is not about how commercial software is evil. To me open source [has] always been about how I think it's a much better way of doing development.

I think it's more productive when you cut through all the bullshit, through the borders between companies, and you can just work together with other people. That's why, obviously, it's so much more fun, too, because you don't have to worry about who you're talking to. You can just take input from anybody who has a good idea.

So I think open source is just a, technically, much better way of doing things. I think the licensing is important but it's important as a way to keep everybody honest, to keep open source open. And that's great. It's never been this religious war for me that it is for some people.

So when Oracle started porting their database, when a lot of big companies came in and started making proprietary programs for Linux, I was like: that's great!

LM: It's just a platform like any other platform. You should be able to run whatever it is, right?

LT: Right. But at the same time, I do think that in the long run open source tends to take over because it's a better way of doing things, especially the infrastructure part. If somebody is doing a database that isn't open source, I would be worried about that in the long run because it's such an infrastructure play; everybody needs it. MySQL obviously already took a lot of it, and there are other open source databases.

In the core infrastructure play, I think open source will take over in the long run. But I don't think it's because of ideological reasons; it's because of technological reasons.

LM: It's already happening. AI and machine learning is touted as the fourth industrial revolution, and a lot of machine learning technologies are being open sourced by Google, Amazon, Microsoft, …. Everyone seems to be open sourcing their stuff ….

LT: If you actually look back historically, basic applications used to be, maybe not exactly open source because people weren't conscious about it, but if you look at how software programming was done in the very early days, people would just ship tapes to each other.

So almost always when you're doing something completely new and the market doesn't really exist, it's very natural to share anyway. And then when the market becomes bigger and economics take over, the sharing often stops. That's where having a good license that ensures the openness can't go away is actually important. It means that when a project grows up from being experimental, from being fun to work with, and becomes a big commercial thing, if you pick the right license it will stay open and actually it will grow. I do believe in that.

LM: Licences are important. If you were to start a new project today what license would you choose? [See the "Free Software Licenses" box.]

LT: I think it is like a personal choice; different people have different opinions on what matters. I don't like the language of GPL v2; it's much too long and much too complicated, but I think the basic ideas are really the right ones. Which is basically saying: "Hey, I give you source code, you have to give me source code back, and this is irrevocable, and nobody has any extra rights, and it stays like that forever." I think that's just wonderful.

I picked the GPL v2 originally because that was what I agreed with. The legalese and all the language is less important to me, I wanted something that was well known, and I still absolutely agree with it. I would still likely choose GPL v2.

I don't like the GPL v3, and the reasons I don't like it are pretty much out there. But I understand people who want to use it. I also understand people who want to use BSD. There are a lot of licenses out there.

LM: I have noticed a pattern that community-driven projects tend to choose GPL-like licenses, whereas big company-backed projects tend to choose BSD or MIT type licenses.

LT: It is true that if you're mostly a commercial entity you tend to go with something like BSD because then it leaves you more options going forward. But that's where I actually think it is a mistake. It does give you options to do other things going forward, but it also means that the community around it does not feel protected because they know that everybody has these options, too, to just take it and go with it. As a developer, if I start a project and I give it away, I really want people to use it, but at the same time I want people to give their improvements back, and that I think is fair. I call it tit for tat.

If somebody else does a project and I want to join that community, I also feel much more protected if that other person chose the GPL because then I feel that any improvements I gave back will be part of this continual improvement. While if you use a BSD license, I think the community is less protected. But, this is my personal opinion; there have been lots of very successful BSD license projects, too. So I don't know.

To some degree the license proliferation has been a huge source of confusion, so I'd rather stick to a couple of well-known licenses and just say: You may not love the GPL v2, but at least everybody knows how to deal with it. You may not love the version 3, but it's one of a handful of licenses. There is a new, shorter version of the BSD license, and everybody knows how that works. It's nice.

By the time you get outside of those three, you have to say which one is it? Is it the Mozilla license, is it the so-and-so? So, I'd rather keep things simple and stay with GPL version 2 just to avoid complexity.

I made a mistake once. I started Sparse for doing static analysis of the kernel. It's not a very well-known project; I started it as a compiler front-end thing. I decided to pick a very modern, simpler license that wasn't well known. (chuckles) I don't even remember which license that was. I thought, it seems to be much simpler than the GPL v2, and it was. But it was not compatible with anything else, and it was different enough that you had to always explain it. We finally got it relicensed. That was many years ago, but it was actually a painful experience to use one of the smaller licenses just because it became such a problem working with people who wanted to use some of the code from Sparse for some other project. It was not compatible with anything else, and it was just not worth the pain to me. So I'd still pick GPL v2.

LM: Let's now change gears from professional life to personal life. You work from home. I also work from home, and it's quite a challenge when you have kids. So how do you maintain the work-life balance?

LT: I've talked to people who cannot work from home because they have kids and they say: "I can't concentrate. I need to go to the office so the kids aren't there and they don't disturb me and they don't distract me when I'm working."

I am not a very nice person. My kids never distracted me or disturbed me when I was working because they realized very quickly that daddy's not a very fun person if you come and scream in his ear when he's working. So they would never

FREE SOFTWARE LICENSES

Several different licenses meet the Free Software Foundation's definition of "Free." Free software licenses fit roughly into two categories:

- Copyleft licenses, like the GNU Public License (GPL) – include a clause that says, if you make changes to the code, you have to share your changes with the community
- BSD-style licenses – do not include a copyleft clause, which means you can adapt the code without sharing your changes.

Parts of Apple's proprietary OS X operating system are based on BSD Unix, which had a BSD license and, therefore, could be integrated into closed-source code. The same thing could not happen with Linux, because it uses the GPL v2 copyleft license.

The GNU project and the Free Software Foundation now recommend GPL v3. Linus and other kernel developers objected to some of the changes with the GPL v3, especially regarding the treatment of Digital Rights Management (DRM) technology, so they stayed with the GPL v2 as a license for the Linux kernel.

come into the office. It was not that the office door was closed and it was not like this is the border you can't cross. There was never that kind of rule. It was more like, that's not the fun room. That's daddy's work area.

Now we have a big house, and the kids are big enough where it's not an issue anyway. But I much prefer working from home because it means that I don't have to plan working. So if something happens, I just go to the computer.

A lot of the time, I don't have to be on my computer; I read mails on my phone or tablet, but not during the merge windows. Merge windows are special times where I have to be at the computer for eight to ten hours a day because that's what it takes.

LM: You always merge from home?

LT: I try. Sometimes I do merge windows when I'm traveling, and it's really painful. Outside of the merge window, a lot of the time I end up waiting for results. For example, there is a bug discussion that's going on, and it wasn't me who found the bug. Someone else found it, and it's being reported and it's really just email – and emails coming in at odd hours, especially if someone is from Europe or the rest of the world. So a lot of the time, I end up waiting around for people. I actually read email a lot on my tablet or my phone, and, when something comes in, I go up to my office; that's when I need to do actual real work in front of a real computer. But I can do the first order filtering of just seeing what's going on from the phone.

LM: Are you like Richard Stallman, where you bring your laptop to the dining table; in case there is pull request, you will move the plate aside and start working ....

LT: No. No. I won't do that. I want to be there. When I'm going away for a week and I can't react to pull requests the same day, I let the top maintainers (not everybody) know that I will be traveling and I will have so little time that I won't be answering that fast. I don't need to feel that if a pull request comes in I will pull it five minutes later. No. I will pull within the next 24 hours.

LM: So is the family time like strict family time, no work?

LT: My wife is stay-at-home mom. When the kids were small (and these were three different kids) in the US you have to drive them to anything after school. So we always had this situation where my wife can only be in so many places at the same time. So I would just know that after school usually somebody needs to be driven. That's much less of an issue now because two of the kids drive themselves and are off to college. But that's kind of the work/life situation I was in. It wasn't like "Okay, dinner time is when the family gets together and then we watch TV afterwards." No.

LM: Do you watch TV?

LT: No. I don't. My wife watches some TV.

LM: Have you heard of *Mr. Robot*?

LT: No but I have heard of *Game of Thrones*. I've never seen it.

LM: Now that your kids are big, they're going to school. They are moving out for college. You will be in a different phase of your life. You will have all the time to yourself. So will you still be doing Linux or will you start something else?

LT: That's odd because when I got my first child, at that point people were worried: Oh, now he has children and he's moving away from university. They (kids) happened about the same time. Some people were saying, "Oh, that's bad. Now he will stop maintaining Linux." And now the children are going away, and you're asking me whether I will stop maintaining Linux.

LM: I am not saying you will stop maintaining Linux, I am saying you have more time now, so will something change?

LT: I don't think it will change a lot. We've had this whole fairly traditional family style where the husband brings in the money, even if he does it in a bathrobe, and the wife takes care of the children. So, it's actually my wife who's getting impacted a lot more by the fact that the kids are going away. But we still have one; she's in high school. She will get her driver's license in a couple of months, so she will be driving herself at that point.

LM: I have two kids and I miss them when I come to these conferences. How do you feel, emotionally, about your kids growing up and moving away?

LT: I'm so happy about it. I thought that babies are not even very interesting. Children really start getting interesting around the age of nine. I'm not saying exactly when they turn nine but at the age of nine, they change. Before nine they're very immediate. They don't think really; they react.

At the age of nine, roughly, they start to have abstract thinking. They start thinking much more interesting things. They turn into humans. I saw that as a huge step forward. When a child turns from something that isn't very smart, maybe slightly smarter than a dog (laughs), don't get me wrong, and turns into a real human and starts thinking at a completely different level. I was very impressed with that. I thought it was very noticeable, too. Now, teenage years, not always pleasant. You get arguments; you get boyfriends

coming home. Ah! Not always the happiest moments. But I'm actually really enjoying seeing them turn into adults. Again, I think when you [grow] into a real independent person, that's another big step and that's happening right now for them. I'm actually very happy; I think it's nice to see them grow up.

LM: Do you have any influence on them as a software person, as a celebrity who is changing the world?

LT: I don't think so.

LM: You're very humble, that we know. But is there any discussion at home with your daughters about what they want to do or what you're doing, any influence like that?

LT: No. Not really. They obviously very much know about the Linux thing. They have computers that run Linux, and that's not an option (chuckles); this is not something they got a choice in.

My oldest daughter is going to a technical college. She will presumably do computer science and engineering. We'll see. But it was not something we pushed them into. We did push them into getting a real degree. We said we'll pay for college, but it has to be a real job. You're not getting some degree where you don't see yourself with a real career afterwards. It was like you don't have to be a tech person.

The second one right now is going into pre-med. She's looking at neuroscience. The third one says she's interested in being a school teacher, which my wife used to be.

LM: School teacher in the US?

LT: In the US, I am not completely convinced it's a good career. But coming from Finland, I know being a teacher is a really very well respected job. I'm not complaining; I'm slightly sad that in the US they're not very well respected.

LM: Finland has an excellent education system; did you ever consider going back for the kids? We lived in Germany, and we were tempted to go back after having kids.

LT: That's where we thought we would go, too. We thought we'd move back to Finland when the kids started school. By the time the kids started school, we had learned how the US system worked. And then we thought by the time they have go to college maybe we'll move back to Finland, because college is free. And even that never happened.

LM: A last question while we are still on the topic of family. What will upset you more, if your daughter brings a boyfriend home or brings an iPad or Windows machine home?

LT: Oh, no. The iPad or Windows machine is much worse. The boyfriends have already happened. It's like you know it's going to happen. It's all fine. It's like whatever. It's just slightly awkward still.

LM: Going back to Linux. These days we hear a lot of security-related stories popping up. What's your perspective: Are there more bugs being discovered or now are more bugs slipping in?

LT: Quite frankly, part of it is that there's this whole scare culture about security. We have bugs, don't get me wrong. To just give an example, fairly recently there were tons and tons of articles being written about how 1.4 billion Android devices were open to vulnerability to this networking bug.

Yes, it was a bug. We actually followed the spec a bit too closely, but from a security angle the vulnerability was that you can do it in the lab, but in the real world it really doesn't matter. Yes it was a bug. In theory 1.4 billion devices were involved, but anybody who actually knew what was going on would ever really care. But then you have these scary stories written about it. That's why I dislike the security community so much because they try to drum up these stories.

We had bad security bugs. We had really nasty ones where I said: "That's just stupid; that was not good." They happen, and we try to fix it. But, more importantly, we are working fairly hard on hardening infrastructure.

LM: What kind of work are you doing in that area?

LT: Bugs will happen, but when they happen, hopefully, all these other safety mechanisms mean that, in practice, they don't end up being security issues, or they are so hard to exploit that it's not usually a problem. So, we do have a very real project going on. It's part of CII [Core Infrastructure Initiative].

Kees Cook has been the main person leading it on the kernel side now. He is instrumental in integrating code to kind of create these fallback safety nets. We're working on that, and we've worked a lot on security issues in general – for example, the supervisor-mode access protection, supervisor-mode execute protection. We are doing things where when bugs happen and a user program fools the kernel into jumping or accessing something in an insecure manner, the hardware will actually catch it because we talked to Intel to make sure that they have the hardware capabilities to do it. So, we've done a lot of things like that. Will we ever get it perfect? No.

LM: But it's not always the kernel, there are many different components.

LT: I think one of the big wake-up calls was all the OpenSSL issues. The kernel is quite special when it comes to security. But when you look at the real security issues, it's often in libraries that are used in all the applications that try to do something secure, and, when those libraries fail, they create issues. We are working on it. I will not say that we'll ever be perfect. We're working on being better.

LM: You are doing this at kernel level. It's true that the kernel community patches bugs quickly, but the problem seems to be with distributions or hardware vendors where these patches don't even reach the devices.

LT: That will be a bigger problem in IoT [Internet of Things], it is not so commercially viable yet. But look at mobile phones; some mobile phones never get updated. Is it annoying? Yes. That is part of why we're trying to do the hardening thing so that even when you don't update, hopefully, it won't be catastrophic. It's a very hard problem to solve.

The good news is a lot of people are doing security on many different levels, which is the only way to do it right in the first place. Yes, we do the best we can do in the kernel, but distributions are trying, too. People are moving on into using containers to limit, when security problems happen, to a smaller part of the system. People are doing a lot of such things. You add all these different layers that, hopefully, make it more and more inconvenient to punch through all of these layers to get to the really serious bugs and exploit them. But, I have to say, some of those attack people are pretty smart people, and clearly they're not all criminals; some of them work for the government.

Part of the issue is that as developers we are looking at giving access to new features, giving access to hardware, and things like that. When we do that, we have a completely different mindset from the people who are looking at attack surfaces. When you read about an exploit you say "Wow, I would have never thought about that," because you come from a different direction.

LM: Let's talk about the desktop. People always ask which distro you use. What's the reason behind that choice?

LT: I try not to care that much about my distro, but what I don't want is to have three distributions that look different because I put the same thing on my wife's machine, my desktop, my laptop, and my kids' machines. For the last 10 years, probably since I switched away from Power PC, I think I've used Fedora.

Right now most of the machines are Fedora 23, a couple of them have been updated to 24, but then I will install my own kernel.

One of the reasons I like Fedora is they tend to be fairly good about new kernels. Red Hat, in general, has been very good about kernel resources. They have been helpful in testing, and, obviously, they do have lot of kernel engineers. One of the reasons I ended up going with Fedora was that they did a good job on the side that I cared about.

LM: The Linux desktop that we wanted has not happened yet, but Chromebooks are doing wonders, especially with the arrival of Android apps. If I can paraphrase, is this the desktop you were looking for?

LT: It's not the desktop I'm looking for, for my needs.

LM: What I meant was Chrome OS as Linux on desktop in consumer space.

LT: If you were looking at things from an end-user standpoint, I think these days it actually makes sense to see mainly the browser. The browser is clearly a big portion of the system and Chrome takes that as the starting point and says "we want you to get the browser experience and then we have some other small things on top." There's a lot of people who seem to really like Chrome OS as a desktop just because they don't have to worry about kind of a traditional desktop.

I think right now it looks like Chrome OS is really taking Linux and making it happen. It's just that the thing I use is the old workstation environment, and it's really doing well. It's doing way much better than the older workstation and what it used to be.

I think that's a sign that the market has become more mature. If you look at why the PC made such a huge difference in the '80s and '90s, it was because the PC finally made a unified platform that could do a bit of everything. You didn't have that before. That's why the PC really changed the world.

But the fact is that back when the PC changed the world, you really needed something that could do a bit of everything because you didn't know what you would need.

What has happened in the last 5 to 10 years is that people have started knowing what they need. Now you can make specialized devices again. The cell phones took a bit longer, but they clearly matured, too. In the last two years not a lot has really changed in cell phones, now you know where the goal post is. Now you don't need the original kind of PC that could do a bit of everything anymore. I think the market has changed, and what I thought of as a desktop was the general purpose PC that doesn't necessarily make sense anymore, economically, where we are now, except as a workstation. There's a lot of people who don't really need that anymore and are perfectly happy doing lots on their phones and maybe on a tablet.

LM: Can we say that the year of the Linux desktop is here with Chrome OS or you are still waiting?

LT: I think, yes, maybe Chrome OS is basically the Linux desktop, but at the same time that whole thing has not played out completely yet. Give it maybe five more years and see where things are. But yes, I do think that we may be in a situation where Chromebook is the desktop for people who don't do development and that's fine. ■■■

A guided tour of some notable and peculiar Linux distributions

One Wild Ride

Linux distributions proliferate like butterflies. Linux Magazine went hunting for some strange and particularly surprising specimens. *By Kristian Kißling*

Even if the syntax suggests otherwise, the name of the classic Linux tool `awk` does not stand for awkward, and it isn't even a reference to the bird known as the Auk, but rather, the name is derived from the names of its authors, Alfred Aho, Peter Weinberger, and Brian Kernighan. Cryptic names are a long-established Unix tradition [1].

Developers often believe the names of their tools are self-explanatory, or they think the name is not of interest to the users anyway. Many developers of the 500+ Linux distributions worldwide clearly subscribe to this school of thought [2] (Figure 1). Even several very early Linux distros bore cryptic abbreviations (LSD, LST, DLD) or really curious names like Yggdrasil (Figure 2).

It isn't difficult to see that humor is a primary motivation for much of the naming that happens around Linux. The Open Source community loves to rebel against the kitschy conventions of market-speak that dominate commercial software companies, with names that seem slightly satirical or, at least, summon up images and associations that would make a marketer cringe.

We're seizing the occasion of Linux's 25th anniversary to arrange an (admittedly incomplete) typology of strange Linux distributions. Although a few of these beauties are still looking fondly into the future, many are relics of the past, and, as you will learn, others are in a curious state that is neither alive nor dead.

Lead Image © Jan Will, 123RF.com

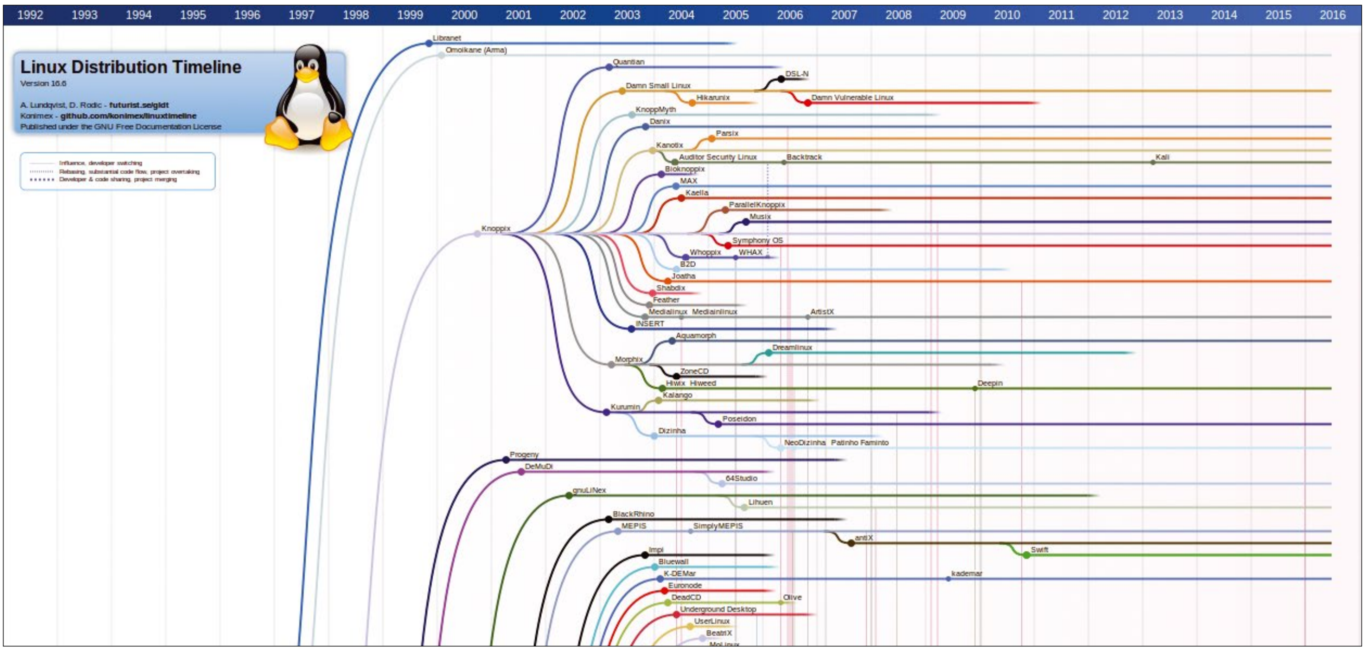


Figure 1: **Only a small excerpt from the extensive family tree of Linux distributions. You'll find this graphic at *http://futurist.se/gldt/*.**

Linux for Zombies

Speaking of undead: The Linux distribution best suited for zombies was very easy to find. Undead Linux, also known as Evil Entity [3], breathed its last back in 2003. The related domains are free, but copies may be haunting the BitTorrent universe, searching for the brains of Linux users. Evil Entity, incidentally, is the arch enemy of cartoon character Scooby Doo, and a "floating mass of dark green tentacles," according to the sci-fi fan site Wikia. If you immediately think of the light-green SUSE as a perfect fit for Evil Entity, you are wrong. Actually, Evil Entity was based on Slackware and sat on a rather gloomy-looking Enlightenment desktop (Figure 3).

The More Broken, the Better

Void Linux [4] was once declared dead by its main developer, but the announcement turned out to be an April Fool's joke [5].

Striking features for the rolling-release Void include its own build system, a self-developed package system called xpbs, and the absence of systemd. Void Linux, which is insistent about not being another distro's fork, relies on the Runit alternative init system [6].

Damn Vulnerable Linux (DVL) [7] is an interesting case. Like a fine wine, this contagious Linux keeps getting better even after its demise. Thorsten Schneider, a lecturer of the University of Bielefeld, published version 1.0 of DVL in 2007 as an admin nightmare so full of security holes that security researchers could use it for experimentation. Since 2012, apparently nobody has further maintained DVL, which is based on Debian and Damn Small Linux. The lack of attention means the number of security vulnerabilities is naturally growing – in this case, a classic win-win situation.

On the Edge

At first, Suicide Linux [8] sounds like an irony-laden and upwardly mobile Linux distro with an optimistic view and

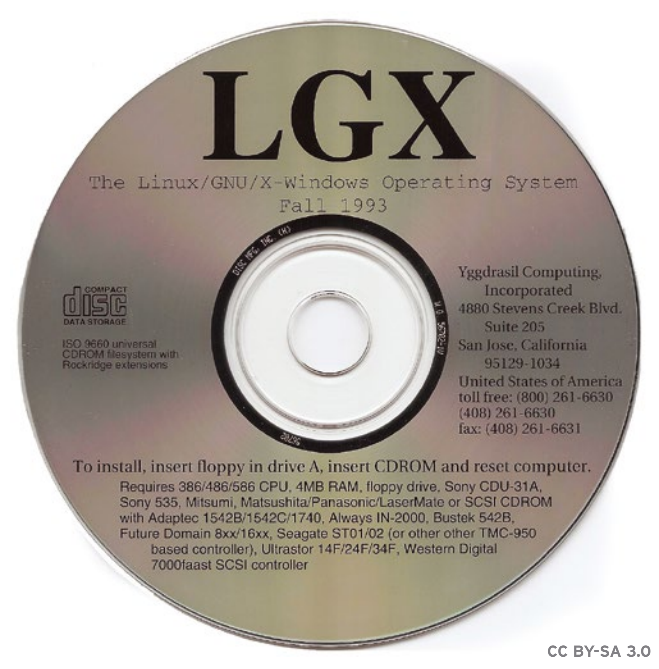
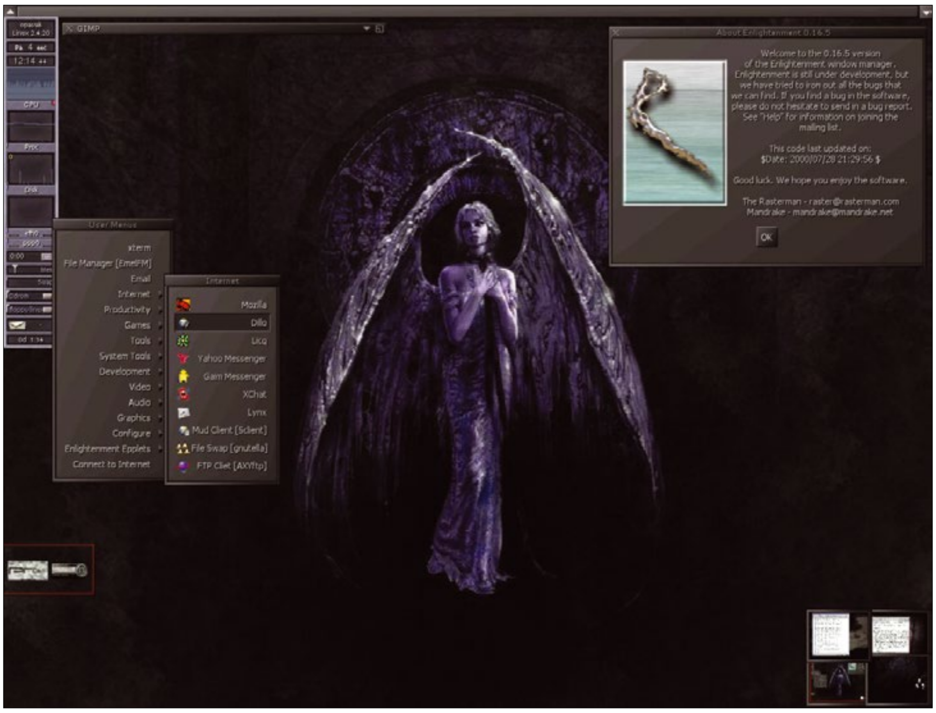

LGX
The Linux/GNU/X-Windows Operating System
Fall 1993
COMPACT disc DATA STORAGE
ISO 9660 universal CDROM filesystem with Rockridge extensions
Yggdrasil Computing, Incorporated
4880 Stevens Creek Blvd.
Suite 205
San Jose, California
95129-1034
United States of America
toll free: (800) 261-6630
(408) 261-6630
fax: (408) 261-6631
To install, insert floppy in drive A, insert CDROM and reset computer.
Requires 386/486/586 CPU, 4MB RAM, floppy drive, Sony CDU-31A, Sony 535, Mitsumi, Matsushita/Panasonic/LaserMate or SCSI CDROM with Adaptec 1542B/1542C/1740, Always IN-2000, Bustek 542B, Future Domain 8xx/16xx, Seagate ST01/02 (or other other TMC-950 based controller), Ultrastor 14F/24F/34F, Western Digital 7000faast SCSI controller
CC BY-SA 3.0

Figure 2: Yggdrasil was among the early Linux distributions, and could be launched from a CD.



Welcome to the 0.16.5 version
of the Enlightenment window manager.
Enlightenment is still under development, but
we have tried to iron out all the bugs that
we can find. If you find a bug in the software,
please do not hesitate to send in a bug report.
See "Help" for information on joining the
mailing list.

This code last updated on:
$Date: 2000/07/28 21:29:56 $

Good luck. We hope you enjoy the software.

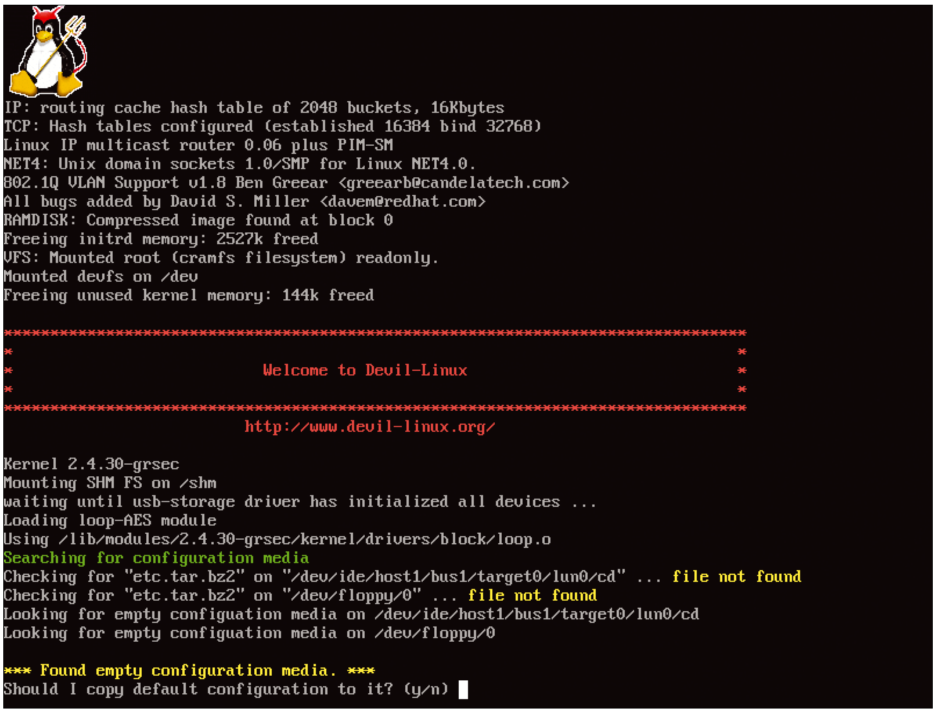
The Rasterman - raster@rasterman.com
Mandrake - mandrake@mandrake.net

Figure 3: A Czech blog still allows a retrospective look at Evil Entity, which once used Enlightenment as its desktop environment but is no longer among us.

plans for the future, but it turns out to be a dream for sysadmins with masochistic tendencies. In fact, Suicide Linux is a Debian package that configures your system so that every incorrectly typed command deletes all the data on the hard drive with an `rm -rf /`. Poisonous pedagogy [9] for Linux administrators.

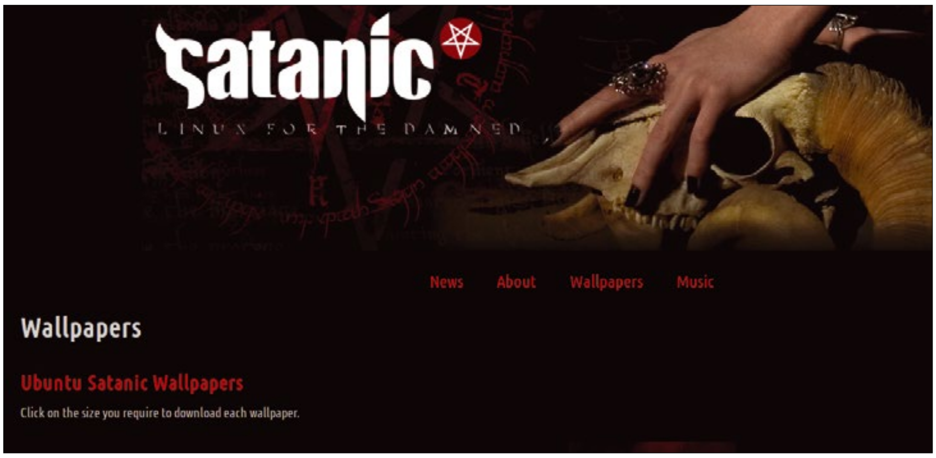
The name Devil-Linux [10] should also induce a hellish sweat among faint-hearted users, but completely without justification. The only diabolical thing about this lightweight Linux distribution, which boots from CDs or USB sticks, is that it is missing a graphical interface (Figure 4). However, that does not matter much; it serves experienced admins as a router and firewall system, and it also serves as a dedicated server for applications.

The focus is on security: Most binaries are compiled with GCC stack-smashing protection, while the kernel relies on grsecurity and PaX. Down to earth, the makers are currently working on version 1.8.0, so exorcists can stay home.



```
IP: routing cache hash table of 2048 buckets, 16Kbytes
TCP: Hash tables configured (established 16384 bind 32768)
Linux IP multicast router 0.06 plus PIM-SM
NET4: Unix domain sockets 1.0/SMP for Linux NET4.0.
802.1Q VLAN Support v1.8 Ben Greear <greearb@candelatech.com>
All bugs added by David S. Miller <davem@redhat.com>
RAMDISK: Compressed image found at block 0
Freeing initrd memory: 2527k freed
VFS: Mounted root (cramfs filesystem) readonly.
Mounted devfs on /dev
Freeing unused kernel memory: 144k freed

*******************************************************************************
*                                                                             *
*                         Welcome to Devil-Linux                              *
*                                                                             *
*******************************************************************************
                        http://www.devil-linux.org/

Kernel 2.4.30-grsec
Mounting SHM FS on /shm
waiting until usb-storage driver has initialized all devices ...
Loading loop-AES module
Using /lib/modules/2.4.30-grsec/kernel/drivers/block/loop.o
Searching for configuration media
Checking for "etc.tar.bz2" on "/dev/ide/host1/bus1/target0/lun0/cd" ... file not found
Checking for "etc.tar.bz2" on "/dev/floppy/0" ... file not found
Looking for empty configuation media on /dev/ide/host1/bus1/target0/lun0/cd
Looking for empty configuation media on /dev/floppy/0

*** Found empty configuration media. ***
Should I copy default configuration to it? (y/n)
```

Figure 4: Devil-Linux dispenses with a graphical interface when booting.



Wallpapers

Ubuntu Satanic Wallpapers

Click on the size you require to download each wallpaper.

Figure 5: The Ubuntu Satanic Edition was aimed predominantly at heavy metal fans, but it has unfortunately already started to rust.

Hellish Spectacle

If Devil-Linux [10] enters the religious realm in name only, other distros are really plunging in. At least, that is what you might think when the spotlight lands on Ubuntu Satanic Edition [11] (Figure 5).

Ubuntu Satanic is not a Linux for devil worshipers, however; in fact, one of the 666 set has already complained. Comments on the About page [12] include this note: "Being a Spiritual Satanist myself a lot of this I find offensive; maybe if you themed it more on satanism and less on dark images and heavy metal, I might be more interested." Hurt feelings aside, that quotation makes the main thrust of the Satanic Edition clear – it was aimed at heavy metal fans.

On the website, wallpaper with fire, skulls, and half-naked women await in random order, as well as heavy metal inspired music collections, bearing names like "Music for the Damned" or "Distro of the Beast." The whole thing was probably intended for humor; a link to the Undead Live CD (version 666.9) was left on the website with a completely outdated Ubuntu 10.10 and Gnome 2 for its desktop. Seriously diabolical!

Heaven Help!

Christians have it slightly better. DistroWatch admittedly considers the Ubuntu Christian Edition [13] (Figure 6) to be "sleeping," although the underlying Ubuntu 12.04 will keep receiving updates until April 2017.

DansGuardian ensures that, while surfing, children do not accidentally land at the aforementioned works of the devil; Christian tools like Xiphos (Bible studies) and OpenLP (presentation platform for churches) are on board. Most pilgrims to the website come from the United States, followed by Poland, incidentally.

Sabily was the name of the Ubuntu Muslim Edition produced in France [14], yet it rolled up the rug in 2011, quite likely because the individual tools found in the Muslim Edition can be installed without any trouble in the standard Ubuntu. According to Wikipedia [15], Sabily preinstalled Arabic language packs and provided programs for Koran studies, an Islamic calendar, and reminder software for prayer times.

Other versions of Sabily were planned, but in fact, Ubuntu 11.10 with Unity desktop is the last available version. Sabily no longer receives support, which is why DistroWatch deems the distribution dead. There is a slightly stuffy coun-

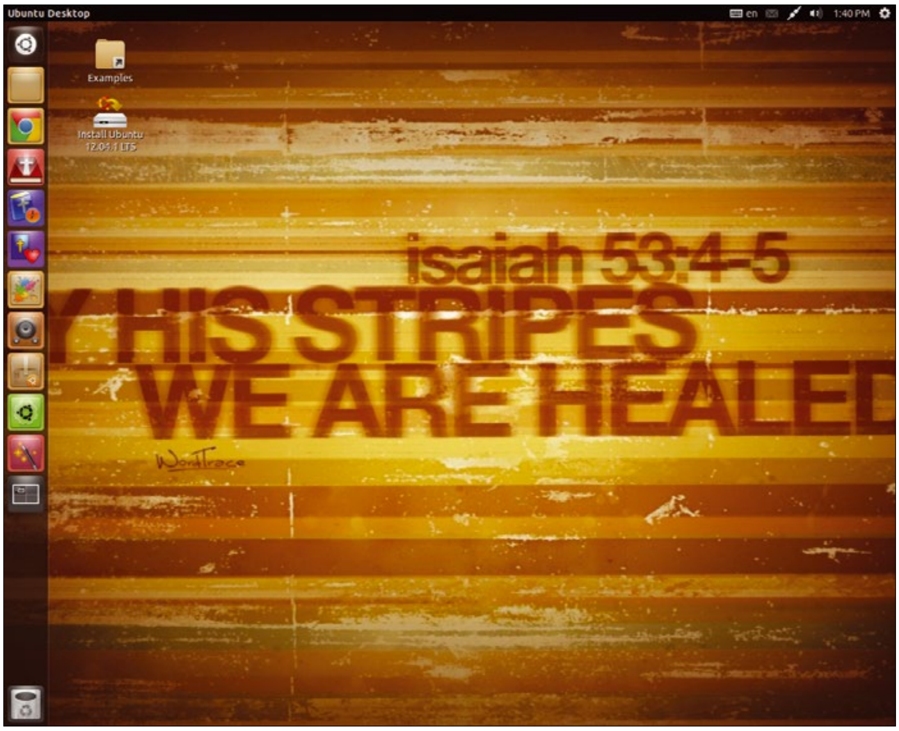


Figure 6: The Christian Edition offers a collection of Bible study software.

terpart to Sabily called Ojuba Linux [16], which is based on Fedora. The latest version of this Arabic distro with some Islamic tools is from March 2014.

Wellness and Pasta

Many of the secular Linux set regard Buddhism as wellness put into practice, but in the case of Bodhi Linux [17], it is not immediately apparent how much religion and wellness are embedded in the Ubuntu-based distribution. Bodhi comes from the same word root as Buddha and represents the Buddhist process of realization. Moksha Desktop (Figure 7) also travels this path, with the word approximately standing for salvation and enlightenment.

On the other hand, illumination is also implicit in the name of the Enlightenment 17 desktop, meaning that perhaps Moksha is just a translation of the word. In any case, Bodhi does not pre-install any special religious software, and the current version 3.2 was released in March 2016. That makes Buddha happy.

However, His Pastalike Divinity, the Flying Spaghetti Monster [18], could not look upon his followers and the landscape of Linux distributions with as much good will. The Linux for Pastafarians, once half-jokingly announced by Linux Format [19], will probably remain a parmesan-sprinkled wish. So far, no sieve-wielding developer or pirate has been found to gather together the central beliefs on a Linux desktop.

If Linux were some kind of religious competition, Buddhism would probably be ahead on points, since only Bodhi Linux awaits with an up-to-date creed.

Supervised Surfing with Kim Jong-Un

Religion and ideology are sometimes two sides of the same coin [20], which is reflected not only in calendars. Whereas Europeans count the year 2016 from the birth of Christ, North Koreans are in the year 105 after Kim Il-Sung's birth, according to the country's Juche political ideology, a fact to which the calendar of Red Star OS [21] pays tribute. Red Star OS is a Fedora-based Linux derivative, similar to OS X and is at home on the North Korean intranet. Because Kim Jong-Un does not allow his citizens onto the real Internet, we can only really speak of Red Star OS as a tool for supervised computer work.

Among the features of Red Star OS 3.0 is providing documents and images with a watermark that uses the encrypted hard disk serial number as its basis. In this way, the regime can understand which path a document takes through a series of computers. In case of doubt, the built-in virus scanner can also censor certain documents using a blacklist.

Hackers are not sure whether the system also includes back doors. Experts are not excluding the possibility that doors are hidden deep in the system or reach the computer via updates. Red Star OS developers have put a lot of brainpower into the system. Fans of ideologically charged art will also find some interesting wallpaper (Figure 8).

Under the Red Flag

For Red Flag Linux [22] (Figure 9), a Fedora-based Linux derivative developed in China since 1999, there are no reports of

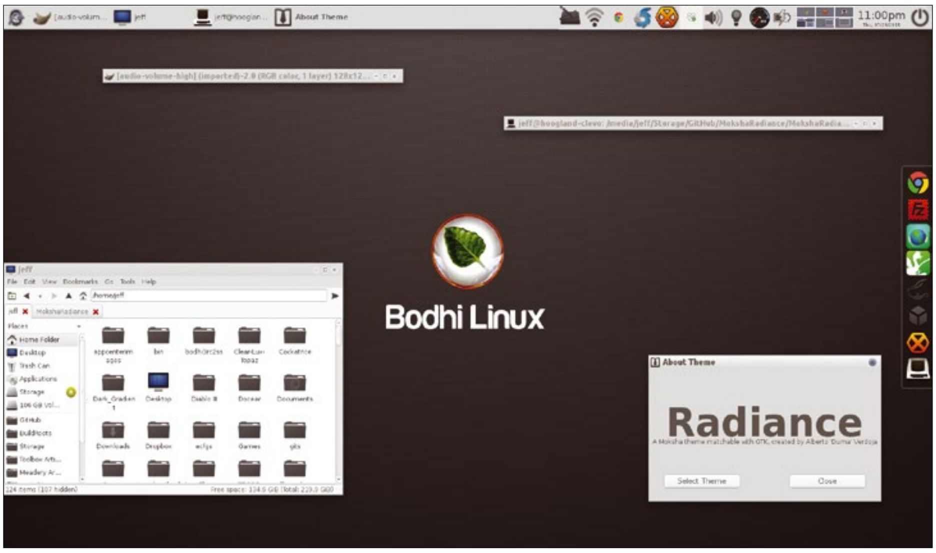


Figure 7: Moksha Radiance is one of the themes for the Moksha desktop.

Figure 8: Celebrating anti-aircraft guns: North Korea's Linux developers understand romance to the max.

built-in monitoring technologies. That said, the Associated Press reported in 2008 that the government wanted to force Internet cafés in Nanchang to switch from Windows to Red Flag. A spokesman for Red Flag declared, however, that their focus was on the server side in order to better manage user numbers and maintain virus protection.

To make China self-sufficient from Windows and, at the same time, maintain a secure operating system, the Academy of Sciences began to work on Red Flag Linux in 1999. In the following years, a suitable company emerged, several commercial releases appeared, and the company ended partnerships with Compaq, Oracle, and HP, among others. The distribution relied on KDE as a desktop, and it had an agreement with Trolltech to create a spin-off for embedded systems.

In 2006, Red Flag Linux was eventually thought to have had an 80% market share among Linux distros, although no one will ever be able to prove that statistic. Indeed, the bankruptcy of the Red Flag Software company, filed in 2014, leads to doubts about whether the Chinese government ever intended to support a nationwide Linux. An IDC analyst pointed to low recognition levels and the competition of Red Hat and SUSE Linux Enterprise as reasons for the firm going bust [23].

The demise of Red Flag did not mean Linux history came to an end in China, however. In 2014, the Chinese government instructed its authorities to no longer use Windows 8 on newly acquired machines. The new star in the Linux sky is Kylin [24], which is available in different versions. One version is even an official Ubuntu derivative.

Cuba Libre

Nova Linux [25], a Cuban flavor of Linux, is also based on Ubuntu, though it originally came from Gentoo. In 2009, Hector Rodriguez, Director of the University of Information Science in Havana (UCI), saw Linux expansion of 20% in Cuba and aimed for the 50% mark for 2014.

The Nova developers have also halted the operation, according to the DistroWatch entry, and the website is no longer accessible, although a new version 5.0 of Nova appeared in the fall of 2015 according to Softpedia [26], and UCI announced that it was switching to Nova 5.0 the end of May 2016.

