

FREE DVD

ubuntu
16.10 Desktop (32-bit)

ubuntuMATE
16.10 Desktop (64-bit)

DIASPORA
Social networking
the FOSS way

NOW FEATURING
LINUXVOICE

LINUX
MAGAZINE

LINUX
MAGAZINE

JANUARY 2017

Double-Sided DVD
INSIDE!

DIASPORA

Social networking without Big Brother or Big Data

Z-Wave

Automate your home with this Linux-ready IoT platform

Lepton

Save disk space with this innovative image compressor

Ardour 5

The free digital audio workstation plays a new melody

5 Formidable screencast tools

Living with Windows

- ChaletOS & Zorin OS pg. 24
- Run Windows apps with Wine pg. 92



Brian Behlendorf
Apache founder on the effort to forge a better blockchain

LINUXVOICE

- KDE thru the ages
- Automotive Linux
- Filesystem monitoring

WELCOME
LINUXVOICE
READERS!
p65



FOSSPicks

- GRUB Customizer
- Attract Mode

Tutorials

- Meticulous monitoring with collectd
- Windows apps in Wine

Issue 194
Jan 2017
US\$ 15.99
CAN\$ 17.99



HETZNER
ONLINE

More Power.

More Performance.

DELLEMC



Dedicated Root Server DX151

Dell PowerEdge™ R730 13th Generation
Single Intel® Xeon®
E5-2600 v3 (2.10 GHz) Octa-Core Haswell Processor
64 GB DDR4 ECC RAM, max. 192 GB at additional cost
Max. 8 hard drives at additional cost
50 TB traffic inclusive*
No minimum contract
Setup fee \$144.50

monthly from \$ **144**.50

Dedicated Root Server DX291

Dell PowerEdge™ R730 13th Generation
Dual Intel® Xeon®
E5-2600 v3 (2.10 GHz) Octa-Core Haswell Processor
128 GB DDR4 ECC RAM, max. 384 GB at additional cost
Max. 8 hard drives at additional cost
50 TB traffic inclusive*
No minimum contract
Setup fee \$211.50

monthly from \$ **211**.50

www.hetzner.de/us

* There are no charges for overage. We will permanently restrict the connection speed if more than 50 TB/month are used. Optionally, the limit can be permanently cancelled by committing to pay \$1.31 per additional TB used.

All prices include VAT and are subject to the terms and conditions of Hetzner Online GmbH. Prices are subject to change. All rights reserved by the respective manufacturers. Intel, Intel Logo, Intel Xeon and Xeon Inside are brands of the Intel Corporation in the USA or other countries.

VOTE TECH

Dear Reader,

The nature of the print publishing industry demands that I write this column some time before you read it. The first copies go on sale two weeks after our deadline, and, depending on where you live in the world, you could be seeing this issue one month or even two months after these words reach layout. Print publishing lives on because it has many admirable qualities, but low latency is not one of those benefits. This introduction is my graceful way of apologizing that what I'm thinking about now is probably not what you're thinking about when you read this. I'm thinking about the election in the US, which is happening the very day I write this column. You already know who won, and you are happily free from having to think about it, but maybe you should.

A few days ago, the security vendor Cylance released a video showing how you can hack a Sequoia AVC Edge voting machine using a PCMCIA card [1]. You can change the votes (and even the names of the candidates) recorded on the machine. The Sequoia AVC Edge is used in assorted precincts in 13 states across the US. The community of activists who follow the electronic voting industry were not impressed with this demo, stating that this hack of the AVC Edge, which will also work on several other touchscreen-style voting machines, has been known for years.

At least the PCMCIA card hack requires physical access. Another voting machine known as the AVS WINVote was in the news a year ago when it was revealed that the WINVote system featured an option for wireless system administration using weak, hard-coded passwords for the admin account and relying on long-discredited WEP encryption. These WINVote systems use a version of Windows XP that hasn't received a security update in years. Experts estimated the system would take about 10 minutes to hack. The State of Virginia published a scathing report on the WINVote system, and, fortunately, all of these systems were taken offline in time for the 2016 election – but many years too late.

Why do these backward and dangerous voting machines still exist long after the experts have pronounced them inadequate? The answer is in the time divide that separates you from me. Here I am at the brink of the election. Election coverage dominates the US news, and even outside the US, the election is receiving significant attention. There you are after the election – after a year of constant and persistently annoying election coverage – and people are just dying for another

topic. No one really wants to talk about insecure and ridiculously obsolete voting machines.

Why isn't anyone addressing this issue in a systematic way? Many outside of the US (and *in* the US for that matter) might not be aware of how decentralized the voting process really is. The headlines refer to a national election, but voters actually choose electors at the state level. The laws for the election are largely managed by the states, and the actual voting and vote tallying happens at the county or municipal level. Decisions about voting hardware might be made by the state, county, or city depending on the jurisdiction.

Buying new voting machines would actually cost a lot of money. Most counties don't have the money for it, and most counties don't have fair vote activists on hand to show up at committee meetings and demand new hardware.

But the problem has reached a new urgency. This is the first election when a major-party candidate has tried to claim the election is "rigged." He is most certainly incorrect. (Despite the problems with these voting machines, the decentralization of the US vote means it would be extremely difficult for a group to systematically hack a national election without detection.) However, the accusation is destabilizing for national unity, and, frankly, the continued existence of clearly deficient voting machines plays into the hands of conspiracy theorists who thrive in a climate of distrust.

The best solution is to *fix the whole mess*. It is time for a uniform policy across the US establishing minimum standards for:

- security of voting machines
- uniformity of the voting process, including a verifiable, hard-copy record of the vote result
- an end to the alarming voter suppression practices that have gained popularity in some jurisdictions
- a non-partisan, scientific methodology for drawing legislative district lines that avoids the partisan gerrymandering tactics that both the major parties seem willing to use when given the opportunity.

Many organizations are already working on these problems. Check out the work of TrustTheVote.org [2] and its sponsor, the Open Source Election Technology Institute [3], for more on creating a fair and verifiable election process using open technologies. The Verified Voting site [4] provides information on voting machines used throughout the US.

Several groups are working on the problem of unfair legislative boundaries through gerrymandering [5], and a group of researchers recently unveiled a technique for determining gerrymandering in a legislative map using high-performance computing [6].

How will uniformity arise from a diverse system with 50 states and thousands of counties? Ultimately, the only way these voting problems will get fixed is if the people who are in a position to fix them will keep their focus after the election – and be willing to put the country's business before whatever personal advantage they get from letting it all stay broken.



Joe Casad,
Editor in Chief

INFO

- [1] Cylance video: <https://blog.cylance.com/cylance-discloses-voting-machine-vulnerability>
- [2] TrustTheVote: <https://trustthetvote.org/>
- [3] Open Source Election Technology Institute: www.osefoundation.org/
- [4] Verified Voting: <https://www.verifiedvoting.org/resources/voting-equipment/avs/winvote/>
- [5] End Gerrymandering: <http://www.endgerrymandering.com/>
- [6] Ending political gerrymandering with HPC: <http://www.admin-magazine.com/HPC/News/Ending-Political-Gerrymandering-with-HPC>



WHAT'S INSIDE

This month we show you social networking the FOSS way with Diaspora and examine a couple Linux distros that will ease the transition from Windows to Linux.

Other highlights:

- **Debian LTS** – the quintessential Free Linux gets long-term support – find out how they did it (page 38).
- **Z-Wave** – use this wireless IoT protocol to automate smart devices in your home (page 50).

Our friends at Linux Voice have another great issue, with tutorials on Wine and Collectd, as well as an illuminating look at 20 years of KDE (page 65).