Unlicensed versions of Windows do still exist in Cuba, but Cuba is hoping for a switch to Linux, at least for universities and public authorities. Even if the country could and would have paid the licensing fees to Microsoft, the long-standing embargo by the United States against Cuba prevented Windows from officially reaching the island. From 2015, the United States started easing the embargo.

We could not ascertain which Ubuntu version the Nova developers used for Nova Lightweight Edition 2015, though Ubuntu 14.04 may have been in line for the job. According to Softpedia, the installer should definitely simplify the installation, and it defaults to the Guano desktop environment, an LXDE fork. But, if the computers in Cuba are as old as the cars driving around there, even the slender Guano desktop should take some old gray Cuban computers to their performance limits.

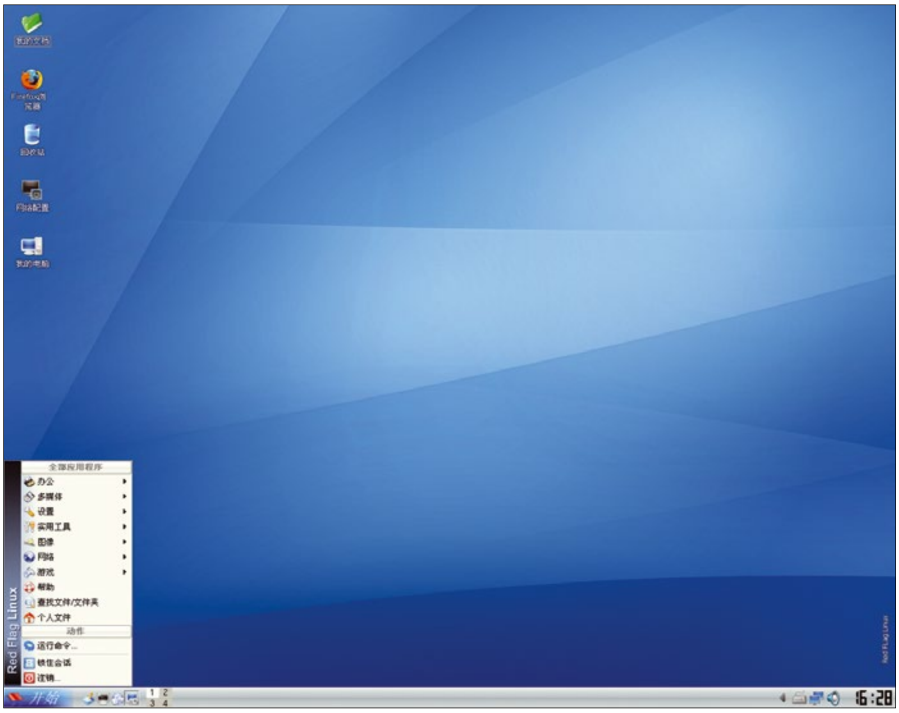


Figure 9: Red Flag Linux only flew its scarlet standard for a decade.

The Dear Little Things

No problem, there are also Linux distributions for old resource-challenged systems. One of the first, H. J. Lu's Boot/Root [27], ran on 5.25-inch floppy disks in 1991. To start the Linux from the hard drive, the user still had to readjust the master boot record by hand. The popularity of Boot/Root was therefore limited.

MCC Interim Linux [28], on the other hand, may be a concept familiar to a few Linux veterans; MCC stands for Manchester Computing Centre. The diskette collection provided a rudimentary Unix environment from 1992 onward. It's been a long, long time, but even in its smallest variant, the kernel would likely hardly fit on a floppy today.

This is probably the reason why Freesco [29], which promises to launch from a floppy disk with 1.44MB of space, relies on an old kernel. Freesco has nothing to do with the SCO Group [30] – the name stands for "Free Cisco." Freesco seeks a firmware alternative for commercial Cisco routers. To start the system, you just need to copy the `fresco.version_no` file to a floppy disk.

Freesco uses a really old 2.0 series kernel; the last version 0.4.5 comes from March 2014. Since then, incidentally, the main developer has been seeking a successor [31] to modernize the underlying library.

Slightly larger, but still puny as ever, is Tiny Core Linux (TCL) [32], whose latest 64-bit variants are available as Core Pure (12MB) and Tiny Core Pure (24MB). The latest editions bear version number 7.2 are from July 2016. TCL does not consider itself a stand-alone Linux distribution, instead, offering only a kernel with BusyBox and a basic FLTK desktop (Figure 10).

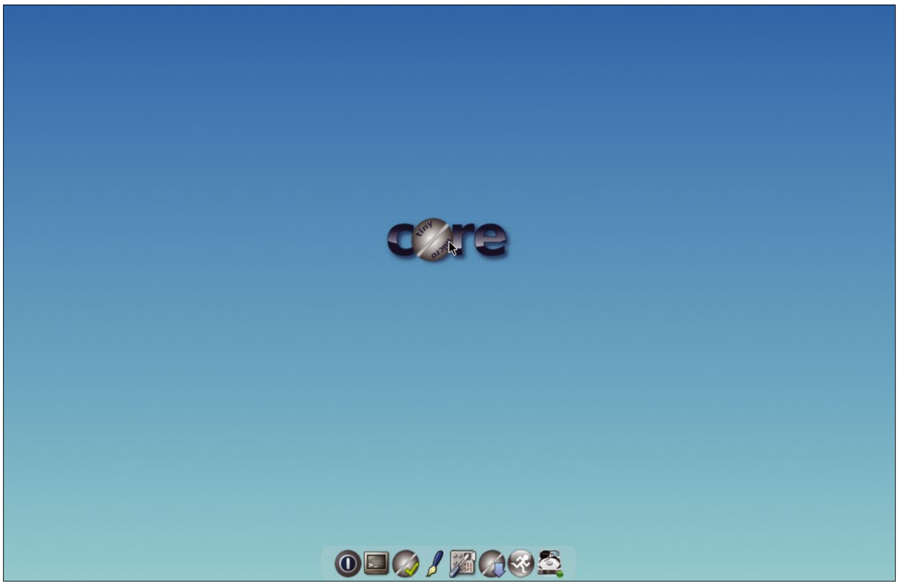


Figure 10: Tiny Core Linux is modular and needs very little space in its simplest variant.

Damn Small Linux [33] really is damn space-saving with its 50MB images, although the last release 4.4.11 dates back to September 27, 2012, and the website has not been accessible since.

The bootable image of the easily configured eisfair server system [34] with kernel 2.22.0 weighs in at only 54MB, but the system is still under active development. The ISOs of the latest

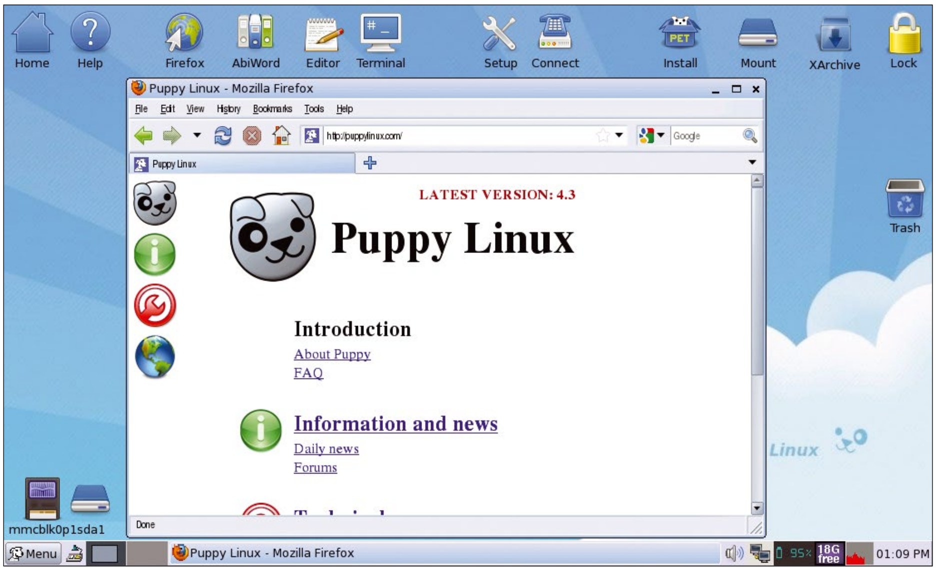


Figure 11: Puppy Linux, which is one of the classics among the Linux pipsqueaks, is named after a developer's chihuahua.

versions of its successor, eisfair-ng, from February, already take up 260MB (64-bit) or 190MB (32-bit). Several different eisfair editions exist today, including a version for the Raspberry Pi.

The Austrian distribution Grml [35], pronounced "grummel," is conspicuous by its name and its popularity with admins. Grml is a Debian-based rescue system, whose small 64-bit variant is 230MB in size. The latest version originates from November 2014. The developers posted an announcement in January 2016 declaring that Grml is still alive and explaining that a new release has been on hold pending resolution of a udev issue that "is visible only if you do not use systemd" [36].

The best-known mini Linux still goes by the name Puppy (Figure 11). Puppy Linux [37] is not a Linux distro, write the developers, but a collection of different Linux distributions. Distros include:

- Official Puppy Linux distributions, which the Puppy team maintains
- Alternative Puppy Linux distributions built for special purposes with the Woof distro-building tool
- Unofficial derivatives (called Puplets), mostly newly mastered by Puppy enthusiasts.

The ISOs vary in the low 200MB range. Usually, they offer a complete Linux desktop, albeit with very rudimentary programs. Puppy can make an old machine completely reusable. Incidentally, the name of the distribution is based on the cute little chihuahua of lead developer Barry Kauler, which, sadly, disappeared into the wilderness one day without a trace.

Gettin' Down on Friday

To avoid all this descending into deep sadness, Listing 1 shows some excerpts from a cheery little song called Friday.

The 13-year-old's song went viral, and the accompanying video [38] received the most negative reviews of all time. Friday was alternatively referred to as the "worst song of all time," "bizarre," and "awkward." That did not stop Nerdopolis, a fan of both of display manager Wayland and Rebecca Black [39], from giving his Wayland test distribution the name Rebecca Black OS [40].

The Debian-based compilation is available for 32- and 64-bit systems, and it boots to a live Wayland session. As well as Wayland toolkits and applications, Rebecca Black OS provides some Wayland-based desktop shells and compositors, such as Hawaii, Orbital, Papyros, and the Weston example desktop. The last version (4.7.16) appeared recently. The whole thing is really quite useful.

LISTING 1: Friday

```
[...]
Everybody's lookin' forward to the weekend, weekend
Friday, Friday
Gettin' down on Friday
Everybody's lookin' forward to the weekend

Partyin', partyin' (Yeah)
Partyin', partyin' (Yeah)
Fun, fun, fun, fun
Lookin' forward to the weekend
[...]
```

It may be Justin Bieber Linux (also called Biebian), but it still beats Windows and Mac.
Biebian is not based off Debian or Ubuntu. It is based off Puppy Lucid 525, and that is part of t
Also, this OS is a joke, inspired by Hannah Montana Linux. The idea began on 4chan's /g/. Anyone who takes it too serious

Support Justin Bieber Linux by adding this banner to your site!

Figure 12: The website of Justin Bieber Linux, which is based on a 4chan gag.

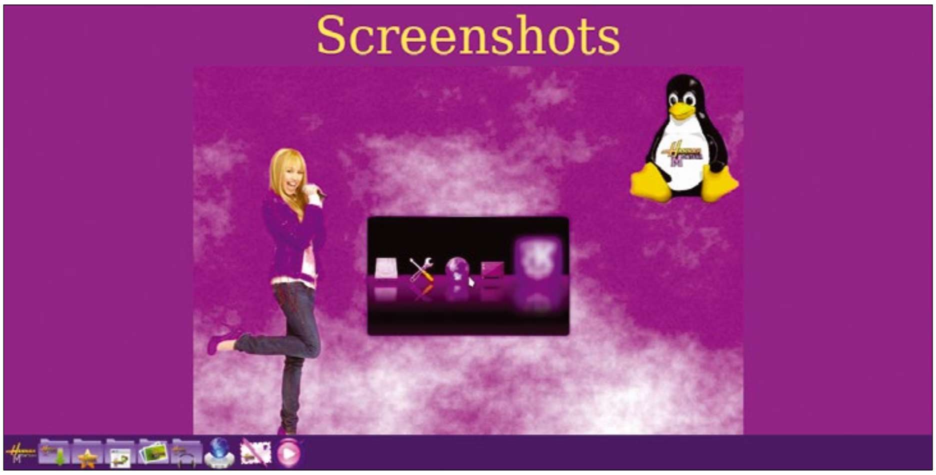


Figure 13: The makers of Hannah Montana Linux pretty well threw themselves into the task of creating their theme.

Bieberism

Useful is not the word for Justin Bieber Linux [41], aka Biebian, which was dedicated to the singing teen star (Figure 12). The nameless inventor of the distribution explains freely that his Linux is a joke. The idea originated on the 4chan board [42]. The sole difference between this distro and the normal Puppy Linux is that it includes some Justin Bieber wallpaper. Part of the joke was that the distribution is based on Puppy Linux Lucid 525, the developer explains. In the distribution year, 2011, Justin Bieber was just 17 years old, so a pup himself. There were never any updates for the distribution.

Biebian, however, was not the first distro that was dedicated to a teen star. The makers of Biebian cited the purple-tinted Hannah Montana Linux (HML) as its role model.

Sitcom Linux

The Disney sitcom *Hannah Montana*, starring Miley Cyrus, was filmed between 2006 and 2010 and enjoyed great popularity among a mostly young and predominantly female audience. According to a survey in 2011, the character was more popular with girls than Barbie. There were movies and video games, and even the Linux distribution, which was not intended to be serious.

HML [43] was based on Kubuntu and came with KDE 4.2. The theme work went much further than that of Biebian, and alongside wallpapers, it included the GRUB menu, boot splash, and a logon screen (Figure 13). The installation wizard greeted our sore eyes with purple and neon green, and even the icons were partly customized. A YouTube video [44] shows HML in all its glory, with its creator writing: “I installed this so you don’t have to.”

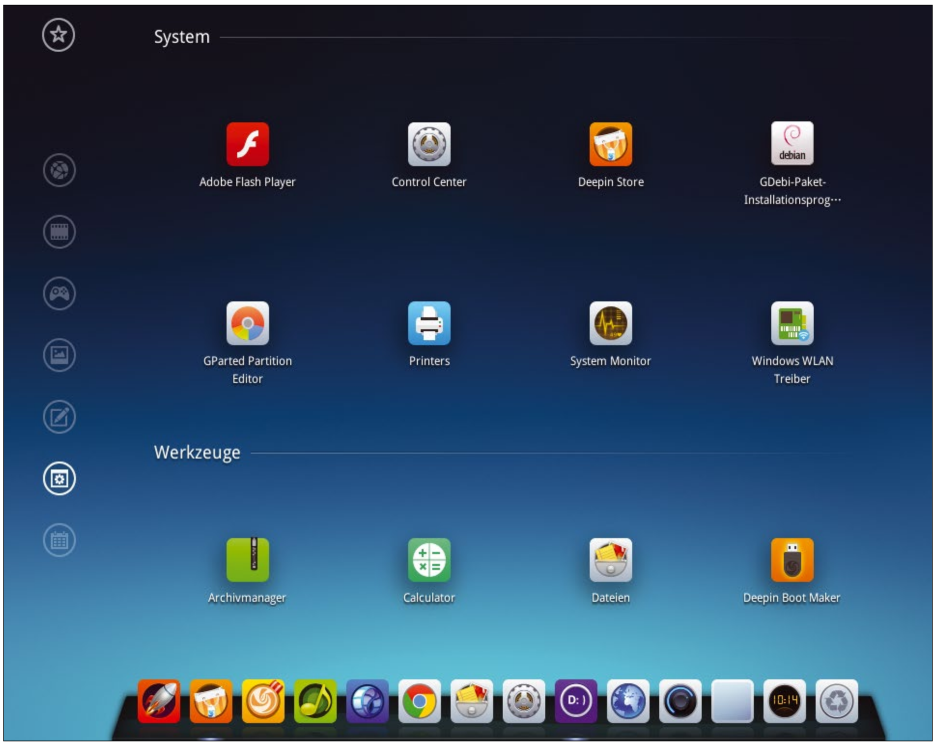


Figure 14: The makers of the Deepin Linux have developed their own desktop.

Life’s No One-Trick Pony

If you have gotten used to the graphical duress of Hannah Montana Linux, Pony OS 3.0 also shocks with its loud-colored pony background and corresponding icons. Pony OS is not a standard Linux, however, since the creators use a custom kernel, so a link to the video [45] should suffice.

Special Agents

Although the Linuxes mentioned all seem to fit into logical categories, there are still some odd and curiously named candidates that are quite unclassifiable.

What exactly the makers of Hiweed Linux 2004 were smoking can no longer be forensically determined. Because it is a Chinese distro, however, the name is probably based on a linguistic misunderstanding. Anyway, the Linux has since been called Deepin [46]; the current version is 15.2. Deepin is based on Debian Unstable, and it uses the Deepin Desktop Environment, a specially developed desktop (Figure 14). Additionally, the developers have also built many tools specifically for the distro.

Yellow Dog Linux [47] bore that name for some time because it ran not only on Apple’s power PC architecture, but also on Sony’s PlayStation 3. It was suitable for operation on clusters. The last known version, 7.0, was released in 2012 and could only be used in conjunction with a fairly expensive IBM PowerLinux 7R2 server, which currently costs around $20,000.

Can It Go Away?

Other Linux distributions have tried to move in new directions to alter features they don't like. Worth mentioning is GoboLinux [48], which seeks to use the filesystem as a database for the package manager and throw the known filesystem hierarchy (FSH) overboard. The developers are expecting "a clean system." GoboLinux uses Enlightenment as its desktop. The latest release was May 2014, although DistroWatch labels the distribution still active.

Another group of makers on a mission are the developers behind the Static Linux distro [49], or stali (not Stalin). Like the GoboLinux developers, the Stali team ignores the official filesystem hierarchy and avoids using systemd. At the same time – and this is probably the most important point – they only offer software as statically linked binary files for performance reasons. Where possible, musl libc is deployed to reduce the sizes of statically linked binaries. Stali should achieve a more effective footprint in the memory, since it dispenses with dynamic linking. At the same time, the project follows its own "suckless" philosophy, a system of principles designed to create "software that sucks less" [50].

Summary

As Justin Bieber Linux illustrates, any joker can build their own Linux distro without great hassle, in the simplest case, by changing background images. Critics have found fault with that repeatedly, but some positive effects emerge from it.

In this evolutionary process, many useful programs and innovative features are also created, whether they be new package managers, desktops, or init systems. It may be true, to stick with the evolutionary imagery, that you get a dodo from time to time. However, the particularly amazing thing is the sheer mass of Linux-based distributions that have seen the light of day in recent decades. And more are still appearing.

Also worth noting is that many of the existing systems do not want to address a wide mass of users at all. Instead, they are optimized to accomplish a specific purpose, fully within the spirit of the Unix philosophy: "Do one thing and do it well."

INFO

[1] Debian Wiki: *https://wiki.debian.org/WhyTheName*

[2] Linux family tree: *http://futurist.se/gldt/*

[3] Undead Linux, AKA Evil Entity: *https://distrowatch.com/table.php?distribution=evilentity*

[4] Void Linux: *http://www.voidlinux.eu*

[5] Obituary for Void Linux: *http://www.voidlinux.eu/news/2014/04/void-is-dead.html*

[6] Runit init system: *http://smarden.org/runit/*

[7] Damn Vulnerable Linux: *https://distrowatch.com/table.php?distribution=dvl*

[8] Suicide Linux: *https://qntm.org/suicide*

[9] Poisonous pedagogy: *https://en.wikipedia.org/wiki/Poisonous_pedagogy*

[10] Devil-Linux: *http://www.devil-linux.org*

[11] Ubuntu SE: *http://ubuntusatanic.org*

[12] Ubuntu SE about page: *http://ubuntusatanic.org/news/about/*

[13] Ubuntuce: *http://ubuntuce.com*

[14] Sabily: *https://distrowatch.com/sabily*

[15] Wikipedia entry on Sabily: *https://en.wikipedia.org/wiki/Sabily*

[16] Ojuba Linux: *https://en.wikipedia.org/wiki/Ojuba_Linux*

[17] Bodhi Linux: *http://www.bodhilinux.com*

[18] FSM: *https://en.wikipedia.org/wiki/Flying_Spaghetti_Monster*

[19] Linux for Pastafarians: *http://www.linuxformat.com/content/new-distro-announcement*

[20] Religion and ideology: *https://en.wikipedia.org/wiki/Ideology*

[21] Red Star OS: *https://en.wikipedia.org/wiki/Red_Star_OS*

[22] Red Flag Linux: *http://www.redflag-linux.com/en/about.php?class1=1&class2=2*

[23] Red Flag filing for bankruptcy: *http://www.zdnet.com/article/chinas-home-grown-linux-os-shutters/*

[24] Kylin: *https://en.wikipedia.org/wiki/Kylin_(operating_system)*

[25] Nova Linux: *http://www.nova.cu*; site is rarely accessible

[26] Nova 5.0: *http://news.softpedia.com/news/cuba-s-national-ubuntu-based-gnu-linux-os-to-get-a-gorgeous-lightweight-edition-491941.shtml*

[27] H J Lu's Boot/Root: *https://www.maketecheasier.com/history-of-early-linux-distros/*

[28] MCC Interim Linux: *https://en.wikipedia.org/wiki/MCC_Interim_Linux*

[29] Diskette Linux Freesco: *http://Freesco.sourceforge.NET*

[30] SCO Group: *https://en.wikipedia.org/wiki/SCO_Group*

[31] Successor search: *http://www.freesco.info/support-forum/viewtopic.php?f=1&t=17703*

[32] Tiny Core Linux: *http://tinycorelinux.net*

[33] DSL: *http://www.damnsmalllinux.org/*

[34] eisfair: *http://www.eisfair.org/en/eisfair/the-eisfair-project/*

[35] Grml: *https://grml.org*

[36] Grml lives on: *http://blog.grml.org/archives/394-Is-Grml-still-alive.html*

[37] Puppy: *http://puppylinux.org*

[38] Friday video: *https://www.youtube.com/watch?v=kfVsfOSbJY0*

[39] Rebecca Black: *https://en.wikipedia.org/wiki/Rebecca_Black*

[40] Rebecca Black OS: *https://sourceforge.net/projects/rebeccablackos/*

[41] Justin Bieber Linux: *http://biebian.sourceforge.net*

[42] Biebian announcement: *http://www.murga-linux.com/puppy/viewtopic.php?t=68377*

[43] Hannah Montana Linux: *http://hannahmontana.sourceforge.net*

[44] HML on video: *https://www.youtube.com/watch?v=aFbKja6rKm0*

[45] Pony OS: *https://www.youtube.com/watch?v=ssEiNhD_5Dk*

[46] Deepin Linux: *https://www.deepin.org*

[47] Yellow Dog Linux: *https://en.wikipedia.org/wiki/Yellow_Dog_Linux*

[48] GoboLinux: *http://www.gobolinux.org*

[49] Static Linux distribution: *http://sta.li*

[50] Suckless philosophy: *http://suckless.org/philosophy*

Charting expletives from the Linux Kernel Mailing List

Climate Study

Kernel amateurs are best advised to read summaries of the heated discussions on the Linux Kernel Mailing List (LKML) before they delve in. We analyze 2.5 million postings to study the density of cursing. *By Mike Schilli and Joe Casad*

Every now and then, a message reaches social media that Linux boss Linus Torvalds has flipped out once again and dressed down kernel colleagues with rude words. Some Linux enthusiasts look on this with amusement, enjoying the tirades of the great dictator over a cool drink after work; others see the harsh nature of the language as representing an intimidating boy's club culture that privileges insiders.

The issue of language on the kernel list has been in the foreground for the last few years. In 2013, Intel developer Sarah Sharp led an effort to improve civility among kernel developers [1], and Red Hat's Lennart Poettering has also spoken up for more politeness and less abusive language [2].

In 2015, Linus responded to criticism by posting a Code of Conflict [3] that affirms the need for civility in the code review process, instructing developers to contact the Linux Foundation's Technical Advisory Board if they feel the process is threatening or abusive, and ending with a directive to not let things get personal:

As a reviewer of code, please strive to keep things civil and focused on the technical issues involved. We are all humans, and frustrations can be high on both sides of the process. Try to keep in mind the immortal words of Bill and Ted, "Be excellent to each other."

Whether you favor the harsh language of some on the kernel list, or whether you still see room for reform, you might have noticed that most of the discussion centers around anecdotes and opinions – no one ever seems to quantify it.

We decided to work through this phenomenon mathematically. For the dataset, we used 2.5 million LKML posts, which were first fed into a MySQL database, and then beaten with Perl and R scripts and presented graphically.

Figure 1 demonstrates the development of the LKML by means of the number of posts over 20 years from 1996 to the present day, with the start of 2016 projected proportionally. The almost linear increase, from 20,000 posts in 1996 to an estimated figure exceeding 270,000 for the current year of 2016, is evidence of the natural growth of the project and its uninterrupted popularity.

Long Tail

What about the number of members; do most of the posts come from a few extra active highfliers, and the rest as a long tail of Linux hobbyists who only write once or twice a year? An R script reads the metadata re-exported from MySQL into CSV format and prints the graphic in Figure 2.

It turns out that a few top posters over the decades have fired off more than 30,000 emails; a few dozen members, Torvalds himself among them, more than 10,000; and then around another 100 have exceeded 5,000. As expected, the curve levels off on its right side.

Lead Image © Cornelius, Fotolia.com

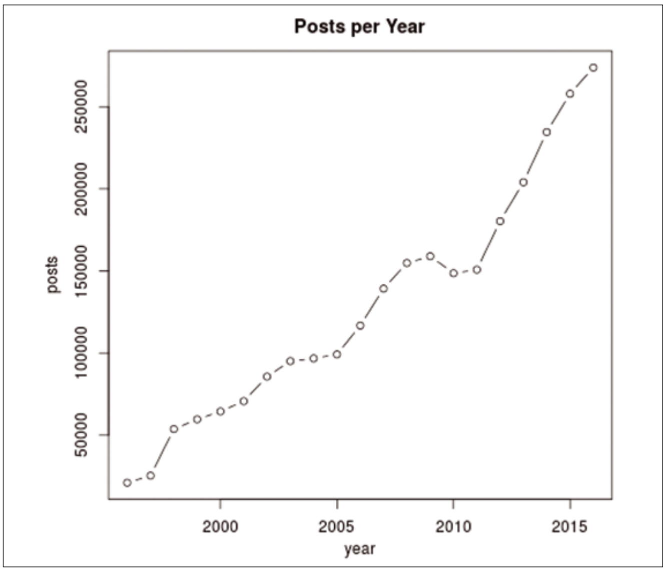


Figure 1: Number of posts on the LKML over 20 years.

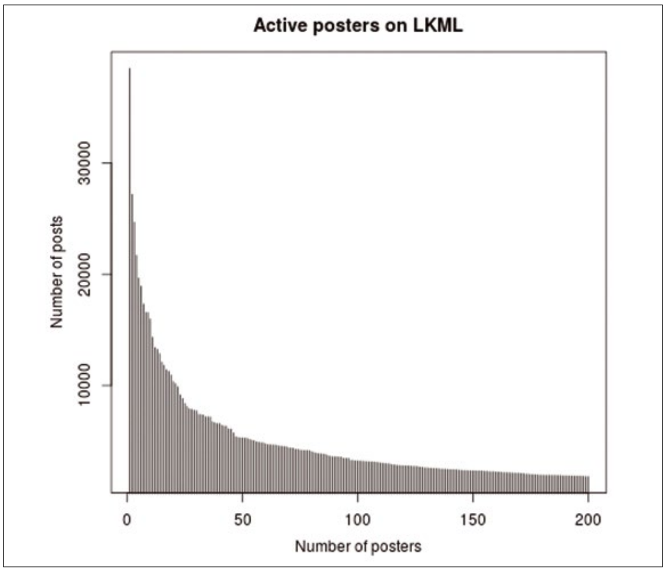


Figure 2: A few dozen members contribute more than 10,000 posts, and a few dozen more, more than 30,000.

Expletives

Before entering analysis of civility on the LKML, it is necessary to clarify when exactly a word is a swear word. Clearly, what is considered profane depends strongly on the cultural environment. One possible approach is offered by the gold standard prevailing in the US: the "Seven Words You can Never Say on Television" compiled by the comedian George Carlin in 1972, referencing words that no publicly aired television or radio stations in the US could send into the ether without first masking them with an annoying 1kHz sound [4] (subscription channels like HBO are the exception).

You can probably guess most of the seven words, which, predictably, center on sex acts, body parts, and bodily functions, but if you have any questions, search for the "seven dirty words" on Wikipedia [5]. If you do not know them all, you are very welcome to use an online dictionary on your own for clarification, but please only do this with your browser set to "incognito" mode.

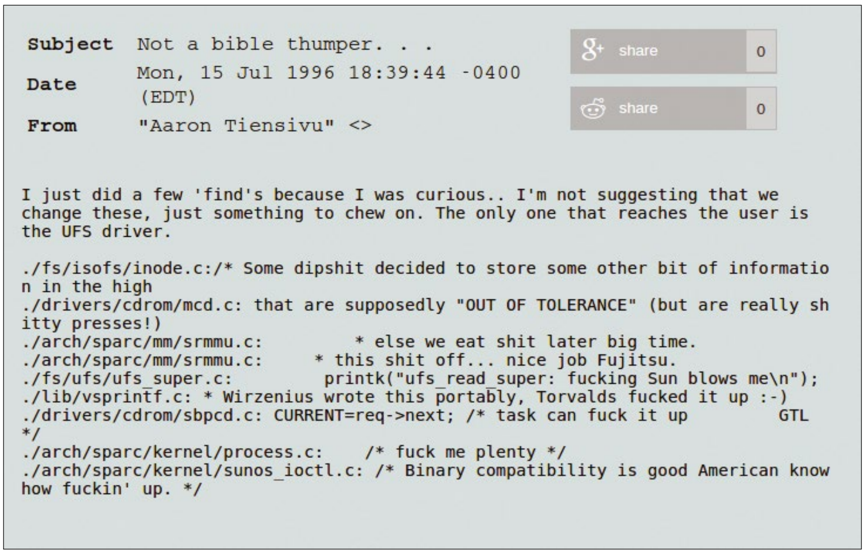
The CPAN Perl module Regexp::Common is available to determine whether a text includes one of the vulgarities; it searches for them at lightning speed with regular expressions using the `profanity` key. The filter, however, will not find coded phrasings or blanked-out words such as *f*ck*; the regular expressions would have to be expanded for this.

But it also finds words that sound offensive to European ears. While an American might think nothing of the expression "a bunch of crap," except perhaps to find it funny depending on the context, Her Britannic Majesty might not be amused at high tea.

If you use regexes to trawl through the historic contributions to the LKML by Linus Torvalds, the filter jumps to July 1996 for the first instance. The member Aaron Tiensivu had written, under the title "Not a Bible Thumper," that the most amazing profanities were concealed in the kernel code (Figure 3). The discussion took its course until Torvalds exercised his authority and stated that, although he was opposed to political correctness, he also didn't see a point in being intentionally rude for no reason, adding ambiguously, "The reason the active kernel messages should be nice is that while I hate politically correct, I do not believe in being actively offensive either except when I _want_ to offend somebody. And there is no point in offending the occasional user."

More recently, Torvalds has also not shied away from arguing with a coarse tone that, if used against work colleagues in an American company, probably would have seen the HR department called to the scene immediately. At the end of 2012, he berated a maintainer who had not, in his opinion, understood the first rule of kernel maintenance: "We do not break userspace." He told the maintainer to "shut the fuck up"; a kernel change that causes problems for a userland program would always be a bug in the kernel (Figure 4).

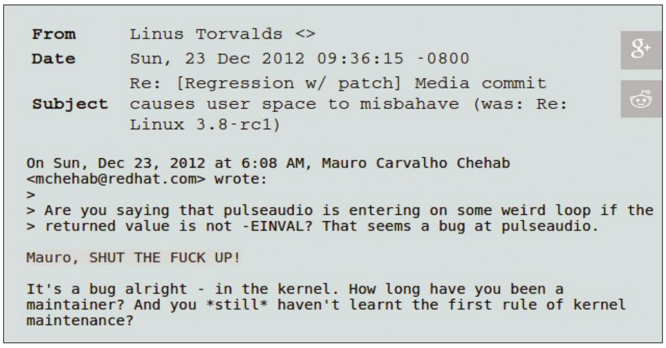
What has been the historical development of profanities on the LKML? Figure 5 shows that there were two peaks in 2000



Subject	Not a bible thumper. . .
Date	Mon, 15 Jul 1996 18:39:44 -0400 (EDT)
From	"Aaron Tiensivu" <>

```
I just did a few 'find's because I was curious.. I'm not suggesting that we
change these, just something to chew on. The only one that reaches the user is
the UFS driver.

./fs/isofs/inode.c:/* Some dipshit decided to store some other bit of informatio
n in the high
./drivers/cdrom/mcd.c: that are supposedly "OUT OF TOLERANCE" (but are really sh
itty presses!)
./arch/sparc/mm/srmmu.c:          * else we eat shit later big time.
./arch/sparc/mm/srmmu.c:      * this shit off... nice job Fujitsu.
./fs/ufs/ufs_super.c:          printk("ufs_read_super: fucking Sun blows me\n");
./lib/vsprintf.c: * Wirzenius wrote this portably, Torvalds fucked it up :-)
./drivers/cdrom/sbpcd.c: CURRENT=req->next; /* task can fuck it up          GTL
*/
./arch/sparc/kernel/process.c:     /* fuck me plenty */
./arch/sparc/kernel/sunos_ioctl.c: /* Binary compatibility is good American know
how fuckin' up. */
```

Figure 3: A post denounces curse words used in the kernel code.



```
From     Linus Torvalds <>
Date     Sun, 23 Dec 2012 09:36:15 -0800
         Re: [Regression w/ patch] Media commit
Subject  causes user space to misbahave (was: Re:
         Linux 3.8-rc1)

On Sun, Dec 23, 2012 at 6:08 AM, Mauro Carvalho Chehab
<mchehab@redhat.com> wrote:
>
> Are you saying that pulseaudio is entering on some weird loop if the
> returned value is not -EINVAL? That seems a bug at pulseaudio.

Mauro, SHUT THE FUCK UP!

It's a bug alright - in the kernel. How long have you been a
maintainer? And you *still* haven't learnt the first rule of kernel
maintenance?
```

Figure 4: Linus Torvalds goes after a maintainer.

and 2008 with around 1,200 expletive emails, with the last decade exhibiting a strongly falling trend. Taking into account that the number of postings per year is constantly increasing, the potty-mouth count is dropping significantly. However, the figure for 2016 only shows the postings up to July, so the adjusted figure would probably be around the 2015 level.

Who uses the most swear words? Listing 1 shows how many posts the ten biggest boors sent out. At the top is the dictator himself. The list includes a number of non-native speakers – in my experience, non-natives often fling around expletives in English with little sensitivity to disguise their limited vocabulary. That said, the top 10 also enshrines some native English speakers.

LISTING 1: Top Swearers

01	Linus Torvalds	1308
02	Alexander Viro	759
03	Peter Zijlstra	548
04	Rik van Riel	397
05	Thomas Gleixner	324
06	Alan Cox	322
07	Andrew Morton	278
08	Ingo Molnar	250
09	Christoph Hellwig	243
10	Benjamin Herrenschmidt	180

What range of words do the maintainers use during their stressful work? Nothing out of the ordinary, as you can see from the pie chart in Figure 6: The list fits pretty closely with the usual repertoire of the American construction worker. The clear favorite is the word "crap."

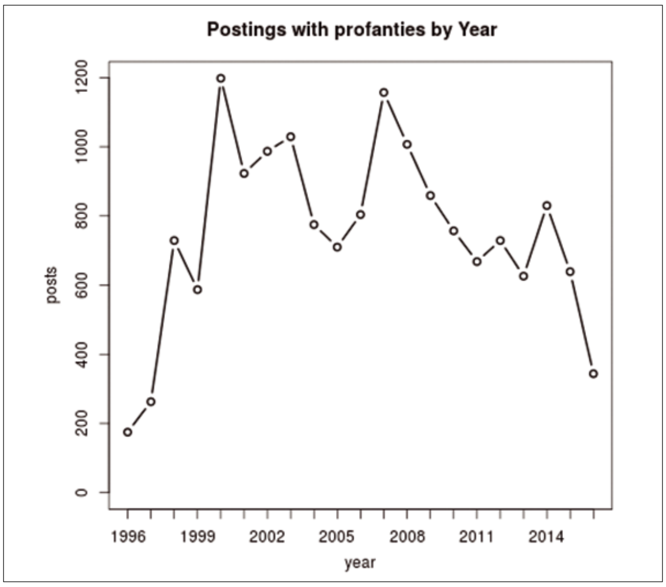


Figure 5: Post count with foul language over the years.

Conclusion

When used in moderation, a strong word can definitely prevent any possible misunderstandings. Linus has said his use of language is intended to keep developers alert and doing their best work – to fix the problems first before sending problematic code up the development tree. On the other hand, Linux bills itself as a meritocracy, and if worthy and potentially productive programmers are choosing not to participate because they are put off by intimidating and sometimes abusive language, the result is a loss for Linux.

Of course, the study described in this article does not attempt to uncover intimidation or abuse but is only searching for the presence of words. As Sarah Sharp points out in a 2013 kernel list post summarizing her position [7], it is possible to use obscenities in a way that is not personally abusive. Saying "If you give a flying fuck about diversity, you should avoid verbal abuse" is not the same as saying "SHUT THE FUCK UP."

Still, real numbers offer real insights into the use of language on the kernel list, and the fact that foul language is on a downward trend should be of some comfort to those who argue for better word choice. ∎∎∎

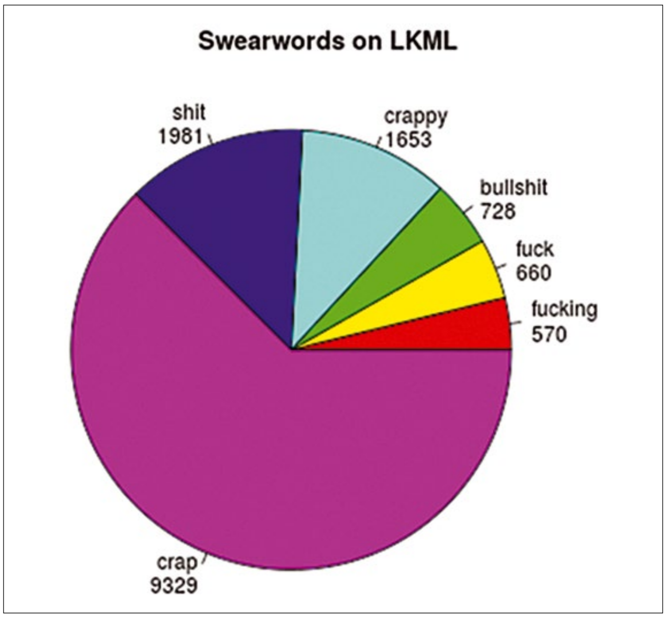


Figure 6: The most popular expletives on the mailing list.

INFO

[1] Sarah Sharp post on civility: *https://lkml.org/lkml/2013/7/15/329*

[2] Lennart Poettering post on civility: *https://plus.google.com/app/basic/stream/z13rdjryqyn1xlt3522sxpugoz3gujbhh04*

[3] Linux Code of Conflict: *https://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/commit/?id=b0bc65729070b9cbdbb53ff042984a3c545a0e34*

[4] Bleep censor: *https://en.wikipedia.org/wiki/Bleep_censor*

[5] Seven Dirty Words: *https://en.wikipedia.org/wiki/Seven_dirty_words*

[6] Linus Torvalds, "Re: Not a bible thumper. . .": *https://lkml.org/lkml/1996/7/20/1*

[7] Sarah Sharp's summary: *https://lkml.org/lkml/2013/7/19/634*

[8] Listings for this article: *ftp://www.linux-magazine.com/pub/listings/magazine/192/Perl*

EmperorLinux

...where Linux & laptops converge

Powerful Linux: The Rhino

- Based on the Dell Precision M7510/M7710
- High performance NVidia 3-D on a 4K/UHD LCD
- High performance Core i7/E3 Quad CPU, 64 GB ECC RAM

- Ultimate configurability — choose your laptop's features:

- 2.7-2.9 GHz Core i7 Quad (8 core) or 2.8-3.0 GHz E3 Xeon (ECC support)
- 15.6" or 17.3" 4K/UHD LCD w/ X@3840x2160
- Up to NVidia Quadro M5000M graphics, 1536 core CUDA
- 256 GB-1 TB NVMe/PCIe solid state drive (up to 3 drives)
- Up to 64 GB RAM (DDR4 2133 MHz)
- Ethernet, 802.11a/g/n, Bluetooth
- One year Linux tech support - phone and email
- Three year manufacturer's on-site warranty
- Choice of pre-installed Linux distribution:

Rugged Linux

Tarantula CF-31

- Panasonic Toughbook CF-31
- Fully rugged MIL-SPEC-810G tested: drops, dust, moisture & more
- 13.1" XGA touchscreen
- Up to 2.6 GHz Core i7 Duo
- Up to 16 GB RAM
- 500 GB hard drive / 512 GB SSD
- 4G/LTE, GPS, CAC card
- **CF-19 and CF-54 also available**

Portable Linux

Toucan T460/T460s

- ThinkPad T460/T460s by Lenovo
- Up to 14" WQHD w/ X@2560x1440
- 2.6 GHz Core i7 Duo
- Up to 32 GB RAM
- 500 GB hard drive / 1 TB SSD
- Mobile broadband 4G/LTE
- 3.0-3.8 pounds
- **P50 (4K/UHD) and X1 Yoga tablet also available**

Tablet Linux

Ant FZ-G1

- Panasonic Toughbook FZ-G1
- Fully rugged MIL-SPEC-810G tested: drops, dust, moisture & more
- 10.1" WUXGA touchscreen & stylus
- 2.3 GHz Core i5 Duo
- 8 GB RAM
- 128 GB-1 TB SSD
- LAN, GPS, serial, SD, CAC card
- **7" FZ-M1 and 12.5" FZ-Q1 available**

www.EmperorLinux.com 1-888-651-6686

Model prices, specifications, and availability may vary. All trademarks are the property of their respective owners.

MythTV, Kodi, Plex, OSMC, and DVBLink tested

Expedited TV

Free and commercial media center programs promise streaming HD videos, television, music, picture galleries, and a few extra tricks for Linux PCs and the Raspberry Pi – all from the comfort of your living room. *By Erik Bärwaldt*

Computer technology and consumer electronics are growing closer and closer, so it is no surprise that, in addition to the traditional multimedia computer programs, software suites now convert computers into home theater centers when combined with an HDMI-compatible TV or projector and a powerful sound system.

Media lovers can do more than just look at pictures and watch movies, they can listen to music, stream content from the web, and scroll through various TV offerings. The recording and editing functions leave little to be desired, but conventional DVD players, digital video recorders, and hi-fi systems might run into difficulties.

The large selection of home theater software for Linux begs the question: Which media center is suitable for me? To help make the choice easier, I compared five leading commercial, free media center solutions that cover almost

TABLE 1: Functional Overview of Media Centers

	MythTV	Kodi	Plex Media Server	OSMC	DVBLink
DVB-T/T2	Yes	Yes	Yes	Yes	Yes
DVB-S/S2	Yes	Yes	Yes	Yes	Yes
DVB-C	Yes	Yes	Yes	Yes	Yes
Analog support	Yes	Yes	Yes	Yes	No
HD support	Yes	Yes	Yes	Yes	Yes
Hides Ads	Yes	Limited	Limited	Limited	No
Schedules recordings	Yes	Yes	Yes	Yes	Yes
Electronic program guide (EPG)	Yes	Yes	Yes	Yes	Yes
DVD view and archive	Yes	Yes	Yes	Yes	No
Child protection	Yes	Yes	Yes	Yes	No
Digital music collections	Yes	Yes	Yes	Yes	No
Image collections	Yes	Yes	Yes	Yes	No
Client-server structure	Yes	Yes	Yes	Yes	Yes
Web administration	Yes	Yes	Yes	Yes	Yes
Add-ons	Yes	Yes	Yes	Yes	Yes (clients)
Mobile apps	Unofficial	Yes	Yes	Yes	Yes
Remote control	Yes	Yes	Yes	Yes	Yes

Lead Image © Paolo De Santis, 123RF.com

every need: MythTV [1], Kodi [2], Plex Media Server [3], OSMC [4], and DVBLink [5] (Table 1).

The Basics

If you want your future media center PC to receive TV programs, you first need a corresponding TV tuner with Linux support. Some common DVB receivers won't run in Linux for want of appropriate modules, so it is worth checking in advance whether Linux provides drivers for the desired device. The Linux TV Project wiki [6], which has a comprehensive hardware database, serves as a first point of contact. It also includes valuable installation tips for users in some countries who need to integrate proprietary firmware for DVB and analog TV components.

Another stumbling block appears when you try to install media center applications manually on an existing Linux system. Because the multimedia software usually works as a client-server application, it requires the usual suspects, such as the Apache web server, PHP7, and the MySQL database back end, which can cause problems for novices and irritate experienced users alike. Both groups are advised to use dedicated media center distributions, because they preconfigure and coordinate the individual components optimally. Specially adapted installation routines often integrate network services into the system, as well.

An Ubuntu Basis

Ubuntu is often used as the basis for these distributions. It is widely used and has very good driver coverage thanks to the numerous Ubuntu and Debian developers. Moreover, the developers maintain the long-term support (LTS) variants over a relatively long period of time so that a media center PC has a long-term stable foundation.

Ubuntu's graphical installation wizard Ubiquity offers you the option of integrating third-party software in the system at install time, including the MP3 plugin and Adobe's Flash technology, which is gradually becoming outdated and prone to serious security flaws. If you don't want to install proprietary codecs later, you should take advantage of this option.

MythTV

MythTV is one of the oldest media center applications and has been constantly developed and maintained since 2002. Released under the GPL, MythTV is aimed more at advanced users and has evolved from a pure disc recorder for television programs into a full-fledged media center by virtue of its sophisticated modular design. Not only does it cope well with analog and digital TV receivers, it also integrates web content and locally stored data.

MythTV is network-enabled and works according on the client-server model. It can burn stored content onto optical discs. In addition to the actual core MythTV programs, many Linux distributions also provide modules for expanding the media center. The current version is available for download as source [7]. Elsewhere on the website, you can find manuals and further information about integrating MythTV [8] into the various Linux distributions. As a complete package with excellent preconfiguration, you can get MythTV in the distributions MythBuntu [9] and LinHES [10], which is based on Arch Linux.

Anyone who uses Ubuntu and its derivatives can install the package in one go using the command:

```
sudo apt-get install ⮐
  mythbuntu-control-centre
```

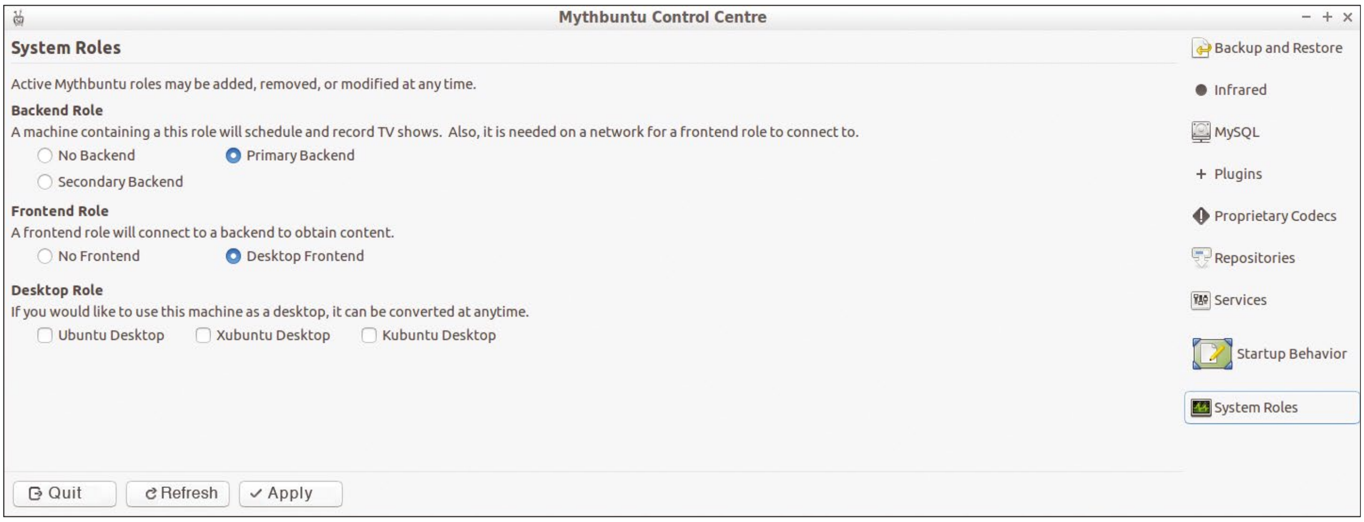
It automatically tracks down the required Apache web server, PHP7, and MySQL Server 5.7 and creates corresponding menu entries. Simple dialogs in the terminal allow for a rough basic configuration.

Role Play

MythTV works with a back-end server and a front-end client. Whereas the back end coordinates the internal work processes in the background, the front end serves as a user interface. A PC can take on multiple roles.

The most convenient way to configure the media center is still via MythTV's own control center. After installation, you just need to call the *System* | *MythBuntu Control Centre* or, depending on the desktop, *System Tools* | *MythBuntu Control Centre*. (Henceforth I will use *System*) The *System Roles* entry is at the bottom right; it lets you define the role of the current system. If you want the home theater PC to work as the sole media center, you need to activate the *Primary Backend* and *Desktop Frontend* roles (Figure 1).

Now you should set up the back end in line with the role definition, which



System Roles

Active Mythbuntu roles may be added, removed, or modified at any time.

Backend Role

A machine containing a this role will schedule and record TV shows. Also, it is needed on a network for a frontend role to connect to.

○ No Backend ● Primary Backend

○ Secondary Backend

Frontend Role

A frontend role will connect to a backend to obtain content.

○ No Frontend ● Desktop Frontend

Desktop Role

If you would like to use this machine as a desktop, it can be converted at anytime.

□ Ubuntu Desktop □ Xubuntu Desktop □ Kubuntu Desktop

Figure 1: Define various functional roles for you media center PC in the MythTV Control Centre.

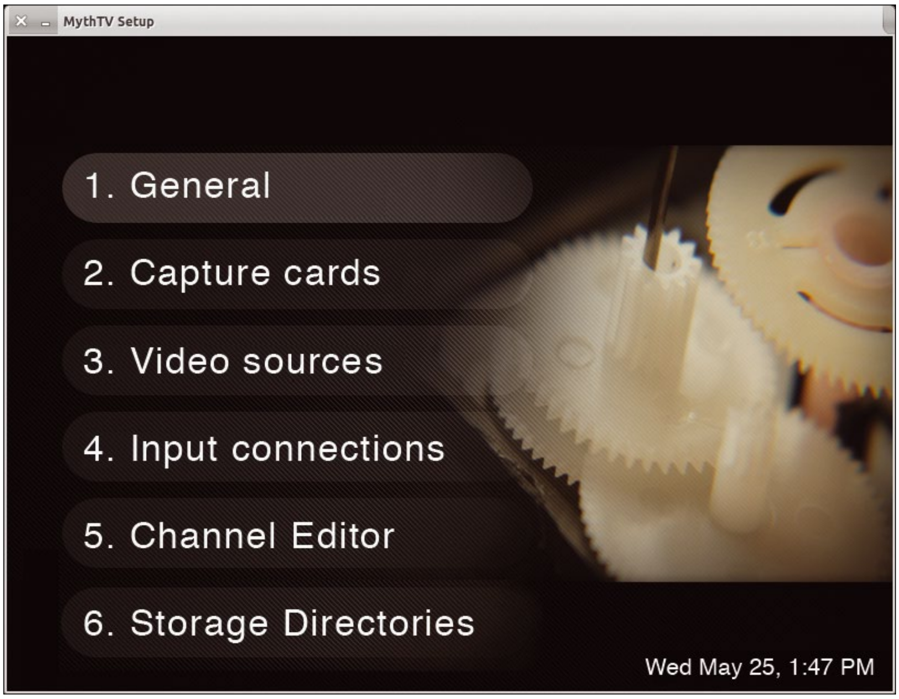


Figure 2: MythTV is configured in the Control Centre.

hides behind *System* | *Administration* | *MythTV Backend Setup* on the desktop. Next, you should carefully configure the *Capture cards*, *Video sources*, *Input connections*, *Channel Editor*, and *Storage Directories* (Figure 2). Although MythTV usually correctly recognizes and addresses TV tuners connected to or in the system, you need to set the video sources themselves and link them to your tuner.

If you have multiple cards attached to your system, you can separate the sources, thanks to this construct. The automated scan for tuners makes work easier and defines paths for the recording function. After completing the rather painstaking configuration, you can start the back end and create the database.

To start the front end, from the desktop click *Applications* | *Multimedia* | *MythTV Frontend*. The media center's complete multimedia performance spectrum is now available on the interface (Figure 3). MythTV provides options to listen to radio stations; watch videos on CD, DVD, or the hard drive; play audio CDs; and look at pictures. The options in the main window in turn branch out into context-sensitive submenus so that no menu hierarchy is overloaded.

The media center supports several remote controls. They use the Linux infrared remote control (LIRC) routines [11] that support several IR controls without any additional configuration.

Kodi

Kodi is probably the best-known media center project in the Linux universe, although it may be more familiar under its old name, XBMC. Kodi, like MythTV, is included in many software repositories of large Linux derivatives and as Kodibuntu [12] in the form of a unique 64-bit-only distribution based on Ubuntu 14.04.

The Kodi project maintains dedicated repositories, so it is worth installing the latest version of the software from these repositories. Alternatively, you can update an older Kodibuntu installation to the latest version of the media center [13].

To install Kodi in Ubuntu, you need to add and update the corresponding repository, install the media center, and trace all necessary dependencies:

```
sudo add-apt-repository ⮯
  ppa:team-xbmc/ppa
sudo apt-get update
sudo apt-get install kodi
sudo apt-get install kodi-pvr-*
```

The final command line installs the add-on clients needed to make use of TV recordings, program playback, and the electronic program guide (EPG).

A corresponding server is used as a back end, for which Kodi provides numerous server alternatives. You can configure the server immediately during installation or use a remote control (Figure 4).

Commissioning

Now you should be able to access the software from the desktop menu *Sound & Video* | *Kodi media center* menu. Several media groups are preset in the media center. Like MythTV, Kodi also plays all common media types. After opening it for the first time, you can select the language in the settings menu. If a TV card is attached to the computer and the personal video recorder (PVR) back ends and front ends are installed, you can configure and

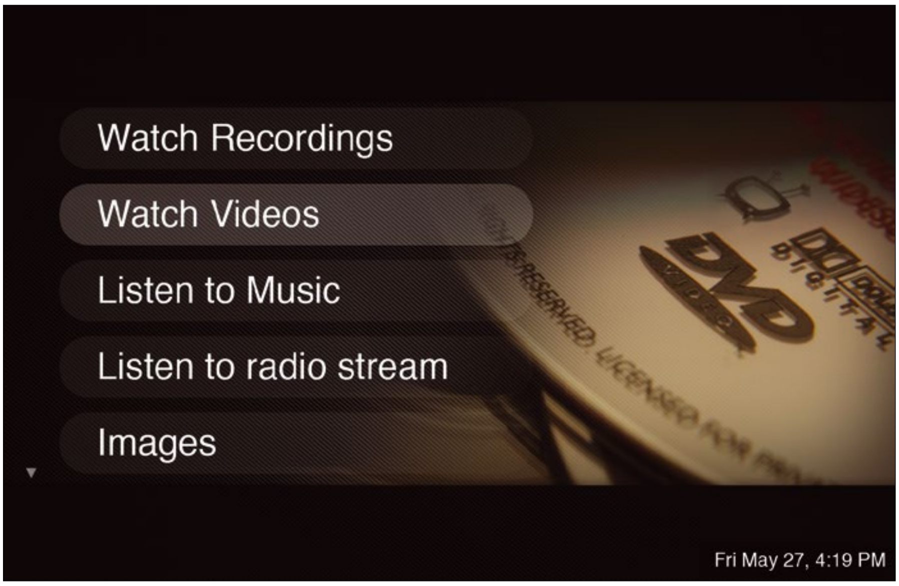
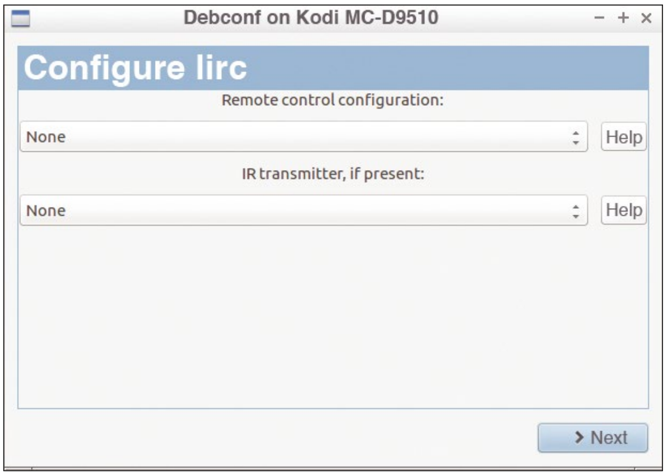


Figure 3: Multimedia content is just a few clicks away.



Debconf on Kodi MC-D9510

Configure lirc

Remote control configuration:

None | Help

IR transmitter, if present:

None | Help

Figure 4: Installing a remote control in Kodi is easy.

activate the PVR client from the *System* | *Add-ons* menu to put the TV card into operation (Figure 5).

This process requires a bit of manual work because not all front ends and back ends play well together. Additionally, I could not install all PVR clients during the test: Under Ubuntu 16.04 “Xenial Xerus,” the Tvheadend client could not be installed on the disk because of missing dependencies. However, once I noticed it malfunctioning after activating a PVR client, I terminated Kodi and then restarted it.

The PVR client tries to import a list of all tuners when starting the software, and it displays a message if there are any problems, in which case, you can choose a different client-server combination.

If the configuration is successful, the *TV* and, if the TV card supports it, *Radio* entries appear in the main menu. Both provide all possible forms of use with the *Channels*, *Guide*, *Recordings*, *Timers*, and *Search* submenus. Additionally, you can set up Internet protocol television (IPTV), call up station media libraries, and play back their content (Figure 6).

Modular

Kodi is strictly modular, and the system fully integrates additional functions with corresponding add-ons (e.g., a weather display that accesses data from a meteorological service on the Internet). With the use of add-ons, you can integrate multimedia codecs, online libraries, streaming services, and YouTube videos. A separate *Add-ons* submenu appears in the main menus for the *Pictures*, *Videos*, and *Music* entries, which you can use to install or disable extensions depending on the context.

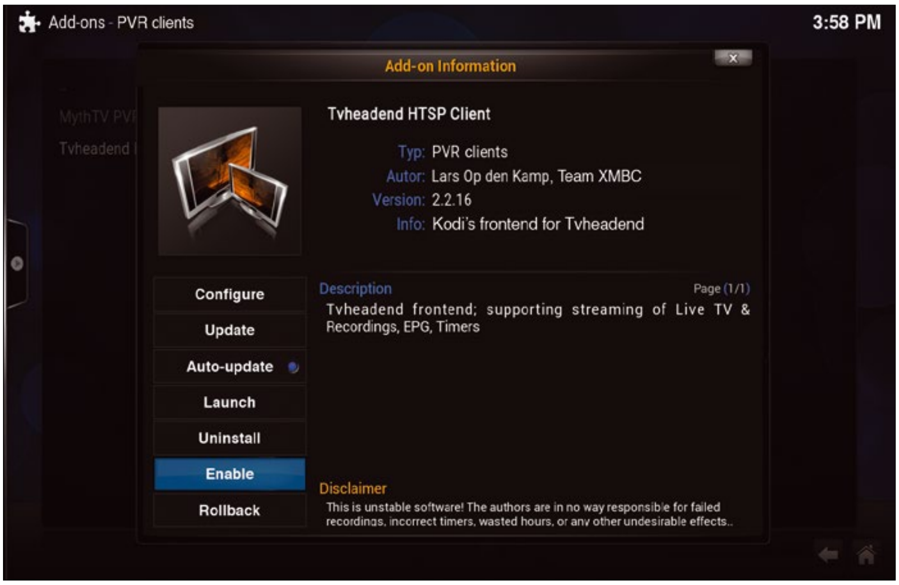
The main menu runs horizontally across the screen and adjusts dynamically according to the existing functionalities: If you start an audio CD, the associated playback controls will appear automatically in the main GUI. Kodi can also play Blu-ray discs on HD screens with no problems.

Correct Settings

The most important basic settings for the media center are hidden in the *System* menu. The *Settings* submenu bundles all the settings together, and the *File Manager* and *Profiles* entries let you take care of ongoing administrative tasks, such as creating/naming/moving folders and defining paths, and create user profiles with user-specific settings. A *System info* viewer and an event viewer round off the interface.

Plex Media Server

Although the basic version of Plex Media Server is free, it is under a proprietary license, and additional services are subject to charge. The software, which is similar to Kodi, is available for free download at



Tvheadend HTSP Client

Typ: PVR clients
Autor: Lars Op den Kamp, Team XMBC
Version: 2.2.16
Info: Kodi’s frontend for Tvheadend

Description Page (1/1)
Tvheadend frontend; supporting streaming of Live TV & Recordings, EPG, Timers

Disclaimer
This is unstable software! The authors are in no way responsible for failed recordings, incorrect timers, wasted hours, or any other undesirable effects..

Figure 5: Manually activating the PVR front end in Kodi.

Figure 6: A TV program shown in full-screen mode.

about 110MB for 32-bit and 64-bit Ubuntu, Fedora, and CentOS packages [14].

The server can also be used for other platforms and for network-attached storage (NAS) systems. Additionally, the manufacturer provides excellent documentation with a very detailed hardware compatibility list [15]. Like Kodi and MythTV, the Plex Media Server can also be supplemented with a client version that is also available for free download as a Plex home theater. Furthermore, you can access and control the server on an intranet from a web interface.

The precompiled packages are installed easily and create a *Sound & Video | Plex Media Manager* entry in the menu structure. The server starts automatically via a graphical interface for the web browser on *localhost*. The dashboard is very clearly structured and requires no learning curve.

You can then make further adjustments by clicking on the *Settings* icon in the upper right corner. However, some options can only be changed if you have a user account with Plex. Certain services can also only be enabled using a fee-based *Plex Pass*, including offline use of the Plex server with mobile devices such as smartphones or tablets, image uploads on mobile systems, cloud synchronization, and multiuser mode at a cost of $5 per month or an annual subscription of $40. A lifelong subscription costs around $150.

The Plex Media Server is strikingly different from the other media centers. For example, the application works well with cloud synchronization. The program also has an interface for transcoding content, which allows it to convert files into other formats. However, these functions depend on the hardware, especially for video editing, and require an implemented encoder in the hardware.

If you want to convert HD (720p or 1280x720) or even FHD (1080p or 1920x1080) material on NAS systems with Atom or Celeron CPUs, it is only possible to a limited extent or not at all because of the increased computational overhead. Older Atom processors aren’t even good enough to transcode SD (480p or 720x480) video material.

The Interface

The simple interface of the Plex Media Server lets you group you multimedia content in libraries. To do so, you need to categorize files in advance. Before equipping the system with multimedia data, you should carefully save different content in different directories: With video files, the Plex server differentiates between conventional *Movies*, *TV shows*, and *Home Videos*. However, it only has one category each for *Music* and *Photos* (Figure 7).

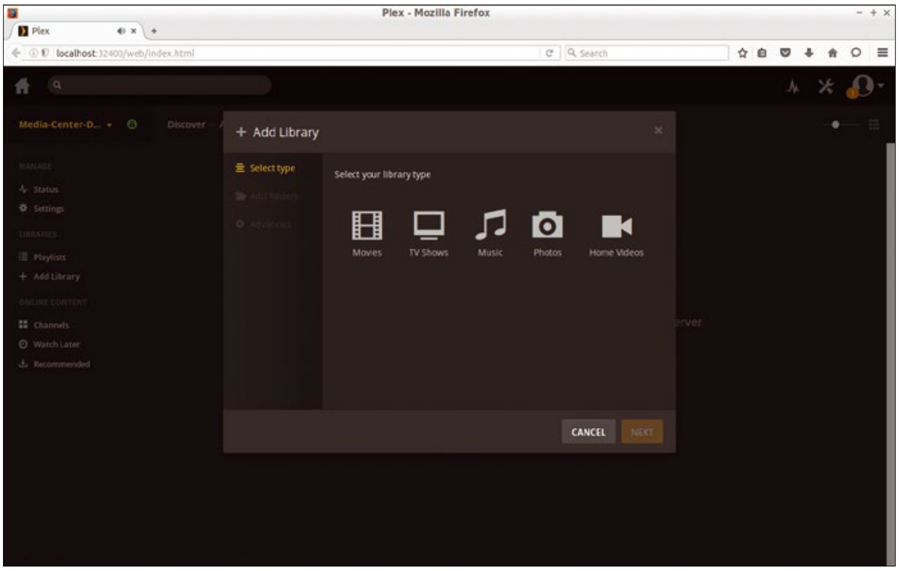


Figure 7: The Plex Media Center – simple and self-explanatory.

Multimedia content on the server needs quite a bit of time to generate the corresponding thumbnails for larger libraries, mainly because it queries various servers on the Internet that contribute metadata, such as summaries and images. The Plex server integrates this metadata into its libraries so you get a quick overview of its multimedia content (Figure 8).

In addition to the media server, the client is available in a native Linux version. The precompiled package of about 37MB is available for Fedora, CentOS, Ubuntu, Debian, and Arch Linux [16]. I used Ubuntu v16.04 for these tests, with the client installed in three steps:

```
sudo add-apt-repository ⮐
  ppa:plexapp/plexht
sudo apt-get update
sudo apt-get install plexhometheater
```

The first command integrates the repository, and then the package sources are updated, with the software installed in the last step. In the *Other* menu you will find a *Plex Home Theater* entry.

After launching the client, you first need to sort out the most important

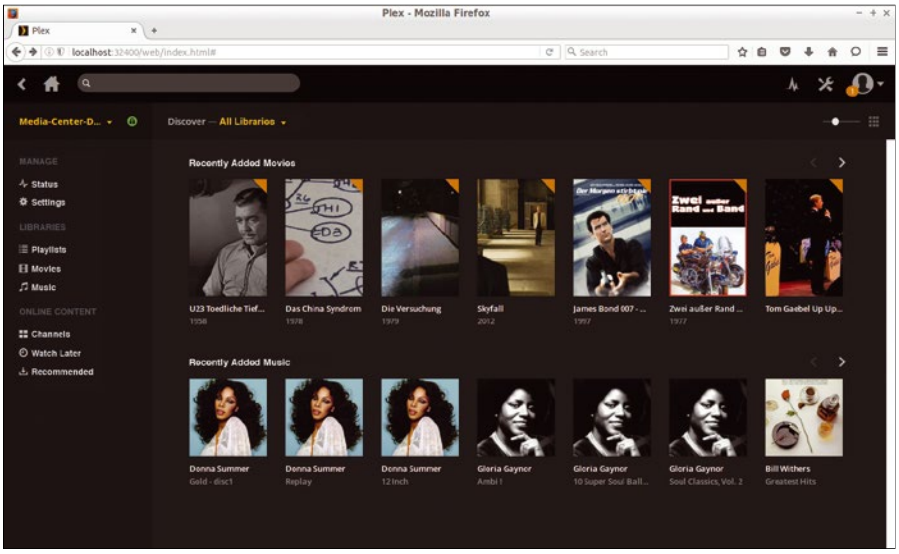


Figure 8: The thumbnail view provides a rough overview of the media center content.

settings. Because the client automatically connects to the existing server on the intranet, the content stored on the server is immediately available. The very intuitive interface also prepares the content in a visually appealing way.

DLNA and UPnP

Many TVs use the DLNA standard to access the content of the Plex Server; however, they are often not able to transcode the content and play it back because of the lack of appropriate hardware encoders.

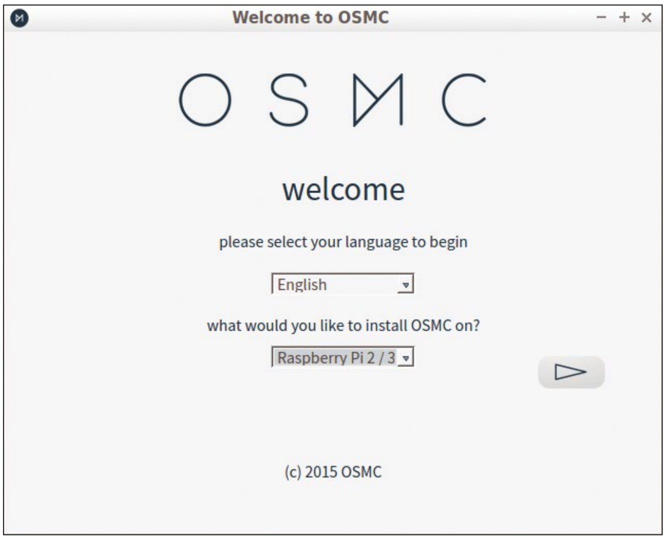
OSMC on the Raspberry Pi

The OSMC (Open Source Media Center) [4] emerged from the Raspbmc project; it has been around since 2014 and takes the Kodi Media Centre to the Raspberry Pi single-board computer. OSMC is not just a media center application, but a complete Debian-based operating system. To start the installation of OSMC on a third-party computer, you transfer the stripped-down Debian derivative to a microSD card that serves to boot the Raspberry Pi.

To install the operating system on the memory card, OSMC takes an unusual, but convenient, approach (from the user's point of view) with a graphical installer. To start the installer, you need to boot your Linux system and then load the installer on your local storage from a repository that first needs to be integrated. On Ubuntu and its derivatives, this is done with the following three commands:

```
sudo sh -c "echo 'deb ↩
  http://download.opensuse.org/↩
  repositories/home:/osmc/↩
  xUbuntu_16.04/ /' >> ↩
  /etc/apt/sources.list.d/↩
  osmc-installer.list"
sudo apt-get update
sudo apt-get install osmc-installer
```

The routine creates the *Sound & Video | OSMC Installer* menu entry; selecting it launches a simple graphical routine (Figure 9). The installer first asks you for your choice of language, storage locations, target computer generation, and version of the required software. After entering the data, it then downloads the OSMC image for the Raspberry Pi off the Internet and automatically installs it on the microSD card. Afterward, you can boot the system on the Raspberry Pi.


Welcome to OSMC

OSMC

welcome

please select your language to begin

English

what would you like to install OSMC on?

Raspberry Pi 2 / 3

(c) 2015 OSMC

Figure 9: Unusual but simple – the OSMC installer.

When first booted, the routine automatically prepares the media center by choosing a language and time zone from a list and then configuring the network. If you are using a Raspberry Pi 2, a problem can occur: You must plug a compatible WiFi USB stick into the computer for it to connect to your WiFi network, which is no problem for the Raspberry Pi 3, thanks to its built-in WiFi hardware.

When you set up the media center for the first time, the process can take some time on both models because of formatting and copying. The speed of progress is determined by the memory card and its limited sequential read and write capabilities. Slow class 4 micro-SD cards significantly slow down the small computer.

All the settings in OSMC are configured in a visually plain, but self-explanatory, interface. The application spectrum corresponds to that of a conventional Kodi interface, although OSMC does without all the resource-hogging gimmicks and animations (Figure 10). Because dialogs and activities strongly resemble those of Kodi, Kodi users will immediately feel at home navigating in OSMC.

You can handle general administrative tasks at the command line via an SSH client running on another computer. OSMC enables the SSH service in the basic installation: The username and password default to *osmc*.

Television

If you want to use OSMC as a TV, you first need to enable a PVR add-on – as in Kodi. The Tvheadend back end is a good choice; a preconfigured installation routine handles the OSMC integration, and you can easily launch the add-on from the media center's GUI and then configure the enabled Tvheadend server in a web browser from your remote computer. To integrate a TV card with your media center, just enter the OSMC system's IP address in the address bar, followed by

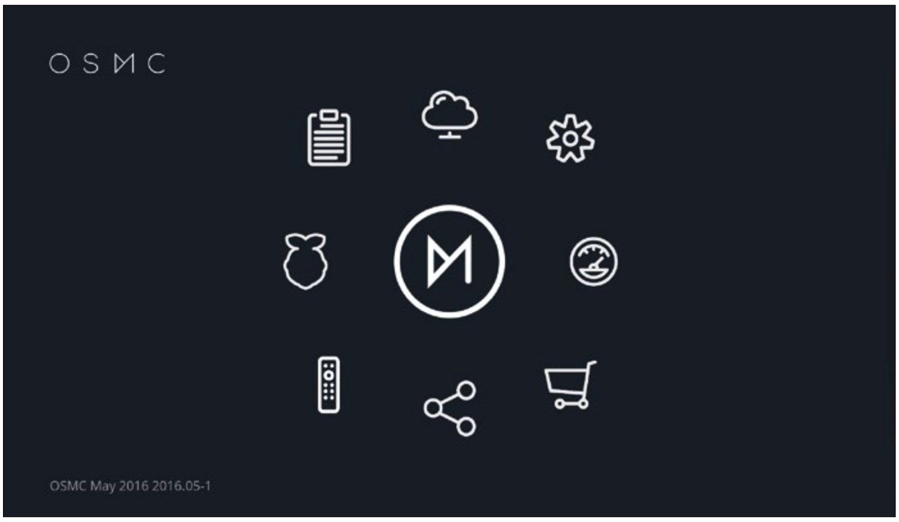
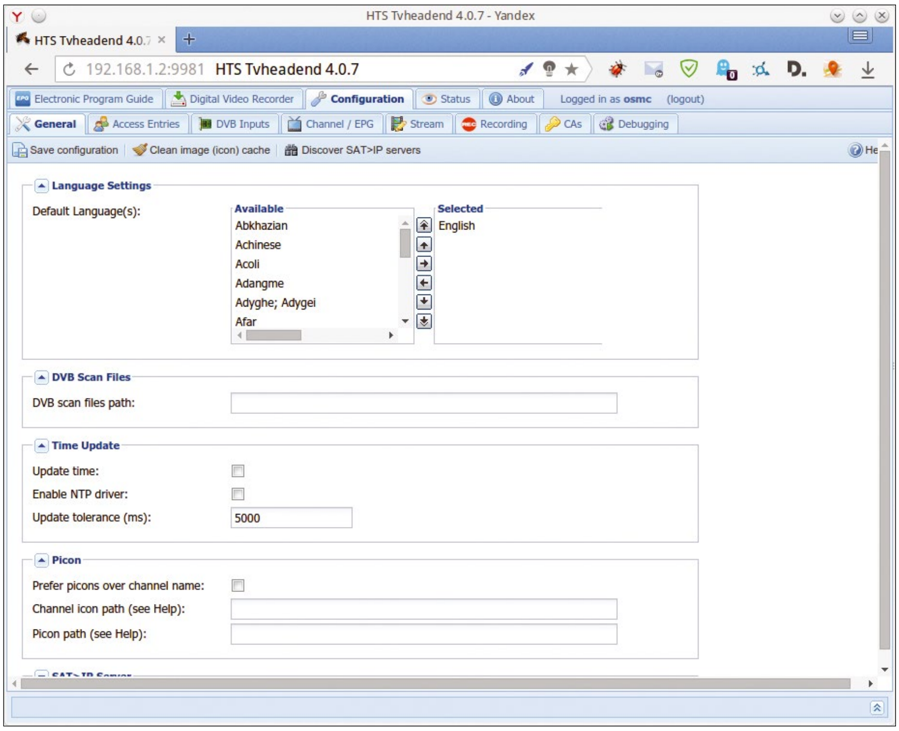


Figure 10: The OSMC interface is lean and simple.



Language Settings

Default Language(s):

Available: Abkhazian, Achinese, Acoli, Adangme, Adyghe; Adygei, Afar

Selected: English

DVB Scan Files

DVB scan files path:

Time Update

Update time: □

Enable NTP driver: □

Update tolerance (ms): 5000

Picon

Prefer picons over channel name: □

Channel icon path (see Help):

Picon path (see Help):

Figure 11: The TV card is easily configured in OSMC via the Tvheadend web interface.

port 9981 to proceed to a configuration GUI (Figure 11).

After enabling the PVR front end that appears in the add-on list, all the TV reception functions and the digital TV recorder should be available.

Web Interface

Although you will probably want to manage OSMC with a remote control supported by the Raspberry Pi, you also have the option of managing the system from a computer on the local network. To do so, you access the Media Center in your web browser and enter port number *8080* along with the IP address of the OSMC system. OSMC outputs the selected content to the TV, not to the computer screen. A web interface with the most important media groups, such as *Movies*, *TV Shows*, and *Music*, as well as a virtual remote control, then appear. Unfortunately, you still cannot control live TV or IPTV with this option.

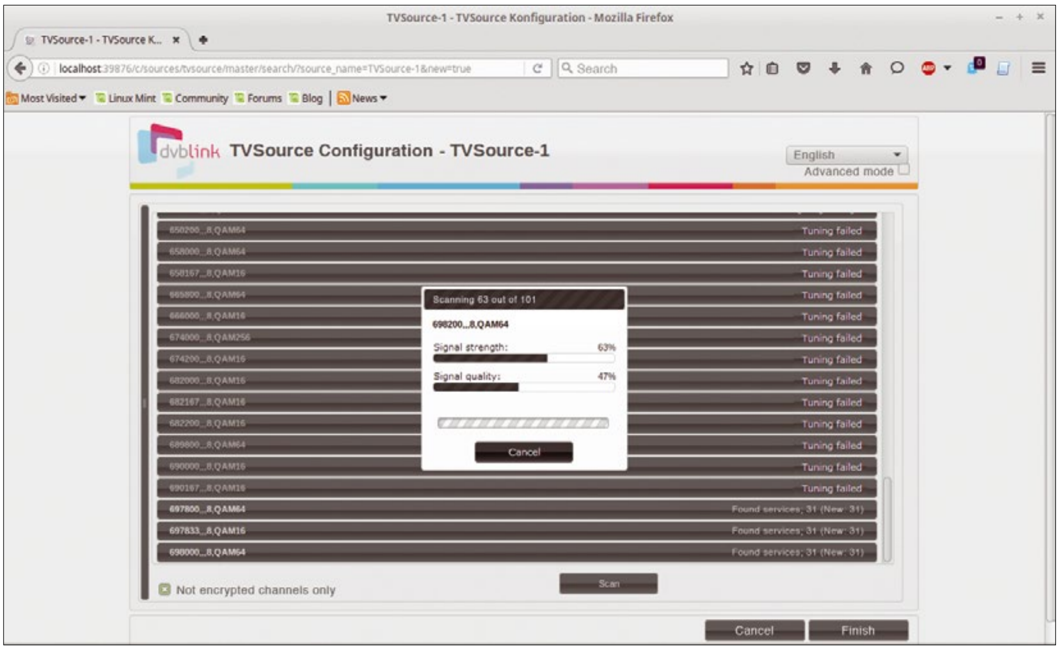
DVBLink

The Dutch company DVBLogic offers a commercial television solution for computers and NAS systems named DVBLink [5]; it supports all current TV standards and runs on standard platforms, including smartphones and tablet PCs. DVBLogic also provides TVButler, a DVBLink on a stick, which works with all major operating systems and NAS storage devices.

DVBLink server packages for 32- and 64-bit systems weigh in at around 32MB and are available for free download [17]; the provider cites only Ubuntu as a platform, but in our lab, the DVBLink server also ran on Linux Mint 17.3 without any trouble. Once you have installed the package, call the IP address of the target computer in your web browser, followed by port number *39876*.

You end up in a simple but clear-cut interface with several horizontal rows. The package does not include an EPG scanner, so you need to use the *DVBLink TVSource* package as a trial version after the install. It is part of the DVBLink software that you can purchase for around EUR50 after the test phase. Then use the *Sources* tab to integrate your TV hardware with the system. After doing so, launch a channel scan; a dialog will prompt you to choose your regional location from a list. Depending on the location, the scan can take some time (Figure 12).

After the scan, select the channels you want to view in the client. A routine then takes you to the web-based client, which you can use to view and record programs. This works on the local network without additionally installing a media center interface and can be used on remote computers and on mobile devices. You just need to call the IP address of the target computer with port number *8100*. In the web client, you can also define certain settings, such as the transfer rate of the streaming service or the window size (Figure 13).


TVSource-1 - TVSource Konfiguration - Mozilla Firefox
TVSource-1 - TVSource K...
localhost:39876/c/sources/tvsource/master/search/?source_name=TVSource-1&new=true
Search
Most Visited
Linux Mint
Community
Forums
Blog
News
dvblink
English
Advanced mode
Cancel
Scan
Cancel
Finish

TVSource Configuration - TVSource-1

```
650200,,,8,QAM64     Tuning failed
658000,,,8,QAM64     Tuning failed
658167,,,8,QAM16     Tuning failed
665800,,,8,QAM64     Tuning failed
666000,,,8,QAM16     Tuning failed
674000,,,8,QAM256    Tuning failed
674200,,,8,QAM16     Tuning failed
682000,,,8,QAM16     Tuning failed
682167,,,8,QAM16     Tuning failed
682200,,,8,QAM16     Tuning failed
689800,,,8,QAM64     Tuning failed
690000,,,8,QAM16     Tuning failed
690167,,,8,QAM16     Tuning failed
697800,,,8,QAM64     Found services: 31 (New: 31)
697833,,,8,QAM16     Found services: 31 (New: 31)
698000,,,8,QAM64     Found services: 31 (New: 31)
```

Scanning 63 out of 101

698200,,,8,QAM64

Signal strength: 63%

Signal quality: 47%

☒ Not encrypted channels only

Figure 12: The channel scan can take a long time depending on your location.

Willing to Cooperate

The DVBLink server also cooperates with the Kodi Media Center, which you can alternatively use as client software. To do so, you only need to install the DVBLink PVR client:

```
sudo apt-get install kodi-pvr-dvblink
```

In the Kodi Media Center, configure and enable the PVR client; after a reboot, Kodi automatically reads the channel and EPG data.

Conclusions

In practice, all the media center solutions were impressive in terms of functionality. They play back various formats, stream on the network, provide user-friendly interfaces, and are available on many platforms. When it comes to processing multimedia content, Linux does not need to hide its light under a bushel, in fact it even sets standards.

In many cases, the documentation and installation routines need more attention. In terms of installation, the commercial DVBLink server in particular offers a viable solution for less experienced users.

MythTV comes in last, however, because it requires expert knowledge to install the software, and, depending on your choice of Linux distro, it can run into problems caused by missing dependencies. The Debian derivative OSMC also was not entirely convincing in terms of installation. In our lab, I was forced first to install proprietary firmware files manually and retroactively to get TV reception to work in many cases. For example, it took a significant amount of manual rework to talk the TV receivers by Sundtek and Hauppauge into cooperating.

As a functionally comprehensive and visually appealing solution, Kodi still stands out in its field, and a combination of Kodi with the DVBLink server and matching PVR client proved the easiest solution to configure while offering robust functionality. ■■■

INFO

[1] MythTV: https://www.mythtv.org

[2] Kodi: https://kodi.tv

[3] Plex Media Server: https://plex.tv

[4] OSMC: https://osmc.tv

[5] DVBLink: http://dvblogic.com

[6] Linux TV project wiki: https://www.linuxtv.org/wiki/index.php/Main_Page

[7] MythTV download: https://www.mythtv.org/download

[8] Instructions for integrating MythTV: https://www.mythtv.org/wiki/External_Links#MythTV_Install_Guides_.2F_Distros

[9] MythBuntu: http://www.mythbuntu.org

[10] LinHES: http://www.linhes.org

[11] LIRC: http://www.lirc.org

[12] Kodibuntu: http://kodi.wiki/view/Kodibuntu

[13] Installing Kodi: http://kodi.wiki/view/HOW-TO:Install_Kodi_for_Linux#Installing_Kodi_on_Ubuntu-based_distributions

[14] Plex downloads: https://plex.tv/downloads

[15] Plex compatibility list: https://docs.google.com/spreadsheets/d/1MfYoJkiwSqCXg8cm5-Ac4oOLPRt-CkgUxU0jdj3tmMPc/edit?pref=2&pli=1#gid=314388488

[16] Plex instructions: https://forums.plex.tv/discussion/87253/plex-home-theater-linux-builds

[17] DVBLogic downloads: http://dvblogic.com/en/

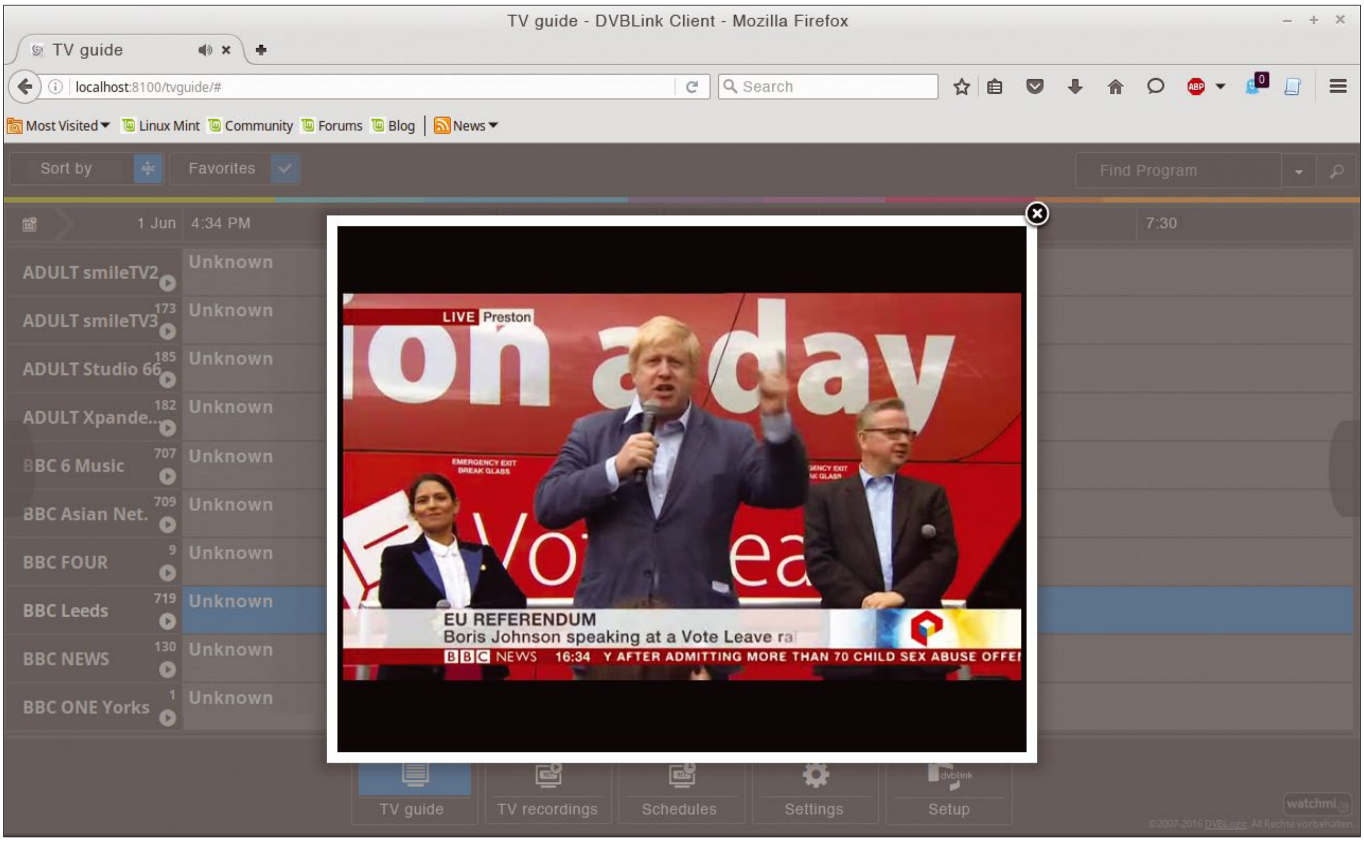


Figure 13: In the web-based viewer, the DVBLink server lets you watch programs, even without external client software.

Your Roadmap to the Open Hardware Revolution ...

An exciting world of projects, tips, and skill-building tutorials awaits you in every issue of Raspberry Pi Geek.