SERVICE

- 3 Comment
- 6 DVD
- 96 Featured Events
- 97 Call for Papers
- 98 Preview

NEWS

8 News

- New board of directors for Linux Foundation
- VeraCrypt 1.19 addresses flaws
- Ubuntu 16.10, Red Hat 7.3, and Ubuntu Core 16 released
- Deep Q-Network Doom bot

11 Kernel News

- Making device-based RAM available to user processes
- Blunting hostile code
- New kernel messaging system
- Clean shutdown on low battery

14 Interview – Meet Apache Creator Brian Behlendorf

Apache developer talks about Hyperledger and blockchains

COVER STORIES

18 Diaspora

Diaspora is a free and open source decentralized social network consisting of millions of independently run pods. Join an existing pod or set up your own with minimal programming skills.

24 ChaletOS and Zorin OS

ChaletOS and Zorin OS are two Ubuntu derivatives designed for users migrating from Windows.



REVIEWS

30 Screencasters

Five screencasting programs reviewed: Kazam, recordMyDesktop, SimpleScreenRecorder, Vokoscreen, and Krut.

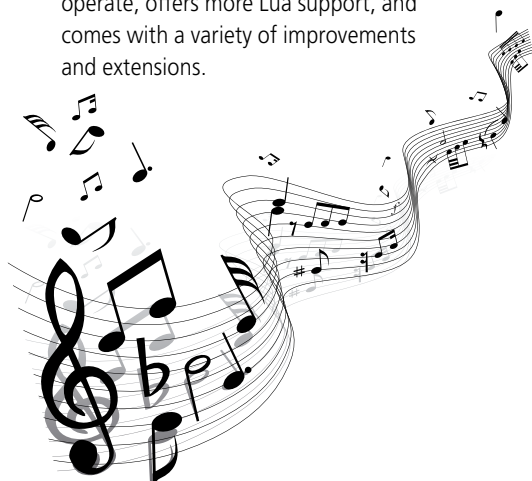
38 Debian LTS

The Debian project is extending its famous development process to offer long-term support.

Funding level (time per month)	15 min.	30 min.	1 h.	2 h.	3 h.	4 h.	6 h.	1 day	2 days	3 days
Benefits (credits)										
Packages to prioritize										
Thanked as sponsor:										
Private mailing list										
Direct contact to LTS staff										
Submit your own test cases										
Prices										
Monthly payment	N/A	€255 / \$300	€380 / \$400	€510 / \$600	€680 / \$800	€1390 / \$1600	€2940 / \$3600	€4980 / \$6000	€9960 / \$12000	€19920 / \$24000
Quarterly payment	N/A	€255 / \$300	€380 / \$400	€510 / \$600	€680 / \$800	€1390 / \$1600	€2940 / \$3600	€4980 / \$6000	€9960 / \$12000	€19920 / \$24000
Biyearly payment	N/A	€255 / \$300	€380 / \$400	€510 / \$600	€680 / \$800	€1390 / \$1600	€2940 / \$3600	€4980 / \$6000	€9960 / \$12000	€19920 / \$24000
Yearly payment	€255 / \$300	€380 / \$400	€510 / \$600	€680 / \$800	€1390 / \$1600	€2940 / \$3600	€4980 / \$6000	€9960 / \$12000	€19920 / \$24000	€39840 / \$48000

40 Ardour 5

The fifth generation of the free Ardour digital audio workstation is easier to operate, offers more Lua support, and comes with a variety of improvements and extensions.

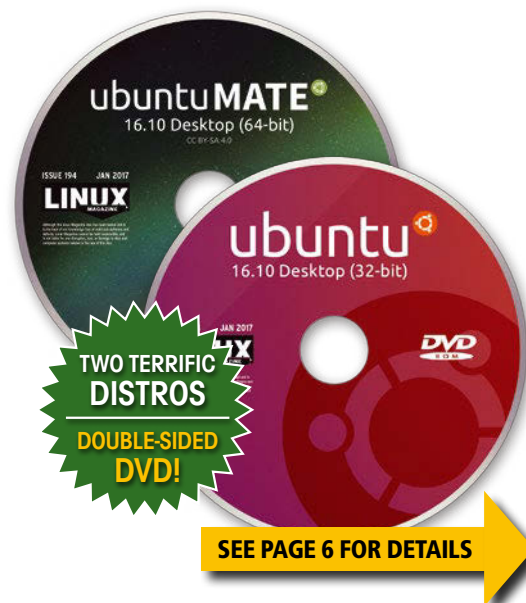
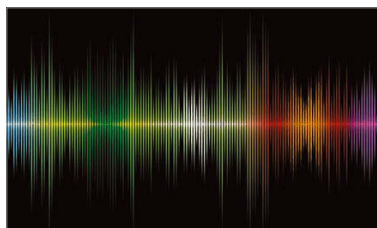


IN-DEPTH

- 46 Lepton Image Compressor**
Lepton lets you compress your photo files to make room for more.



- 60 Command Line – Audio File Conversion**
Convert your music files to Ogg Vorbis, or use Flac for more refined playback.



- 48 Charly – Searx**
This privacy-respecting, hackable meta search engine is the answer to commercial search engines that don't have your best interests at heart.



- 50 Z-Wave**
We examine the range and security of the Z-Wave protocol and offer a few basics to get you started on a smart home.



- 54 Browse Your Source**
If you've ever struggled to get a sense of someone else's code, the right tool could save you hours of grepping.

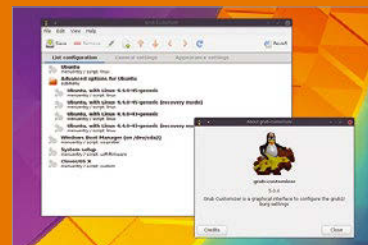


- 58 Ask Klaus!**
Controlling brightness, forensic mode, and proprietary Broadcom WiFi drivers.



LINUXVOICE

- 65 Welcome**
Looking forward to a third decade of the KDE desktop.
- 66 SEOscerer's Apprentice**
Beware the SEO "expert." You might call forth problems far worse than a low search rank.
- 67 Cyber-Glossary**
The UK's National Cyber Strategy doc is worth every penny of the free download cost.
- 68 20 Years of KDE**
The ups and downs of the "Kool" desktop over the past two decades.
- 72 FAQ – Connected Car**
Automotive Grade Linux: Will 2017 be the year of Linux on wheels?
- 74 Core Tech – Filesystem Monitoring**
Notification APIs in Linux.
- 80 FOSSPicks**
The QupZilla QtWebEngine browser, Powerline prompts, GRUB Customizer, Attract Mode emulator front end, and more.
- 86 Gaming**
Starting over with Mad Max, Aragami stealth ninjas, and Warhammer RTS.
- 88 Tutorial – Collectd**
A tool that harvests and stores your system stats for colorful graphs.
- 92 Tutorial – Wine**
Vintage Windows apps on Linux.



On the DVD

ubuntu
16.10 Desktop (32-bit)

Ubuntu 16.10 Desktop 32-bit Live

Ubuntu features the Unity desktop with a Dash and favorites along the edge of the desktop. Version 16.10, Yakkety Yak, comes with nine months of security and maintenance updates, LibreOffice 5.2 – the GTK3 version, and Nautilus file browser 3.20. Systemd is now used for user sessions as well as system sessions. Other highlights:

- Linux kernel 4.8
- Changelog entries for PPAs
- Gnome apps v3.20 or later
- Snap packages

ubuntuMATE
16.10 Desktop (64-bit)
CC BY-SA 4.0

Ubuntu Mate 16.10 Desktop 64-bit Live

The Mate desktop is a stable, traditional desktop forked from the Gnome 2 codebase. Mate's modest hardware requirements make it suitable for modern workstations, single-board computers, and older hardware. As the community says, "Ubuntu Mate makes modern computers fast and old computers usable." Mate is the first major distribution to ship a full GTK3+ implementation of the upgraded Mate Desktop 1.16.0. In addition to many bugfixes, this version of Yakkety Yak brings updated Mate themes and translations.

TWO TERRIFIC DISTROS
DOUBLE-SIDED DVD!



ADDITIONAL RESOURCES

- [1] Ubuntu 16.10 Yakkety Yak: <https://www.ubuntu.com/desktop>
- [2] Release notes: <https://wiki.ubuntu.com/YakketyYak/ReleaseNotes>
- [3] Snaps: <https://www.ubuntu.com/desktop/snappy>
- [4] Ubuntu Mate: <https://ubuntu-mate.org/about/>

Defective discs will be replaced. Please send an email to subs@linux-magazine.com.