Order your subscription today and tune in to the revolution!

shop.linuxnewmedia.com

Set up Bluetooth on your Raspberry Pi 3
My Raspberry Pi controls everything!
raspberry pi GEEK
PI TOO SLOW? We explore the souped-up single-board systems BPI-M3, HiKey, and CubieTruck
Automate Your Home with FHEM
Halloween Build a scary ghost for your trick or treaters
PiXtend Add analog I/O and ... to your Raspberry
Cayenne An IoT platform without the co...
Free DVD! NOOBS 1.9.2
Hot Beans Rasp Pi coffee roaster controlled with a web app
WWW.RASPBERRY-PI-GEEK.COM
Add a Touchscreen to your Raspberry Pi
This magnificent experiment depends on my Raspberry Pi
raspberry pi GEEK
BlinkStick LEDs without the GPIO
Sunrise Sunset Pi Zero speeds up time on a model train set
littleBits Pet Feeder Feeding Fifi when you're away
Pi 3 is Here! with Bluetooth and built-in wireless networking
Track Your Trek Logging bike data with a Pi and sensors
Free DVD!
We compare the NEW Raspberry Pi 3 with earlier models
I can't wait to get started!
raspberry pi GEEK
Pi 3 Up Close Bluetooth, built-in wireless, and what you need to know
Print a Laptop Case with FreeCAD and a 3D printing service
Happy Birthday Remembering 4 years of the Raspberry Pi
Switch Tricks Start your Pi with a WiFi smart plug
upribox: Protect your online privacy
Free DVD! NOOBS 1.9.0
Scripting Class We solve the same practical problem using Awk, Perl, Python, and TLC
WWW.RASPBERRY-PI-GEEK.COM

Print Sub
Carry our easy-to-read print edition in your briefcase or backpack – or keep it around the lab as a permanent reference!

Digital Sub
Our PDF edition is a convenient option for mobile readers.

6 print issues with 6 DVDs or
6 digital issues for only

$59.95 £37.50 €44.90

Download on the App Store

GET IT ON Google play

iTunes Store

Google US

Google UK

Apple, the Apple logo, iPhone, iPad, and iPod touch are trademarks of Apple Inc., registered in the U.S. and other countries. App Store is a service mark of Apple Inc.

Putting free digital assistants to the test

Friend and Helper

Researchers from the University of Michigan have built an intelligent personal assistant akin to Siri and Cortana from free components. Although the Sirius Project focuses on the server load created by digital assistant software, we are interested in the usability of Sirius and its successor Lucida.

By Peter Kreußel

What does the chief engineer of a spaceship in the 23rd century do to operate a computer from the 20th century? He picks up the mouse and says, "Hello, computer" (*Star Trek IV: The Voyage Home*, Paramount Pictures, 1986). During his journey through time, Montgomery "Scotty" Scott nonetheless had to hit the keys eventually.

Owners of modern smartphones, on the other hand, can go a long way with `OK, Google`, `Hey, Siri`, or `Hey, Cortana`; the speech assistants understand many questions or instructions formulated in everyday language. You can only guess how many algorithms are behind the proprietary marvels.

Things are quite different with the open source intelligent personal assistant Sirius [1], which was developed in 2015 by the research group Clarity Lab at the University of Michigan [2]. The software, published under the BSD license, bundles together the free speech recognition systems CMU Sphinx [3] (PocketSphinx and Sphinx4), Kaldi [4], image recognition based on OpenCV [5], the question-answering system OpenEphyra [6], and UC Berkley's deep learning framework Caffe [7]. A Wikipedia dump forms the basis for OpenEphyra's data corpus. With aid from all of these components, Sirius is in a position to answer typed or spoken questions and to recognize objects in images (Figure 1).

The developers at Clarity Lab formulated the aim of the software in an abstract [8] for the Sirius tutorial that took place during the Twentieth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-20). They proceed from the assumption that the demand for intelligent personal assistants (IPAs) will increase in the future and ask what server architectures will have to look like to handle the workload of these programs. Because of a lack of open source IPAs to calculate the load, they developed Sirius so they could represent the resource requirements realistically.

How does Sirius fare in practical use? Is the program a suitable helper on the Linux desktop? Those running the test considered these questions, and carefully examined Sirius and its successor, Lucida [9]. They installed the software on Ubuntu 14.04 and Ubuntu 16.04, used the Sirius speech recognition, tested its question-answering system, and scrutinized its image recognition abilities. Lucida is not yet as far along. So far, only a simple question and answer game has operated in its demo version, which the testing team briefly exercised.

Ready-to-Assemble Kit

The Clarity Lab website offers a download that includes the Sirius application, Sirius Suite, and the web front-end server; the Sirius Suite alone with a

Lead Image © iofoto, 123RF.com

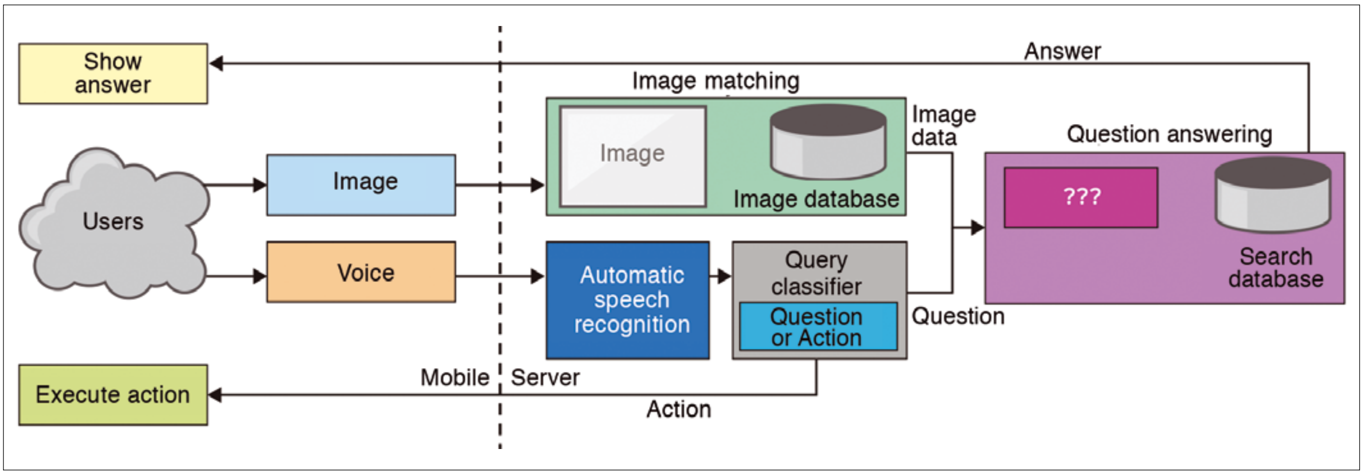


Figure 1: Components of the Sirius intelligent personal assistant (based on an image at the Clarity Lab website [1]).

Caffe snapshot; and the Wikipedia dump for the question-answering system [10].

After unpacking the Sirius archive, you switch to the `sirius-1.0.1/sirius-application` directory. A few scripts here import the software expected by Sirius, load components from the Internet, and compile and install them. The scripts are written for Ubuntu 14.04; if you use this somewhat older LTS version (that is nevertheless supported until 2019), you should enter the following four commands:

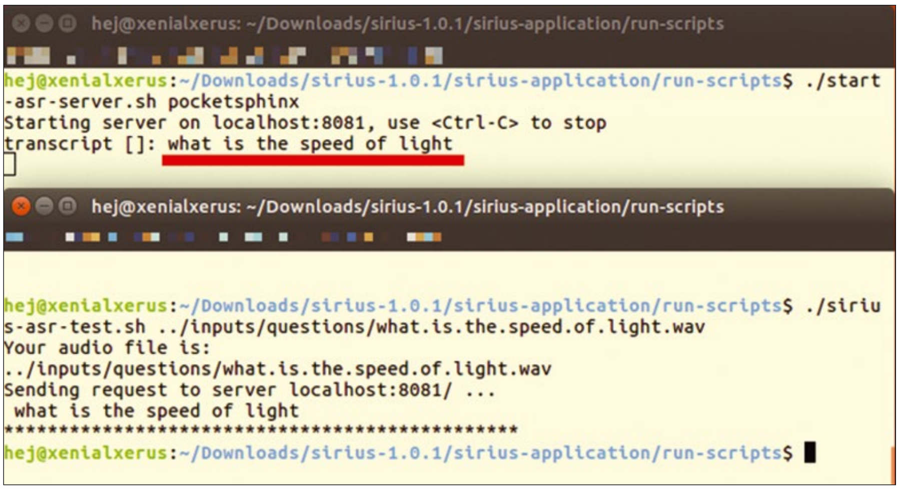
```
sudo ./get-dependencies.sh
sudo ./get-opencv.sh
./get-kaldi.sh
./compile-sirius-servers.sh
```

If you use the current Ubuntu 16.04, adjust the `get-dependencies.sh` script in the text editor beforehand and comment out the entry for adding the external FFmpeg repository (`ppa:kirillshkrogalev/ffmpeg-next`). The external package source is no longer necessary because FFmpeg is in the official Xenial repositories.

Next, execute the first three commands, but before you call up `./compile-sirius-servers.sh`, place a symbolic link from `/usr/bin/libtoolize` to `/usr/bin/libtool`, because the Kaldi makefile searches for this binary.

A fast Internet connection is an advantage, because the scripts download a whole host of software. With the OpenCV download, around 3GB of data are copied onto the disk; Kaldi takes up 2GB. The Sirius archive itself is 470MB in size, and the Wikipedia dump encompasses some 11GB. When completely installed, Sirius and its components occupy around 25GB of disk space.

The scripts that bring the speech recognition, image recognition, and question-answering system into the arena are in the `sirius-application/run-scripts` directory with `start` at the beginning of their file names. All three components are implemented as server services. The scripts you use to direct your requests to the servers are also found here with `test` in their file names.


hej@xenialxerus: ~/Downloads/sirius-1.0.1/sirius-application/run-scripts

```
hej@xenialxerus:~/Downloads/sirius-1.0.1/sirius-application/run-scripts$ ./start
-asr-server.sh pocketsphinx
Starting server on localhost:8081, use <Ctrl-C> to stop
transcript []: what is the speed of light
```

hej@xenialxerus: ~/Downloads/sirius-1.0.1/sirius-application/run-scripts

```
hej@xenialxerus:~/Downloads/sirius-1.0.1/sirius-application/run-scripts$ ./siriu
s-asr-test.sh ../inputs/questions/what.is.the.speed.of.light.wav
Your audio file is:
../inputs/questions/what.is.the.speed.of.light.wav
Sending request to server localhost:8081/ ...
 what is the speed of light
**************************************************
hej@xenialxerus:~/Downloads/sirius-1.0.1/sirius-application/run-scripts$
```

Figure 2: After WAV files are sent to Sirius ASR, you see what was understood by the back end.

Good Listener

In their first attempt, the test team fed a few of the WAV files stored in the `sirius-application/inputs/questions` directory to Sirius automatic speech recognition (ASR) and started the ASR server in a terminal in succession with one of the three available back ends (Kaldi, PocketSphinx, and Sphinx4):

```
./start-asr-server.sh kaldi
./start-asr-server.sh PocketSphinx
./start-asr-server.sh sphinx4
```

We then called up the `sirius-asr-test.sh` script in a second terminal together with a question (Provided) and saw the result from Sirius (Figure 2). Sometimes it worked well, sometimes only after waiting a while, and sometimes not at all; the communication with Sphinx4 using Ubuntu 16.04 completely misfired. For the comparison, the test team recorded the sentences themselves (Recorded) with a microphone and sent them to all three back ends. With the aid of five example sentences, Table 1 shows what Kaldi, PocketSphinx, and Sphinx4 understood.

The quality of text recognition is very patchy: With the WAV files provided, only the Sphinx4 back end worked almost flawlessly. On the other hand, with the testers' own recordings, the correctly recognized sentences remain a strange exception. The developers may have trained their speech recognition libraries primarily with the files they enclosed, which are spoken with an American accent throughout. With the test team's own recordings (in British English with a German accent), Sphinx4 particularly was unable to cope; the other engines at least recognized individual words.

Quality of the audio should not explain the lack of understanding, because a decent microphone was used. The testers recorded their sentences at random with a headset and a different frequency response, and the recordings still delivered inferior results. The Google and Apple speech recognition engines recognized almost all the questions on the test team's smartphones.

Answer Me

If the digital assistant understands a question, it would be great if it could answer it as well. The Sirius developers employ the question-answering system OpenEphyra [6] for this step.

A Wikipedia dump without semantic distinctions serves as the data corpus. The developers created this with Indri [11], a search engine specialized for large text corpora. You can download the Wikipedia knowledge database from the Sirius download page and extract it into the `sirius-application/question-answer` directory.

Now start the QA server with the `start-qa-server.sh` script from the `sirius-application/run-scripts` directory. On the Ubuntu 16.04 test machine, this did not work without further ado; a call to `ant` – which uses the XML build files for OpenEphyra and documentation files – in the `sirius-application/question-answer` directory was necessary before the server started working. If you receive an *insufficient threads configured* warning, you can fix it with a simple hack and comment out this line in the `sirius-application/question-answer/src/info/ephyra/OpenEphyraServer.java` file:

```
con1.setThreadPool(↩
  new QueuedThreadPool(NTHREADS));
```

After taking care of this problem, you must call up the `compile-sirius-servers.sh` script once more and restart the QA server.

Now you can ask questions in a second terminal; for example:

```
./sirius-qa-test.sh ↩
  "what is the speed of light"
```

After a confirmation that the question has come through, a message appears stating that the question has gone to the server. After a short wait, the answer pops up in the terminal (Figure 3).

Because spoken and typed questions are both possible, it would be great if you could combine these. That is no problem with Sirius; you simply start the ASR service along with the QA server and use the following script for communication:

```
./sirius-asr-qa-test.sh ../inputs/real/↩
  who.is.the.current.president.of.the.↩
  united.states.wav
```

Depending on the ASR back end, the analysis then continues. After this part has successfully transcribed the question, however, the QA service still requires some time to find the answer, so patience is needed.

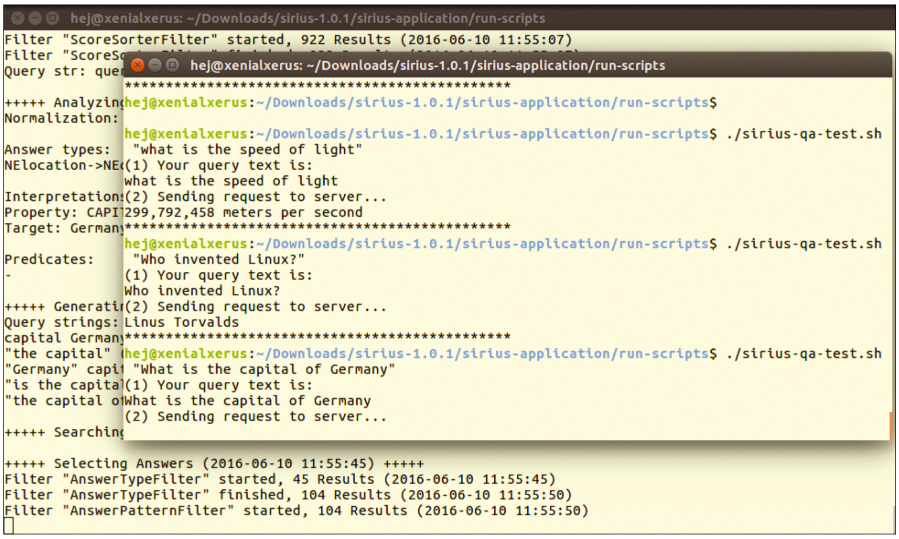
Vague Information

OpenEphyra goes beyond a pure keyword search. The system finds answers that are not written in the Wikipedia text. Broadly speaking, OpenEphyra analyzes and classifies both the questions and the data corpus available in textual form. Consequently, for the question "Who wrote *Hitchhiker's Guide*?," it is ready with the right answer: "Adams." If you are interested in the precise configuration, you will find a diagram of the underlying architecture online [12].

The test team fired off a total of 12 questions to Sirius from various fields of knowledge. For comparison, we questioned two of the proprietary competitors, Google Now and Apple's Siri.

TABLE 1: Sirius ASR Back Ends

Recording	Source	Kaldi	PocketSphinx	Sphinx4
Who invented the telegraph?				
	Provided	who invented the telegraph	who invented the telegraph	who invented the telegraph
	Recorded	we went at the telegraph	we're going to the telegraph	with only scowled
Where is the Louvre Museum located?				
	Provided	where is the liberal museum love the change yeah	where is the liver uneasy and located	where's the louvre museum located
	Recorded	where was the little free museums okay tent	where is the u. over a museum located	london back while passengers are
Where did John Lennon die?				
	Provided	where do you john lennon dot	where did john lennon got	where did john lennon died
	Recorded	when it it's john lennon die	where did john lennon die	only after all how often run
What is the population of France?				
	Provided	what is the population of france	what is the population of forms	what is the population of france
	Recorded	uh what is the population of france	what is the population of trunks	in a half and unload newark crown
What is the speed of light?				
	Provided	which is the speed of light	what is the speed of light	what is the speed of light
	Recorded	well just the speed of flights	what does the speed of light	the injury to half moon last



```
hej@xenialxerus: ~/Downloads/sirius-1.0.1/sirius-application/run-scripts
Filter "ScoreSorterFilter" started, 922 Results (2016-06-10 11:55:07)
Filter "ScoreSo
Query str: quer

+++++ Analyzing
Normalization:

Answer types:
NElocation->NE(

Interpretation:
Property: CAPI
Target: Germany

Predicates:
-

+++++ Generati
Query strings:
capital Germany
"the capital" (
"Germany" capi
"is the capital
"the capital o

+++++ Searching

+++++ Selecting Answers (2016-06-10 11:55:45) +++++
Filter "AnswerTypeFilter" started, 45 Results (2016-06-10 11:55:45)
Filter "AnswerTypeFilter" finished, 104 Results (2016-06-10 11:55:50)
Filter "AnswerPatternFilter" started, 104 Results (2016-06-10 11:55:50)
```

```
hej@xenialxerus: ~/Downloads/sirius-1.0.1/sirius-application/run-scripts
************************************************
hej@xenialxerus:~/Downloads/sirius-1.0.1/sirius-application/run-scripts$

hej@xenialxerus:~/Downloads/sirius-1.0.1/sirius-application/run-scripts$ ./sirius-qa-test.sh
 "what is the speed of light"
(1) Your query text is:
what is the speed of light
(2) Sending request to server...
299,792,458 meters per second
************************************************
hej@xenialxerus:~/Downloads/sirius-1.0.1/sirius-application/run-scripts$ ./sirius-qa-test.sh
 "Who invented Linux?"
(1) Your query text is:
Who invented Linux?
(2) Sending request to server...
Linus Torvalds
************************************************
hej@xenialxerus:~/Downloads/sirius-1.0.1/sirius-application/run-scripts$ ./sirius-qa-test.sh
 "What is the capital of Germany"
(1) Your query text is:
What is the capital of Germany
(2) Sending request to server...
```

Figure 3: Once the OpenEphyra server is running in one terminal window, you can enter your question in another and receive your answer there, as well.

The results, summarized in Table 2, unfortunately relegate the question-answering system in Sirius to the status of nice gimmick without major practical uses. The open source assistant lands a few serious hits that do command respect, given the background of a data corpus developed without semantic or ontological input. For a reliable helper in everyday use, however, the number of hits is certainly not sufficient.

During the comparison with Google Now on Android and Siri on the iPhone, you should bear in mind that many answers have clearly been prepared on them with paid editorial work. Who should know better than Google what questions users frequently ask? Even the often emphasized "personality" of Apple's assistant hardly comes from machine intelligence, but from careful human development. Since OpenEphyra is very wide of the mark on a few questions, and the software is clearly difficult to compile on modern Linux distributions, the developers are already considering alternatives for Lucida [13].

Look Closely

Image recognition needs a database that includes known images before it can accept your service. Users can shift to this in the `sirius-application/image-matching` file and call up the `make-db.py` script. This accepts the name of the database (`landmarks`) and the directory of images (`matching/landmarks/db`) as parameters:

```
./make-db.py landmarks matching/↩
  landmarks/db
```

The name of the database is hard-coded in the start script `start-imm-server.py`. If you choose an identifier different from `landmarks`, you need to adjust the script accordingly.

When the `./start-imm-server.sh` command from the `run-scripts` directory is called up, the image recognition server starts up. Afterward, `sirius-imm-test.sh` accepts an image file as a parameter. The name of the recognized object appears in the terminal output.

The test team added a few of their own images to the database for the comparison. For this, we used photos from the Freeimages service [14] and chose depictions of buildings that resembled Sirius' test images. With a few of the photos, the perspective changed somewhat, and it quickly became clear that the image matching only worked as long as the shooting location was almost the same. Stepping up against Sirius and the underlying OpenCV are Google Goggles [15] on Android and the CamFind app [16] on iPhone. Figure 4 shows the result.

Until now, only the individual Sirius components could be seen in action. This means that the software is still a long way from a personal assistant à la Google Now, Siri, or Cortana. If you are looking for a handy, complete product, you can naturally combine the services using shell script.

TABLE 2: Question-Answering Performance

Question	Sirius (OpenEphyra)	Google Now	Siri
What is the speed of sound?	1,500 meters per second	340.29 m/s	[brings up] Wikipedia article (speed of sound)
What is the speed of light?	299,792,458 meters per second	299 792 458 m/s	1.079.252.848,8 kph
What is the diameter of the earth?	149597870700 meters	12l 756.2 kilometers	The diameter of the Earth is about 7913 miles.
How old is the Taj Mahal?	1400 years	368 years	The answer is 368 years.
How old is the Earth?	100,000 years	4,543 x 10^9 years old	–
Who discovered America?	Rogers	–	Input interpretation, first humans reach North America via Beringia.
Who composed the Eroica?	Beethoven	Ludwig van Beethoven	–
Who composed "Penny Lane"?	Lennon	Paul McCartney/John Lennon	–
What is the capital of Germany?	Soviet Union	Berlin	Berlin is the capital of Germany.
How long is the Danube?	200,000 years	2860 km	The answer is about 2850 kilometres.
Who wrote *Hitchhiker's Guide*?	Adams	Douglas Adams, Eoin Colfer	It looks like the author of The Hitchhiker's Guide to the Galaxy was Douglas Adams.
Who invented Linux?	Linus Torvalds	–	–

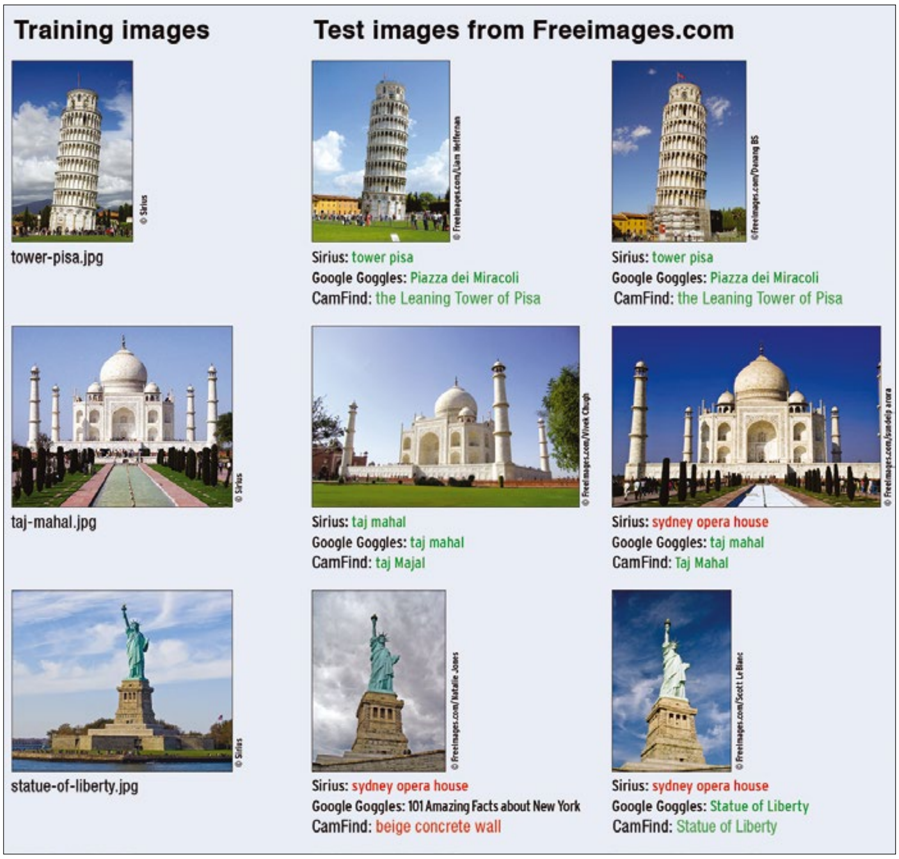

Training images
Test images from Freeimages.com
© Sirius
tower-pisa.jpg
© Freeimages.com/Liam Heffernan
Sirius: tower pisa
Google Goggles: Piazza dei Miracoli
CamFind: the Leaning Tower of Pisa
©freeimages.com/Danang BS
Sirius: tower pisa
Google Goggles: Piazza dei Miracoli
CamFind: the Leaning Tower of Pisa
© Sirius
taj-mahal.jpg
© Freeimages.com/Vivek Chugh
Sirius: taj mahal
Google Goggles: taj mahal
CamFind: taj Majal
© Freeimages.com/Sundeip arora
Sirius: sydney opera house
Google Goggles: taj mahal
CamFind: Taj Mahal
© Sirius
statue-of-liberty.jpg
© Freeimages.com/ Natalie Jones
Sirius: sydney opera house
Google Goggles: 101 Amazing Facts about New York
CamFind: beige concrete wall
© Freeimages.com/Scott Le Blanc
Sirius: sydney opera house
Google Goggles: Statue of Liberty
CamFind: Statue of Liberty

Figure 4: If you submit images from different angles for Sirius image recognition, you find little joy. Only Google Goggles was almost flawless.

Form a Team

Another possibility is to use the web server enclosed in the `sirius-web` folder. The README file unfortunately still contains dated information about the dependencies; the necessary software was copied onto the disk with `pip install` in the `get-dependencies.sh` script used earlier.

The web server started up successfully on Ubuntu 14.04, but it looked meager on the current system, and although the display worked in the browser (Figure 5), interaction with the controls was not possible.

Out of the Box

Sirius is not being further developed, which is reason enough to cast a glance at its successor, Lucida [9]. Since compiling from the sources [17] did not succeed, the test team opted for Docker containers that had a trimmed-down demo version. After installing `docker` and `docker-compose`, you should run the following two commands:

```
sudo docker pull claritylab/lucida:latest
sudo docker pull claritylab/lucida-asr
```

In the process, around 17GB copies to the disk. Next, access the main container with this command:

```
sudo docker run -i ☡
  -t claritylab/lucida /bin/bash
```

The `docker-compose.yml` file, which is necessary to start the demo version on the host system, is in the root directory. You are advised to copy its content onto the clipboard and paste it into a new `docker-compose.yml` file after leaving the Lucida container with `exit`; then, start all the Lucida services from the same directory:

```
sudo docker-compose up
```

The developers recommend using the Lucida web interface (*http://localhost:8081*) with the Chrome browser or free Chromium variant. A Wikipedia data dump is not present. The web front end instead asks for text to form the evidence base for the question-answering system. You enter this yourself and then click on `Submit`.

Lucida then asks for the access privileges for the microphone, because the demo version exclusively communicates by speech. You click on the microphone symbol to activate this, and speak your question. Kaldi operates in the background and attempts to understand the text. The result is shown in the speech bubble on the left, with the answer opposite on the right (Figure 6).

As previously seen in the Sirius experiment with Kaldi (Table 1), dictating a sentence and having it correctly recognized was only possible with great difficulty. In this Docker edition, it is not possible to send questions to the system from your keyboard, and image recognition is likewise lacking.

A process to exchange the Kaldi speech recognition back end is not provided, meaning that the test team was also unable to experiment with PocketSphinx or Sphinx4. In the GitHub repository, the developers state that they will publish the next generation of Lucida toward the end of summer 2016 [18]. Along with a new command center, they should especially enclose a better question-answering system and a guide for how users can exchange individual components.

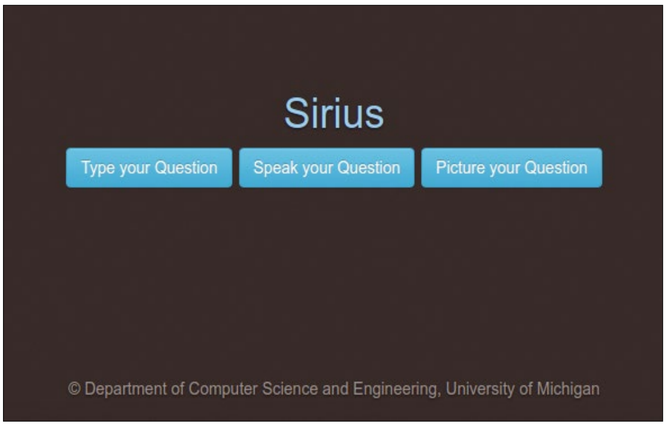
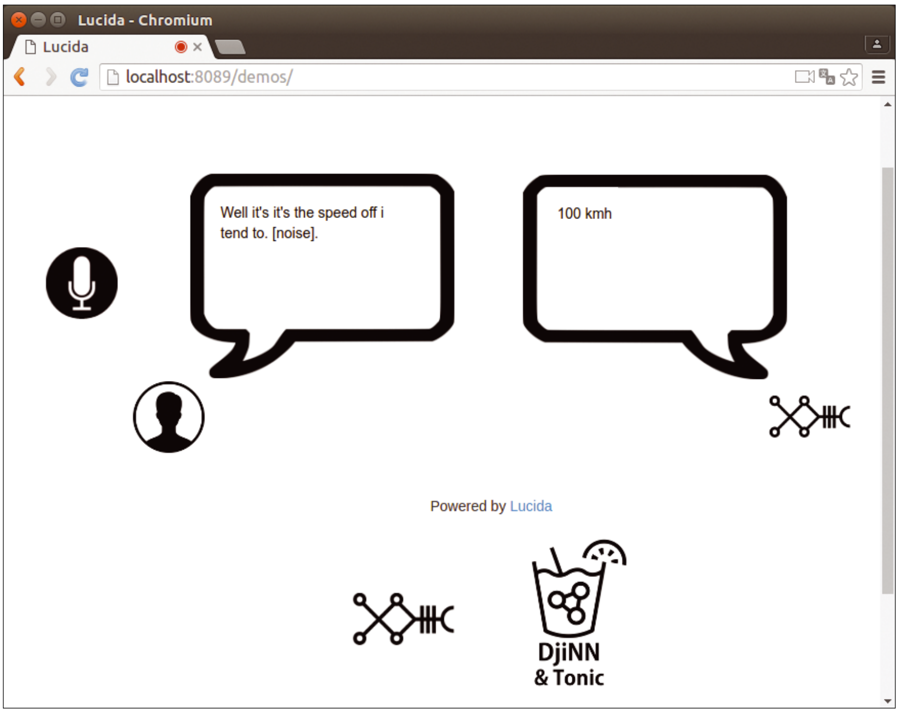

Sirius
Type your Question
Speak your Question
Picture your Question
© Department of Computer Science and Engineering, University of Michigan

Figure 5: The web front end included with Sirius in the `sirius-web` directory was unfortunately not functional in the test.

Future Helpers

Sirius and Lucida are not suitable for serious use to support you in everyday life, so Linux users will probably have to wait a while before they can get reasonable answers or real help from a digital assistant. The performance of the free programs – still a long way behind that of commercial alternatives – presumably is not because of an inferior quality of software, however. Firms such as Google or Apple have undoubtedly invested a large amount of money into training efforts. In the case of speech recognition, for instance, training consists of tedious tasks providing hours of recordings and phonetic transcription.

In principle, as a user, you can also train the free components and accustom them to your voice. However, linguistic knowledge and, most of all, staying power are needed to make that happen. ■■■



Well it's it's the speed off i tend to. [noise].

100 kmh

Powered by Lucida

Figure 6: The new Lucida web front end functions in Chrome and Chromium. It accepts questions by microphone, although the quality of the answers remains limited by the Kaldi speech recognition.

INFO

[1] Sirius: *http://sirius.clarity-lab.org/sirius*

[2] Hauswald, Johann, Michael A. Laurenzano, Yunqi Zhang, et al. "An open end-to-end voice and vision personal assistant and its implications for future warehouse scale computers." *In: Proceedings of the Twentieth International Conference on Architectural Support for Programming Languages and Operating Systems* (ASPLOS, 2015), New York: ACM, pp. 223-238.

[3] CMU Sphinx: *http://cmusphinx.sourceforge.net*

[4] Kaldi: *http://kaldi-asr.org*

[5] OpenCV: *http://opencv.org*

[6] OpenEphyra: *http://www.ephyra.info*

[7] Caffe: *http://caffe.berkeleyvision.org*

[8] Sirius seminar during ASPLOS-20: *http://sirius.clarity-lab.org/tutorial*

[9] Lucida: *http://lucida.ai*

[10] Sirius downloads: *http://sirius.clarity-lab.org/downloads/#sirius*

[11] Lemur's Indri: *http://www.lemurproject.org/indri.php*

[12] OpenEphyra architecture: *https://mu.lti.cs.cmu.edu/trac/Ephyra/wiki/Docs/ArchitectureOverview*

[13] Status of OpenEphyra: *https://github.com/claritylab/lucida/issues/89*

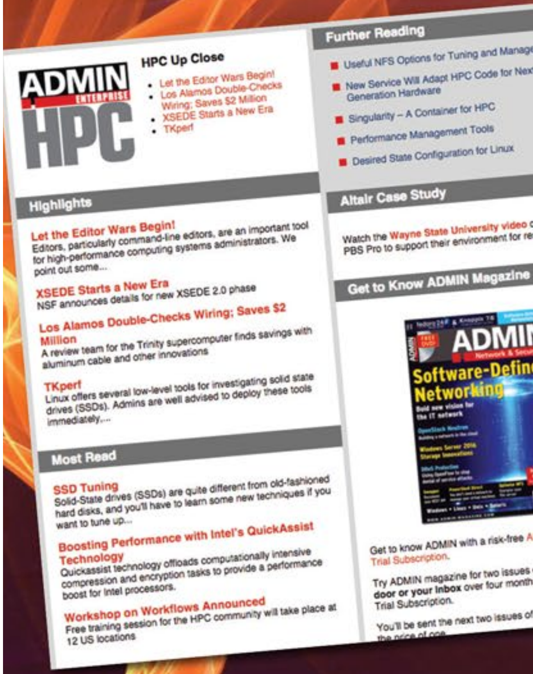
[14] Freeimages: *http://www.freeimages.com*

[15] Google Goggles: *http://www.google.com/mobile/goggles*

[16] CamFind: *http://camfindapp.com*

[17] Lucida GitHub repo: *https://github.com/claritylab/lucida*

[18] Next Lucida version: *https://github.com/claritylab/lucida/issues/116*



Klaus Knopper answers your Linux questions

Ask Klaus!

By Klaus Knopper

Downgrading Packages

Hi Klaus, how can I select a specific version, possibly an older one, of a package to install in Debian if I want to avoid the newest version of, for example, Chromium from the "unstable" branch?

To see the current available versions of a package, you have to update the package catalog first. I sometimes even delete the old package catalog lists (first line) to make sure all of them get downloaded again,

```
sudo rm -f /var/lib/apt/lists/*
sudo apt-get update
```

although this step is optional. The second line does not actually update anything, it just fetches the software catalogs from all Debian branches listed in `/etc/apt/sources.list` and `/etc/apt/sources.list.d/*`.

To check for available versions in each Debian branch, the command

```
apt-cache policy <package-name>
```

can be helpful. Because it only queries the software catalog, no root permissions are required (Listing 1). On some systems, the command `apt-show-versions` is installed and shows more verbose information. However, as opposed to `apt-cache policy`, it is not included in standard Debian.

The Chromuim version currently installed is 52.0.2743.116-2 from the Debian/testing branch. The `apt-get install` command would install version 53.0.2785.92-2 from Debian/unstable, so if I would rather use version 53.0.2785.89-1 ~ deb8u1 from the Debian/stable security updates, for example, the command for download and installation would be:

```
sudo apt-get install ↩
  chromium= 53.0.2785.89-1~deb8u1
```

Another option for specifying the branch rather than the specific version number is adding the branch name right after the package name separated by a slash, which is syntactically not supported for branches with a subbranch like stable/updates. I'll chose the unstable branch here:

```
sudo apt-get chromium/unstable
```

If `apt-get` now complains about dependencies (e.g., packages that Chromium depends on) not being installable, you can also tell `apt-get` which Debian branch to go for those with the `-t` switch:

```
sudo apt-get install -t testing ↩
  chromium=53.0.2785.89-1~deb8u1
```

You can specify dependency packages to install in addition to `chromium` by adding them to the same command line, each with versioning information. In this way, you can possibly resolve more complicated dependencies with mixed-branch versions of installed software:

```
sudo apt-get install -t testing ↩
  chromium=53.0.2785.89-1~deb8u1 ↩
  chromium-l10n=53.0.2785.89-1~deb8u1 ↩
  libnspr4/unstable
```

This is what I frequently do in Knoppix to get the newest packages for Xorg and the desktop, yet get the more tested versions for services like `apache2` and `samba` from the stable branch.

To keep the installed version permanently without automatic upgrades, you can put it on "hold" by issuing:

```
echo chromium hold | ↩
  sudo dpkg --set-selections
```

The `hold` status can be removed again by simply upgrading or downgrading the package anytime later. ■■■

LISTING 1: Chromium Versions

```
apt-cache policy chromium

chromium:
  Installed:   52.0.2743.116-2
  Candidate:   53.0.2785.92-2
  Version table:
     53.0.2785.92-2 500
        500 http://ftp.debian.org/debian unstable/main i386 Packages
     53.0.2785.89-1~deb8u1 990
        990 http://security.debian.org stable/updates/main i386 Packages
 *** 52.0.2743.116-2 500
        500 http://ftp.debian.org/debian testing/main i386
        100 /var/lib/dpkg/status
Packages
     50.0.2661.94-1~deb8u1 990
        990 http://ftp.debian.org/debian stable/main i386 Packages
```

KLAUS KNOPPER

Klaus Knopper is an engineer, creator of Knoppix, and co-founder of LinuxTag expo. He works as a regular professor at the University of Applied Sciences, Kaiserslautern, Germany. If you have a configuration problem, or if you just want to learn more about how Linux works, send your questions to: *klaus@linux-magazine.com*

A bag o' tricks from the Perlmeister

Undercover Information

If you have been programming for decades, you've likely gathered a personal bag of tricks and best practices over the years – much like this treasure trove from the Perlmeister. By Mike Schilli

For each new Perl project – and I launch several every week – it is necessary to first prime the working environment. After all, you don't want a pile of spaghetti scripts lying around that nobody can maintain later. A number of template generators are available on CPAN. My attention was recently drawn to App::Skeletor, which uses template modules to adapt to local conditions. Without further ado, I wrote Skeletor::Template::Quick to adapt the original to my needs and uploaded the results to CPAN.

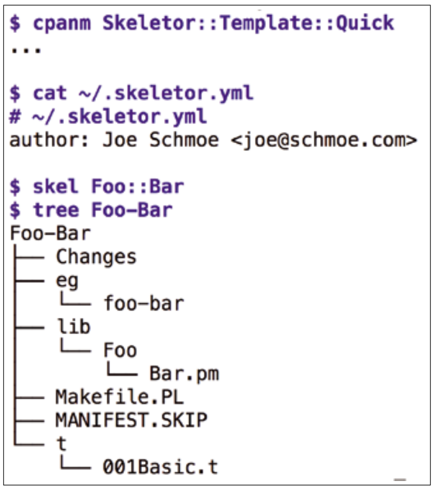
If you store the author info, as shown in Figure 1, in the `~/.skeletor.yml` file in your home directory and, after installing the Template module from CPAN, run the `skel Foo::Bar` command, you can look forward to instantly having a handful of predefined files for a new CPAN distribution dumped into a new directory named `Foo-Bar`. Other recommended tools for this scaffolding work would be the built-in Perl tool `h2xs` or the CPAN Module::Starter module.

To help recruit future users of a new module on CPAN as code contributors, the Perl-typical `Makefile.PL` generated by the tool contains a link to the Github repository with the source code. On the CPAN package repository site, *search.cpan.org*, the reference is later seen next to the link for downloading the module, and module authors are looking forward to receiving GitHub pull requests for improving the code (Figure 2).

Solid Foundation

In the newly created code directory, the template generator has created everything you need to get started: from the module files (`lib/Foo/Bar.pm`) to sample scripts, such as `eg/foo-bar` and a test suite (`t/001Basic.t`). Everything is ready for use right from the start. The code files not only contain handy code snippets but also define templates for the documentation. This is important, because documenting how to use scripts and libraries should never be considered a chore.

When I design new code, I first write down how potential users will use my wonders of technology – as far as I can

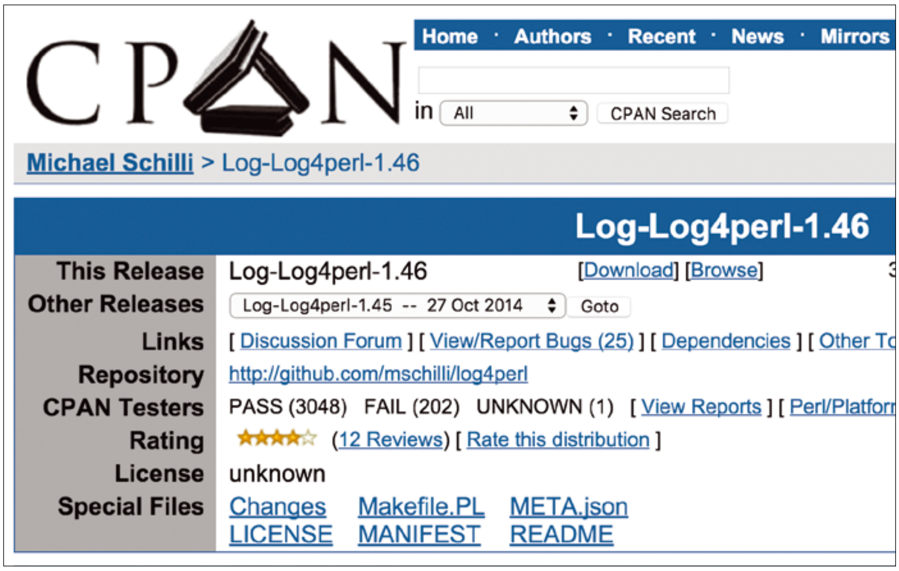


```
$ cpanm Skeletor::Template::Quick
...

$ cat ~/.skeletor.yml
# ~/.skeletor.yml
author: Joe Schmoe <joe@schmoe.com>

$ skel Foo::Bar
$ tree Foo-Bar
Foo-Bar
├── Changes
├── eg
│   └── foo-bar
├── lib
│   └── Foo
│       └── Bar.pm
├── Makefile.PL
├── MANIFEST.SKIP
└── t
    └── 001Basic.t
```

Figure 1: Skeletor produces the raw skeleton for a new CPAN module.

MIKE SCHILLI

Mike Schilli works as a software engineer in the San Francisco Bay Area. He can be contacted at *mschilli@perlmeister.com*. Mike's homepage can be found at *http://perlmeister.com*.



Home · Authors · Recent · News · Mirrors

in All CPAN Search

Michael Schilli > Log-Log4perl-1.46

Log-Log4perl-1.46

This Release Log-Log4perl-1.46 [Download] [Browse] 3
Other Releases Log-Log4perl-1.45 -- 27 Oct 2014 Goto
Links [Discussion Forum] [View/Report Bugs (25)] [Dependencies] [Other To
Repository http://github.com/mschilli/log4perl
CPAN Testers PASS (3048) FAIL (202) UNKNOWN (1) [View Reports] [Perl/Platforr
Rating ★★★★☆ (12 Reviews) [Rate this distribution]
License unknown
Special Files Changes Makefile.PL META.json
LICENSE MANIFEST README

Figure 2: The package page on the CPAN Repository site features a link to the GitHub project hosting the source code under `repository`.

tell in advance. These are mostly object-oriented modules, and before typing the first line of code, I tend to the `SYNOPSIS` section in the `POD` area of the source code to describe how the new class is instantiated and what any subsequently invoked methods will do:

```
my $m = MyModule->new;
$m->dosomething( 42 );
```

This light exercise often helps me find out whether the imaginary interface really is a smart idea. If it is awkward to handle or feels wrong, the problem is quickly corrected, because still there is no code to rework.