NEW

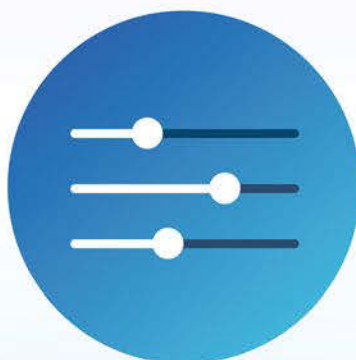


CloudNX

next generation Cloud Servers



**UK data centres,
UK support**



Flexible



**Pay as you
use pricing**

Enterprise features

- Load balancing
- VPN
- External firewall

Set up in seconds

- Easy to use control panel

Cloud Servers from:

£15.00

per month ex VAT charged at 20%

Call **0808 1686 777**

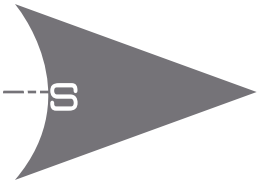
or visit [fasthosts.co.uk/cloud-servers](https://www.fasthosts.co.uk/cloud-servers)

SERVERS • WEB HOSTING • DOMAIN NAMES • EXCHANGE EMAIL



NEWS

Updates on tech



THIS MONTH'S NEWS

08 The Linux Foundation Announces New Board Members

- The Linux Foundation Announces a New Board of Directors
- Open Stack Summit in Barcelona

09 VeraCrypt Addresses Flaws

- VeraCrypt 1.19 Addresses Some Critical Flaws
- Ubuntu 16.10 Released
- More Online

10 RHEL 7.3 Released

- Red Hat Releases Red Hat Enterprise Linux 7.3
- Ubuntu Core 16 Released
- Carnegie Mellon Students Build Doom Bot Based on Google Deep Q-Network

The Linux Foundation Announces a New Board of Directors

The Linux Foundation has announced three new additions to its board of directors. The new members are Erica Brescia, co-founder and COO of Bitnami; Jeff Garzik, co-founder of Blog; and Nithya A. Ruff, Director of Western Digital's Open Source Strategy Office.

Brescia and Ruff will take the place of Larry Augustin and Bdale Garbee as At-Large Directors. Garzik replaces Matt Jones of Jaguar Land Rover.

Ruff said in a statement, "The Linux Foundation is working to create an inclusive open source culture that stretches across organizations. I anticipate being able to help deepen that work as more industry professionals of all backgrounds get involved in open source."

Jim Zemlin, executive director of the Foundation welcomed the new directors and said, "They will help guide the strategy of The Linux Foundation, the home to some of the most successful open source projects and largest shared technology investment in history. The open source community at large and our nearly 800 members will benefit from the insight and expertise each of these individuals brings. We thank Larry, Bdale, and Matt for their long and faithful service on the Board and look forward to their continued participation in the community."



OpenStack Summit in Barcelona

The European edition of the OpenStack Summit concluded in Barcelona, Spain, October 28. Back in 2010, the first OpenStack Summit in Austin, Texas, had only 75 attendees. Six years later, the Barcelona event welcomed more than 5,000 attendees from all around the globe.



OpenStack is often compared with the Linux kernel: What Linux has been to servers, OpenStack has been to infrastructure and cloud. Giants like AT&T are building their future networks on top of OpenStack, and compa-

nies like Volkswagen are using OpenStack to build their infrastructure. Anni Lai of China Mobile drew a direct comparison between OpenStack and Linux. She said, "OpenStack is a little bit over 6 years old compared to Linux [at] 25 years old. We are still a youngster; however, our speed of innovation and our presence in data centers are not less than Linux, which is mostly due to our open collaborative platform and culture."

During the event, Mark Collier, COO of the OpenStack Foundation, announced four new gold members, including City Network and Deutsche Telekom in Europe and 99Cloud, and China Mobile in China.

The Foundation also announced the winner of the OpenStack Superuser Award, China Mobile, which did some massive deployments of OpenStack. In a later interview, a China Mobile representative told us they are building a public cloud on OpenStack that competes with AWS.

Some of the highlights of the event included keynote speeches from members of the scientific community. Tim Bell from CERN gave a talk about how the organization is using OpenStack to handle massive amounts of data generated by the Large Hadron Collider. Dr. Rosie Bolton of SKA (the Square Kilometre Array project) also gave a keynote presentation around the possible use of OpenStack in the radio telescope project.

VeraCrypt 1.19 Addresses Some Critical Flaws

VeraCrypt, the open source full-disk encryption program, has released version 1.19, which fixes many critical flaws in previous editions.

Between August 16 and September 14, 2016, an audit of VeraCrypt 1.18 was conducted by the French cybersecurity firm Quarkslab. The audit, which was sponsored by the Open Source Technology Improvement Fund (OSTIF), found eight critical vulnerabilities. All of the serious issues have been fixed in the latest version of VeraCrypt, and users are urged to immediately upgrade to version 1.19.

The report cites other issues with VeraCrypt that still need fixing. "The availability of GOST 28147-89, a symmetric block cipher with a 64-bit block size, is an issue. This algorithm must not be used in this context. Compression libraries are outdated or poorly written. They must be updated or replaced. If the system is encrypted, the boot password (in UEFI mode) or its length (in legacy mode) could be retrieved by an attacker."

If you are a VeraCrypt user, please upgrade to the latest release immediately.

Ubuntu 16.10 Released

Canonical, the parent company of Ubuntu, has released the latest version of their open source operating system, Ubuntu 16.10.

According to Canonical, Ubuntu 16.10 "Yakkety Yak" comes with "hybrid cloud operations, bare-metal cloud performance, and the ability to lift-and-shift 80% of Linux VMs to machine containers." Also included in the latest release are "Kubernetes for process-container coordination, full container support in OpenStack, and telco-grade networking latency enhancements."

Ubuntu 16.10 includes Juju 2.0 for managing hybrid cloud operations and features MaaS (Metal as a Service) 2.0, which allows a physical data center to "feel like a cloud," with on-demand availability of machines with custom images.

On the consumer front, Ubuntu 16.10 gives a preview of Unity 8 that is geared toward mobile devices and offers device convergence. Canonical said in a press release that the Unity 8 developer preview includes apps that scale from phone to desktop and from mouse to touchscreen, setting a precedent for the next wave of Linux devices.

Ubuntu 16.10 is available for free download: <http://cdimage.ubuntu.com/daily-live/current/>.



ubuntu

MORE ONLINE

Linux Magazine

www.linux-magazine.com

Off the Beat • Bruce Byfield

Crowdfunding vs. Patronage

Originally, free software was the work of volunteers. At first, they sought the support of sympathetic companies, and were often given a free hand so long as they showed results. However, more recently, free software developers have sought the support of crowdfunding – a development that leaves me with mixed feelings.

Replacing Google Maps with Marble

Several years ago, Google Maps was ranked as the most popular app on smart devices, installed on over half of all devices. The convenience is obvious, but few stop to investigate privacy concerns. For that reason, I have been using Marble, KDE's geographical application, instead.

Paw Prints • Jon "maddog" Hall

Free and Open Source: Economics, not Politics

"Open Source" has been prominent in Brazil for more than 10 years. During that time, Free and Open Source Software and Hardware (FOSSH) has become associated with the PT political party. This paper outlines why FOSSH should be a major policy of every Brazilian political party. While it is answering a particular current problem in Brazil, it can be used in almost any country.

ADMIN HPC

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

Modern Fortran – Part 1: Fortran 90

Jeff Layton

Fortran comprises a huge body of software in high-performance computing. We look at Fortran's evolution into a modern HPC language.

ADMIN Online

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

SDN and the Future of Networking • Udo Seidel

Networking is a central focus of IT. With all the virtual machines running on today's networks, it stands to reason that the experts would someday find a way to virtualize the network itself. Software-Defined Networking (SDN) is an architecture that promises to cope with new demands.

A REST Interface for FreeIPA • Thorsten Scherf

Access to the FreeIPA identity management framework is usually handled via a graphical web interface or a command-line tool, but the framework can also be queried directly via the JSON-RPC API.

Red Hat Releases Red Hat Enterprise Linux 7.3

Red Hat has announced the release of Red Hat Enterprise Linux 7.3, the latest version of their flagship operating system. The new release features improvements across the board, from performance to security.

Much of the focus is on the data center networking experience, with new support for lightweight tunnels that enable more secure, scalable, and efficient guest instances. Additionally, enhancements to the bulk packet memory allocator improve performance for both 40GB and 100GB networking interfaces.

High-transaction applications like databases, event-processing systems, and virtual machines need a high input/output (IO) rate and low latency. RHEL 7.3 improves performance by adding support for high speed, low latency non-volatile memory devices. In addition, version 7.3 makes performance-sensitive Parallel NFS (pNFS) clusters easier to manage with support for the Block SCSI and Flex Files layout format.

RHEL 7.3 also introduces new capabilities around Linux containers and the Internet of Things (IoT). Two key features aimed at IoT include support for communication with Bluetooth LE (Low Energy) and kernel support for the Controller Area Network bus (CANbus) protocol used in automobiles and high-end industrial controllers.

RHEL 7.3 offers better support for Linux containers through enhanced management tools and an updated container runtime (docker engine). RHEL 7.3 also comes with new container signing capability as a Technology Preview.

Ubuntu Core 16 Released

Canonical, the parent company of Ubuntu, has released Ubuntu Core 16. Ubuntu Core is Canonical's minimal edition for embedded devices, Internet of Things, and cloud deployments. The latest version comes with Update Control, which allows software publishers and manufacturers to validate updates across the ecosystem before they are applied. The greatest feature of Ubuntu Core is that its updates are transactional, which means that failures are automatically rolled back so there are no bricked or compromised devices.

Canonical claims the new Ubuntu Core delivers security, management, operations, and upgradability in a compact, developer-friendly platform, thanks to its use of Snap packages. Snaps are securely confined, digitally signed, read-only, tamper-proof application images.

In the month of October, millions of IoT devices were used to launch a massive wave of Distributed Denial of Service (DDoS) attacks on the managed DNS service Dyn. The attack brought down a big chunk of the Internet in the United States. Services for Amazon, Twitter, and other major sites were down for many users.

Experts believe that the attack could have been avoided if these IoT devices were running operating systems that focus on security of IoT devices through regular and secure updates without user intervention.

Carnegie Mellon Students Build Doom Bot Based on Google Deep Q-Network

Two Carnegie Mellon students created a Doom AI agent that is capable of beating human players, as well as built-in AI agents, in the classic Doom computer game. Devendra Chaplot and Guillaume Lample used Google's DeepMind deep-learning technology to develop their Doom-playing bot, which they have nicknamed "Arnold."

According to the story at the CMU website, Doom is harder to play than other games that are the focus of AI research because the player can only see part of the playing field. The game exists in a 3D world but plays out on a 2D screen. The built-in AI agents included with Doom cheat by consulting maps and other background data to compete. Chaplot and Lample's bot, on the other hand, uses visual information that would be available to a human.

Although other Doom bots exist in the world (Arnold actually took second place in a recent world-wide competition), the Carnegie Mellon project is attracting attention as an application of a Deep Q-Network (DQN), part of the DeepMind platform, which is billed as Google's answer to IBM Watson and other similar tools.



Zack's Kernel News



Chronicler Zack Brown reports on the latest news, views, dilemmas, and developments within the Linux kernel community.

By Zack Brown

Making Device-Based RAM Available To Arbitrary User Processes

Anshuman Khandual wanted to come up with a way for all attached devices to make their on-board RAM available to a running Linux system as just a normal region of RAM that any process could use without having to do anything special.

This is a tough nut to crack, because it's not just a simple matter of mapping the memory and making it available for allocation by user processes. As Anshuman put it, "To achieve seamless integration between system RAM and coherent device memory it must be able to utilize core memory kernel features like anon mapping, file mapping, page cache, driver managed pages, HW poisoning, migrations, reclaim, compaction, etc."

Each of the above requirements had its own set of difficulties, which Anshuman enumerated. For one thing, unlike regular RAM, device RAM couldn't be made available until after a given device had been initialized. That constraint would need to be handled properly. Likewise,

each issue had its own constraints and caveats that would need to be papered over so that regular user code would perceive the RAM as simply being available for use.

Anshuman went over a wide-ranging array of steps that he felt would need to be taken to accommodate device RAM.

One requirement he identified was the need for the device's memory page data structures to be used cleanly within the kernel's LRU (least recently used) lists of available memory. However, Jerome Glisse felt that this might not be as important as it seemed. The problem, as he saw it, was that the kernel's page cache code would sometimes need to add any given memory page to the LRU lists. If this wasn't available, the page cache code would be flummoxed.

However, Jerome thought there was a reasonable workaround available for this. In his own work on HMM (heterogeneous memory management), he explained, "I am using `ZONE_DEVICE` and because memory is not accessible from CPU (not everyone is blessed with a decent system bus like CAPI, CCIX, Gen-Z, ...) so in my case a file back page must always be spawned first from a regular page and once read from disk then I can migrate to GPU page. So if you accept this intermediary step you can easily use `ZONE_DEVICE` for device memory. This way no lru, no complex dance to make the memory out of reach from regular memory allocator."

Jerome suggested that he and Anshuman pool their efforts, as it seemed there was a good opportunity to simplify and enhance both projects.

While Jerome was suggesting that some of Anshuman's requirements might not be strictly necessary, Dave Hansen felt that Anshuman had left out some crucial requirements, specifically support for `autonuma` (automatic non-uniform memory architecture) and `hugetlbfs`. Both of these relate to gathering up available RAM and making it available to the system, so presumably they'd be relevant to Anshuman's work.

David Nellans, working with Anshuman on this project, replied that this had been a conscious choice, to make the device RAM pages migratable between CPUs via an explicit decision, rather than by `autonuma`'s automated method. As for ignoring `hugetlbfs`, David said that Anshuman's code would rely more on THP (transparent huge tables), which performed a similar function as `hugetlbfs`.

Dave wasn't satisfied with this explanation and said that in particular it would force some complexity into the `hugetlbfs` code to exclude this type of device RAM explicitly.

Various folks descended into a technical debate of the issues surrounding this and Jerome's suggestion. The question of what to support and what to work around was crucial to identifying whether the feature could be implemented at all: Support too much, and the code becomes too complicated/slow/unmaintainable. Work around too much, and the code becomes too inconsistent/unusable/breakable. The goal is ultimately to create features that everyone can use without too many headaches and that can be maintained without too much insanity.

For example, at one point Anshuman objected to Jerome's earlier suggestion, saying that "there are problems now migrating user mapped pages back and forth between LRU system RAM memory and non LRU device memory which is yet to be solved. Because you are proposing a non LRU based design with `ZONE_DEVICE`, how we are solving/working around these problems for bidirectional migration?" And he added, "Before non LRU migration support patch series from Minchan, it was not possible to migrate non LRU pages which are generally driver managed through `migrate_pages` interface. This was affecting the ability to do compaction on platforms which has a large share of non LRU pages. That series actually solved the migration problem and allowed compaction. But it still did not

solve the migration problem for non LRU *user mapped* pages. So if the non LRU pages are mapped into a process's page table and being accessed from user space, it can not be moved using `migrate_pages` interface."

To which Jerome replied at some point, saying, "Minchan is trying to allow migration for device driver kernel allocated memory ie not memory that end inside a regular vma (non special vma) but only inside a device driver file vma if at all. So we are targeting different problem. Me I only care about "regular" process memory is private anonymous, or share memory (either back by regular file or pure share memory). I do not want to mess with any of the device driver vma or any special vma that are under control of an unknown device driver. Trying to migrate any such special memory is just not going to work. Moreover I believe it is not something we care [about] in the first place."