Halt, TDD Police!

When test-driven development (TDD) [1] raised its head about 10 years ago, many folks enthusiastically jumped on the bandwagon. According to its principles, developers first write a test case and then add the new feature. New code was created in pair-programming, and the test case preceding each new feature first failed before being implemented (red bar) or passed when completed (green bar).

This procedure ran out of road a while ago, and many programmers went back to the old routine. I kept two things: I make changes in the code according to the theory of minimum viable product. First, I add the desired function and run tests, and then I use refactoring methods to keep the project clean.

Whenever I find a bug and correct the code, I also try to add a test case, which raises an alarm without the bug fix and runs smoothly after patching. This is priceless to avoid the kind of regressions that inevitably occur when the code becomes more complex or when the project is approaching its tenth release.

In this way, some of my modules sport surprisingly extensive test suites that have built up over the years, simply by following that routine, and at a relatively low cost. Now they're in a state where not even the most diligent of programmers could stomp this out of the ground in reasonable time.

IDE for Old Hands

Opinions differ on development environments. Some prefer a comfortable, mouse-controlled monster tool like Eclipse, which identifies the program syntax and links together all variables and functions so that developers can jump back and forth between definitions and use cases, as well as between files, simply by clicking on code snippets.

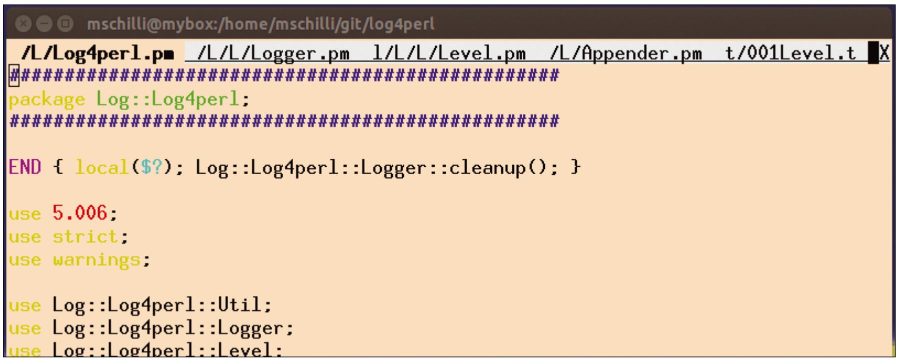
When it comes to practical IDEs, I'm not exactly spoiled as an old hand, but I do require lightning fast performance. I use the `vim` editor with a trick for switching back and forth between a project's files: If you call `vim -p File1 File2 […]`, `vim` displays all the files passed in on the command line as tabs, and you can toggle between them with `gt` (go to tab, right) and `gT` (left) (Figure 3). To make things even faster, I mapped `gt` with Vim's `map` command to Shift + L, and `gT` to Shift + H (see `.vimrc` [2]).

I can easily remember that a lowercase h moves the cursor to the left in `vim`, while a lowercase l moves it to the right. So, I move back and forth between the tabs simply by typing the corresponding uppercase letters. If you have many files open at the same time, you can't quit them all at the same time with `ZZ` or `:wq`; instead you need to type `:qall`, or – like I did – map the latter to Shift + Q.

Infrastructure as Code

To make sure that code does not only work in the development environment, but also for the users in their worlds, the release process must ensure two things. The generated artifact is exclusively allowed to pick up the source code from the Git repository and is not allowed to rely on local files; this means that the build can always be reproduced. And, before the product is launched into the wild, it has to pass through the accompanying test suite, which simulates the end user's view.

Professional developers use build servers for this. They automatically wake up with tools like Jenkins or similar, if new sources appear in the Git repository. They grab the new code, start the build, run the tests, and put together an artifact



mschilli@mybox:/home/mschilli/git/log4perl

```
 /L/Log4perl.pm  /L/L/Logger.pm  l/L/L/Level.pm  /L/Appender.pm  t/001Level.t  X
##################################################
package Log::Log4perl;
##################################################

END { local($?); Log::Log4perl::Logger::cleanup(); }

use 5.006;
use strict;
use warnings;

use Log::Log4perl::Util;
use Log::Log4perl::Logger;
use Log::Log4perl::Level;
```

Figure 3: Four source files and a test suite open in the Vim editor with tabs.

such as a tarball or an RPM package if successful. They then upload the latter to the distribution server in one fell swoop. Open source projects often use Travis-CI [3] for this. It's an excellent build hosting provider that sets up a build server for a GitHub project at the push of a button and is happy with simple three-liner configurations that live alongside in the source code.

A virtual environment, such as a Vagrant VM [4] provisioned with Ansible, or a Docker container [5], which generates artifacts and runs tests, is equally fine for home use. If all goes well, the release is made and the `cpan-upload` script from the CPAN::Upload CPAN module then uploads a tarball created by `make tardist` to CPAN.

Listing 1 [6] shows a Docker configuration that produces a clean room based on the latest Ubuntu distro. A call to Docker's `build` command picks up the local Dockerfile, pulls in the lean Ubuntu base image from the Docker mothership, and adds more layers to it according to each `RUN` instruction in the file:

```
docker build -t testimg .
```

The statements in the Dockerfile tell Docker to run `apt-get update` to point Ubuntu's package manager to the latest repository versions, and to install packages for build support, such as `make`. Later calls to the same `build` command will reuse the previously created content from the cache, as long as the lines in the Dockerfile haven't changed according to a checksum comparison.

LISTING 1: Dockerfile

```
FROM ubuntu

RUN apt-get -y update
RUN apt-get -y install cpanminus
RUN apt-get -y install make
RUN apt-get -y install libwww-perl
```

The build script in Listing 2 first runs the Docker `build` command, which creates a new image, and then invokes the `run` command that launches a container based on the image. The `-v` option makes the host's source directory for the module available inside the container below `/mybuild` for read and write access.

Because the `build` script is checked as `adm/build` into the module's Git repository, Perl's `FindBin` module first identifies its absolute location. The module code resides in the parent directory, and `Path::Tiny` changes the location accordingly and then recreates a minimal absolute path using `realpath`.

Line 13 calls `bash` as a command in the container; it uses the `-c` option to pass a string with the typical Perl-style triple jump of `perl Makefile.PL; make test; make tardist` to it. This in turn puts together the distribution tarball from the module code under clean-room conditions. Subsequent build steps should copy the tarball to new clean rooms and test whether it can be installed and used, as this is not automatically the case, especially if it needs more modules from CPAN at run time.

Automatically Error-Free

The important thing is that each level of the build process runs automatically and immediately pulls the emergency ripcord if unexpected events occur. The automatic part is essential because human operators tire easily and start making mistakes when they continuously repeat the same steps. Broken releases are often the consequence, causing embarrassment and user frustration. If you do invest time in automating the build process, you will learn to enjoy the ability to push a button after making a change to the code, before heading off for lunch, in the assurance that everything will follow a tried and trusted path.

LISTING 2: Build Script

```
01 #!/usr/local/bin/perl -w
02 use strict;
03 use Sysadm::Install qw(:all);
04 use FindBin qw( $Bin );
05 use Path::Tiny;
06
07 my $tag = "build";
08 my $dir = path( "$Bin/.." )->realpath;
09
10 sysrun "docker", "build", "-t", $tag, ".";
11
12 sysrun qw( docker run --rm --name buildc -v ),
13     "$dir:/mybuild", $tag, "bash", "-c",
14     "cd /mybuild; perl Makefile.PL; make test; make tardist";
```

Tagging Releases

To later determine which state of the source tree a release is based on, the build process needs to mark the status in Git, usually with a tag that contains the release number:

```
git tag release_1.01
git push --tags origin
```

If `origin` refers to the remote Git repository, the following `push` command with the `--tags` argument ensures locally applied tags in the Git repository get copied to GitHub for everyone to see. If you want to reproduce bugs present in previous releases later, you can get the source code's historical state during the release in question back, by checking it out with

```
git checkout -b testbug release_1.01
```

The new `testbug` branch then contains the status quo at the time of authoring.

Creating Packages

CPAN accepts tarballs, but to give your users more convenience, you may want to convert the build into a package for the target distributions such as Debian or RPM. If you are not worried about downloading about half of all CPAN modules ever created as dependencies, you can go for Dist::Zilla. For those who prefer a more lightweight approach, I recommend the Ruby tool `fpm` [7].

With a reasonably fresh Ruby version, the practical tool installs itself after you type `gem install fpm`. It supports numerous options, but to bundle a number of files into a package using one of the supported formats (RPM, Debian, or OS X), you will only need to specify the package type with `-t` and point `fpm` to the source files under `usr` to be included with `-s usr`.

In the directory, as the `tree` command in Figure 4 shows, the files are stored exactly as they will be placed on the system when installed, that is with the `foo` script in `usr/bin/foo` and the Perl module `Foo.pm` in `usr/lib/perl5/site_perl/Foo.pm`. The example specifies Debian as the package format (`-t deb`); a `.deb` file with the specified version number is thus created.

The tool is fantastically easy to use and shields developers from what are sometimes quite tragic implementations of specific package bundlers, such as `rpmbuild`. It can also resolve dependencies on other packages, and it is certainly something that should not be missing from any tool box.

Into the Box

Some new projects may have dependencies on a whole bunch of CPAN modules, for which packages for the end user's distribution may not exist. The time-consuming workaround of opening a CPAN shell and initializing, followed by installing a few dozen modules, which in turn have dependencies on more modules, is something end users shouldn't be burdened with. This is especially true if the target group normally has nothing to do with Perl, and only wants to use a new command-line utility provided by the project.

In these cases, the Carton CPAN module provides invaluable services, because it automatically resolves dependencies stored in a `cpanfile` with the following format:

```
requires 'Log::Log4perl', '1.0';
requires 'Pod::Usage', '0.01';
```

After onboarding indirectly dependent modules, with carton install, it launches a build process for the whole enchilada in a newly created `local` directory. If you then push the results into the directory of the utility discussed in the previous section, `fpm`, the tool puts together a package that will work on the target systems regardless of installed modules, thus resulting in a robust release.

Bugs? What Bugs?

All programmers make mistakes, and sometimes you need to put on your crime scene cleaner suit and trudge through the code with Perl's debugger to see where the problem lies. If the problem does not occur at the beginning of the script, but in the middle of a module, you can set a break point in the associated function, step through to the point in question manually, or use my favorite Perl trick: A statement planted at the desired location in the code, like this:

```
$DB::single = 1;
```

stops the debugger started with `perl -d <script>` and `c` at precisely this point. But, sometimes `perl` rummages through code even before the actual show in the main program begins. For example, the ORM wrapper Rose::DB creates its data structures while a database module is being loaded with the `use My::Data` command.

How can you launch a script that uses this module, so that the debugger stops in My::Data and not just in the first line of the main program? If you add the following line

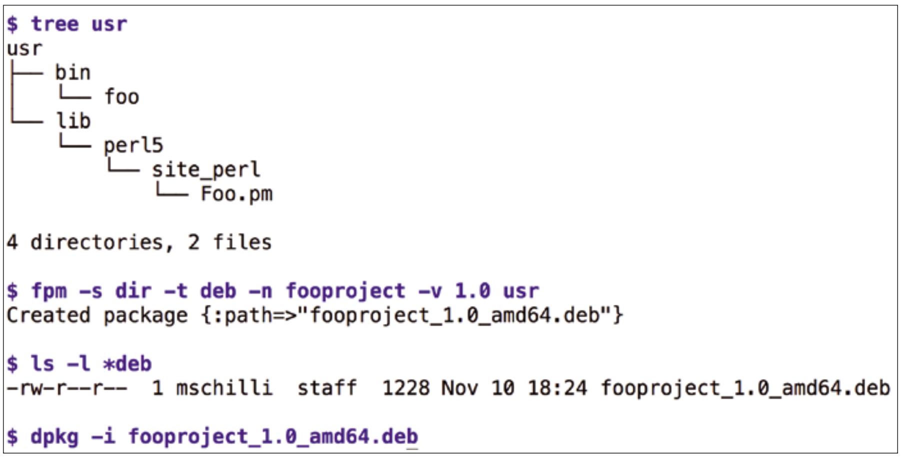
```
BEGIN { $DB::single = 1 }
```

at the beginning of the script, the debugger stops at the first executable line, regardless of whether it's in the main program or somewhere else.

Write Down Everything

Often, a developer stumbles across ugly code constructs such as duplications or obvious bugs that output ugly warnings at run time. Then, a small distraction happens, such as the discovery of even more catastrophic failures that must be resolved immediately or a work disruption in the form of a meeting with the boss – and the previously discovered error falls through the cracks. Nothing is more annoying than to see errors that you already discovered pop up later during a live demo or in production, because you let them slip through.

Therefore, make a note of everything, either in a note system such as Evernote or, if you have the discipline to repeatedly look for to-do tags in your code, with such keywords. Beware, however; many developers never actually look for these later, as is evident by many an open source module that contains forgotten to-do constructs, even if the code has seen years of production. Automatic search as part of the release process makes sure the cleanup actually happens later.



```
$ tree usr
usr
├── bin
│   └── foo
└── lib
    └── perl5
        └── site_perl
            └── Foo.pm

4 directories, 2 files

$ fpm -s dir -t deb -n fooproject -v 1.0 usr
Created package {:path=>"fooproject_1.0_amd64.deb"}

$ ls -l *deb
-rw-r--r--  1 mschilli  staff  1228 Nov 10 18:24 fooproject_1.0_amd64.deb

$ dpkg -i fooproject_1.0_amd64.deb
```

Figure 4: The package bundling tool `fpm`, which dumps the files residing below `usr` into a Debian package.

INFO

[1] "Agile, Test Driven Development" by Mike Schilli, *Linux Magazine*, Issue 154, 2013: *http://www.linux-magazine.com/Issues/2013/154/Perl-Test-Driven-Development*

[2] The author's Vimrc file: *https://github.com/mschilli/dotfiles*

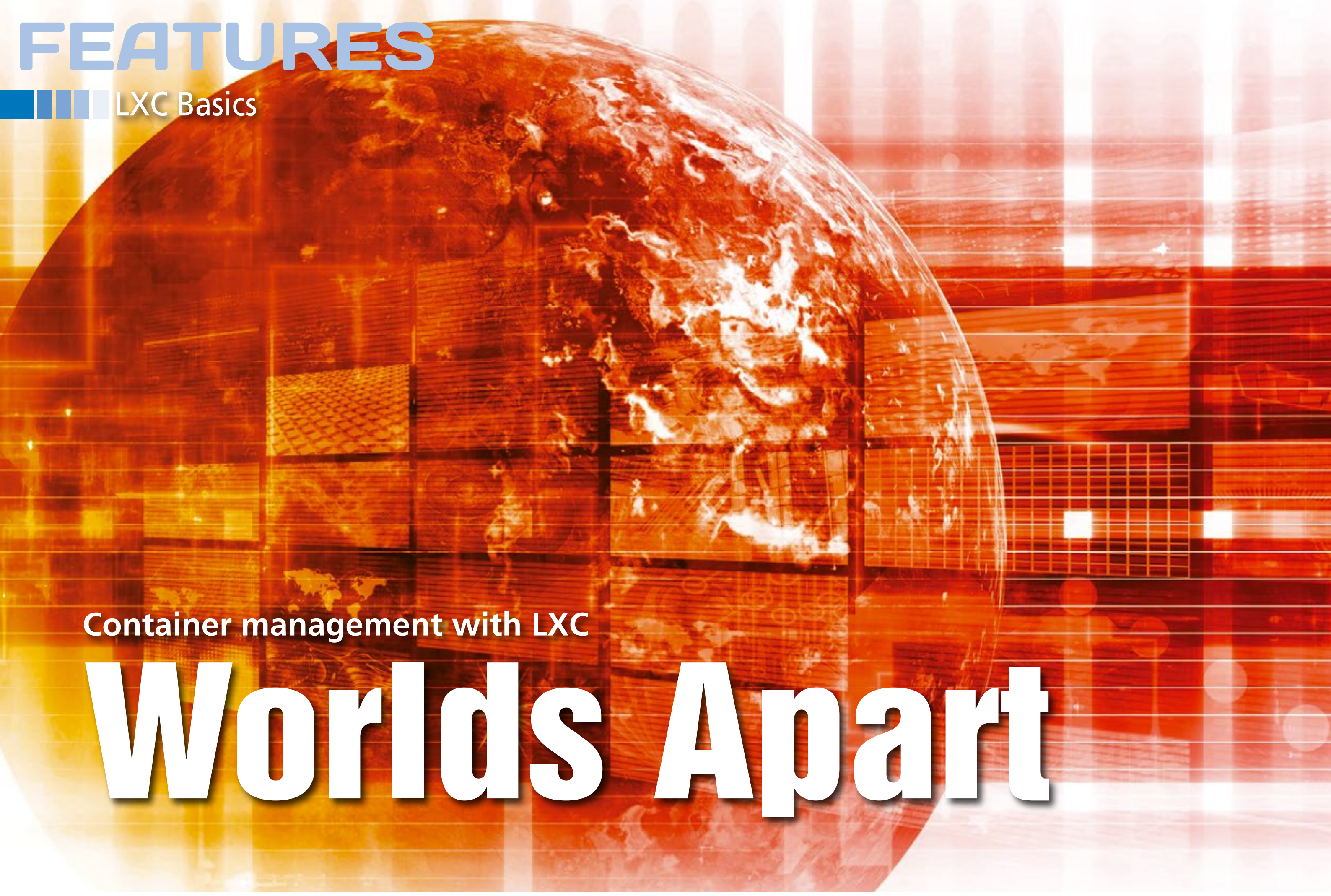
[3] "Headache-Free Continuous Integration" by Mike Schilli, *Linux Magazine*, Issue 140, 2012: *http://www.linux-magazine.com/Issues/2012/140/Perl-Travis-CI*

[4] "Perl Script as a Sniffer with Built-In Statistics" by Mike Schilli, *Linux Magazine*, Issue 180, 2015: *http://www.linux-magazine.com/Issues/2015/180/Perl-Build-a-Sniffer*

[5] "A Lifting Experience at the Dock" by Mike Schilli, *Linux Magazine*, Issue 163, 2014: *http://www.linux-magazine.com/Issues/2014/163/Perl-Testing-Modules-with-Docker*

[6] Listings for this article: *ftp://ftp.linux-magazine.com/pub/listings/magazine/192/Perl*

[7] Effing Package Management: *https://github.com/jordansissel/fpm/wiki*



Container management with LXC

Worlds Apart

Chroot lets you run a largely autonomous guest system on a computer, without the need to emulate hardware. The LXC container management tool refines this technique with a comprehensive range of additional functions. *By Harald Zisler*

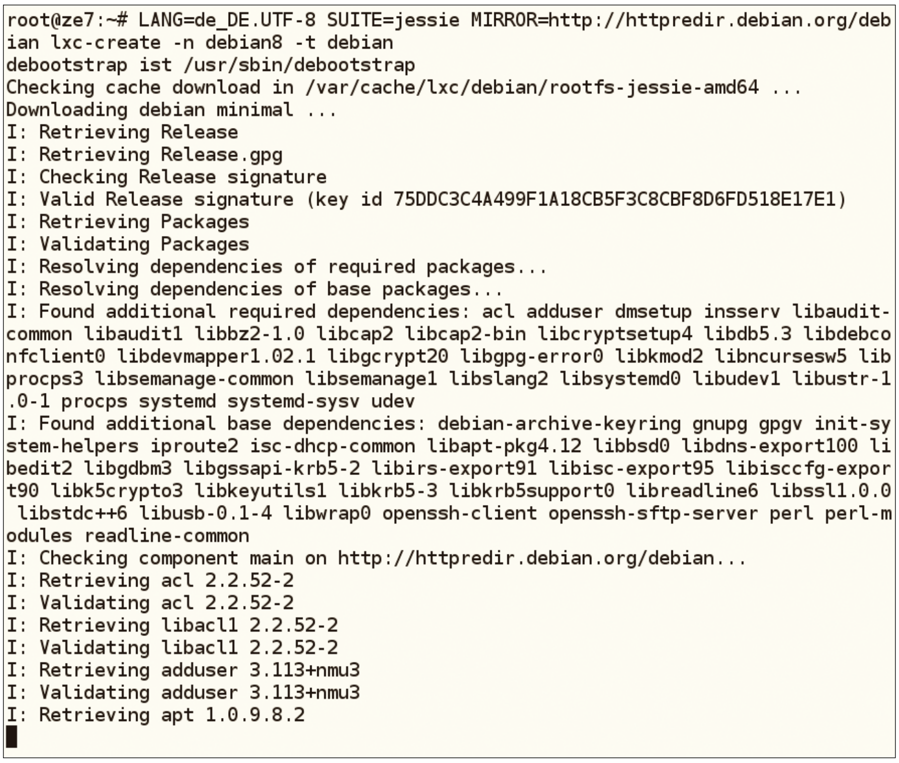
Chroot seals off parts of a directory tree against a break-out, thus isolating users, groups, or special services in an area that is harmless for the rest of the system. FreeBSD refined the concept with “Jails,” Solaris with “Zones,” and Linux with container services such as LXC [1]. LXC does not use virtual machines, like VMware, but creates a virtual environment with its own processes, but using a shared kernel on the host system.

Getting Started

The following exercise uses Debian 8.0 “jessie” as the host, but it can also be transferred to other systems like openSUSE or Ubuntu with minor changes.

AUTHOR

Harald Zisler has focused on FreeBSD and Linux since the early 1990s. He writes magazine articles and books on engineering and computer-related topics and offers Linux and database training in small groups.



```
root@ze7:~# LANG=de_DE.UTF-8 SUITE=jessie MIRROR=http://httpredir.debian.org/deb
ian lxc-create -n debian8 -t debian
debootstrap ist /usr/sbin/debootstrap
Checking cache download in /var/cache/lxc/debian/rootfs-jessie-amd64 ...
Downloading debian minimal ...
I: Retrieving Release
I: Retrieving Release.gpg
I: Checking Release signature
I: Valid Release signature (key id 75DDC3C4A499F1A18CB5F3C8CBF8D6FD518E17E1)
I: Retrieving Packages
I: Validating Packages
I: Resolving dependencies of required packages...
I: Resolving dependencies of base packages...
I: Found additional required dependencies: acl adduser dmsetup insserv libaudit-
common libaudit1 libbz2-1.0 libcap2 libcap2-bin libcryptsetup4 libdb5.3 libdebco
nfclient0 libdevmapper1.02.1 libgcrypt20 libgpg-error0 libkmod2 libncursesw5 lib
procps3 libsemanage-common libsemanage1 libslang2 libsystemd0 libudev1 libustr-1
.0-1 procps systemd systemd-sysv udev
I: Found additional base dependencies: debian-archive-keyring gnupg gpgv init-sy
stem-helpers iproute2 isc-dhcp-common libapt-pkg4.12 libbsd0 libdns-export100 li
bedit2 libgdbm3 libgssapi-krb5-2 libirs-export91 libisc-export95 libisccfg-expor
t90 libk5crypto3 libkeyutils1 libkrb5-3 libkrb5support0 libreadline6 libssl1.0.0
 libstdc++6 libusb-0.1-4 libwrap0 openssh-client openssh-sftp-server perl perl-m
odules readline-common
I: Checking component main on http://httpredir.debian.org/debian...
I: Retrieving acl 2.2.52-2
I: Validating acl 2.2.52-2
I: Retrieving libacl1 2.2.52-2
I: Validating libacl1 2.2.52-2
I: Retrieving adduser 3.113+nmu3
I: Validating adduser 3.113+nmu3
I: Retrieving apt 1.0.9.8.2
█
```

Figure 1: When you create a new container, LXC downloads all the required elements and more or less sets up a turnkey system.

Lead Image © Kheng Ho Toh, 123RF.com

To begin, you need to install the *lxc*, *lxctl*, and *libvirt-bin* packages, including the dependencies required by the package manager (e.g., *debootstrap*). If you have a custom kernel, you need to enable control groups (cgroups), which you can check with the command:

```
lxc-checkconfig
```

Depending on the installation source, creating the root filesystem for a container can take some time. On Debian, perform this task with the command:

```
$ LANG=en_US.UTF-8 SUITE=jessie MIRROR=⮐
  http://httpredir.debian.org/debian ⮐
  lxc-create -n debian8 -t debian
```

The `LANG` switch defines the language settings, and `SUITE` defines the desired template. The `-n` option lets you assign the container name (Figure 1). Subsystem templates created with the `-t` option come from the `/usr/share/lxc/templates/` directory. However, you have shorter way to accomplish the same thing, as an alternative proposal from the Debian wiki [2] for creating the container shows:

```
$ lxc-create -n debian8 ⮐
  -t debian -- -r jessie
```

After successful installation, the procedure ends by outputting a randomly generated root password, which you should write down and change with the `passwd` command when you first log in to the container. Entering `lxc-ls` lists all existing containers.

To access the container you just created at a local level, launch it with the

```
lxc-start -n debian8 -d
```

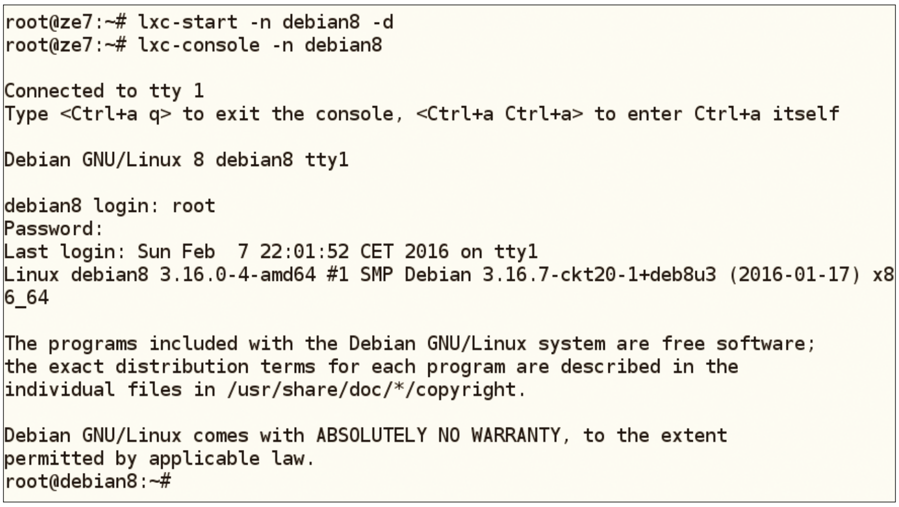
command. As with almost all `lxc` commands, you use the `-n` option to name the container (Figure 2). Make sure you enter the `-d` option when starting; otherwise, the login prompt appears, and you cannot kill it in the same terminal! Now request a console with the login prompt:

```
lxc-console -n debian8
```

To help you discover the status and process ID of an active container and to stop the current container, use the

```
lxc-info -n <container>
lxc-stop -n <container>
```

commands. For a selection of the most important control programs, see Table 1.



```
root@ze7:~# lxc-start -n debian8 -d
root@ze7:~# lxc-console -n debian8

Connected to tty 1
Type <Ctrl+a q> to exit the console, <Ctrl+a Ctrl+a> to enter Ctrl+a itself

Debian GNU/Linux 8 debian8 tty1

debian8 login: root
Password:
Last login: Sun Feb  7 22:01:52 CET 2016 on tty1
Linux debian8 3.16.0-4-amd64 #1 SMP Debian 3.16.7-ckt20-1+deb8u3 (2016-01-17) x8
6_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@debian8:~#
```

Figure 2: After launching the container, you can log in via a terminal.

Network

To access the container on the network, you need to adjust the network settings on the host and guest. In both cases you need to edit the `/etc/network/interfaces` file. Additionally, you need to configure the network settings for the container in `/var/lib/lxc/<Container>/config`. To address the container with its own name on the LAN, add an entry to `/etc/hosts` (see the box "Network Configuration"); then, run `adduser <user>` to create a user in the container. Now enter

```
systemctl status sshd.service
```

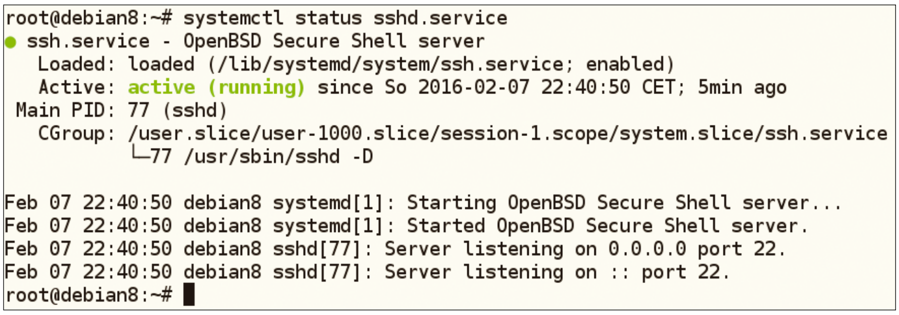
to check whether the SSH server is running in the container (Figure 3). When you pop up a new terminal, log in with

```
ssh <user>@<container>
```

and respond to the prompt for the new SSH keys; the login to the shell should work. You can log off again by typing `exit`.

TABLE 1: Important LXC commands

Action	Command
Create, Clone, Delete	
Create a container	`lxc-create -n <container> -t <template>`
Clone a container	`lxc-clone -o <container> -n <NewContainer>`
Delete a container	`lxc-destroy -n <container>`
Starting and Stopping	
Start a container	`lxc-start -n <container>`
Freeze a container	`lxc-freeze -n <container>`
Continue a container	`lxc-unfreeze -n <container>`
Stop a container	`lxc-stop -n <container>`
Information	
List all containers	`lxc-ls`
List all active containers	`lxc-ls ---active`
List all stopped containers	`lxc-ls ---stopped`
Info for a container	`lxc-info -n <container>`
IP address of the container	`lxc-info -i -n <container>`
Process ID of the container	`lxc-info -p -n <container>`
Monitor a container	`lxc-monitor -n <container>`
Running Applications	
Start an application in the container	`lxc-attach -n <container> --<command>`



```
root@debian8:~# systemctl status sshd.service
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled)
   Active: active (running) since So 2016-02-07 22:40:50 CET; 5min ago
 Main PID: 77 (sshd)
   CGroup: /user.slice/user-1000.slice/session-1.scope/system.slice/ssh.service
           └─77 /usr/sbin/sshd -D

Feb 07 22:40:50 debian8 systemd[1]: Starting OpenBSD Secure Shell server...
Feb 07 22:40:50 debian8 systemd[1]: Started OpenBSD Secure Shell server.
Feb 07 22:40:50 debian8 sshd[77]: Server listening on 0.0.0.0 port 22.
Feb 07 22:40:50 debian8 sshd[77]: Server listening on :: port 22.
root@debian8:~# █
```

Figure 3: Checking the status of the SSH daemon inside the container.

Cloning Containers

To create a clone of the `debian8` container, you just type:

```
lxc-clone -o original -n new
```

But this only works if you stop the guest system first. Cloning in our lab took about 20 seconds. Next, change the IP address under `/var/lib/lxc/debian_clone/config` and `/var/lib/lxc/debian_clone/rootfs/etc/network/interfaces` to suit your needs. You also need to add an appropriate entry for the cloned container to the `hosts` files. When you start the cloned container, try to open a connection via SSH. The configuration file of the clone might have more entries than those of the original – this is normal.

Program Management and Updates

Using the container console, update the installed packages in the usual way with `apt-get update` followed by `apt-get upgrade`. By using the package manager, you can install additional software packages from the listed sources. To access a wider program selection, add the desired repositories to the `/etc/apt/sources.list` file in the container (Listing 4). You do not need to start the updates from within the running container: You can also do this with `lxc-attach` from the host [3].

Preferably, you should run the updates in one fell swoop on the host system. In Listing 5, the shell script updates the host, terminates all active containers, and then restarts them individually for updating. The `-qq` switch that follows `apt-get` suppresses all output – use it only after a successful test run. The same applies for `-y` ("yes"), which waives all commands without prompting. The `lxc-wait` command in the update loop ensures that the container is ready for operation.

Future

In the Debian Administrator's Handbook [4] you will find some valuable information on using LXC with Debian and Debian-based distributions. Additionally, you can find detailed information about the network configuration for containers online [5], as well on display redirection for GUI applications [6]. ■■■

NETWORK CONFIGURATION

In the following examples, the network is 192.168.0.0/24, with an IP address of 192.168.0.37 for the host system and 192.168.0.90 for the guest; the container name is `debian8`. You need to adjust the values from the listings to match your local conditions.

Listing 1 shows the changes to the `/etc/network/interfaces` file on the host; for correct name resolution in this case, also add the line `192.168.0.90 debian8` to the `/etc/hosts` file. Then, working on the host system, open the `/var/lib/lxc/debian8/config` file, and add the entries from Listing 2. Finally, modify the `/var/lib/lxc/debian8/rootfs/etc/network/interfaces` file as shown in Listing 3.

The settings configure the network card in the host as a bridge (`br0` instead `eth0`). The container has a virtual Ethernet (`veth`) interface to which `br0` is connected.

After completing the changes, you first start the network on the host with the

```
/etc/init.d/networking restart
```

command. Now test the configuration by pinging the IP address of the container. To verify name resolution, you can specify the hostname instead of the IP address.

LISTING 1: Host /etc/network/interfaces

```
iface eth0 inet manual
auto br0
iface br0 inet static
   bridge_ports eth0
   bridge_stp off
   bridge_fd 0
   bridge_maxwait 0
   address 192.168.0.37
   netmask 255.255.255.0
   network 192.168.0.1
   broadcast 192.168.0.37
   gateway 192.168.0.1
   dns-nameservers 192.168.0.1
```

LISTING 2: Host /var/lib/lxc/debian8/config

```
# Network:
lxc.network.type=veth
lxc.network.link=br0
lxc.network.flags=up
lxc.network.hwaddr=00:16:3e:10:20:30
lxc.network.ipv4=192.168.0.90
```

LISTING 3: Container Network Setup

```
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet static
   address 192.168.0.90
   netmask 255.255.255.0
   network 192.168.0.0
   broadcast 192.168.0.90
   gateway 192.168.0.1
   dns-nameservers 192.168.0.1
```

LISTING 4: Adding Repositories

```
# /etc/apt/sources.list
deb http://httpredir.debian.org/debian jessie main contrib non-free
deb http://ftp.de.debian.org/debian/ jessie main contrib non-free
deb http://security.debian.org/ jessie/updates main contrib non-free
deb http://ftp.de.debian.org/debian/ jessie-updates main contrib non-free
```

LISTING 5: Updating Host System

```
#! /bin/sh
echo "--- Update the host system -----"
apt-get -qq update
apt-get -qq -y upgrade
apt-get -qq -y clean
apt-get -qq -y autoclean
apt-get -qq autoremove
echo "--- Terminate container -----------"
for i in $(lxc-ls --active -1); do
  echo $i
  lxc-stop -n $i
done
echo "--- Start/update container ---"
for n in $(lxc-ls --stopped -1); do
  echo $n
  lxc-start -n $n -d
  lxc-wait -n $n -s 'RUNNING'
  sleep 5
  lxc-attach -n $n -- apt-get -qq update
  lxc-attach -n $n -- apt-get -qq -y upgrade
  lxc-attach -n $n -- apt-get -qq -y clean
  lxc-attach -n $n -- apt-get -qq -y autoclean
  lxc-attach -n $n -- apt-get -qq autoremove
  lxc-stop -n $n
done
echo "--- Upgrade completed ---"
```

INFO

[1] LXC: *https://linuxcontainers.org*

[2] LXC in the Debian wiki: *https://wiki.debian.org/LXC*

[3] Software updates for containers: *http://www.cyberciti.biz/faq/how-to-update-debian-or-ubuntu-linux-containers-lxc*

[4] Debian handbook: *https://debian-handbook.info/browse/en-US/stable/sect.virtualization.html#sect.lxc*

[5] Network setup with LXC: *http://unix.stackexchange.com/questions/50201/how-to-configure-external-ip-addresses-for-lxc-guests*

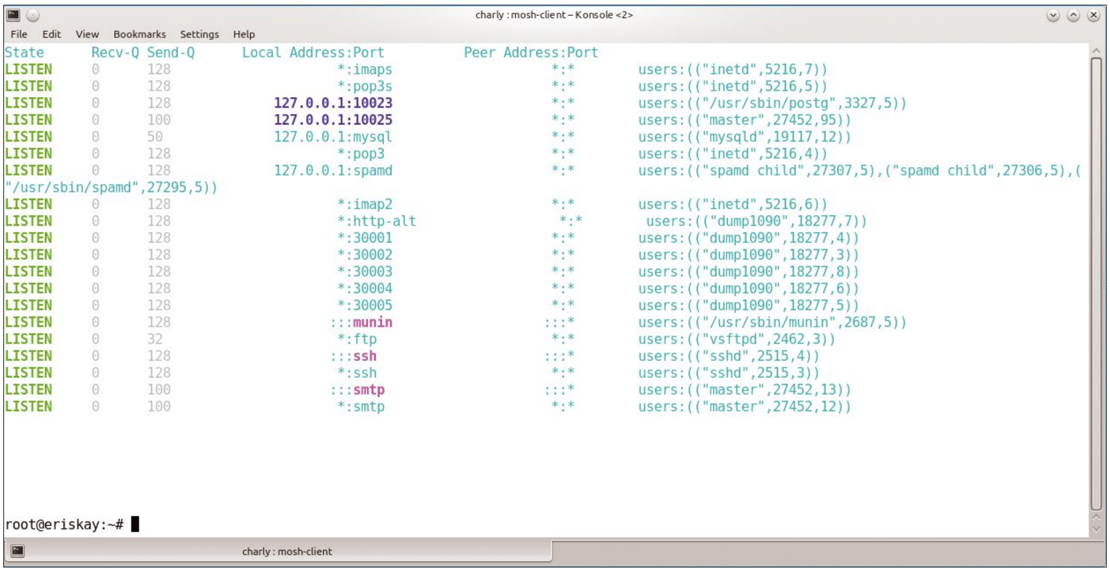
[6] Display redirection for GUI applications: *https://newspaint.wordpress.com/2015/09/14/how-do-i-get-x11-applications-running-in-a-lxc-container/*

The sys admin's daily grind: socket statistics

Short Cut

Most sys admins use netstat to find out about the status of network sockets, but Charly knows a good shortcut. *By Charly Kühnast*

A bunch of parameters control the behavior of `netstat`. Administrators can also happily combine options so that some `netstat` calls look like an armadillo has rolled across the keyboard. With `ss` (like socket statistics), there is an even more specific tool for this purpose. It comes from the *iproute2* package [1] – just like its siblings `bridge`, `rtacct`, `rtmon`, `tc`, `ctstat`, `lnstat`, `nstat`, `routef`, `routel`, `rtstat`, and `ip`.



```
State      Recv-Q Send-Q     Local Address:Port          Peer Address:Port
LISTEN     0      128                *:imaps                    *:*        users:(("inetd",5216,7))
LISTEN     0      128                *:pop3s                    *:*        users:(("inetd",5216,5))
LISTEN     0      128          127.0.0.1:10023                  *:*        users:(("/usr/sbin/postg",3327,5))
LISTEN     0      100          127.0.0.1:10025                  *:*        users:(("master",27452,95))
LISTEN     0      50           127.0.0.1:mysql                  *:*        users:(("mysqld",19117,12))
LISTEN     0      128                *:pop3                     *:*        users:(("inetd",5216,4))
LISTEN     0      128          127.0.0.1:spamd                  *:*        users:(("spamd child",27307,5),("spamd child",27306,5),(
"/usr/sbin/spamd",27295,5))
LISTEN     0      128                *:imap2                    *:*        users:(("inetd",5216,6))
LISTEN     0      128                *:http-alt                  *:*        users:(("dump1090",18277,7))
LISTEN     0      128                *:30001                    *:*        users:(("dump1090",18277,4))
LISTEN     0      128                *:30002                    *:*        users:(("dump1090",18277,3))
LISTEN     0      128                *:30003                    *:*        users:(("dump1090",18277,8))
LISTEN     0      128                *:30004                    *:*        users:(("dump1090",18277,6))
LISTEN     0      128                *:30005                    *:*        users:(("dump1090",18277,5))
LISTEN     0      128               :::munin                   :::*        users:(("/usr/sbin/munin",2687,5))
LISTEN     0      32                 *:ftp                      *:*        users:(("vsftpd",2462,3))
LISTEN     0      128               :::ssh                     :::*        users:(("sshd",2515,4))
LISTEN     0      128                *:ssh                      *:*        users:(("sshd",2515,3))
LISTEN     0      100               :::smtp                    :::*        users:(("master",27452,13))
LISTEN     0      100                *:smtp                     *:*        users:(("master",27452,12))

root@eriskay:~#
```

Figure 1: The socket collector `ss`, color-supported here, too, provides a good overview.

Because a running Linux uses several hundred or more ports, `ss` comes with plenty of filters. I mostly need the `-t` and `-u` parameters, which restrict the results to TCP or UDP sockets. Other parameters limit the list to raw, Datagram Congestion Control Protocol (DCCP), package, and Unix domain sockets.

Caution: The tool only displays TCP sockets for existing connections by default (`established` or `close_wait`). If I also want to see sockets in the `lists` status – as I usually do – I have to type `ss -ta`. If I only want the tool to pay attention to listening TCP sockets and suppress all others, I use `ss -lt`. If I add `-p` here, I can also see which process opened the socket. I need to use root privileges for this; otherwise, `-p` has no effect.

Numbers, Please!

A further default is that `ss` translates the port numbers from the `/etc/services` file into names. I can switch this off using `-n` (for numeric). If I want the tool to resolve the IP addresses concerned into names, I just add an `-r`. Using `-4` and `-6`, I can limit it to one of the two TCP/IP versions.

I find it useful to filter by source (`sport`) and destination port (`dport`). For example,

```
# ss -natp6 sport = :25
LISTEN   0   100   :::25   :::* ↩
  users:(("master",27452,13))
```

shows whether the mail server is listening via IPv6 (yes) and which process is on port 25. Logical links are also okay:

```
# ss -t4 dport = :443 or dport = :80
ESTAB 0 0 10.0.0.201:53389 ↩
  10.0.0.118:http
```

`ccze` [2] sorted the coloration in Figure 1, but `grc` [3] or `rainbow` [4] would have managed it, too. ■■■

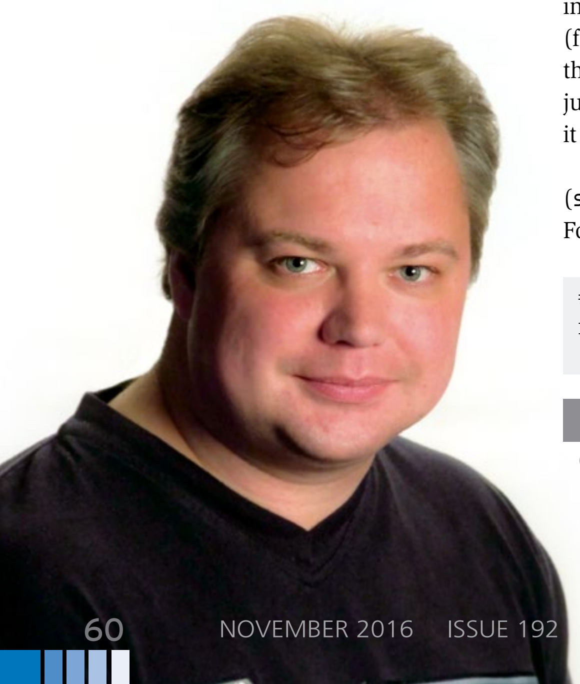
INFO

[1] Iproute2: *http://www.linuxfoundation.org/collaborate/workgroups/networking/iproute2*

[2] Ccze: *http://freecode.com/projects/ccze/*

[3] Grc: *http://kassiopeia.juls.savba.sk/~garabik/software/grc.html*

[4] Rainbow: *https://github.com/nicoulaj/rainbow*



CHARLY KÜHNAST

Charly Kühnast manages Unix systems in the data center in the Lower Rhine region of Germany. His responsibilities include ensuring the security and availability of firewalls and the DMZ.

Emacs for writers

Text Mode

With the pointers and useful tips in this article, you can turn Emacs into a powerful writing environment. *By Dmitri Popov*

DMITRI POPOV

Dmitri Popov has been writing exclusively about Linux and open source software for many years, and his articles have appeared in Danish, British, US, German, Spanish, and Russian magazines and websites. Dmitri is an amateur photographer, and he writes about open source photography tools on his Scribbles and Snaps blog at *scribblesandsnaps.wordpress.com*.

Emacs can be anything you want it to be: a coding environment, an outliner, a task manager, and everything in between. You can transform this versatile text editor into a tool for writing articles, drafting blog posts, and penning your next great novel (Figure 1). As is often the case, the exact Emacs configuration and customizations depend on your specific needs and usage scenarios, but there are several tweaks and plugins you might find useful no matter what writing tasks you plan.

Configuring and Extending Emacs

The `.emacs` file in your home directory specifies Emacs' default behavior and configures a wide range of settings. The example configuration in Listing 1 contains several useful commands and options. The default monospaced font and line spacing make regular text difficult to read, and the `set-default-font` and `setq-default line-spacing` expressions let you specify the desired font (it must be installed on your system), font size, and line spacing. In this case, Emacs is set to use the Mononoki font at the 17pt size.

By default, Emacs doesn't do soft word wrap – a de rigueur feature of any decent word processor. But, you can enable this functionality by toggling the visual line mode using the `M-x visual-line-mode` expression. To avoid doing this manually every time you work with text in Emacs, add the `global-visual-line-mode t)` expression to the `.emacs` file. On-the-fly spellcheck is another indispensable feature for any writer, and Emacs supports this functionality via the flyspell mode. You'd probably want to enable spellcheck only when working in the text mode, and this can be done by adding the `(add-hook 'text-mode-hook 'flyspell-mode)` hook expression.

Besides enabling some clever features, you'll also want to turn off functionality that is not very useful for writing. For example, Emacs starts with the Auto Fill mode enabled, which automatically breaks lines when they become too wide. This can be useful for working with code, but it can quickly become an annoyance when editing regular text. To disable this mode, add the `turn-off-auto-fill` and `remove-hook 'text-mode-hook 'turn-on-auto-fill` expressions to the `.emacs` file. The latter expression removes the hook that enables Auto Fill in the text mode.

By default, Emacs automatically saves changes and creates a backup copy of the currently edited file. This functionality litters directories with temporary and backup files, and if you prefer to keep things tidy, you can disable the auto-save and backup features using the `(setq backup-inhibited t)` and `(setq auto-save-default nil)` expressions.

The default Emacs configuration lacks a feature crucial for any writing professional – word count. The *wc.el* [1] plugin provides a simple solution to this problem. To install the plugin, download the wc.el file and move it to the ~/.emacs.d/plugins directory (create the directory manually if it doesn't exist). Then, add (add-to-list 'load-path "~/.emacs.d/plugins") and (load "wc.el") expressions to the .emacs configuration file. As you might have guessed, the first expression points Emacs to the specified plugin directory, and the second expression loads the wc.el plugin. Using the plugin couldn't be easier: Simply run the M-x count-words command to view the line, word, and character count in the minibuffer.

LISTING 1: .emacs Example Configuration

```
(set-default-font "Mononoki 17")
(setq-default line-spacing 11)
(global-visual-line-mode t)

(add-hook 'text-mode-hook 'flyspell-mode)
(turn-off-auto-fill)
(remove-hook 'text-mode-hook 'turn-on-auto-fill)
(setq backup-inhibited t)
(setq auto-save-default nil)

(add-to-list 'load-path "~/.emacs.d/plugins")
(load "wc.el")
(load "xah-lookup")
```

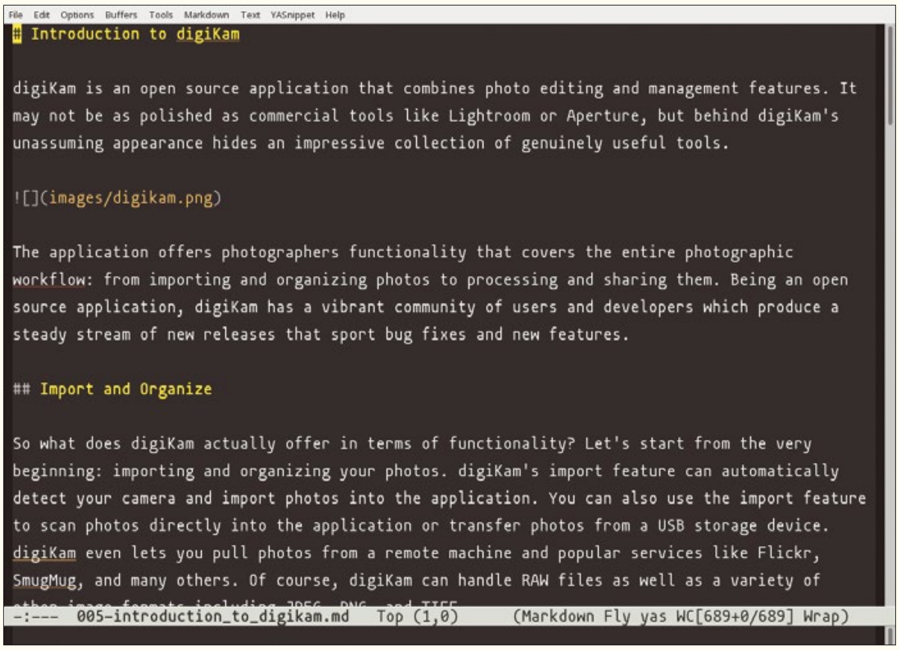
If you prefer to see live word count in the status bar, the *wc-mode* plugin [2] is what you need. To install it, clone the project's GitHub repository and move the wc-mode.el file to the ~/.emacs.d/plugins directory. Add the (require 'wc-mode) and (global-set-key "\C-cw" 'wc-mode) expressions to the configuration file to load the plugin and assign the C-x w keyboard shortcut to it.

Open a text file in Emacs, type C-c w, and you should see the word count in the status bar. The clever part is that the plugin shows not only the total word count but also the initial word count (i.e., the word count at the moment you opened the text file), plus the number of words written since. This functionality can come in handy when you set daily writing goals. For example, if you aim to write 1,000 words per day or per session, the wc-mode plugin can help you track your progress.

Being able to look up word definitions and query various resources directly from Emacs can be useful in many situations, and the *xah-lookup.el* plugin [3] puts several online references at your fingertips. To install the plugin, download the xah-lookup.el file, move it to the ~/.emacs.d/plugins directory, add the (load "xah-lookup.el") expression to the .emacs file. The plugin conveniently provides keyboard shortcuts for running queries for the word under the cursor as a search term. The F1 9 keyboard shortcut, for example, looks up the word's definition in the Free Dictionary, and the F1 8 combination performs a Wikipedia search.

Instead of the xah-lookup.el plugin, or in addition to it, you can specify a simple lookup function directly in the .emacs file. The example function below makes it possible to look up the word under the cursor in the WordNet online database and display the result in the Emacs built-in browser (Listing 2).

If you prefer to use your system's default browser, add the (require 'browse-url) expression somewhere in the .emacs file and replace the eww qUrl command in the function with (browse-url qUrl).



```
# Introduction to digiKam

digiKam is an open source application that combines photo editing and management features. It
may not be as polished as commercial tools like Lightroom or Aperture, but behind digiKam's
unassuming appearance hides an impressive collection of genuinely useful tools.

![](images/digikam.png)

The application offers photographers functionality that covers the entire photographic
workflow: from importing and organizing photos to processing and sharing them. Being an open
source application, digiKam has a vibrant community of users and developers which produce a
steady stream of new releases that sport bug fixes and new features.

## Import and Organize

So what does digiKam actually offer in terms of functionality? Let's start from the very
beginning: importing and organizing your photos. digiKam's import feature can automatically
detect your camera and import photos into the application. You can also use the import feature
to scan photos directly into the application or transfer photos from a USB storage device.
digiKam even lets you pull photos from a remote machine and popular services like Flickr,
SmugMug, and many others. Of course, digiKam can handle RAW files as well as a variety of
other image formats including JPEG, PNG, and TIFF
-:---  005-introduction_to_digikam.md   Top (1,0)      (Markdown Fly yas WC[689+0/689] Wrap)
```

Figure 1: Emacs can be transformed into a powerful writing environment.

LISTING 2: Look Up Words in WordNet

```
;Look up word in WordNet
(defun wordnet-define ()
  (interactive)
  (let (qWord qUrl)
    (setq qWord
          (if (use-region-p)
              (buffer-substring-no-properties (region-beginning) (region-end))
            (thing-at-point 'word)))
    (setq qUrl (concat "http://wordnetweb.princeton.edu/perl/webwn?s=" qWord))
    (eww qUrl)
    ))
```

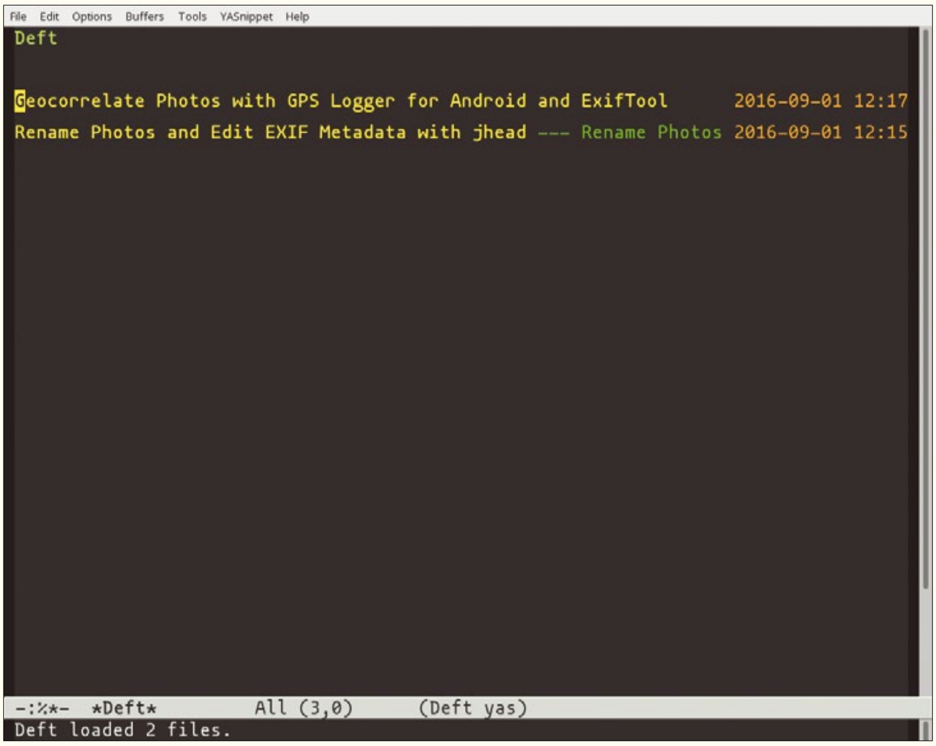
Adding Markdown Support

The basic configuration in Listing 1 takes care of the basics, but there is much more you can do to transform Emacs into a complete writing environment. If you use Markdown for text formatting, you'll appreciate the *markdown-mode* [4] plugin.

The easiest way to install the plugin is through the MELPA Stable package archive (an online plugin repository for Emacs). To do this, you need to enable MELPA Stable by adding the following configuration to the `.emacs` file:

```
(require 'package)
(add-to-list 'package-archives
             '("melpa-stable" . "https://stable.melpa.org/packages/"))
(package-initialize)
```

Restart Emacs and run the `M-x package-install RET markdown-mode RET` command (RET refers to the Return key press). Once the plugin has been installed, it automatically enables the markdown mode when you open a Markdown-formatted text. Besides Markdown syntax highlighting, the plugin provides keyboard shortcuts for frequently used commands and formatting options. For example, the `C-c C-a l` command inserts an inline link, and the `C-c C-i i` command inserts an inline image. The plugin also features commands for working with lists, navigating outlines, previewing, and exporting Markdown-formatted files, etc.



```
Deft

Geocorrelate Photos with GPS Logger for Android and ExifTool       2016-09-01 12:17
Rename Photos and Edit EXIF Metadata with jhead --- Rename Photos  2016-09-01 12:15

-:%*-  *Deft*          All (3,0)      (Deft yas)
Deft loaded 2 files.
```

Figure 2: Deft transforms Emacs into an efficient note-taking tool.

Note Taking with Deft

If you want to use Emacs for note taking, Deft [5] is right up your alley (Figure 2). To install Deft, download the `deft.el` file, move it into the `~/.emacs.d/plugins` directory, and add the `(load "deft.el")` expression to the `.emacs` file. Launch Emacs and run the `M-x deft-setup` command to create the `~/.deft` directory for storing notes.

To launch Deft, use the `M-x deft` command. Alternatively, you can specify a global keyboard binding to launch Deft using a keyboard shortcut. Add, for example, `(global-set-key [f8] 'deft)` to the `.emacs` file to start Deft by pressing the F8 key. When Deft is running, it opens a dedicated buffer that lists the titles of all text files in the default Deft directory. Each item in the list contains a short summary and modification time. The items are sorted by last modified date, from newest to oldest.

To create a new note, either run the M-x deft-new-file command or use the C-c C-n keyboard shortcut. You can also rename notes using the C-c C-r shortcut and delete them with the C-c C-d combination. Instead of deleting notes, you can choose to archive them using the C-c C-a shortcut. By default, all archived notes are stored in the ~/.deft/archive directory.

Deft's primary function is search and filtering. Simply start typing, and Deft displays the notes that match the entered query in real time. To open the first matching item, press Return. To open any other item, use the C-p and C-n shortcuts to move up and down the filtered list.

Although Deft features sensible defaults, you can easily modify its settings. For example, if you prefer to store notes in a different directory, you can add (setq deft-directory "~/path/to/notes") to the .emacs file. By default, Deft works with notes that have .txt, .text, .md, .markdown, and .org file extensions. The (setq deft-extensions '("txt" "md")) expression lets you specify file extensions that you want Deft to recognize.

Manage Snippets with Yasnippet

Yasnippet [6] is described as "a template system for Emacs," which doesn't say a lot until you realize that this plugin expands abbreviations into text snippets assigned to them. Yasnippet is a rather powerful tool that offers a wide range of features and customization options. However, you don't need to learn all its intricacies to put this tool to some basic uses.

Of course, you need to install Yasnippet first. Use the M-x package-list-package command to fetch a list of all available packages, and then locate and install the *yasnippet* package. Create the snippets subdirectory in ~/.emacs.d and add the following expressions to the .emacs file:

```
(require 'yasnippet)
(yas-global-mode 1)
(yas-load-directory "~/.emacs.d/snippets/")
```

Before you can use Yasnippet, you need to populate the ~/.emacs.d/snippets directory with snippet files. Start with creating the text-mode directory for storing snippets that work in Emacs' text mode. Create a text file and paste the following text into it:

```
# name: Tagline
# key: tgl
# --
Better is the enemy of good enough
```

Save the file under the tgl name. As you can see, a basic snippet file consists of three parts: the name directive that provides a short description of the snippet, the key directive that defines the snippet's abbreviation, and the text that replaces the abbreviation. In this case, when you type tpl and press the Tab key, the abbreviation is replaced with the specified text. Yasnippet has other clever tricks up its sleeve, too. For example, instead of the key directive, you can use binding to assign a snippet to a keyboard shortcut:

```
# binding: C-c C-c C-d
```

This way, you can insert a snippet using the specified key combination.

Final Word

The described configuration settings and plugins make Emacs more writer-friendly, but to unleash its full potential, it's worth mastering Emacs' fundamentals. Plenty of Emacs tutorials are floating around, but The Woodnotes Guide to Emacs for Writers [7] provides a comprehensive introduction to Emacs functionality relevant for writing professionals. ■■■

INFO

[1] wc.el word count plugin: *www.emacswiki.org/emacs/wc.el*

[2] wc-mode plugin: *github.com/bnbeckwith/wc-mode*

[3] xah-lookup plugin: *ergoemacs.org/emacs/emacs_lookup_ref.html*

[4] markdown-mode plugin: *jblevins.org/projects/markdown-mode*

[5] Deft plugin: *jblevins.org/projects/deft*

[6] Yasnippet plugin: *github.com/joaotavora/yasnippet*

[7] The Woodnotes Guide to Emacs for Writers: *therandymon.com/woodnotes/emacs-for-writers/emacs-for-writers.html*

Subscribe today and join the revolution!

Each issue of Raspberry Pi Geek is an adventure, with ingenious applications and cool projects for Raspberry Pi, Arduino, and other maker-board systems!

Discover the secrets that will empower you to envision and build your own Raspberry Pi inventions.

Features

Special articles on new technologies and topics of interest to the Rasp Pi community.

RASPBERRY PI GEEK

raspberry pi GEEK

In this issue we celebrate the Raspberry Pi's fourth birthday. We also explore some new features of the newest Pi and present some projects that take advantage of these faster, more capable machines, such as 3D modeling and distributed compiling.

Features

Service

Skills

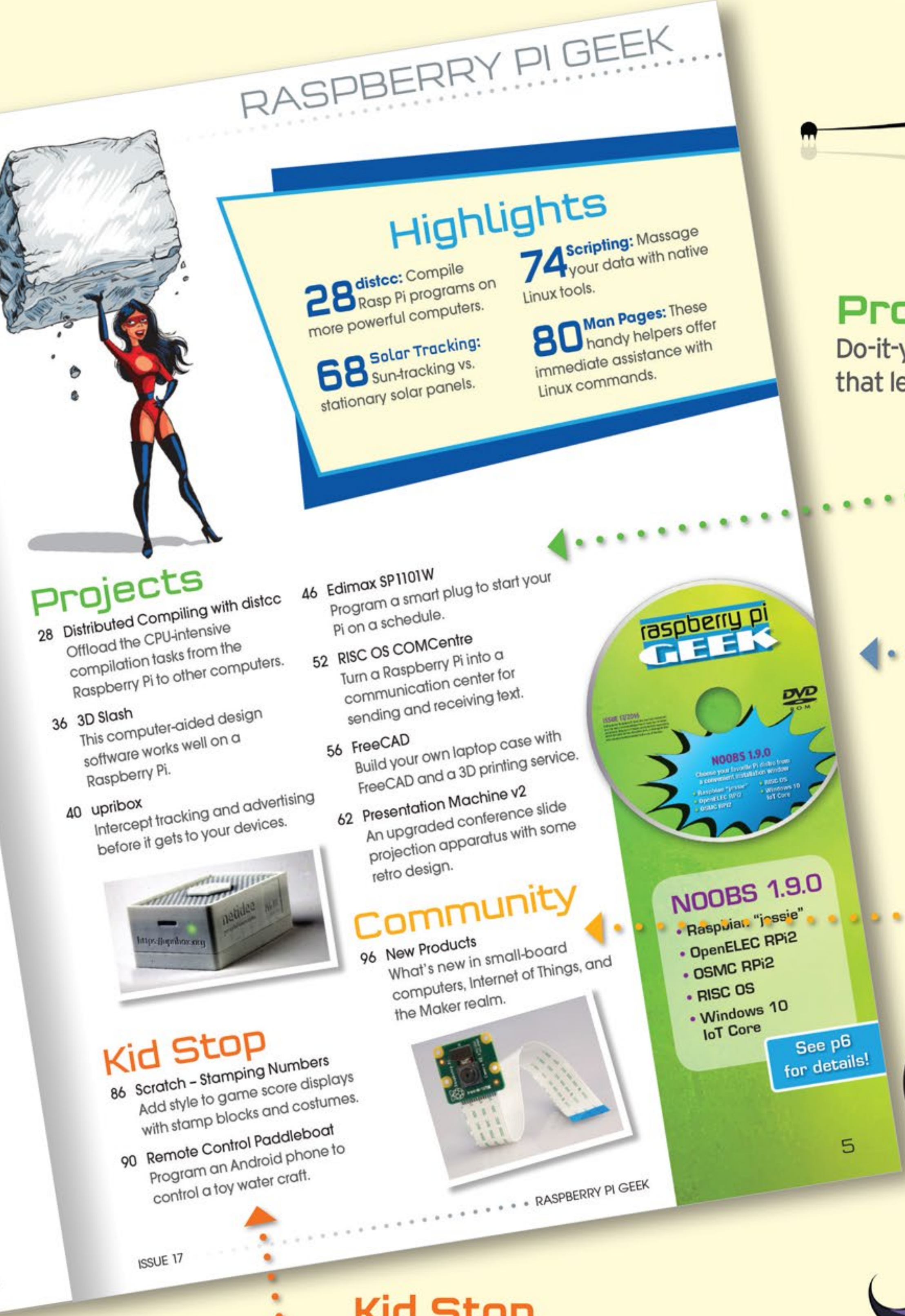
ISSUE 17

RASPBERRY PI GEEK

4

6 print issues with 6 DVDs or 6 digital issues for only

$59.95 £37.50 €44.90 shop.linuxnewmedia.com



RASPBERRY PI GEEK

Highlights

28 distcc: Compile Rasp Pi programs on more powerful computers.

68 Solar Tracking: Sun-tracking vs. stationary solar panels.

74 Scripting: Massage your data with native Linux tools.

80 Man Pages: These handy helpers offer immediate assistance with Linux commands.

Projects

Community

Kid Stop

NOOBS 1.9.0

- Raspbian "jessie"
- OpenELEC RPi2
- OSMC RPi2
- RISC OS
- Windows 10 IoT Core

See p6 for details!

5

RASPBERRY PI GEEK

ISSUE 17

Projects

Do-it-yourself, real-world projects that let you learn by doing.

DVD

Every print issue comes with a free DVD!

Community

Interviews and reports on Raspberry Pi meet-ups around the world.

Kid Stop

Special projects for kids, including a new Scratch programming exercise in every issue.

Skills

Timely tutorials to help you build your skills with Linux and other underlying technologies.

Neatly managing and handling PGP/GnuPG keyrings

Key Service

PGP/GnuPG is becoming increasingly popular, thanks to digital crime and government surveillance. We take a look behind the scenes and show how you can keep your keyring current and valid. *By Frank Hofmann*

If you want to encrypt your data traffic, you first have to gain a general understanding of the concepts and individual steps necessary. Pretty Good Privacy (PGP) or GNU Privacy Guard (GnuPG) let you create key pairs – public and private – and group your keys in keyrings. In this article, I discuss the tools for managing these keyrings and keeping them up to date, including checking them regularly for validity.

Once you have your key pair, you pass your public key to your communication partners. Traditionally, that either occurs directly as a character string in an email attachment [1], in printed form (e.g., during a key-signing party) [2], or by downloading the key from a web page. For some time, it's been possible to use QR codes that you can receive, check, and manage on your smartphone. In Figure 1, you can see the Monkeysign [3] GUI with the QR code key.

Individually exchanging keys with each communication partner is a bit complicated. To simplify the process for everyday use, key servers were established [4] (Table 1) to store your public key in a key server so that anyone can download it, validate it, and immediately use it to communicate with you. An example call using GnuPG [5] looks like this:

```
$ gpg --keyserver hkp://keys.gnupg.net ⮐
  --send-keys <key ID>
gpg: sending key <key ID> to hkp server
```

The program is simply called `gpg` on the command line, or `gpg2` in version 2. You can state the ID of the key that you want to share with the `--send-keys` switch. The `--keyserver` switch specifies the key server to be used. It does not matter which key server you initially designate or have already set up in the `~/.gnupg/gpg.conf` file [6]. All the larger key servers [7] work together and regularly compare the database of stored keys with one another [8]. Within 48 hours, the shared key generally spreads so far that it can automatically be checked for validity worldwide.

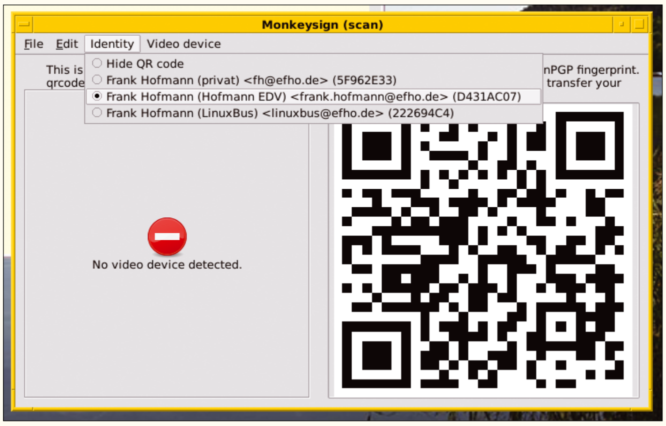
Please note that you can only upload keys for which you also hold the private key. When you use the `--list-secret-keys` switch, GnuPG delivers an overview of the private keys. Listing 1 shows an extract for one of my keys. Here, *sec* denotes the key's procedure and identification, *uid* is the user ID, and *ssb* is the secret key's procedure and identification [9]. Additionally, you can see in the readout that GnuPG analyzes the `.gnupg/secring.gpg` file, where it saves your private key.

Integrating Email

To exchange data in signed or encrypted form [10] from your email program, you need to add a suitable cryptographic signature (see "Navigating the Signature Jumble" box). As a result, you are in a position to immediately exchange your data in signed

AUTHOR

Frank Hofmann (*http://www.efho.de*) works in Berlin as a service provider at Büro 2.0, an open source experts' network specializing in printing and typesetting. Since 2008, he has coordinated the regional meeting of LUGs from the Berlin-Brandenburg Region and is co-author of the Debian-Paketmanagement-Buch (Debian Package Management Book, *http://www.dpmb.org*).



○ Hide QR code
○ Frank Hofmann (privat) <fh@efho.de> (5F962E33)
● Frank Hofmann (Hofmann EDV) <frank.hofmann@efho.de> (D431AC07)
○ Frank Hofmann (LinuxBus) <linuxbus@efho.de> (222694C4)

No video device detected.

Figure 1: A GnuPG QR code key.

TABLE 1: A Selection of Key Servers

Name	URL
SKS	`hkp://pool.sks-keyservers.net`
MIT	`hkp://pgp.mit.edu`
PGP	`ldap://keyserver.pgp.com`
GnuPG	`hkp://keys.gnupg.net`

LISTING 1: Listing Secret Keys

```
$ gpg --list-secret-keys
/home/frank/.gnupg/secring.gpg
-----------------------------
sec   4096R/D431AC07 2014-09-05
uid   Frank Hofmann (Hofmann EDV) <frank.hofmann@efho.de>
ssb   4096R/3B074F29 2014-09-05
```

NAVIGATING THE SIGNATURE JUMBLE

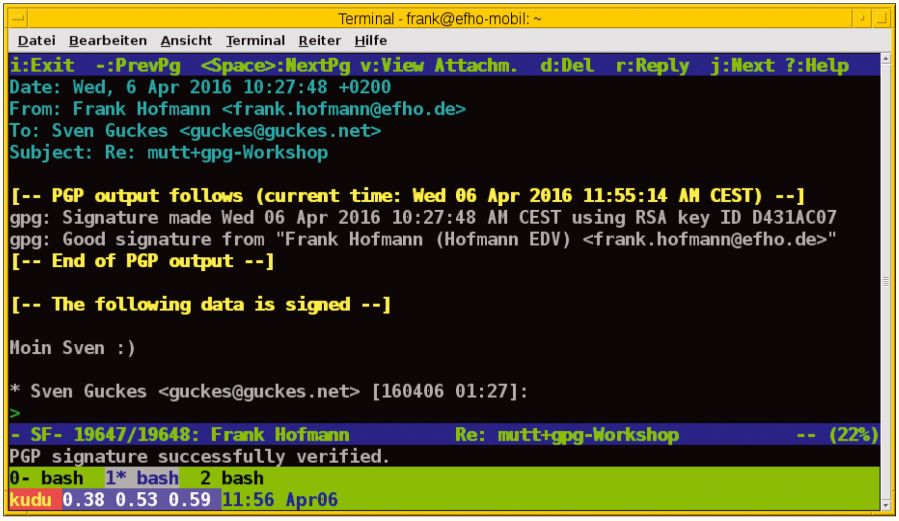
The cryptographic signature is completely different from the similar-sounding electronic signature [12] or email signatures for commercial transactions [13]. With a cryptographic signature, you extend the outgoing email by incorporating an additional block of text, including a header. Along with the ID of your GnuPG/PGP key, it contains the cryptographically secured checksum of the email content, thus protecting both the sender and the content. Both the electronic signature and signature for email in commercial use are variants of X.509 certificates [14]. The signatures sit at the end of the email as a block of text in plain language and more or less fulfil the function of a business card or document header on a company's letterhead.

and/or encrypted form. Figure 2 shows an example using the text-based email client Mutt [11]. The text lines in yellow show that the sender of this message has signed cryptographically. The PGP key has the ID *D431AC07*, is based on the RSA procedure, and is recognized as valid. In short, this email very likely came from the sender designated in the message.

All mail clients currently offer a suitable extension for dealing with signed and encrypted email. They differ clearly, however, in terms of the complexity of setting up and using the extension. You can find a comprehensive step-by-step guide for Mozilla Thunderbird/Icedove combined with the Enigmail [15] encryption plugin on the Debian wiki [16].

Displaying Existing Keys

Figure 3 shows a selection of the public keys with the name of the author; all interested parties have unrestricted access to these keys on the key server. This form of Internet-based request can serve as a reliable reference point for everyday use. Each line contains the key type (*pub*, or the public part of the key), the key length with the procedure (e.g., *4096R* indicates a 4096-bit RSA key), the key identifier (*ID*), the date of key creation, and the name (*User ID*) and email address.


Terminal - frank@efho-mobil: ~
Datei Bearbeiten Ansicht Terminal Reiter Hilfe

```
i:Exit  -:PrevPg  <Space>:NextPg v:View Attachm.  d:Del  r:Reply  j:Next ?:Help
Date: Wed, 6 Apr 2016 10:27:48 +0200
From: Frank Hofmann <frank.hofmann@efho.de>
To: Sven Guckes <guckes@guckes.net>
Subject: Re: mutt+gpg-Workshop

[-- PGP output follows (current time: Wed 06 Apr 2016 11:55:14 AM CEST) --]
gpg: Signature made Wed 06 Apr 2016 10:27:48 AM CEST using RSA key ID D431AC07
gpg: Good signature from "Frank Hofmann (Hofmann EDV) <frank.hofmann@efho.de>"
[-- End of PGP output --]

[-- The following data is signed --]

Moin Sven :)

* Sven Guckes <guckes@guckes.net> [160406 01:27]:
>
- SF- 19647/19648: Frank Hofmann          Re: mutt+gpg-Workshop          -- (22%)
PGP signature successfully verified.
0- bash  1* bash  2 bash
kudu 0.38 0.53 0.59 11:56 Apr06
```

Figure 2: Email with verified cryptographic signature.

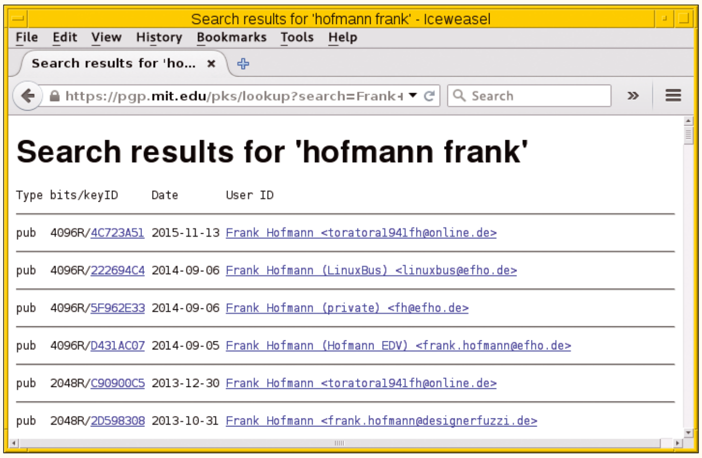
If you compare Figures 2 and 3, you will note that the email's key ID in Figure 2 is identical to the one listed in line 4 of Figure 3. As is apparent from Figure 3, more than one key is associated with "Frank Hofmann," but only the keys in lines 2-4 belong to the author.

Checking the Other Side

All communication partners can automatically verify cryptographically signed email, provided the public PGP/GnuPG keys are stored on a public key server or the participants have a copy of the originator's public key.

GnuPG hangs the keys you received from the key server on your locally saved keyring. To check this, you need an existing Internet connection, unless you have access to a complete copy of the public key server's data pool (rather unlikely).

Listing 2 shows the full call on the command line to download a key. The `--keyserver` switch designates the server; here, `hkp://keys.gnupg.net` gives the key server network a generic specification. Specify the ID of the public key you want to receive with the `--recv-keys` switch, such as for the RSA key with the ID `C76E337A`.


Search results for 'hofmann frank' - Iceweasel
File Edit View History Bookmarks Tools Help
Search results for 'ho... ×
https://pgp.mit.edu/pks/lookup?search=Frank+
Search

Search results for 'hofmann frank'

```
Type bits/keyID     Date       User ID

pub  4096R/4C723A51 2015-11-13 Frank Hofmann <toratoral941fh@online.de>

pub  4096R/222694C4 2014-09-06 Frank Hofmann (LinuxBus) <linuxbus@efho.de>

pub  4096R/5F962E33 2014-09-06 Frank Hofmann (private) <fh@efho.de>

pub  4096R/D431AC07 2014-09-05 Frank Hofmann (Hofmann EDV) <frank.hofmann@efho.de>

pub  2048R/C90900C5 2013-12-30 Frank Hofmann <toratoral941fh@online.de>

pub  2048R/2D598308 2013-10-31 Frank Hofmann <frank.hofmann@designerfuzzi.de>
```

Figure 3: The public key with the name of the author.

Reading the Keyring

As already discussed, you can group several keys together during everyday use with the help of a keyring. GnuPG maintains separate files for public and private keys, each ending with the `.gpg` suffix. The `pubring.gpg` file contains the keyring with the public keys, and the `secring.gpg` file encompasses the private keys. You can visually display the content of the keyring of public keys (Listing 3) with the `gpg -k` command (long option, `--list-public-keys`). The secret keys use the equivalent `-K` or `--list-secret-keys`.

LISTING 2: Downloading a Key

```
$ gpg --keyserver hkp://keys.gnupg.net --recv-keys C76E337A
gpg: request key C76E337A from hkp-Server keys.gnupg.net an
gpg: key C76E337A: public key "Wolfram Eifler <adagio@weifler.in-berlin.de>" imported
gpg: 3 marginal-needed, 1 complete-needed, PGP trust model
gpg: depth: 0  valid:   3  signed:   2  trust: 0-, 0q, 0n, 0m, 0f, 3u
gpg: depth: 1  valid:   2  signed:   2  trust: 2-, 0q, 0n, 0m, 0f, 0u
gpg: total number of keys processed: 1
gpg:                              imported: 1  (RSA: 1)
```

LISTING 3: Displaying Public Keys

```
$ gpg -k
/home/frank/.gnupg/pubring.gpg
------------------------------
pub   4096R/D431AC07 2014-09-05 uid       [ unlimited] Frank Hofmann (Hofmann EDV)
<frank.hofmann@efho.de>
sub   4096R/3B074F29 2014-09-05

pub   1024D/DFA0A4D4 2008-01-10
uid        [ unknown] Gerold Rupprecht (home email key) <geroldr@bluewin.ch>
sub   2048g/F9E8DE2F 2008-01-10

pub   4096R/612616B5 2009-07-12
uid        [ full] Axel Beckert <abe@deuxchevaux.org>
uid        [ full] Axel Stefan Beckert
uid        [ full] Axel Beckert (FSFE fellow) <abe@fsfe.org>
uid        [ full] Axel Beckert (Symlink) <xtaran@symlink.ch>
uid        [ full] Axel Beckert (E-mail + Jabber) <abe@noone.org>
uid        [ full] Axel Beckert (Debian developer) <abe@debian.org>
uid        [ full] [jpeg image of size 3155]
sub   4096g/004AB7CC 2009-07-12
```

Validating Keys

GnuPG has two ways to validate the key: the `--edit-key` [17] and `--check-sigs` switches. To use either, you also need the name of the key you want to check.

For `--edit-key`, GnuPG clarifies whether a suitable secret key exists and opens a shell in which you can make modifications to the key with GnuPG-specific commands. Checking for validity is the only thing of interest here, meaning that the `check` command comes into play. In Listing 4, the readout shows that the key is valid.

In the second, shorter call in Listing 5, GnuPG identifies an expired key. In the readout, the program signals the status of the check with a flag directly following the *sig* text. The exclamation point indicates a successful validation, a minus sign a bad signature, and a percentage sign an error that arose during validation.

To check the whole keyring, call up GnuPG again with the `--check-sigs` switch. Do not designate a specific key name this time, however, and you will receive an overview of the status of all the keys on the keyring.

Updating the Local Keyring

Keys already received can expire or be revoked, expanded, or modified by the owner. To update one of several entries in the local keyring, GnuPG has the `--refresh-keys` switch. If you designate a key name, the tool only updates that one; otherwise, it works for all existing entries in the keyring (Listing 6). In this context, a signature means that another user has signed and so confirmed the key.

GUI Key Management

If graphical programs appeal more to you than tools for the command line, you should take a look at Seahorse (Gnome) [18] or KGpg (KDE) [19], as well as the GNU Privacy Assistant (GPA) [20].

You can manage certificates (Figures 4 and 5), as well as GnuPG and SSH keys, with the compact Seahorse application. It is integrated within the Ubuntu and Gnome desktops. To keep your local GnuPG keyring up to date, Seahorse can synchronize the available keys with the key server network on request with the *Remote* | *Match and share keys* menu item.

On first sight, the GPA operates more comprehensively than Seahorse, but concentrates on GnuPG key management (Figure 6). In GPA, for example, you can add keys, modify and sign them, or match your keyring with the key server network.

Conclusion

Your keyring is always up to date with little effort using

LISTING 4: Opening a GPG Shell

```
$ gpg --edit-key "Frank Hofmann"
gpg (GnuPG) 1.4.18; Copyright (C) 2014 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Secret key is available.

pub  4096R/D431AC07  created: 2014-09-05  expired: never     call: SC
                     Trust: unlimited Validity: unlimited
sub  4096R/3B074F29  created: 2014-09-05  expired: never     call: E
[ unlimited] (1). Frank Hofmann (Hofmann EDV) <frank.hofmann@efho.de>

gpg> check
uid  Frank Hofmann (Hofmann EDV) <frank.hofmann@efho.de>
sig!3        D431AC07 2014-09-05  [self-authentication]

gpg> quit
```

LISTING 5: Checking Keys

```
$ gpg --check-sigs "Wolfram Eifler"
pub   2048R/C76E337A 2011-05-11 [expired: 2016-05-09]
uid                  Wolfram Eifler <adagio@weifler.in-berlin.de>
sig!3        C76E337A 2011-05-11  Wolfram Eifler <adagio@weifler.in-berlin.de>

1 Certification not checked due to missing key
```

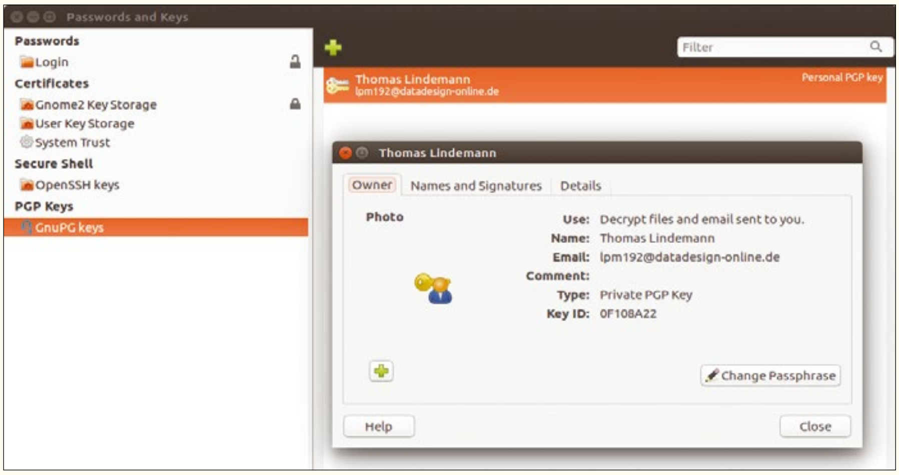
LISTING 6: Refreshing Keys

```
$ gpg --keyserver hkp://keys.gnupg.net --refresh-keys
gpg: 8 Keys updated with hkp://keys.gnupg.net
gpg: request key D431AC07 from hkp-Server keys.gnupg.net
gpg: request key 5F962E33 from hkp-Server keys.gnupg.net
gpg: request key 222694C4 from hkp-Server keys.gnupg.net
gpg: request key DFA0A4D4 from hkp-Server keys.gnupg.net
gpg: request key 612616B5 from hkp-Server keys.gnupg.net
gpg: request key 4A2E162B from hkp-Server keys.gnupg.net
gpg: request key 95930EDE from hkp-Server keys.gnupg.net
gpg: request key C76E337A from hkp-Server keys.gnupg.net
gpg: key D431AC07: "Frank Hofmann (Hofmann EDV) <frank.hofmann@efho.de>" unmodified
gpg: key 5F962E33: "Frank Hofmann (private) <fh@efho.de>" unmodified
gpg: Key 222694C4: "Frank Hofmann (LinuxBus) <linuxbus@efho.de>" unmodified
gpg: key DFA0A4D4: "Gerold Rupprecht (home email key) <geroldr@bluewin.ch>" 1 new signature
gpg: key 612616B5: "Axel Beckert <abe@deuxchevaux.org>" 190 new signatures
gpg: key 612616B5: "Axel Beckert <abe@deuxchevaux.org>" 1 new subkey
gpg: key 4A2E162B:"Axel Beckert (ISG D-PHYS) <beckert@phys.ethz.ch>" 2 new signatures
gpg: key 95930EDE: "Axel Beckert (no legal relevance) <abe@deuxchevaux.org>" 14 new signatures
gpg: key C76E337A: "Wolfram Eifler <adagio@weifler.in-berlin.de>" unmodified
gpg: total number of keys processed: 8
gpg:                         unmodified: 4
gpg:                         new subkeys: 1
gpg:                         new signatures: 207
gpg: 3 marginal-needed, 1 complete-needed, PGP trust model
gpg: depth: 0  valid:   3  signed:   2  trust: 0-, 0q, 0n, 0m, 0f, 3u
gpg: depth: 1  valid:   2  signed:   2  trust: 2-, 0q, 0n, 0m, 0f, 0u
```

GnuPG and its graphical front ends. To browse for more information, I recommend the GnuPG wiki [21] and an article on OpenPGP best practices [22]. The question of where and how you can store private keys and their related files securely has remained unanswered to this point. ■■■

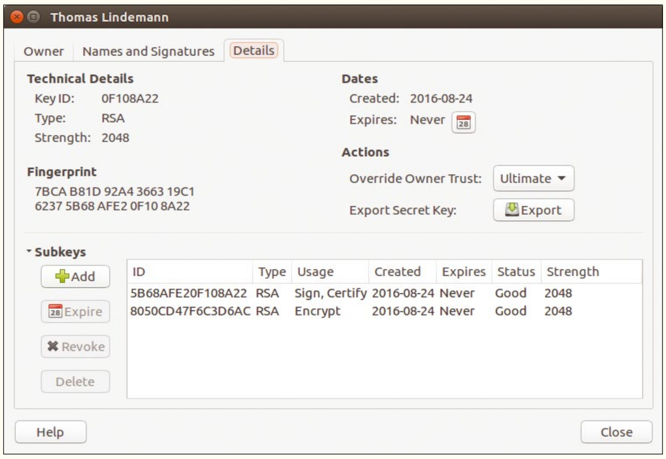
NOTE OF THANKS

The author thanks Sebastian Andres, Wolfram Eifler, Sven Guckes, Gerold Rupprecht, and Martin Ebnöther for their suggestions and criticism before the publication of this article.



Use: Decrypt files and email sent to you.
Name: Thomas Lindemann
Email: lpm192@datadesign-online.de
Comment:
Type: Private PGP Key
Key ID: 0F108A22

Figure 4: Seahorse delivers an overview of keys and signatures.