So, there are a range of issues surrounding the various aspects of support for device-based RAM in the general memory pool. Different ongoing kernel projects have a vested interest in seeing a compatible implementation, and there are never any easy ways to discern which would be the easiest approach to thread a given needle.

Ultimately, projects of this type will inevitably attract the attention of hardcore kernel hackers who can definitively rule out certain approaches. At that point I'd expect a certain amount of goal realignment, especially if the folks involved in the initial patches have gotten too far off into a world of feature compromise and requiring user code to be too aware of kernel internals. This is exactly the sort of situation where we might see Linus Torvalds come along and propose a dramatically simpler approach that satisfies none of the original proponents but that allows device-based RAM to be used by certain processes in simple ways.

Blunting Hostile Code

Juerg Haefliger posted some code to implement eXclusive Page Frame Ownership (XPFO), which he felt would help guard against `ret2dir` attacks. A `ret2dir` attack occurs when hostile code is written into kernel memory, such that a function call will return to the location

of that hostile code, thus executing it. It's a sibling to a `ret2usr` attack, in which the hostile code is written into user memory instead of kernel memory.

XPFO addresses `ret2dir` by creating exclusive ownership of memory pages, so that user code would not be able to allocate memory from the kernel, insert hostile code, and then cause the kernel to attempt to run that code.

Laura Abbott had some technical suggestions, mostly about portability and how to improve locking efficiency in Juerg's code, and she offered suggestions for how to contact the relevant maintainers.

Juerg liked Laura's code review and pointers and started posting updated patches. The technical discussion went on for a bit, but there were no serious objections, and it looked as though Juerg was making good progress toward putting together an acceptable patch.

Some argue that little security fixes like this don't make much difference in the grand scheme of things, especially when they guard against attacks that can only occur when user code does something dumb; however, others think that any opportunity to remove an attack vector should be taken. Ultimately, there's a give-and-take between usefulness and bloat, but it does look as though Juerg's code is likely to make it into the kernel eventually.

New Kernel Messaging System

David Herrmann introduced the idea of a kernel messaging bus called `bus1.ko`, inspired by the existing `kdbus` project but going its own way in terms of design. The `bus1.ko` project would implement interprocess communication (IPC) that would be completely divorced from user space.

Communication would take the form of nodes containing the message data and handles containing references to nodes. Peer processes would send messages to one another via handles that would give the target process access to the referenced node. Once a process was in possession of a handle, the only way to cut off that process's access to the referenced node would be for the original process to destroy the node in question. There would be no way to modify or revoke the handle once it was sent.

Linus Torvalds took a look at this and gave his assessment:

The thing that tends to worry me about these is resource management.

If I understood the documentation correctly, this has per-user resource management, which guarantees that at least the system won't run out of memory. Good. The act of sending a message transfers the resource to the receiver end. Fine.

However, the usual problem ends up being that a bad user can basically DoS a system agent, especially since for obvious performance reasons the send/receive has to be asynchronous.

So the usual DoS model is that some user just sends a lot of messages to a system agent, filling up the system agent resource quota, and basically killing the system. No, it didn't run out of memory, but the system agent may not be able to do anything more, since it is now out of resources.

Keeping the resource management with the sender doesn't solve the problem, it just reverses it: Now the attack will be to send a lot of queries to the system agent, but then just refuse to listen to the replies – again causing the system agent to run out of resources.

Usually the way this is resolved is by forcing a "request-and-reply" resource management model, where the person who sends out a request is not only the one who is accounted for the request, but also accounted for the reply buffer. That way the system agent never runs out of resources, because it's always the requesting party that has its resources accounted, never the system agent.

David replied that for this version of the design, `bus1.ko` did all resource accounting via user ID in order to match POSIX specs as well as possible. He said, "More advanced accounting is left as a future extension."

He went on to explain that the tricky part of resource accounting came when users needed to transfer resources. He said, "Before `SEND`, a resource is always accounted on the sender. After `RECV`, a resource is accounted on the receiver. That is, resource ownership is transferred. In most cases this is obvious: Memory is copied from one address-space into another, or file-descriptors are added into the file-table of another process, etc."

David added, "at the time of `SEND` resource ownership is transferred (unlike

sender-accounting, which would transfer it at time of RECV). The reasons are manifold, but mainly we want RECV to not fail due to accounting, resource exhaustion, etc. We wanted SEND to do the heavy-lifting, and RECV to just dequeue. By avoiding sender-based accounting, we avoid attacks where a receiver does not dequeue messages and thus exhausts the sender's limits."

Finally, David said, "The issue left is senders DoS'ing a target user. To mitigate this, we implemented a quota system. Whenever a sender wants to transfer resources to a receiver, it only gets access to a subset of the receiver's resource limits. The inflight resources are accounted on a uid <-> uid basis, and the current algorithm allows a receiver access to at most half the limit of the destination not currently used by anyone else."

Richard Weinberger asked whether messages could be passed between cgroup containers as well as between processes within a single system; and Tom Gundersen replied, "There is no restriction with respect to containers. The metadata is translated between namespaces, obviously, but you can send messages to anyone you have a handle to."

The discussion petered out around there. It's unclear whether Linus was satisfied with David's answer regarding DoS possibilities, so it's difficult to tell

what would become of David's design. It's also unclear who would be the big users of this message-passing system. So, overall, it's too soon to predict the fate of any patches coming out of this design.

Clean Shutdown When Battery Power Is Too Low

Pavel Machek was annoyed by the fact that Nokia's N900 smartphone would drain the battery right down to a system crash. He proposed doing a clean shutdown if the system detected that the lithium ion battery dropped below 3V of usable power.

He posted a patch to do this. Tony Lindgren liked the concept but felt that 3V seemed slightly arbitrary. He proposed making the exact voltage configurable, and he added, "Also, the shutdown voltage can depend on external devices connected. It could be for example 3.3V depending on eMMC on some devices while devices with no eMMC could have it at 3.0V."

Pavel groaned, in a friendly way, and agreed that, yes, making it configurable would probably be the way to go. He said, "we probably need to create battery object in the device tree, then add properties there."

Pali Rohár suggested, "bq27x00 has the EDVDF flag which means that the battery is empty. Maemo with bq27x00 driver is configured to issue system

shutdown when EDVDF is set. Maybe kernel should issue emergency shutdown e.g. a minute or two after the EDVDF flag is set?"

Pavel then took a look and noticed that "it prioritizes battery cold over battery dead. IMO we don't need to shutdown on battery cold (we just may not charge the battery), but we need to shutdown on battery dead." So he posted a short patch to catch that particular case.

Later, Pavel had second thoughts about whether or not to let this whole feature be configurable at all. He said, "Actually, do we need to make it configurable? It looks like we should respect hardware telling us battery is dead, and only use (low) hardcoded voltages as a fallback."

He and Pali went back and forth a little on technical considerations, but there was no ultimate design resolution. Even so, it seems clear that something along the lines of this feature will get into the kernel as soon as it seems ready. ■■■

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**.

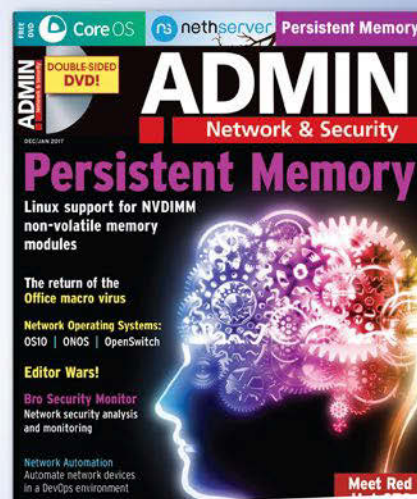
REAL SOLUTIONS FOR REAL NETWORKS

Each issue delivers technical solutions to the real-world problems you face every day.

ADMIN magazine covers Windows, Linux, Solaris, and popular varieties of the Unix platform.

Learn the latest techniques for better network security, system management, troubleshooting, performance tuning, virtualization, cloud computing, and much more!

**FREE
CD or DVD
in Every Issue!**



6 issues per Year!

ORDER ONLINE AT: shop.linuxnewmedia.com

An interview with Brian Behlendorf

Back on the Block

By Swapnil Bhartiya

Brian Behlendorf is nothing short of an open source celebrity. Behlendorf was one of the primary developers of an open source project that revolutionized the web: the Apache web server. In 2003, MIT Technology Review listed Behlendorf among the top 100 innovators in the world under the age of 35.

I first met Behlendorf back in 2005, when I lived in India, and I would often seek his input for my stories. But Behlendorf took a sabbatical from the open source world for several years. Early this year, The Linux Foundation appointed Behlendorf as the executive director to head their Hyperledger project. Hyperledger, which launched in December 2015, is an industry-wide effort to develop and enhance open standards for blockchain transaction technology. Blockchain is best known as the technology at the core of the Bitcoin cryptocurrency, but many experts believe it has much wider application to a range of tasks, from banking, to medicine, to real estate.

I asked Behlendorf what he has been doing all these years – and what brought him to The Linux Foundation.

Linux Magazine: You have been in the open source world for a very long time. How much has open source achieved in all these years?



Brian Behlendorf: I think open source has become the default. If you're launching a new platform or trying to promulgate a new technology, you have to be open source.

IBM knew there wasn't much of a commercial future for a proprietary blockchain platform. It would have been a hard time for them selling against the already market-winning, open source equivalents. The innovation that you see in open source would be really hard to achieve in a venture capital-funded private domain. This is for broad market, not just some specific domain.

The financial services world was an early adopter of open source without realizing it. A lot of their technologists brought in Linux and Apache under the bar, but still thought of themselves as

buying Big Iron and big technologies from a company like SAP or Oracle.

The financial world is also waking up to move quickly into these new domains, like blockchain. Their involvement on Hyperledger has been partly a journey of helping their executives understand what does it mean to be writing code and then giving it away.

In some ways, open source is now a default; in other ways, there's still a whole new terrain and whole new groups who need to learn how open source works. Those conversations are shorter now than before, but they're still happening.

LM: You worked at the White House and also on the Obama campaign. How important do you think it is for government to use and open-source their own infrastructure?

BB: I think if taxpayer money is being spent on the development of new software that should all be open source. No question.

I think when government developers are building code, that should also be open source. It should be open source, but also shared upstream with open-source communities, such as those housed at Apache, or The Linux Foundation, or elsewhere.

In fact, government should try to find homes for their code very much like the homes for Linux or Apache, and if they don't exist, create new ones. I think the nonprofit structure that we've figured out with many of these large organizations is really really important to making these communities work. Government can spend a million dollars towards a body of code, then just throw it out there. No one ever sees it again and evolves it. It is as good as dead. It has to be about more than just putting a license on it and shipping it out.

By the way, the license shouldn't be GPL. It shouldn't be an MIT or Apache license. MIT and Apache also give you all the patent rights. Most people don't know this, but governments can actually file patents.

LM: Linux often gets the credit for making companies comfortable with the open source development model. How will this comfort with open source help in the adoption of blockchain?

BB: Banks are an important part of this open source blockchain technology. Fifteen years of running Linux helps them understand that open source is reliable and is not a fad. They are more comfortable with the idea that, when you need to build something new, you build it collectively with other banks.

The other thing is that banks have always collaborated on standards. They worked on financial products market language and other XML-based systems, but those standards then had to be implemented separately. Those implementations always lead to conflict and struggles, especially when you wanted to upgrade.

Now people are realizing that, if we can evolve the standards and the software in parallel, at the same time, and in conjunction with each other, we can

evolve these platforms more quickly. I think that's what has driven this interest in blockchain. It's hard right now to tear apart blockchain standards from blockchain implementation. Blockchain, in particular, is a tool that only makes sense when you're talking about a community of companies. That requires thinking about sharing code and pushing that code out.

LM: Can you please explain what blockchain is and how it will affect our lives?

BB: Blockchain is a shared, multi-master database resilient to hostile actors. It's a ledger where each entry in the ledger is agreed to by consensus by everyone else in the network. Everyone has the same copy of the database. It's in the same order. Events that happen, that are written to it, can't later be refuted. They can't later be censored or changed. That creates a system of record – a common body of truth that you can use to implement immediate settlement in a banking network.

You can also use blockchain to build a land title registry for a country, where everyone can see who owns what property. Then as a property changes hands, you can see the history of the changing ownership. If a corrupt politician tries to steal that land, you'd be able to catch it and keep that from happening.

Once you have this shared database, you can build smart contracts that execute on shared databases across the network. The accompanying software can add new entries to that ledger based on whatever variables you want.

LM: What are banks and other institutions doing now (or prior to adopting a blockchain solution)? Don't they need something similar to ensure no one is tampering with records?

BB: Previously banks would either depend upon a single company at the center of that network to maintain point-to-point connections with everybody, but then depend upon that central provider to arbitrate when there were conflicts or if there were just differences in how things were recorded. That was a problem. If you have to depend upon a central provider, that's a weakness. That's the main idea.

DON'T MISS A SINGLE ISSUE!

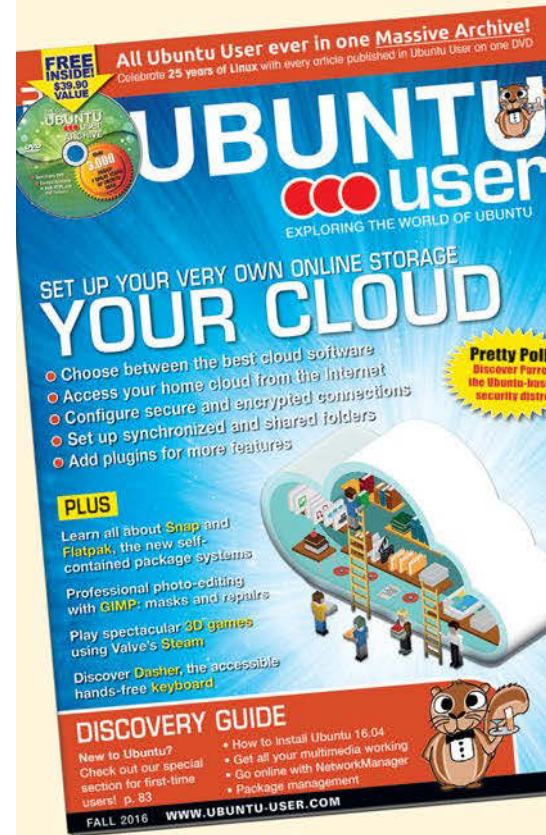
The first print magazine created specifically for Ubuntu users!

Ease into Ubuntu with the helpful Discovery Guide, or advance your skills with in-depth technical articles, HOW-TOs, reviews, tutorials, and much, much more.

SUBSCRIBE NOW!

4 issues per year for only
£ 24.90 / EUR 29.90 / US\$ 39.95

- ✓ Don't miss a single issue!
- ✓ Huge savings – Save more than 35% off the cover price!
- ✓ Free DVD – Each issue includes a Free DVD!



LM: What other industries will blockchain affect?

BB: It's also hard to imagine which industry is *not* going to be affected. For example, many companies are already working on the question of storing health data in blockchains. There's obviously a lot of privacy concerns when it comes to sharing health data; not just a personal medical record. You could share things like who's asked for permission records because sometimes just knowing who's asked for that permission is important enough. It's worth recording somewhere in an indelible ledger, so that you can go back later and audit to see who asked for records across the network. I know there's one company working on recording patient consents for clinical trials.

Blockchain offers paperwork reduction benefits. It is also a way to give patients better visibility when the data of a trial that they've participated in is studied later when making a case for a new trial. These medical uses are some really interesting applications. The privacy concerns mean it's not as obvious or as easy as putting all the big data into one big pool.

LM: What's in there for developers to get excited about blockchain?