Technical Details
Key ID: 0F108A22
Type: RSA
Strength: 2048

Fingerprint
7BCA B81D 92A4 3663 19C1
6237 5B68 AFE2 0F10 8A22

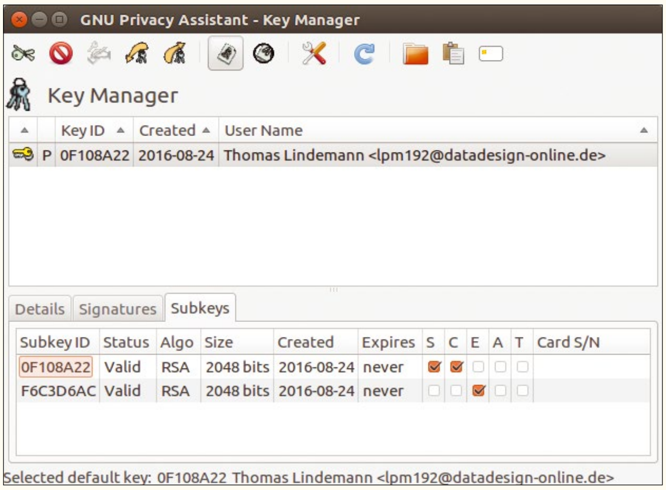
Dates
Created: 2016-08-24
Expires: Never

Actions
Override Owner Trust: Ultimate
Export Secret Key: Export

Subkeys

ID	Type	Usage	Created	Expires	Status	Strength
5B68AFE20F108A22	RSA	Sign, Certify	2016-08-24	Never	Good	2048
8050CD47F6C3D6AC	RSA	Encrypt	2016-08-24	Never	Good	2048

Figure 5: Seahorse presents the details of a GnuPG or SSH key on request.



		Key ID	Created	User Name
	P	0F108A22	2016-08-24	Thomas Lindemann <lpm192@datadesign-online.de>

Subkey ID	Status	Algo	Size	Created	Expires	S	C	E	A	T	Card S/N
0F108A22	Valid	RSA	2048 bits	2016-08-24	never	☑	☑	□	□	□	
F6C3D6AC	Valid	RSA	2048 bits	2016-08-24	never	□	□	☑	□	□	

Selected default key: 0F108A22 Thomas Lindemann <lpm192@datadesign-online.de>

Figure 6: The GPA keeps all your keys in sight.

INFO

[1] "PGP with GnuPG" by Martin Loschwitz, *Ubuntu User*, issue 24, 2015, pg. 56, *http://www.ubuntu-user.com/Magazine/Archive/2015/24/Reliably-encrypting-emails-using-GnuPG*

[2] Key signing party: *https://en.wikipedia.org/wiki/Key_signing_party*

[3] Monkeysign: *http://web.monkeysphere.info/monkeysign/*

[4] "Submitting your GPG key to a keyserver": *https://debian-administration.org/article/451/Submitting_your_GPG_key_to_a_keyserver*

[5] GnuPG: *https://www.gnupg.org*

[6] GPG options: *https://www.gnupg.org/documentation/manuals/gnupg/GPG-Options.html*

[7] Key servers: *https://sks-keyservers.net/status/*

[8] GnuPG server pools: *https://sks-keyservers.net/overview-of-pools.php*

[9] "Creating a new GPG key with subkeys": *https://www.void.gr/kargig/blog/2013/12/02/creating-a-new-gpg-key-with-subkeys/*

[10] Sending encrypted emails using Thunderbird and PGP: *http://www.wefightcensorship.org/article/sending-encrypted-emails-using-thunderbird-and-pgphtml.html*

[11] Mutt and GnuPG: *https://dev.mutt.org/trac/wiki/MuttGuide/UseGPG*

[12] The difference between electronic signatures and digital signatures: *https://www.globalsign.com/en/blog/electronic-signatures-vs-digital-signatures/*

[13] Tank, Margo H. K., Sara E. Emley, and R. David Whitaker. A Brief Guide to Using Electronic Signatures in Securities Transactions, *http://www.buckleysandler.com/uploads/1082/doc/A-Brief-Guide-to-Using-Electronic-Signatures-in-Securities-Transactions.pdf*

[14] X.509 certificates: *https://en.wikipedia.org/wiki/X.509*

[15] Enigmail (DEB): *https://packages.debian.org/jessie/enigmail*

[16] Setting up OpenPGP encryption in Thunderbird/Icedove, *https://wiki.debian.org/Icedove#Setting_up_OpenPGP_Encryption*

[17] The GNU Privacy Handbook, Chapter 3, Key Management: *https://www.gnupg.org/gph/en/manual/c235.html*

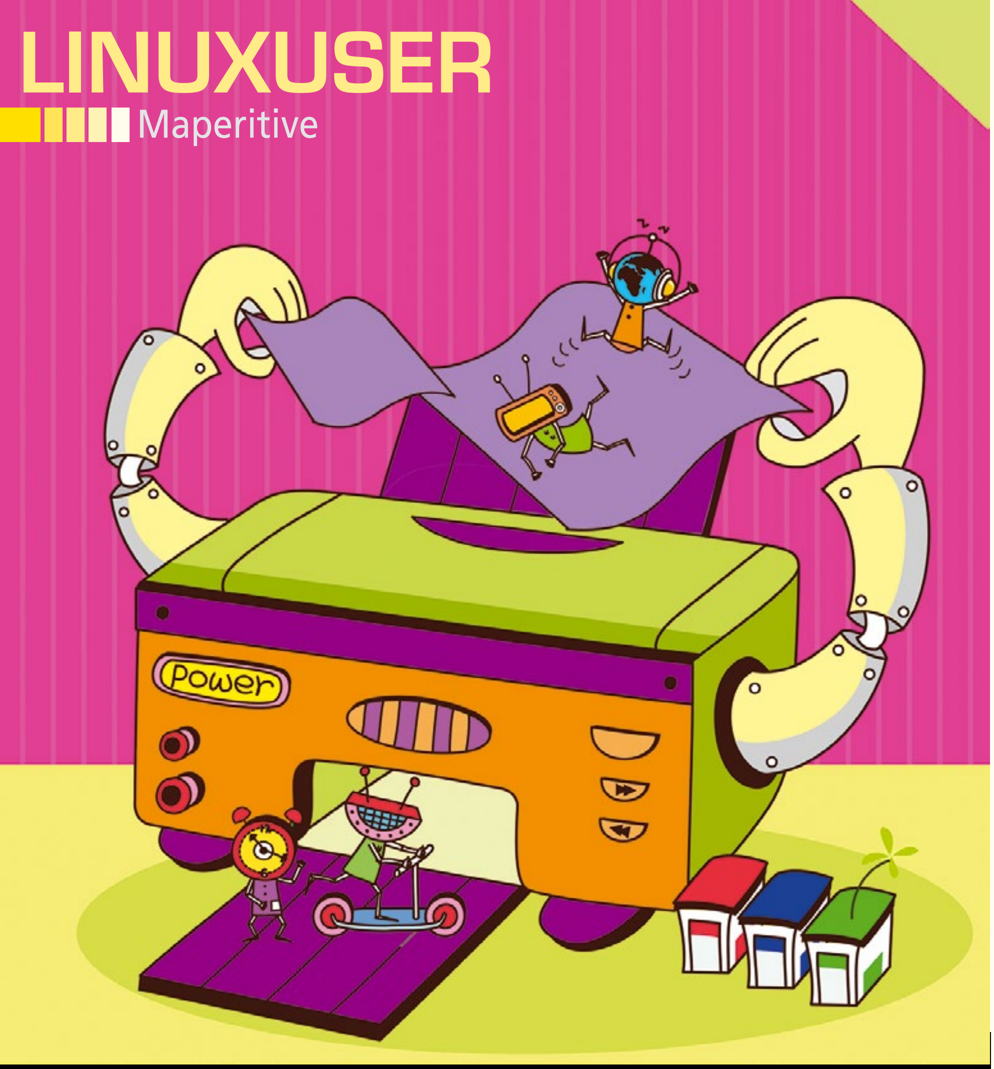
[18] Seahorse: *https://wiki.gnome.org/Apps/Seahorse*

[19] KGpg: *https://utils.kde.org/projects/kgpg/*

[20] GNU Privacy Assistant: *https://www.gnupg.org/related_software/gpa/index.html*

[21] GnuPG wiki: *https://wiki.gnupg.org*

[22] OpenPGP best practices: *https://help.riseup.net/en/security/message-security/openpgp/best-practices*



Enrich OSM maps with additional information

Custom Print

Maperitive helps users create high-quality maps with additional information from OpenStreetMap data.

By Karsten Günther

The word has spread that OpenStreetMap (OSM) offers high-quality data for many areas. Even established GPS manufacturers are now using this data source – at least for outdoor devices, where users require minute detail. This makes using the data provided by the OSM project for your own maps all the more desirable. Currently, only printed maps display sufficiently large areas at a high resolution with enough details for planning trips.

Another benefit is that you can modify the OpenStreetMap data yourself. If you need a specific cycling route or want to see the picnic areas on the maps, for example, you have two options. First, you can add the information directly to your project with the online editor or with programs like Merkaartor [1], Josm, or others. Second, tools such as Maperitive [2] let you generate maps in high resolution as image files, which you can then edit with Gimp or Inkscape.

The first method – that is, editing the OSM data – is preferred because it benefits the entire project. However, for very specific tasks, the second method is better and faster because you do not need to wait for the data to update on the server. On the other hand, you do need to re-enter your own changes each time you reload the database.

In many online publications, you will find OSM maps as screenshots, but this is only useful for ebooks. For printed publications, the resolution (typically under 100 dpi) of the images provided in the browser is too low to ensure legible results. The classic way of producing better quality images is via a separate Mapnik server, but this causes significant overhead in terms of the installation and operation.

Tools such as Maperitive enter the scene here: They generate bitmap (tiles) or SVG maps directly from the OSM data, giving you extensive control over the content.

Geodata

Maperitive reads and interprets the geodetic data and computes the maps. You can customize the data loaded from various sources [3] using styles and save the results in different formats. Several predefined styles exist for the basic types of maps; these styles are available immediately after installation. If necessary, you can customize them to produce personalized maps.

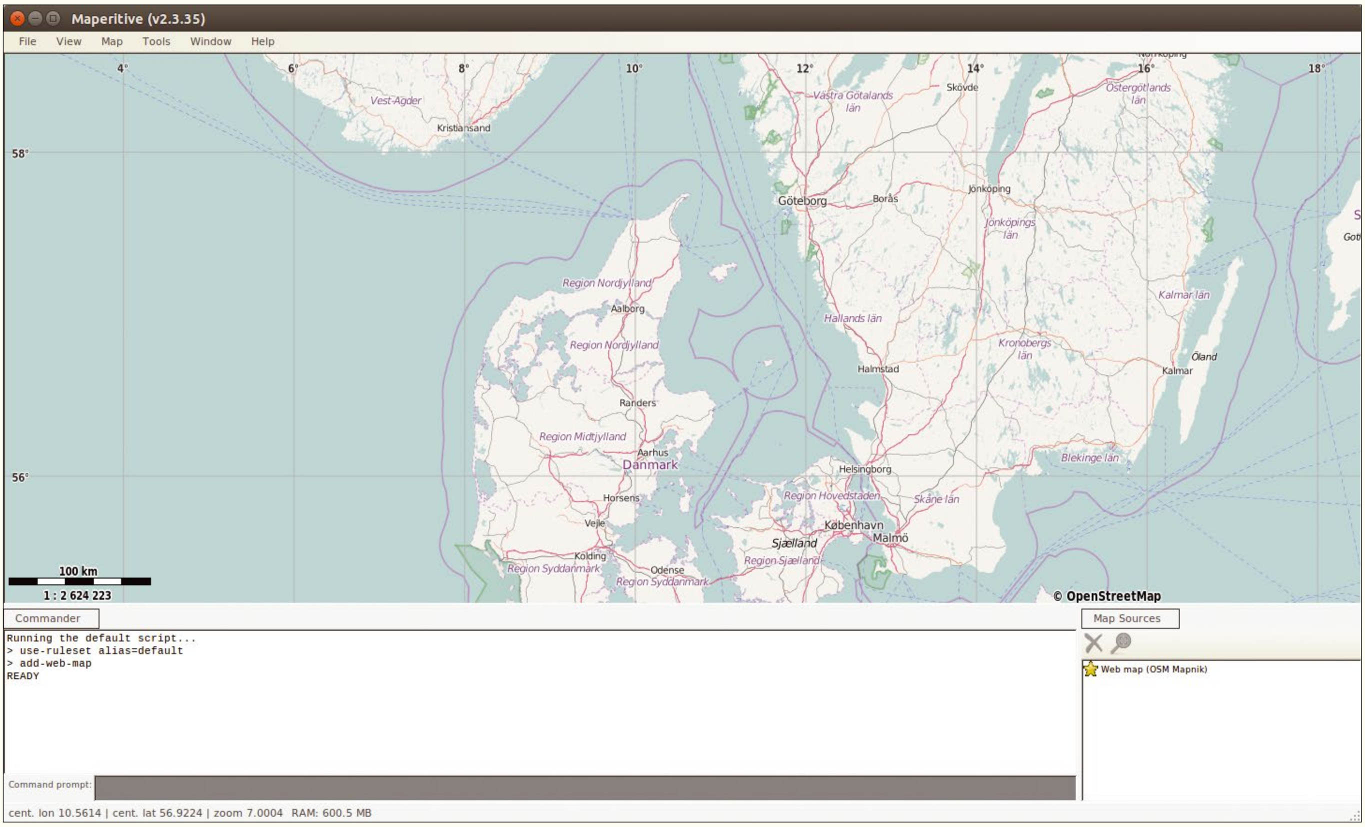
As a Mono application, Maperitive can be used across platforms, but the installation is typically a manual process (see the “Installation” box). The program makes heavy use of resources, especially CPU time and memory, for working with large maps.

Launch

After launching, Maperitive comes up with a simple interface (Figure 1). The menu at the top of the window provides basic functions; the image section offers a preview of the map. The *Commander* at the bottom left lets you enter special

INSTALLATION

Maperitive is not found in the repositories of popular distributions. Despite its dependence on Mono, the installation is quite simple: First, install the Mono framework – the package name varies between distributions. In addition, Maperitive needs *libgdiplus* version 3.8 and Python. On Ubuntu, you can then retroactively install the *libmono-winforms2.0-cil* and *mono-devel* packages if necessary. Then, unzip the zip archive containing the current version of Maperitive, preferably in `/opt/`. Next, start the Maperitive script `Maperitive.sh`. If you move the script to `/usr/local/bin/`, you will need to modify the path to `Maperitive.exe`.



```
Running the default script...
> use-ruleset alias=default
> add-web-map
READY
```

cent. lon 10.5614 | cent. lat 56.9224 | zoom 7.0004   RAM: 600.5 MB

Figure 1: Maperitive launches into a classical OSM (web) map.

commands; the *Map Sources* to the right provides an overview of the data source and lets you select a source. The status line at the bottom of the window shows the current *Zoom level,* as well as the current memory consumption, among other things.

Only a few of the functions and commands offered by Maperitive are reflected in the menu. The Commander, a shell-style interface, provides full access to all functions. Here you can define all the details of the functions, using command arguments and parameters.

In the image window, you can select the visible section with the mouse. To do this, you typically need to click on the section, because the Commander is normally active. In the selected image window, you can change the section (*Zoom level*) using the mouse wheel. If you hold down the Shift key, you can select areas of the preview and zoom them to full screen.

When launched, the program shows you the OSM map from the web referred to in the sources as *Web map (OSM Mapnik)* by default. This map uses prefabricated bitmap tiles provided by a Mapnik server whose appearance you can change with Maperitive. Because you will normally be using the software to create your own maps or test your own rules, you will likely want to disable or remove this source. To do this, select the unwanted source and press the button with the cross above the window.

Interaction with the program is mainly via the Commander window. It consists of an area for the

TABLE 1: Important Commands

Command	Function
`add-web-map`	Add web presentation
`apply-ruleset`	Apply rule set
`change-dir`	Change working directory
`clear-map`	Remove map sources
`clear-results`	Delete results of a search
`download-osm`	Read OSM data for the current bounding box (standard method XAPI)
`download-osm-overpass`	Read OSM data for the current bounding box from the Overpass Server
`edit-rules`	Open the rule set editor
`export bitmap`	Export map as bitmap graphic
`export-svg`	Export map as SVG graphic
`export-osm`	Export map as OSM file (XML)
`help`	Show help
`list-settings`	Show settings
`list-sources`	View sources
`load-source`	Download OSM, GPX, or bitmap graphic
`reload-ruleset`	Reapply the current rules
`run-script <script>`	Run Maperitive script
`save-map-script`	Save commands as script
`set-home`	Remember display
`set-setting`	Configure settings
`use-ruleset`	Load a set of rules
`use-script-dir`	Change script directory
`zoom-map-scale`	Zoom map according to scale

output (top) and an input line (*Commander prompt*). The output is also used as the log window and displays the responses to the executed commands. At the prompt, you then enter commands with arguments and run them (Table 1). The command line has a history and an auto-complete function to make working with the program more convenient.

The command line collaborates directly with the wizard (Figure 2). You can enable the wizard in the *Window* menu by selecting *Wizard* or pressing F1. It acts as an online help on the one hand, explaining the commands and their arguments. On the other hand, it inserts a command and its arguments in the command line if you double click.

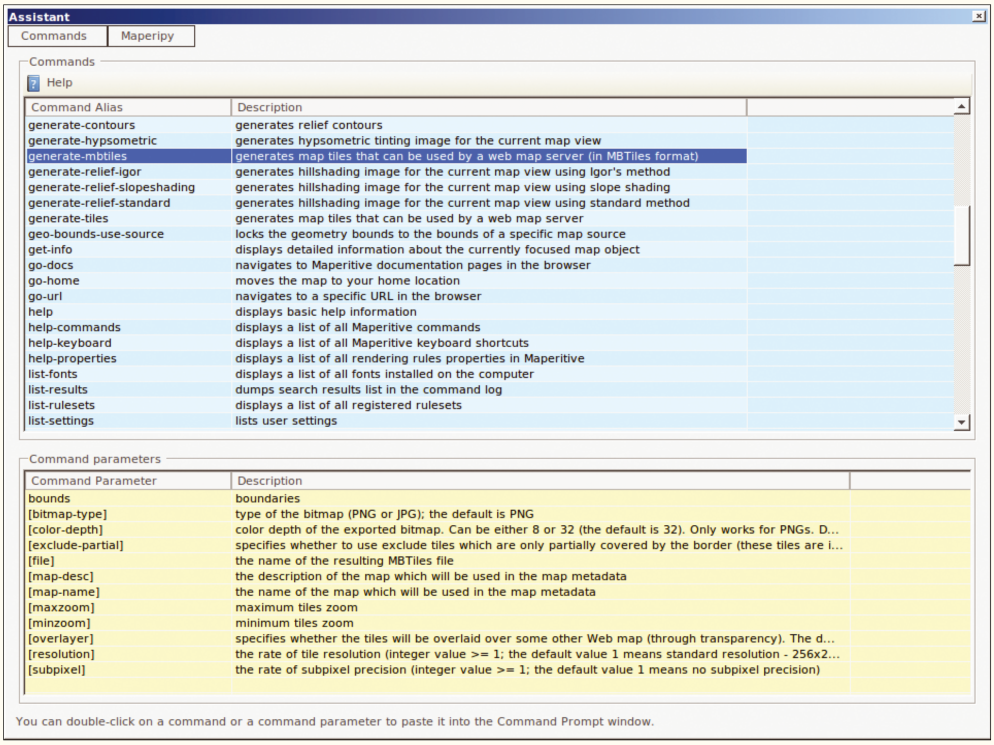
To quickly view the loaded data, the program has a couple of pre-built rule sets. You will find these styles in the *Map* menu below *Switch to rules*. After calling this function, the software computes a new view of the map, which can take some time.

Practice

The following descriptions essentially refer to a project that was created for a forthcoming ebook. However, the procedure is suitable in many parts for other projects. For this workshop, I used Maperitive version 2.3.35. The following steps create a new map:

- Download the map data.
- Create an initial overview.
- Define what the map shows (create rules).
- Test, adapt, and improve the rules.
- Install the finishing touches, such as contour lines.
- Create an output file (SVG file).
- Finishing: Edit and print the SVG file.

The data for the maps come from two sources: either directly from the OSM servers or from prefabricated OSM files. The current version of Maperitive can directly download the OSM data and even save it locally. The benefit of using locally stored maps should not be underestimated: Although the online data constantly changes, you can evaluate and adjust local records as needed. However, not all OSM file formats are suitable for all editors – Merkaartor does not read PBF files.



Command Alias	Description
generate-contours	generates relief contours
generate-hypsometric	generates hypsometric tinting image for the current map view
generate-mbtiles	generates map tiles that can be used by a web map server (in MBTiles format)
generate-relief-igor	generates hillshading image for the current map view using Igor's method
generate-relief-slopeshading	generates hillshading image for the current map view using slope shading
generate-relief-standard	generates hillshading image for the current map view using standard method
generate-tiles	generates map tiles that can be used by a web map server
geo-bounds-use-source	locks the geometry bounds to the bounds of a specific map source
get-info	displays detailed information about the currently focused map object
go-docs	navigates to Maperitive documentation pages in the browser
go-home	moves the map to your home location
go-url	navigates to a specific URL in the browser
help	displays basic help information
help-commands	displays a list of all Maperitive commands
help-keyboard	displays a list of all Maperitive keyboard shortcuts
help-properties	displays a list of all rendering rules properties in Maperitive
list-fonts	displays a list of all fonts installed on the computer
list-results	dumps search results list in the command log
list-rulesets	displays a list of all registered rulesets
list-settings	lists user settings

Command Parameter	Description
bounds	boundaries
[bitmap-type]	type of the bitmap (PNG or JPG); the default is PNG
[color-depth]	color depth of the exported bitmap. Can be either 8 or 32 (the default is 32). Only works for PNGs. D...
[exclude-partial]	specifies whether to use exclude tiles which are only partially covered by the border (these tiles are i...
[file]	the name of the resulting MBTiles file
[map-desc]	the description of the map which will be used in the map metadata
[map-name]	the name of the map which will be used in the map metadata
[maxzoom]	maximum tiles zoom
[minzoom]	minimum tiles zoom
[overlayer]	specifies whether the tiles will be overlaid over some other Web map (through transparency). The d...
[resolution]	the rate of tile resolution (integer value >= 1; the default value 1 means standard resolution - 256x2...
[subpixel]	the rate of subpixel precision (integer value >= 1; the default value 1 means no subpixel precision)

You can double-click on a command or a command parameter to paste it into the Command Prompt window.

Figure 2: The wizard has two functions: It explains the commands and lets you transfer them to the command line with a double click. There is no need to specify all the arguments for the commands.

To load the online data, first move and scale the map so that it shows the desired area. In the *Map* menu, select *Set Geometry Bounds*, the function for defining the limits. To do so, shrink the section slightly until the software shows the boundaries of the area in the preview. Handles let you precisely set the desired section in the corners. In densely mapped areas, it is well worth precisely choosing the section you need; this saves disk space and computing time later.

Next, select *Download OSM data (Overpass)* to start the download. Depending on the location and size of the area, this can take a while. Load the generated files into the program via *File | Open map sources…* . To test the data for correctness and completeness, you can just apply the standard rules via

Switch to rules in the *Map* menu. To do so, disable the *Web map* below *Map sources* to see the results.

The next step is the most complicated: You need to create or modify the rules to make individual adjustments. In the *Map* menu, *Edit rendering rules* launches an editor with the currently active rules. Basically, you have two ways to proceed here. You can edit the pre-built rules, which can often be quite complex and difficult to understand. A tip for doing this: First, look at the files below `./rules/` in the Maperitive installation directory, instead of just editing the default rules. Alternatively, you can gradually build up your own rule set. If you need inspiration, you can look at some existing rules [4].

About Rules and Commands

Two terms always occur in Maperitive: rules and commands. Both affect the way maps are displayed but at different levels. Commands refer to the program functions and call them, whereas rules control the appearance of objects on the map, by controlling the interpretation of the data used [5].

Rules in particular define which objects from the data upload are displayed. The rule set is usually stored in a specially formatted file, with the `.mrules` (“Maperitive rules”) extension. To tell the program to use these rules for rendering, you can enable them using the `use-ruleset` and `apply-ruleset` commands in the Commander. This process takes a while – *Running a task* in the preview shows you the current status. You can see the effect of changed rules using `reload-ruleset` after you have rewritten the rules file.

A rule file consists of several parts. As a special feature, these files use indentation to distinguish different structures. The software works with three main kinds of objects:

- Point features (represented by OSM nodes) for items such as mountain peaks, viewpoints, but also city centers.
- Line features for objects such as OSM ways, that is, roads of all types, routes, rivers, and
- Area features, areas generated by closed OSM ways (buildings, squares, forests, or lakes).

Listing 1 shows the principle structure of a rule file on the basis of extracts from various MRULES files. You do not indent the `features` keyword, but you do indent the following keywords: `points`, `areas`, `lines`, and the other specific targets. Interestingly, several `points` areas can occur in a file, which improves the readability, as does inserting comments (“routes”).

Because of the variety of tags [6], it is not easy to find the correct tags for specific objects in an OSM database. One typical example is forests, sometimes designated as `natural=wood`, and sometimes also as `landuse=forest`. It's even harder with expanses of water: These can be found under `natural=water`, `waterway=riverbank`, `landuse=reservoir`, `landuse=basin`, `waterway=dock`, and many other tags. To find the typical tags for structures that are definitely in the maps,

LISTING 1: Structure of a Rule File

```
features
  points, areas
    hospital : amenity=hospital
    [...]
  areas
    building : building=yes
    water : natural=water OR waterway=riverbank OR landuse=reservoir OR
            landuse=basin OR waterway=dock
    [...]
  lines
    oneway : oneway=yes OR oneway=true OR oneway=-1
    river : waterway=river
    [...]
    gps tracks : gpstrack[]
[...]
// Routes
    route ncn : relation[type=route AND route=bicycle AND network=ncn]
    route foot : relation[type=route AND route=foot]
[...]
  points
    supermarket : shop=supermarket
[...]
```

LISTING 2: Hiking.mrules

```
  target: shelter
    define
      icon-image : icons/SJJB/png/accommodation_shelter2.n.32.png
      icon-width : 24
      min-zoom : 10
    draw : icon
```

LISTING 3: Define a Target

```
target: supermarket
  define
    icon-image : icons/shopping_supermarket.p.32.png
    min-zoom : 1
    font-size: 10
    icon-width : 16
  draw : icon
```

you need to check out sites such as the OSM overview [7] and a list of special objects [8].

Many objects appear as icons or text, as the following examples show. Because Maperitive evaluates the OSM database, the program can basically view all the objects that exist in the database. A multi-stage process controls the rendering of the objects. This is demonstrated by a typical problem when creating rules: the absence of certain entries in the maps.

If you want to add shelters of various types to a map, you need a target definition for these objects. They are maintained in the OSM database as `amenity=shelter`. You can copy the required definitions, for example, from `Hiking.mrules` (Listing 2). However, just adding this definition to your MRULES file in not enough; you also need to declare a corresponding feature:

```
shelter : amenity=shelter
```

If this entry is missing under `features` in the `points` area, or if it is not correctly indented, then the shelters will not appear in the map. The indentations are a frequent source of error in these definitions. Make sure that the `draw` statement appears at the same level as `define` and indent the additional information `icon-width`.

Unfortunately, the error messages give hardly any information about why something is not correctly displayed. The same applies for commented-out entries; these also must be at the appropriate level, if you want to avoid the Maperitive parser tripping over them.

There is in fact yet another level for rules, which you also need to note: You see precisely the subset of a target that you output in the target definition with `draw`. If you want to include the available supermarkets, this is done with a target like the one in Listing 3.

Supermarkets usually have a name, such as the name of the chain to which they belong, and it would be handy to see this in addition to the identical symbol for all supermarkets. You can achieve this through an entry in the form `draw : text` (same indentation depth as for `draw : icon`).

Maperitive so far only supports PNG graphics as icons. You must also make sure that the dimensions of the icons match the desired destination, for example, that arrows actually point to the right spot. If the application cannot display an icon, it is simply dropped without comment from the output.

Another special feature: The rules usually use fixed sizes regardless of the zoom level of the map to provide identical output of draw structures. When displaying birds eye views of the map, the software displays the icons and text at the same size as in views from a shorter distance.

There is only one way to modify the display as a function of the zoom level: To find the places where there is, for example, a supermarket at greater distances, you can enable the appropriate icons at low zoom levels. Do this with `min-zoom in the target definition: 5`.

To prevent the icon from looking too large and covering other important information, you need to modify the `icon-width`. Instead of `icon-width: 24`, reduce the icon width to, say, `icon-width: 5:8;10:12;13:24`. The program interprets this as: Up to

TABLE 2: SVG Export

Argument	Parameter	Meaning
`file=`	`<file>`	Define output file, default `Maperitive/output/output.svg`.
`zoom=`	`<zoom level>`	Define section for output. This parameter conflicts with `map-scale`.
`map-scale=`	`<scale>`	Define scale of output map. `map-scale=10000` generates a map with a scale of 1: 10,000.
`compatibility=`	`inkscape` or `illustrator`	Compatibility of the SVG data, default `inkscape`.
`compressed=`	`<file>`	Create compressed SVG file (`svgz`).
`scale=`	`<scaling>`	Scaling used, default 1.
`copy-images`	`true` or `false`	Copy images to the output file (default). `copy-images=false` includes references only.
`ai-autorescale=`	`true` or `false`	Special specification for Adobe Illustrator.
`precision-typo=`	`true` or `false`	Convert text to path to achieve optimum quality. Usually not necessary.

zoom level 5, use the 8pt icons; up to level 10, use the 12pt icons; and up to level 13, use the 24pt icons.

Exporting Maps

The maps produced with Maperitive so far exist only as patterns in the memory space reserved by the program. To use the maps elsewhere, you need to export them. There are two possibilities: Exporting as a bitmap (`export bitmap`) stores the current preview in the pixel format (by default as a PNG). This results in lower quality, but it is perfectly suitable except for tasks where you need map images – e.g., for the web maps of navigation systems.

You can also export as an SVG graphic (`export-svg`), which creates a scalable graphic whose components exist as editable objects. This variant type is useful for high-resolution maps and downstream processing. Maperitive provides numerous arguments for the SVG export (Table 2).

The order of the arguments is arbitrary, as long as you specify the arguments with values. In the case of arguments that normally expected values after the equals sign, you can specify the arguments without these if they come first. You would thus create a map with a scale of 1:10000 with the following command:

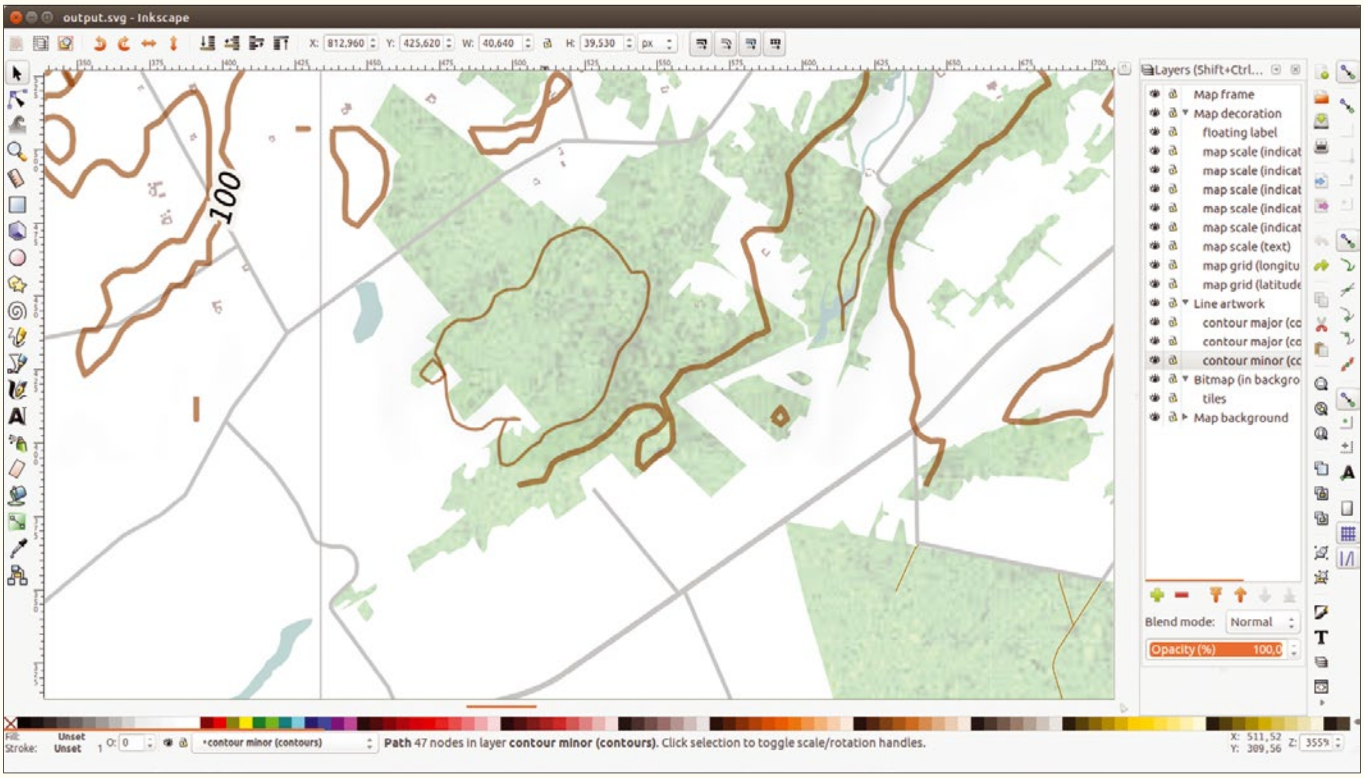
```
export-svg compressed compatibility=inkscape map-scale=10000 copy-images=false
```

Because SVG graphics consist of vectors, you can reproduce them at any resolution. You can thus create an overview map, as well as maps of sections in any quality without special adaptations. You can set the exported section of the loaded map in advance. To do so, define the desired area independently of the OSM data scope using *Maps | Set Print Bounds*.

The ability to export the current map as an OSM file in XML format is still relatively new. It offers the possibility to create multiple variants of the current map from the same data and to edit the variants again.

Contour Lines

Sometimes the OSM data is not sufficient to generate all the information you need. The routes and POIs in the maps are great, but to gain a realistic impression



Path 47 nodes in layer **contour minor (contours)**. Click selection to toggle scale/rotation handles.

Figure 3: **A small error in the contour lines has little effect on the overall result.**

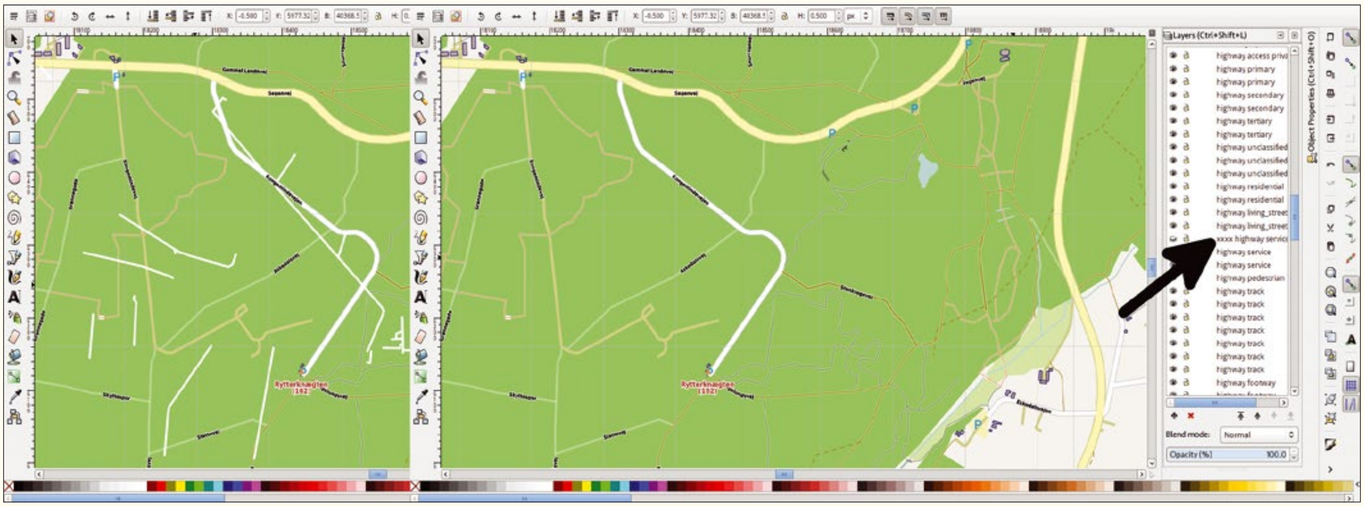


Figure 4: It's easy to remove unnecessary elements from the card. Just hide or delete the layer marked *xxxx*.

of a location, you definitely need to add contour lines at least. To do this, Maperitive uses a Digital Elevation Model (DEM) server from which the application queries the altitude data. When using the default DEM server SRTM3, you should mention the data source on any maps you create, however.

Maperitive's algorithm for calculating the contour lines does not work perfectly (Figure 3). Contour lines are equipotential lines that are always shown as closed. In Maperitive, the `generate-contours` command generates contour lines. The `<limits>`, `<interval>`, `<maximum>`, and `<minimum>` arguments let you adjust the contour lines to the requirements of the terrain.

Using `<borders>` lets you limit the edited area with a bounding box (the default is the preview pane). `<Interval>` defines the density of the contour lines, specified in meters (default: 10). `<Maximum>` defines the highest contour, and `Minimum` the lowest.

Contour labeling is done automatically, but the algorithm can quickly become overwhelmed in case of deviating values for the interval. There is room for improvement here. Contour lines appear in the SVG export as various levels with the *contour major* prefix in the *Line artwork* group.

Finishing

The cards created with Maperitive will look pretty good as is, depending on the database and the rules. However, you will likely need to rework the details of the exported data. This applies in particular to cards with large amounts of information at scales greater than 1:50000. In terms of this example, there turned out to be several incomplete regular trails declared as "Service Ways" scattered over the entire island (Figure 4).

To improve the map, it's now Inkscape's turn. You can either remove the planes with defective items or complete them by adding more elements. However, Inkscape is so slow when editing maps that it is little fun to edit larger images: Even on a quad-core machine with 8GB RAM, the computer swaps constantly. You will need to plan your changes well in advance.

Conclusions

Maperitive makes it relatively easy to create excellent maps based on OSM data. However, the procedure requires a huge amount of time and much experience. Maperitive helps convert the OSM data into a usable form and does most of the work for you. After that, however, significant manual work is required to produce really good results. ■■■

INFO

[1] Merkaartor: *http://wiki.openstreetmap.org/wiki/Merkaartor*

[2] Maperitive: *http://maperitive.net/download/Maperitive-latest.zip*

[3] OSM download: *http://geofabrik.de*

[4] Sample rules: *http://wiki.openstreetmap.org/wiki/Category:Maperitive/Rules*

[5] Building rules: *http://maperitive.net/docs/Rendering_Rules_Advanced.html*

[6] Information about tags: *https://taginfo.openstreetmap.org*

[7] Tags (overview): *http://wiki.openstreetmap.org/wiki/Map_Features*

[8] Tags for special objects: *http://wiki.openstreetmap.org/wiki/How_to_map_a*

Managing log files with logrotate

Log Rolling

Regularly switching log files keeps them from filling up the filesystem, but the logrotate tool can do much more.

By Bruce Byfield

BRUCE BYFIELD

Bruce Byfield is a computer journalist and a freelance writer and editor specializing in free and open source software. Bruce's most recent book, *Designing with LibreOffice*, was released under a Creative Commons License in March 2016. You can buy or download his book at *http://designingwithlibreoffice.com/download-buy/*. In addition to his writing projects, he also teaches live and e-learning courses. In his spare time, Bruce writes about Northwest Coast art. You can read more of his work at *http://brucebyfield.wordpress.com*.

Linux log files do not take up much space. Even with systemd, many logs continue to be text files. All the same, if left unattended, log files would eventually fill their entire filesystem – and, if that filesystem is root, crash the system. To avoid this eventuality, Linux uses *logrotate*, a system of multiple logs that are switched regularly, either at a specified time or when the file size reaches the designated limit.

Logrotate is installed by default in every distribution of which I am aware. You can usually take for granted that logrotate is installed, but you can check by running the command `ls /var/log`. If you see multiple files with numbered suffixes, then logrotate is installed. Alternatively, in some distributions, you can run `less /var/lib/logrotate.status` to see the latest logrotate activity.

In fact, logrotate is required by a Linux system, because many standard applications use logs by default, including apache, aptitude, cups-daemon, and dpkg. Ordinarily, such logs exist as much for the system as the administrator, but they can be invaluable for troubleshooting. You may want to modify the default setups for a log or rotate a log for your own scripts.

For example, on my system with a solid state drive, I have a script with a log for `fstrim`, the utility run weekly to discard unused blocks and to keep the drive running at top efficiency. By setting up the fstrim script for logrotate, I can keep its size from ballooning out of control without any further action on my part.

Logrotate works with several files:

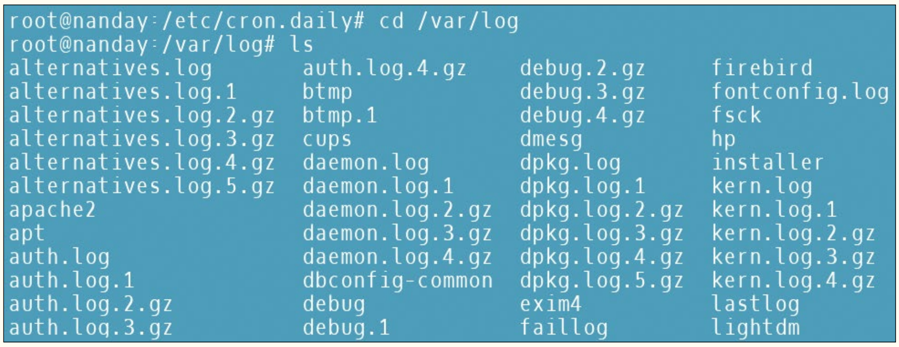
- `/etc/logrotate.conf`: The global configuration file.
- `/etc/logrotate.d`: A directory of logrotate settings for particular applications. Entries in this directory override the configuration in `/etc/logrotate.conf`.
- `/etc/cron.daily/logrotate`: The scheduler in which an entry for logrotate is entered so that all the applications with an entry in `/etc/logrotate.d` are processed. Usually, the scheduler is daily, but it can be hourly, monthly, or weekly instead.

The logs themselves are kept in `/var/log` (Figure 1).

Note that, unlike many commands, the `logrotate` command does not override the settings in `/etc/logrorate.conf` or `/etc/logrotate.d`. Instead, the command's options affect only how the command is run.

Setting Log Options

Values for log rotation can be set globally in `/etc/logrotate.conf` (Figure 2) or in a separate file in `/etc.logrotate.d` (Figure 3). Fields can be set in any order in the file, but if fields contradict each other, the one closest to the bottom of the field is used. Otherwise,



```
root@nanday:/etc/cron.daily# cd /var/log
root@nanday:/var/log# ls
alternatives.log        auth.log.4.gz    debug.2.gz    firebird
alternatives.log.1      btmp             debug.3.gz    fontconfig.log
alternatives.log.2.gz   btmp.1           debug.4.gz    fsck
alternatives.log.3.gz   cups             dmesg         hp
alternatives.log.4.gz   daemon.log       dpkg.log      installer
alternatives.log.5.gz   daemon.log.1     dpkg.log.1    kern.log
apache2                 daemon.log.2.gz  dpkg.log.2.gz kern.log.1
apt                     daemon.log.3.gz  dpkg.log.3.gz kern.log.2.gz
auth.log                daemon.log.4.gz  dpkg.log.4.gz kern.log.3.gz
auth.log.1              dbconfig-common  dpkg.log.5.gz kern.log.4.gz
auth.log.2.gz           debug            exim4         lastlog
auth.log.3.gz           debug.1          faillog       lightdm
```

Figure 1: Logrotate uses a series of logs to prevent the size of logs becoming large enough to destabilize the system.

fields can be in any order, with one per line, although the usual arrangement is to have global settings at the start of the file, and values that apply to specific logs contained within curly braces. For example, `/etc/logrotate.d/` defines one log this way:

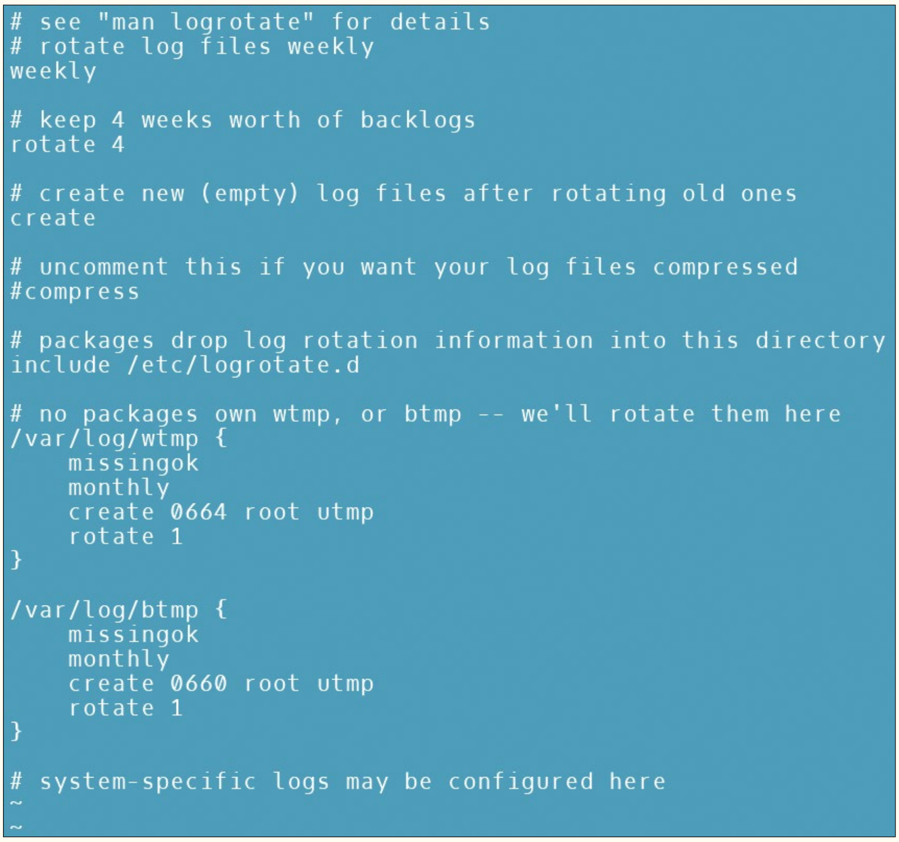
```
/var/log/apt/term.log {
  rotate 12
  monthly
  compress
  missingok
  notifempty
}
```

The same file, incidentally, defines another apt-related log in the same file. Most distributions further organize the settings by adding a comment above each field so that it is easily read. Field values can also be disabled by adding a hashtag (#) at the start of the line.