BB: If you're simply curious about where this technology is heading, that's one reason to start to dig in and get your hands dirty. If you work for a company that is involved in some sort of operation where they're sharing data with other firms, and you think you might benefit from blockchain technology, now may be the time to start thinking about building a proof of concept. It still might be a year or two away before you're launching something into production, but build a proof of concept now, to start thinking about the right kinds of data to record in a chain – and the right kinds of smart contracts to define the transactions.

This is 1994 in the web. In '94, a lot of companies could afford to ignore the web. By '97, most could not afford to ignore the web. By '99, if you were ignoring the web, you were writing your own death certificate. We're still 5 years away from that, but now's the time to get started, if you want to be on the leading

edge. There's so much employment demand now for developers who understand blockchain technologies.

LM: You were in a kind of retirement phase; you took a break from the limelight?

BB: Sort of. I just had a career where I felt like I wanted to take a little bit of break from pure open source. I was at CollabNet and enjoying it, but ultimately, it went in a different direction from what I would have liked. I've moved on. I actually gave my boss 2 years notice and then left in 2007.

In 2007, I spent a lot of time traveling and speaking about open source. Then I worked in the 2008 Obama campaign. I was a policy advisor, thinking about open source software, but I also addressed how to combine open data and open government. After President Obama was elected, I worked at the White House on those issues. All I could do really was plant some seeds that later blossomed. I didn't want to stay in government forever. I just wanted to go long enough to have an impact, but I spent 2 years there.

While I was in government, I started a consulting relationship with an organization I've always been very close to and really liked, which was the World Economic Forum. That consulting relationship turned into a full-time job. I moved to Geneva as CTO for WEF. Open source wasn't really the focus of my job at the WEF. I was responsible for a team of 30 people. Spent half my time trying to build a social network for world leaders. The other half of my time I spent trying to get the WEF off of LotusNotes and everything in between. I did a lot to plant open-source seeds in a bunch of the operations and in the communities at the World Economic Forum. I got to see a lot about how the world works. I also got to do a lot of skiing in the French Alps.

Then, after that, I came back to San Francisco and took maybe even more of a departure. I decided to work as a venture capitalist – mainly because a friend of mine had started a fund that seemed interesting. The fund seemed different, distinct from a lot of the flash of the venture capital heyday: you know, all the random apps. It wasn't about the

early stages. It was about the later stage. Instead of being Uber for Dogs or something, our goal was to fund clearly defined technologies, such as a nuclear fusion energy plant or an energy technology company or a cure for diabetes. That was intellectually very interesting and I had fun there about 2 1/2 years, but I felt a gravitation pulling me back to open source.

LM: What brought you to the Hyperledger project?

BB: Jim Zemlin [Linux Foundation CEO], and he's been trying to hire me constantly. Then he finally got an office right next to mine.

LM: So he moved closer to you. He wanted you that badly?

BB: (Laughs) He opened an office in San Francisco in the Presidio (which is a collection of buildings near the Golden Gate Bridge), right next to where I was working at a venture capital firm. I joked with him that he did that as a recruiting tool to try to bring me in, but it worked. I joined in May of this year.

LM: Did you really feel that Hyperledger was as big and important as Apache, or did you join just to make Jim happy?

BB: (Laughs) No, no. I didn't just take it to make Jim happy. It had to be more than that. My personal motivation was that the blockchain space had some problems. There were problems in the way the communities around it were working. I felt like, if I can play a role constructively in helping with solving some of these structural issues, that would be a valuable thing. I just wanted to get back to something related to open source.

It really did go back to this feeling that we're in 1994 and this is the web. Blockchain is such a foundational technology that could affect so many things. We don't know exactly how it will. We don't know exactly to what depth. It feels likely to me that blockchain technologies will affect the way societies build their IT systems – not *company* systems so much as *society* IT systems. That's what I want to play a role in helping re-invent. ■■■

STOP SHARING!

1&1 VIRTUAL SERVER CLOUD

starting at **\$4.99** per month*



Trusted Performance.
Intel® Xeon® processors.

NEW

1&1 eliminates the "noisy neighbor effect": Ideal for beginners as a web and mail server, but also for more demanding projects like database applications, the new 1&1 Virtual Server Cloud is 100% yours to use! Take advantage now of the latest cloud technology.

- No shared resources through VMware virtualization
- Full root access
- SSD storage
- Unlimited traffic
- High performance
- Maximum security
- Best price-performance ratio
- 24/7 expert support
- Choice between Linux/Windows
- Plesk ONYX



☎ 1-844-296-2059



1and1.com



Social networking the FOSS way with Diaspora

Pod People

Diaspora is a free and open source decentralized social network consisting of millions of independently run pods. Join an existing pod or set up your own with minimal programming skills. *By Nate Drake*



Social networking and privacy hardly go together like peanut butter and jelly. Social network giants such as Facebook routinely track your location; the Chrome extension *Marauder's Map* will even plot your friend's movements based on data from Facebook Messenger [1].

Even those people who switch off location services or connect via Facebook's "dark web" address are required to register an account in their own name instead of an alias or nickname [2], which has caused trouble for people who happen to share the same name as a celebrity or have a name that doesn't follow western naming conventions.

Worse still, in the past, Facebook has faced allegations of using customers' personal data in unethical ways. This includes allegedly keeping account data even after it's been deleted, tracking users' movements across other websites, using facial recognition to tag people in photos that they didn't upload themselves, a lack of transparency about government requests for data, and even using user "Likes" in advertisements [3]. In brief, in the words of Andrew Lewis [4], "If you are not paying for it, you're not the customer; you're the product being sold."

Fortunately, Facebook is not the only social network. The Diaspora Social Network [5] has been in active development since late 2010 and describes itself as an "online social world where you are in control."

AUTHOR

Nate Drake is a freelance journalist specializing in cybersecurity and retro tech.

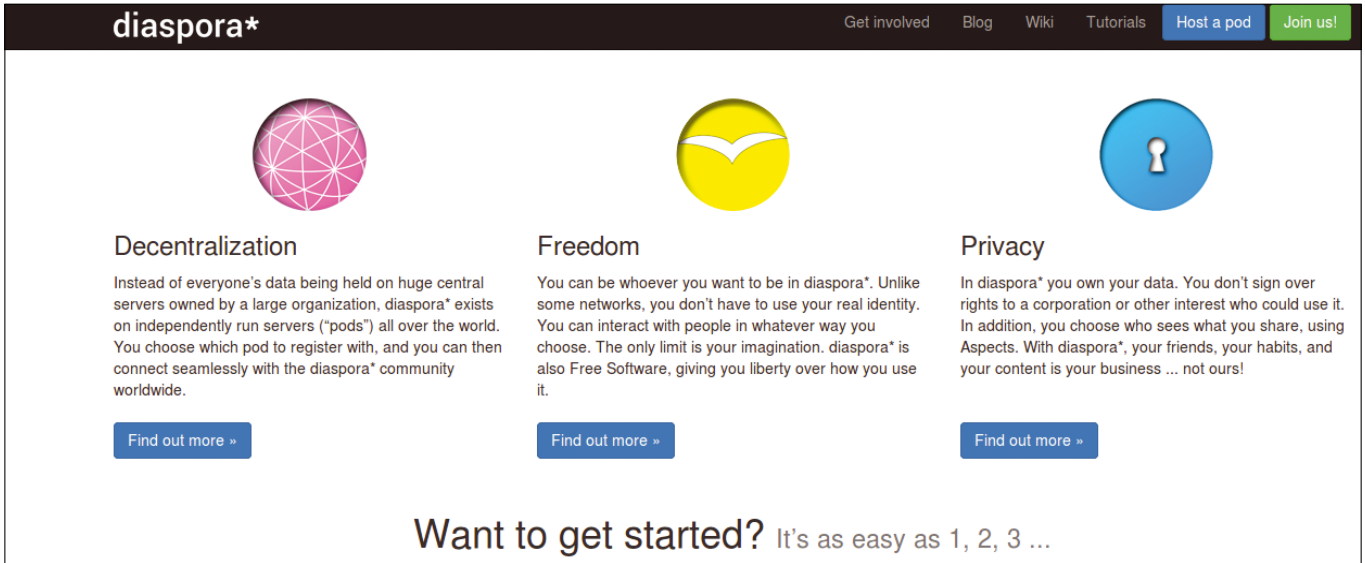


Figure 1: Diaspora emphasizes control of user data and freedom through decentralization and open source software.