Regardless of where the log options are set, a couple of options must be set. Each set of values must have a regularity, such as monthly, weekly, or daily, corresponding to the cronjob in which it will run. Although not required, each set should also have a maximum number of rotations, which is set using `rotate NUMBER`.

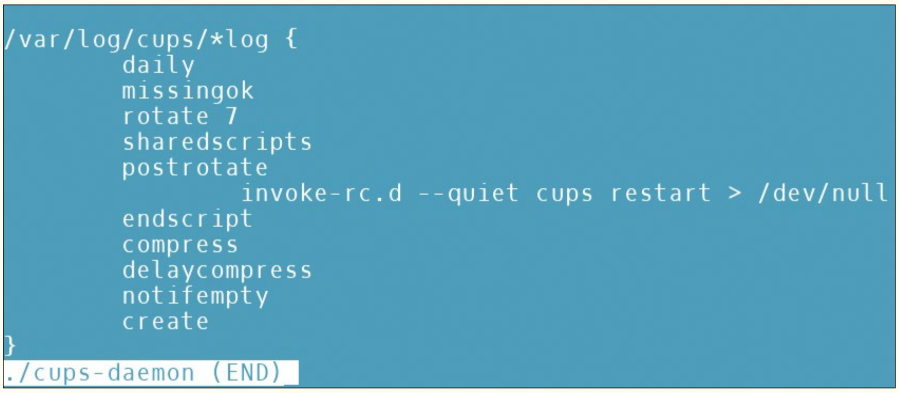
Alternatively, logs can be rotated whenever the current one reaches a current size with `size SIZE`. Other options are `maxsize SIZE` or `minsize SIZE`. The size can be set using `k`, `M`, or `G` (kilobytes, megabytes, gigabytes). Because logs are plain text, you can probably use kilobytes.

Regardless of which method is used to determine rotation, when a log is rotated, it is renamed by adding a number as a suffix, so that the current log becomes `log.1`, and the previous one `log.2`, and so on. When the maximum number of rotations is reached, the oldest log is deleted and replaced by the next oldest.



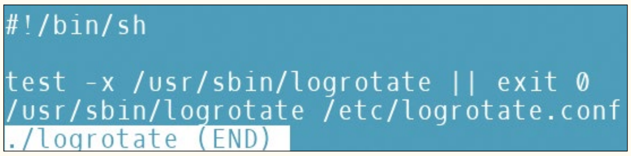
```
# see "man logrotate" for details
# rotate log files weekly
weekly

# keep 4 weeks worth of backlogs
rotate 4

# create new (empty) log files after rotating old ones
create

# uncomment this if you want your log files compressed
#compress

# packages drop log rotation information into this directory
include /etc/logrotate.d

# no packages own wtmp, or btmp -- we'll rotate them here
/var/log/wtmp {
    missingok
    monthly
    create 0664 root utmp
    rotate 1
}

/var/log/btmp {
    missingok
    monthly
    create 0660 root utmp
    rotate 1
}

# system-specific logs may be configured here
~
~
```

Figure 2: The `/etc/logrotate.conf` file contains the global options for logrotate.



```
/var/log/cups/*log {
        daily
        missingok
        rotate 7
        sharedscripts
        postrotate
                invoke-rc.d --quiet cups restart > /dev/null
        endscript
        compress
        delaycompress
        notifempty
        create
}
./cups-daemon (END)
```

Figure 3: The configuration file for the CUPS daemon from `/etc/logrotate.d/`.



```
#!/bin/sh

test -x /usr/sbin/logrotate || exit 0
/usr/sbin/logrotate /etc/logrotate.conf
./logrotate (END)
```

Figure 4: Once you add logrotate as a cronjob, you may not need to edit it again.

If rotate is set to 0, then the older log is deleted, instead of being temporarily stored. If you prefer, with `dateext`, older logs can have a date extension such as `YYYYMMDD`, with the exact date format set using `dateformat`. No matter what the naming method, there is no requirement about how many rotations to use, except that, the more important the information, the more rotations should generally be used.

Ordinarily, rotated logs are stored in the same directory as the current one – usually, `/var/log`. However, you also have the option of using `olddir Directory` to store a log rather than deleting it when its turn come. Another choice is to mail a log being rotated out of existence instead of renaming it, using `mail ADDRESS`. If you prefer, use `mailfirst` to mail the just-rotated file. If you want to be sure that no log is mailed, use `nomail` instead.

By default, old logs are compressed using `gzip`. However, you can store old logs without depression by using `nocompress`. You can also use another command for archiving, such as `tar` using `compresscmd`, which requires a matching `uncompresscmd` so that the logs can be properly accessed. With any compression method, `compressionoptions` passes on to logrotate the options to use during compression. For example, when using `gzip`, passing on the option `-6` will maximize compression at the price of speed.

Still other options include `missingok`, which ignores a missing log file and continues processing, and `shred` and `shredcycles NUMBER`, which overwrites deleted logs, making them unreadable. Further flexibility can be added to how a particular set of options is handled by adding scripts that begin with `postrotate`, `prerotate`, `firstaction`, or `lastaction` on a separate line, each of which ends with `endscript`.

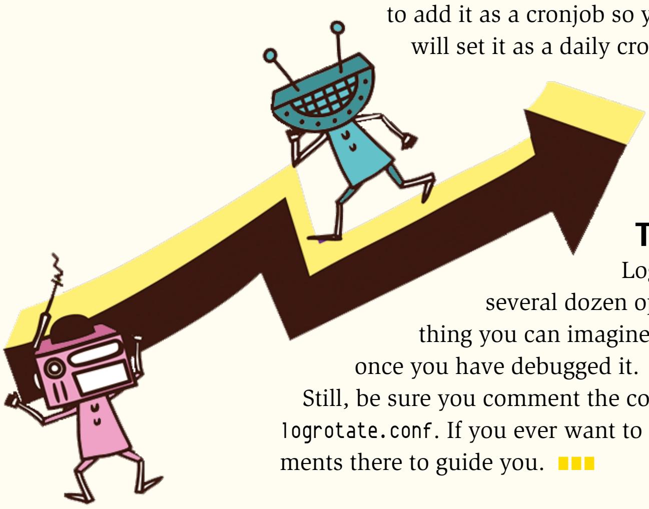
Choosing Command Options

When logrotate runs, it works with all the logs defined in `/etc/logrotate.conf` and `/etc/logrotate.d/`. The options for the command are about how it runs, not how individual sets of logs are handled. If you mail any logs when they are rotated, use `--mail (-mail) COMMAND` to change the email application from the default `/usr/bin/mail -s`.

When you add new configuration entries, or if you have removed files manually instead of using logrotate, you may need to use `--force (-f)`. In these situations or with any other troubleshooting, I suggest using `--verbose (-v)`, especially when testing, so that you can be sure that the command and the configuration files are interacting in the way that you hope.

Setting Up a Cronjob

Although you might test your command as you set up logrotate, its main purpose is to add it as a cronjob so you can forget about log rotation. Typically, you will set it as a daily cronjob, running a command to confirm the existence of log rotation, then running logrotate itself, as shown in Figure 4. This command can be easily modified if you want to run logrotate at another frequency.



Then Forget About It

Logrotate has a simple basic purpose. However, with several dozen options, it can be fine-tuned to do just about anything you can imagine – and, moreover, do so with no input from you once you have debugged it.

Still, be sure you comment the configuration files as you edit them, especially `/etc/logrotate.conf`. If you ever want to edit them, you will be grateful to have the comments there to guide you.

7:05 AM
iPad
ADMIN
Network & Security
Backup and replicate VMs on VMware ESXi
Clone Freedom
VMware ESXi virtualization is free, but it does not have a software tool to back up virtual servers. The free XSIBackup can help.
Setting Up the Source
Installation
Target Host and S
8
9
10
11
12
13
14
15

The thrilling saga of a database gone bad

True Tails

In which our hero relates a "True Tail" where lack of testing almost created a divorce.

By Jon "maddog" Hall

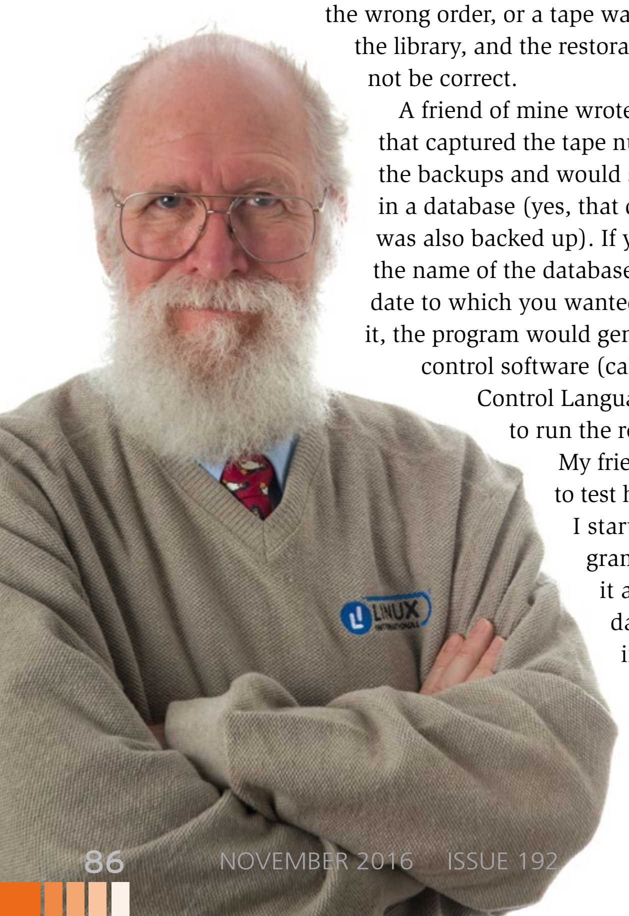
Recently, I wrote that programmers typically make some of the worst testers of their own software, based on the logic that if programmers knew enough about the bugs, they would write the program correctly in the first place. Of course, this is a generalization and led to lots of discussion about unit tests, regression tests, and many other types of tests used to validate code.

On the other hand, many people agreed with me, based on experience of sending the code out to end users and hearing a programmer scream, "I never thought someone would use my code that way." To illustrate this last point, I will relate one of maddog's True Tails™.

The year was 1975, and I was working for a very large insurance company. We had 500,000 12-inch, nine-track tapes in our on-site tape library, numbered from 000001 to 500xxx, and each of those tapes had an external, human-readable number with the same internal machine-readable number on the tape.

We had large databases that were backed up every night, and it took seven or eight of these tapes to back up the database. If something went wrong, you had to restore the database from those tapes, in the same order that they had been produced.

Sometimes, things did go wrong because the operators (being human) lost the record of which tapes had been used for which database, or put the tapes into the restore program in the wrong order, or a tape was missing in the library, and the restoration would not be correct.



A friend of mine wrote a program that captured the tape numbers from the backups and would store them in a database (yes, that database was also backed up). If you typed in the name of the database and the date to which you wanted to restore it, the program would generate the control software (called Job Control Language, or JCL) to run the restoration.

My friend asked me to test his program. I started his program, and when it asked for the date, I typed in *July 5*. The program crashed.

"You are not supposed to type in the date like that, you should type in *Year, Month, Day* in numbers," my friend said. I told him that there was no prompt, so I did not know.

He then changed the program to properly prompt for the date. I tried the program again, and this time when it asked for the date I typed in the word *Junk*. The program crashed again. My friend said I should have typed in what was asked. I pointed out that when the database is down the operators are under stress, and they might type in something incorrect. Thus, he should test to see if the input is a correct form and that it's a *probable* date (not later than today, not 300 years ago). He went away again.

He later came back proudly and asked me to test his program once more. This time, I just hit the Enter key without typing in anything. His program crashed again, due to "buffer underflow."

Two days later, he came back with a sneer on his lips. I tested his program again, and it did seem to do proper editing on the input, and it did generate the proper output, so he submitted the program to the pool of operational programs.

A few days later, I had to do some tricky work on the database, so I asked if he would be around in case I needed to restore the database and his program failed. He said that night was his first wedding anniversary and "please do not call me."

Later that night, I encountered problems with the process; the database was corrupted, and I tried to use the program to restore it, but the program failed. It was 2 o'clock in the morning.

I called my friend and heard a small whisper, "Jon, it is my wedding anniversary, do you know what that means?" I said, "It means you will not have a job in the morning if you do not get in here and fix this problem."

He hung up, and I looked at the program again. It did not do proper editing on the tape numbers of the restore tapes. One tape had only five digits, not six, in its internal number, which caused a blank in the JCL (which was not allowed). I punched some cards with the corrected JCL and restored the database. At 3:30am, my friend showed up, still in his robe and slippers.

"What are you doing here?" I asked.

Proper testing may be a life saver (or a marriage saver). ■■■

THE AUTHOR

Jon "maddog" Hall is an author, educator, computer scientist, and free software pioneer who has been a passionate advocate for Linux since 1994 when he first met Linus Torvalds and facilitated the port of Linux to a 64-bit system. He serves as president of Linux International®.

Attention All Drupal Webmasters and Developers!
Drupal Watchdog
DATA & ANALYTICS
Google Analytics for Drupal admins
Creating Simple Online Forms
Data Migration with the Migrate API
Drupal Modules for better site security
Dissecting 1 line of PHP code
Choosing and Pairing Fonts
Canvas Fingerprinting Learn about a weird new ad tracking trick
WWW.DRUPALWATCHDOG.COM
COMMUNITY

Highlights of LinuxCon NA 2016

The State of the Penguin

The Linux community gathered at LinuxCon NA 2016 in Toronto to celebrate the 25th anniversary of Linux.

By Swapnil Bhartiya

LinuxCon North America – one of the biggest Linux and open source events – took place in Toronto, Canada, August 22-24, 2016. What made this LinuxCon important was the fact that it fell during the same week that Linux celebrated its 25th anniversary.

Keynote Highlights

The first day of the event began with a welcome keynote by the executive director of the Linux Foundation, Jim Zemlin. "We made it. 25 years. Quite an accomplishment," said Zemlin as he took the stage.

Zemlin reflected on the 25 years of Linux and said how Linux has changed the way we collaborate. "Today, you literally cannot make anything by building all the software you need by yourself." And, today, the vast majority of code in any modern technology product or service is open source.

The success of Linux has proved that "you can better yourself by bettering others at the same time," said Zemlin.

Dr. Ainissa Ramirez, science evangelist and author, spoke about the important role technology plays in shaping and changing our society. As an example, she described how before the invention of the telegraph, people wrote with exquisite detail in very long, elaborate sentences. After the telegraph, they had a more terse, sparse style. In a nutshell, new technology can have a deep, unintentional impact. Technology has become an extension of our brain, she said.

She made a call to the open source developers asking them to consider the result of different devices and inventions.

"You already have the right mindset for how to look at technology," she said. "You share it. You're open. You're inclusive. You know that many, many minds make things better than one mind."

Red Hat CEO Jim Whitehurst pointed out the close relationship between Linux and Red Hat. He said "…as I started to reflect on the history of Red Hat, it's really hard to talk about the history of Red Hat without talking about Linux. It's hard to talk about Linux without talking about Red Hat because the two, for so long, so closely have worked together and continue to work well together."

It was an event of historical magnitude as Wim Coekaerts, Corporate Vice President of Enterprise Open Source, Microsoft, delivered his keynote. Before joining Microsoft, Coekaerts was senior Vice President, Linux and Virtualization

Images courtesy of Swapnil Bhartiya

Engineering at Oracle and was known as Mr. Linux. It was interesting to see a hard-core Linux user talking about leading open source at Microsoft. In the talk, he described how Microsoft is contributing to open source and how its creating a level playing field for Windows and Linux.

Author and copyright activist Cory Doctorow delivered a popular talk about digital rights management (DRM) and the importance of licenses for the success of open source.

Brian Behlendorf, executive director of the HyperLedger project, talked about the importance of blockchain in the modern economy. Donna Dillenberger, fellow, Watson Research Center at IBM, explained how blockchain is being used by different industries and gave an impressive demo showing how IBM is using blockchain for its own supply chain transactions.

Conversation with Linus Torvalds

One highlight of every LinuxCon is the conversation between Linus Torvalds and Dirk Hohndel, a friend of Torvalds and now VP and chief of open source at VMware. During the conversation, Torvalds talked about a few times when he almost gave up on the kernel. He admitted that the arguments on the mailing list would get heated, and he would get so angry that he would consider giving up. "I decided I would be offline for a week because I [couldn't] take this anymore. Usually, the next day I'm back and things are better," said Torvalds. He also added that all of this occurred a long time ago, not within the last 10 years.

At the same time, Torvalds praised the Linux community. "We have thousands of people involved, and it's working great," he said. "We still argue. We're not all happy people. We don't love each other." But, he added, "there's a lot of respect for the technical side and things are working very well, in ways things have not always worked."

"You can just feel the fact that people want to work together and want to make a better system. That really makes for a great community," he said.

Hohndel asked what might happen to Linux if Torvalds weren't there. Torvalds said that this would have been a bigger problem some 15 years ago, but now there are many people who could take up the work.

"Most of the people who could do it, wouldn't want to do it, I think," said Torvalds. "There are people who have been around almost as long as I have been. There are people who are universally trusted and do a lot of the work today that could step up. My traditional answer is, hey, I will be there."

Women in Open Source Networking Event

Another notable event at the conference was the women's networking luncheon. It was exclusively for women, so I couldn't go, but my wife, who works for the U.S. Department of Defense and is a member of organizations like Lean In, attended. She found that it's a great opportunity for women in tech to connect with each other. Even more interesting was the format of the event. Instead of having an individual speaker or panel discussion, all the women in the room had the opportunity to introduce themselves and describe their role in the field. This approach turned all the women into speakers and listeners, and like-minded women swiftly connected with each other.

Another highlight of the event was Kids Day, a workshop for children. It was organized in partnership with MakerKids and Kids on Computers. I met with a few kids and their parents, and I learned that most kids were not even aware of the fact that they were using Linux on their parent's computers. Some kids, (not all), were excited to spend time on computer with other kids.

Childcare

This year the Linux Foundation set up a childcare program, where they hired caregivers who played with and cared for kids in one of the conference rooms within the venue. I brought my kids with me – one is 4 years old and the other 11 months – and it was a great way for the kids to play in a safe and secure environment while parents participated in the conference.

Adios, LinuxCon – Hello, Open Source Summit

On the last day of the conference, Jim Zemlin announced that the name of the event is changing from LinuxCon to Open Source Summit (OSS). This broader scope will allow the Foundation to bring more events under the same brand without any conflict. The next LinuxCon… er, Open Source Summit will be in Los Angeles in 2017, and I am already looking forward to it. ■■■

Zack's Kernel News

Chronicler Zack Brown reports on the latest news, views, dilemmas, and developments within the Linux kernel community.

By Zack Brown

Intelligent Networking Policies

Kan Liang wanted to simplify Linux networking configuration to get better performance on a variety of workloads. The problem, he said, was that the default configuration options didn't really work well, and tweaking them properly took a lot of skill.

His approach was to create a set of policies that would provide hints to the kernel that would then be used to tweak the more fine-grained networking controls automatically. Kan's idea would involve per-socket, per-task, and per-device policies that the kernel would interpret to give near optimal performance.

Kan specified that his approach would not be an attempt to optimize networking performance fully, but that it would hopefully bring a system to 90 percent networking efficiency.

One question he anticipated was whether any concept of networking policies belonged in the kernel or in user space. He felt the kernel was the right place for his code, because it would need to handle requests from multiple users and could do the job more simply and efficiently from within the kernel.

He also pointed out that, as much as possible, the net policy code would rely on existing kernel infrastructure, rather than coding the entire thing separately; for example, it would interact with networking hardware using existing interfaces.

Stephen Hemminger replied, saying that he agreed on all points except the need to do this in kernel space. His argument was, whatever *could* go outside the kernel *should* go outside the kernel. Alexei Starovoitov also thought that Kan's code didn't belong in the kernel for the same reason.

Kan stuck to his point that the kernel implementation would be much simpler and easier, but there was no further discussion on the mailing list.

The problem as I see it is that there are lots of things that would be easier to do in the kernel. Kernel code is just inherently less encumbered than user code. If everything that was easier to do in the kernel were actually implemented in the kernel, Linux would be far more monstrously huge than it is now. It's only by insisting that absolutely everything go into user space, if at all possible, that Linux is able to remain only as massively huge as it is now, and no huger.

Of course, microkernel people make the argument that a lot more could be stripped out of the kernel and put into user space, and that's true. The Linux philosophy, however, isn't simply to exclude everything from the kernel that it possibly can. There's also the problem of speed. It's possible that if Kan's code would run significantly faster in kernel space than in user space, then kernel people might be willing to accept the code. But if it's just a question of maintainer convenience, that's not enough to justify accepting code into the tree.

Securing Memory Locations

As a security measure, William C. Roberts wanted to randomize the locations of memory allocations in the kernel, so they couldn't be predicted by hostile code. He submitted some code to do this, but Nick Kralevich objected. As he understood it, William's code "adds a random gap between various `mmap()` mappings, with the goal of ensuring that both the mmap base address and gaps between pages are randomized."

He pointed out that Android systems had experienced problems with that kind of memory fragmentation in the past. He said, "After a program runs for a long time, the ability to find large contiguous blocks of memory becomes impossible, and mmap()s fail due to lack of a large enough address space."

Nick gave links to the various patches that had previously been needed to undo the fragmentation that he felt William's code now was trying to reintroduce. He said, "If this behavior was re-introduced, it's likely to cause hard-to-reproduce problems, and I suspect Android based distributions would tend to disable this feature either globally, or for applications which make a large number of mmap() calls." Jason Cooper agreed that Nick had identified the key problem with this type of feature and urged William to address the fragmentation concerns before putting much more work into the code.

William agreed that fragmentation was definitely a problem and that one of the goals of his code would be to implement the memory randomization without the added burden of memory fragmentation.

ZACK BROWN

The Linux kernel mailing list comprises the core of Linux development activities. Traffic volumes are immense, often reaching 10,000 messages in a week, and keeping up to date with the entire scope of development is a virtually impossible task for one person. One of the few brave souls to take on this task is **Zack Brown**.

As a stopgap, Dave Hansen suggested simply disabling William's randomization feature on all 32-bit systems. As he put it, "All of the Android problems seemed to originate with having a constrained 32-bit address space." Pavel Machek agreed that 32-bit systems would be hard hit by William's code. Meanwhile Nick also said, "I like Dave Hansen's suggestion that this functionality be limited to 64 bits, where concerns about running out of address space are essentially nil. I'd be supportive of this change if it was limited to 64 bits." Jason also agreed that a 64-bit-only approach would resolve all of his objections, and William said he liked this idea for his code as well.

It's very rare when a security debate identifies a problem and a solution in the same breath. This seems to be one of those instances.

Early Access to Firmware

Luis R. Rodriguez posted some patches to continue the ongoing effort to end support for device firmware availability at initialization, partly motivated by a desire to simplify the incredibly variable bootup procedure in Linux. Once these changes go through, firmware will only be accessible from a mounted filesystem (i.e., a late stage of the bootup procedure). As of Luis' post, he said that only the Dell RBU driver still offered early access to firmware.

As Luis put it, "Thou shalt not make firmware calls early on init or probe."

For subsystems or devices that need access to firmware as early as possible, Luis suggested either including the firmware directly in the kernel binary or within the initramfs image used to boot the system.

Julia Lawall and Josh Boyer both pointed out areas of the kernel beyond the Dell RBU driver that still needed early access to firmware, and Luis replied that this would make things a bit more difficult. He said, "it seems we may want to allow for these type of calls within probe in the end but in order to vet for drivers that fw is available through the direct filesystem lookup we may need help from userspace."

Luis added that some of those instances made it all the more urgent to deprecate and remove the user mode helper code that allowed early access to firmware.

At the same time, Daniel Vetter pointed out that all GPU drivers that depend on firmware would still need early access, because, as he put it, "people are generally pissed when they boot their machine and the screen goes black." He tentatively suggested "loading the different subsystems of the driver in parallel (we already do that largely), and then if one firmware blob isn't there yet, simply stall that async worker until it shows up" – although he did acknowledge that some kernel folks had already told him not to do that.

Luis replied that he wanted to know more about Daniel's needs in this area, since one of the goals of his cleanup was to keep the kernel actually functioning and not break anyone's code too badly.

The discussion grew more technical, but it began to be clear that there were various use cases requiring early access to firmware. In particular, embedded systems did not want to have to include an initramfs image, because every byte would add to the footprint of their products.

There was no ultimate solution during the discussion on the mailing list. At various points, several people suggested conversing in person at the Kernel Summit or the Linux Plumbers conference.

This may be one of those things like the Big Kernel Lock, that everyone wants to get rid of, but that many parts of the kernel depend on. The reliance on early availability of firmware may have to be

pruned out gradually over a course of years, rather than all at once by a single patch.

Driver Dependencies

Rafael J. Wysocki wanted to create a new set of dependencies associated with the Linux kernel: specifically, the dependency that one device and its associated driver might have for another. His idea was that a device and its driver would register a dependency on another device and driver, thus linking them together logically. This way, an attempt by the user to unload a driver that was needed by another device could be forestalled by the system. The driver in question could only be unloaded after all other drivers released their links to it.

Tomeu Vizoso thought this would be a great idea, as long as the dependencies could be figured out properly. In particular, Rafael wanted to distinguish between two types of dependency links between drivers: those that were created when the device was registered with the system and those that only came into being when the device probed the system to see what other hardware was available. Tomeu felt there was a strong danger of code duplication in the latter case between the probing code and the linking code. Also, he felt that if the kernel couldn't be refactored to avoid that duplication, it would be a tough problem to keep both sets of code in sync with each other.

Rafael replied, "That almost certainly is going to be a fair amount of work, but that doesn't mean we should avoid doing it. If it leads to better code eventually, it's worth doing."

Tomeu was also concerned that Rafael wanted to make things too complicated by having drivers construct dependency links before probing the system based on whatever information was already available. He thought it would be easier simply to do the probe. Rafael replied that it would be easier to understand the order of operations if the system used information that was already available, although he acknowledged that if it proved very difficult to collect that information, it wouldn't be worth it.

Meanwhile, Linus Walleij said he liked Rafael's proposal very much, although he also pointed out that circular dependencies could become a problem.

Greg Kroah-Hartman also liked Rafael's plan and thought that Rafael had also successfully identified the implementation requirements.

Thierry Reding was also very supportive. He said, "I think this would be a huge step towards making the kernel more robust with little driver or subsystem code having to be duplicated. Currently most provider/consumer subsystems are fragile in that there isn't proper reference counting. Many subsystems will happily allow you to remove any of the provider, regardless of whether or not it has consumers. Most of the subsystems will make sure that modules can't be unloaded, but beyond that won't be able to prevent drivers from being unbound (either when a device is unplugged or unbound via sysfs). Even with proper reference counting there is no easy way to deal with devices going away (you'd need some sort of revoke semantics implemented for all providers, and consumers must be able to handle that situation gracefully)."

Thierry added, "I can also imagine subsystems where a reliable mechanism is in place to properly hotplug and -unplug providers. The good thing about the functional dependencies mechanism you propose here is that it's an optional mechanism that drivers use from `->probe()`. Subsystems where a better mechanism exists can simply choose to do without functional dependencies."

Elsewhere, Andrzej Hajda pointed out an issue he had with Rafael's idea: He felt that certain drivers should be able to have "weak" dependencies, where the driver could still function in a limited capacity even if some of its dependencies weren't met, although later he acknowledged that, "It should be enough if provider notifies consumers that the resource will be unavailable."

There were various technical questions and comments, mostly having to do with exactly how and when drivers might communicate their dependencies to each other. Eventually Rafael posted some actual code to implement what he was talking about, and the conversation petered out.

It seems as though, in general, nobody objects to drivers having these sorts of dynamic and enforced dependencies. The issues seem to be code duplication and the need to keep specific behaviors sane and usable. It seems as though once the code gets a bit further along, it will sail into the main source tree.

Speeding Up the Balloons

Liang Li wanted to improve the speed at which the VirtIO balloon driver could inflate and deflate. A balloon driver is one technique of allowing a virtual Linux system to gain access to more memory. To the virtual system, the driver gives the appearance of using a lot of memory, while in fact the memory isn't used at all and is simply available to the host to use for other processes. When the virtual system needs access to more memory, the balloon driver seems to "free up" some of the memory it has allocated. In fact, the newly available memory is being allocated by the host. It's all just a blue pill to let the virtual system think it's not plugged into a Matrix feeding pod.

However, the act of maintaining that illusion is slow, at least according to Liang. To inflate the balloon to 3G on a virtual system that believed itself to have 4G available took about 1.5 seconds on Liang's setup. He posted a patch to reduce this to a ninth of the speed, down to 175 milliseconds. The key, he said, was to avoid passing heavyweight page frame numbers around and to use a sleek bitmap instead.

Cornelia Huck had some technical suggestions, and Michael S. Tsirkin felt that Liang's implementation had some race conditions that needed to be ironed out. After some back and forth, Liang posted another patch that addressed Michael's races, but at a speed cost. The new code took 338 milliseconds instead of 175. This didn't seem like a necessary cost to Michael, and the two of them went back and forth on it. Michael also submitted his own versions of the patch to illustrate some of his implementation ideas.

At some point the discussion petered out. Clearly, though, even at 338 milliseconds, Liang's patch is a big improvement over the existing code. With modern corporations running vast farms of virtualized systems, this kind of speedup makes a big difference. ■■■

FEATURED EVENTS

Users, developers, and vendors meet at Linux events around the world. We at *Linux Magazine* are proud to sponsor the Featured Events shown here.

For other events near you, check our extensive events calendar online at *http://linux-magazine.com/events.*

If you know of another Linux event you would like us to add to our calendar, please send a message with all the details to *events@linux-magazine.com*.

LinuxCon Europe

Date: October 4–6, 2016

Location: Berlin, Germany

Website: *http://events.linuxfoundation.org/events/linuxcon-europe*

Learn from the leading maintainers, developers, and project leads in the Linux community from around the world. Choose from among 100+ sessions, ranging from tutorials to deep technical dives and everything in between.

OpenStack Summit

Date: October 25–28, 2016

Location: Barcelona, Spain

Website: *https://www.openstack.org/summit/*

Welcome to users, developers, and administrators of OpenStack cloud software. This five-day conference includes keynotes, breakout sessions, and working sessions. Learn about new products in the ecosystem and participate in hands-on workshops.

Node.js Interactive

Date: November 29–December 2, 2016

Location: Austin, Texas

Website: *http://events.linuxfoundation.org/events/node-interactive*

Node.js Interactive is the marquee event for JavaScript developers, companies that rely on Node.js, and vendors that support them with tools, training, and other services. Join the fun with Breakout, Code & Learn, and Un-conference sessions, as well as the usual booth crawl.

EVENTS

LinuxCon Europe	October 4–6	Berlin, Germany	http://events.linuxfoundation.org/events/linuxcon-europe
ContainerCon Europe	October 4–6	Berlin, Germany	http://events.linuxfoundation.org/events/containercon-europe
Cornell Drupal Camp	October 6–7	Ithica, New York	http://camp2016.drupal.cornell.edu/
Icinga Camp San Diego 2016	October 18	San Diego, California	https://www.icinga.org/community/events/icinga-camp-san-diego/
Linux Presentation Day	October 22	Berlin, Germany	http://www.linux-presentation-day.org/
EclipseCon Europe	October 25–27	Ludwigsburg, Germany	https://www.eclipsecon.org/europe2016/
OpenStack Summit 2016	October 25–28	Barcelona, Spain	https://www.openstack.org/summit/
OpenPOWER Summit Europe	October 26–28	Barcelona, Spain	http://openpowerfoundation.org/openpower-summit-europe/
Web Summit	November 7–10	Lisbon, Portugal	https://websummit.net
ApacheCon Europe	November 16–18	Seville, Spain	http://events.linuxfoundation.org/events/apachecon-europe
UbuCon Europe	November 18–20	Essen, Germany	http://ubucon.org/de/events/ubucon-europe/
LinuxDay in Vorarlberg	November 26	Dornbirn, Austria	http://www.linuxday.at
AnDevCon	November 29 December 1	Austin, Texas	http://www.andevcon.com–
Node.js Interactive North America	November 29–December 2	Austin, Texas	http://events.linuxfoundation.org/events/node-interactive
Open Source Monitoring Conference	November 29–December 2	Nürnberg, Germany	https://www.netways.de/events_schulungen/osmc/uebersicht/
SPTechCon	December 5–8	San Francisco, California	http://www.sptechcon.com/venue

Images © Alex White, 123RF.com

CALL FOR PAPERS

We are always looking for good articles on Linux and the tools of the Linux environment. Although we will consider any topic, the following themes are of special interest:

- System administration
- Useful tips and tools
- Security, both news and techniques
- Product reviews, especially from real-world experience
- Community news and projects

If you have an idea, send a proposal with an outline, an estimate of the length, a description of your background, and contact information to *edit@linux-magazine.com*.

The technical level of the article should be consistent with what you normally read in *Linux Magazine*. Remember that *Linux Magazine* is read in many countries, and your article may be translated into one of our sister publications. Therefore, it is best to avoid using slang and idioms that might not be understood by all readers.

Be careful when referring to dates or events in the future. Many weeks could pass between your manuscript submission and the final copy reaching the reader's hands. When submitting proposals or manuscripts, please use a subject line in your email message that helps us identify your message as an article proposal. Screenshots and other supporting materials are always welcome.

Additional information is available at:
http://www.linux-magazine.com/contact/write_for_us.

AUTHORS

Erik Bärwaldt	34
Swapnil Bhartiya	8, 12, 88
Zack Brown	92
Bruce Byfield	82
Joe Casad	3
Karsten Günther	74
Jon "maddog" Hall	86
Frank Hofmann	68
Kristian Kißling	20
Klaus Knopper	50
Peter Kreußel	44
Charly Kühnast	60
Dmitri Popov	62
Mike Schilli	30, 52
Harald Zisler	56

CONTACT INFO

Editor in Chief
Joe Casad, jcasad@linux-magazine.com

Managing Editor
Rita L Sooby, rsooby@linux-magazine.com

Localization & Translation
Ian Travis

News Editor
Swapnil Bhartiya

Copy Editors
Amber Ankerholz, Amy Pettle

Layout
Dena Friesen, Lori White

Cover Design
Dena Friesen

Cover Image
CC BY-SA 4.0

Advertising – North America
Ann Jesse, ajesse@linuxnewmedia.com
phone +1 785 841 8834

Advertising – Europe
Brian Osborn, bosborn@linuxnewmedia.com
phone +49 8999 341148

Publisher
Brian Osborn, bosborn@linuxnewmedia.com

Marketing Communications
Gwen Clark, clark@linuxnewmedia.com
Linux New Media USA, LLC
616 Kentucky St.
Lawrence, KS 66044 USA

Customer Service / Subscription
For USA and Canada:
Email: cs@linuxpromagazine.com
Phone: 1-866-247-2802
(Toll Free from the US and Canada)
Fax: 1-785-856-3084

For all other countries:
Email: subs@linux-magazine.com
Phone: +49 89 99 34 1167
Fax: +49 89 99 34 1198

www.linuxpromagazine.com – North America

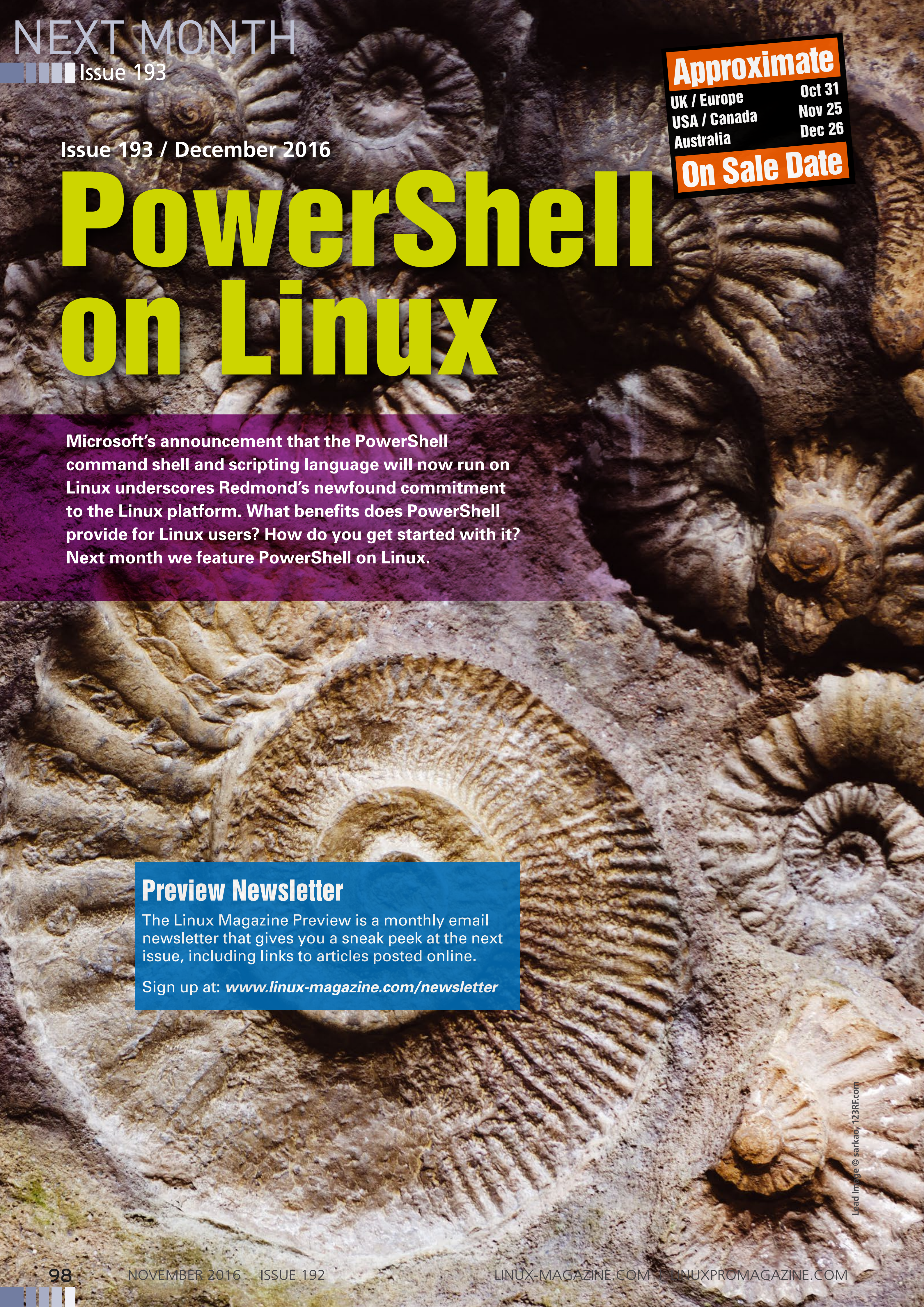
www.linux-magazine.com – Worldwide

While every care has been taken in the content of the magazine, the publishers cannot be held responsible for the accuracy of the information contained within it or any consequences arising from the use of it. The use of the disc provided with the magazine or any material provided on it is at your own risk.

Printed in Germany

Distributed by COMAG Specialist, Tavistock Road, West Drayton, Middlesex, UB7 7QE, United Kingdom

Published in Europe by: Sparkhaus Media GmbH, Putzbrunner Str. 71, 81749 Munich, Germany.



Issue 193 / December 2016

PowerShell on Linux

Approximate

UK / Europe	Oct 31
USA / Canada	Nov 25
Australia	Dec 26

On Sale Date

Microsoft's announcement that the PowerShell command shell and scripting language will now run on Linux underscores Redmond's newfound commitment to the Linux platform. What benefits does PowerShell provide for Linux users? How do you get started with it? Next month we feature PowerShell on Linux.

Preview Newsletter

The Linux Magazine Preview is a monthly email newsletter that gives you a sneak peek at the next issue, including links to articles posted online.

Sign up at: ***www.linux-magazine.com/newsletter***

Lead Image © sarkao, 123RF.com

Ubuntu User ever in one Massive Archive!
Celebrate 25 years of Linux with every article published in Ubuntu User on one DVD
UBUNTU
user
EXPLORING THE WORLD OF UBUNTU
SET UP YOUR VERY OWN ONLINE STORAGE
YOUR CLOUD
Choose between the best cloud software
Access your home cloud from the Internet
Configure secure and encrypted connections
Set up synchronized and shared folders
Add plugins for more features
PLUS
Learn all about Snap and Flatpak, the new self-contained package systems
Professional photo-editing with GIMP: masks and repairs
Play spectacular 3D games using Valve's Steam
Discover Dasher, the accessible hands-free keyboard
DISCOVERY GUIDE
New to Ubuntu? Check out our special section for first-time users! p. 83
How to install Ubuntu 16.04
Get all your multimedia working
Go online with NetworkManager
Package management
FALL 2016
WWW.UBUNTU-USER.COM
Pretty
THE COMPLETE
UBUNTU
user
ARCHIVE
DVD ROM
While this Ubuntu User disc has been tested and is to the best of our knowledge free of malicious software and defects, Ubuntu User cannot be held responsible, and is not liable for any disruption, loss, or damage to data and computer systems related to the use of this disc.
Searchable DVD
Content Available in Both HTML and PDF Formats
Over 3,000 PAGES! 7 GREAT YEARS OF UBUNTU USER

Capacity Optimized Software Defined Storage Solutions

60/90-Bay 4U Storage Servers and Enclosures with up to 9 Petabytes per 42U Rack

SC946SE2C/1C-R1K66JBOD *(JBOD Enclosure)*
SSG-6048R-E1CR60N/L *(Server)*

SC946ED-R2KJBOD *(JBOD Enclosure)*
SSG-6048R-E1CR90L *(Server)*

Intel® Xeon® Processor E5-2600 v4/v3 product families supported (except JBODs)

Over 50 Software Defined Storage Optimized Products to Choose From

Maximum Efficiency
High capacity 1U-4U form factors. Leading the industry with up to 50% greater density. Highest rated - 95% efficient Platinum power supplies

Maximum Performance and Expandability
All NVMe support with storage servers and direct attach storage platforms delivering up to 20 GB/s throughput

Mission Critical Reliability
Capable of fully redundant and fault-tolerant operation with hot swappable drive bays, power supplies, fans and redundant server boards with remote management.

2U 12 Bay 3.5"

1U 12 Bay 3.5"

4U 24 Bay 3.5"

2U 24 NVMe

2U 48 NVMe

4U 90 Bay

Learn more at www.supermicro.com/storage

Intel Inside®. Powerful Productivity Outside.