The Diaspora Foundation, which is responsible for the project's source code, further elaborates that the core philosophies behind the network are decentralization, freedom, and privacy (see Figure 1).

Decentralized Diaspora

One of the most exciting features of Diaspora is that user data is not held on any one central server. The Diaspora Foundation itself is part of the Free Software Support Network (FSSN) [6] and as such is much less susceptible to hostile corporate takeovers or gag warrants from shadowy government actors.

In practical terms, decentralization is also achieved through independently run servers or pods throughout the world. Each pod has its own domain name and is managed by volunteers on different servers in different locations [7]. Pods are federated in terms of sharing data, so it doesn't matter with which

server users register. For example, if Bob registers an account at US-based `diasp.org`, he can still share content with Alice who registered at Germany-based `despora.de`.

Although the fact that the pod is run by volunteers in a country with liberal data privacy laws isn't an absolute guarantee that your content is safe, users remain in control of their data and can delete it at any time. A list of available pods complete with uptime and active users is available on Pod Uptime [8], of which the most popular is Framasphere [9]. Although the servers are based in Germany, the website itself is run by a French community of open source developers known as Framasoft.

When choosing a pod on Pod Uptime, it's advisable to try to find a location without any Orwellian surveillance laws but also to keep an eye on the software version they are running. At the time of writing, the current version is `v0.6.99.0`. Not all pods are open for registration of new users. Anyone who has

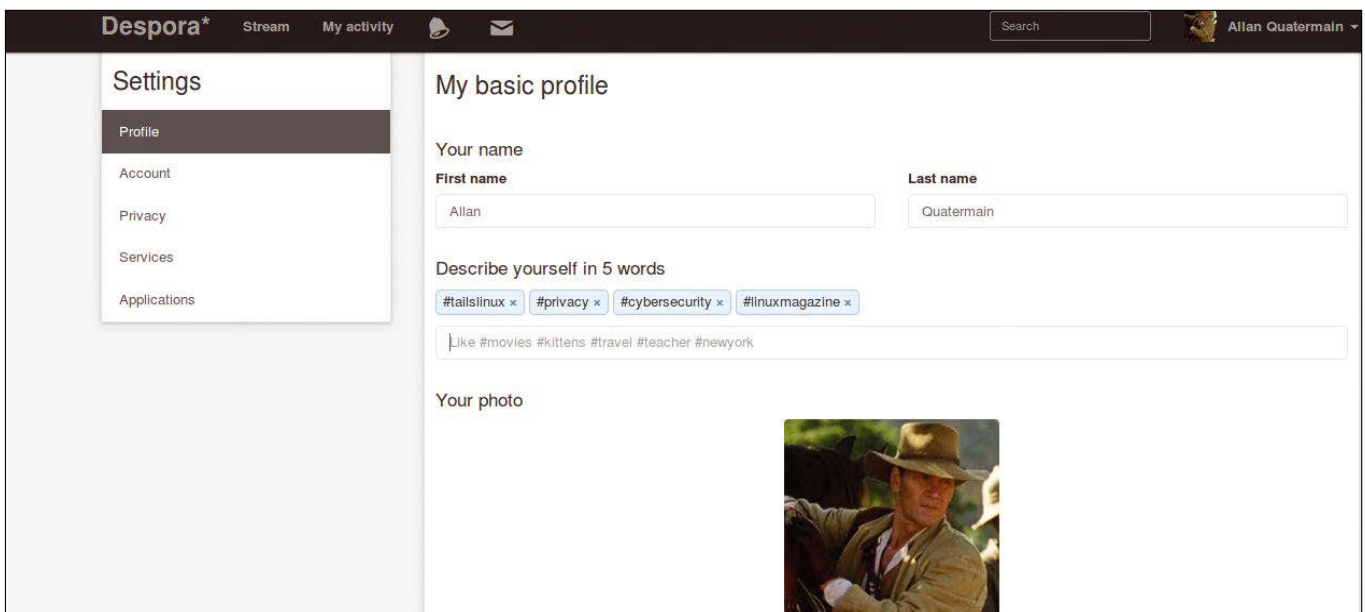


Figure 2: The author's Diaspora account in the name of the fictional character Allan Quatermain. Users can add hashtags to their profile to allow others with similar interests to find them, such as `#linuxmagazine`.

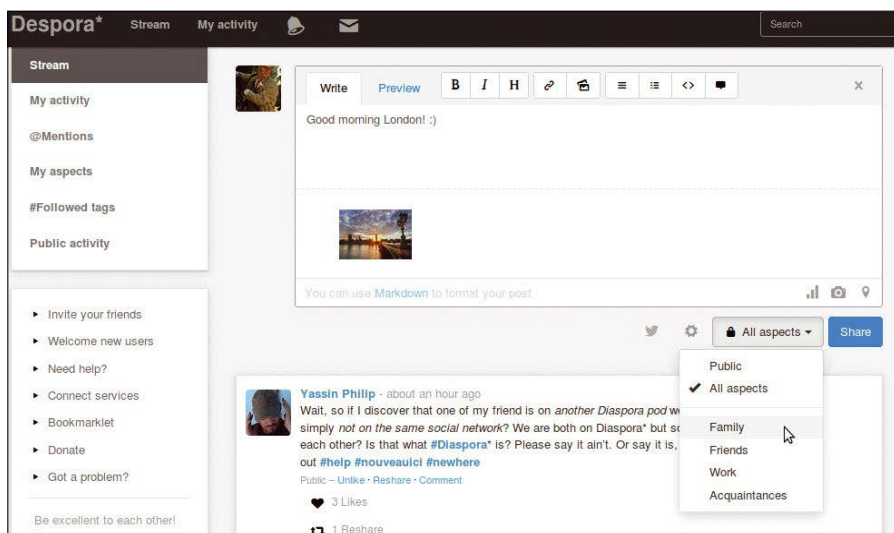


Figure 3: By default, posts are visible to all of a user's aspects. This can be changed to a public post or one or more aspects as needed.

trouble choosing a pod can simply click the large red button at the top, and the website will choose one automatically.

Diaspora Freedom

Aside from using free and open source software, Diaspora also protects user's privacy by allowing them to choose an alias or nickname. The only limit is one's own imagination.

The limits of this freedom do not extend only to using an imaginary name or username. Users are free to follow topics of interest through use of hashtags to see what famous people and companies have to say without alerting others. Users can also include tags in their "public profile" to allow others with similar interests to follow them [10] (see Figure 2).

As with Facebook, some of the data on users' profiles is public and some is private. Users who have specifically chosen to share information with one another can see their "extended profiles" (i.e., biography, gender, location, and birthday). Unlike with other social networks, users do not have to fill in this information. Public profiles are visible to any visitors to the website and simply consist of a user's chosen name and profile photo, as well as any posts they've specifically chosen to make public. These settings can be changed from the account page so that everyone can see a user's extended profile.

Diaspora Privacy

Diaspora's Privacy features go much further than masking profiles. The chief difference between Diaspora and other major social networks like Facebook is that there is no advertising and therefore no money to be made in analyzing your browsing activities. The source code can be examined to verify that it's used for no other purposes than connecting and sharing data with users' family and friends. Only admins of pods with which you interact can view user data [11]. Privacy can be further ensured by hosting your own pod (see below).

By default, searches for other users are accomplished by searching for user@pod (e.g., macumazan@despora.de) or, if two people are using the same pod, simply by searching for their username. Each user remains in control of whether their

profile can be found by other users through the search bar, resulting in much greater anonymity [12].

Sophisticated control of data is also possible through the use of aspects. Aspects are simply the way contacts are grouped in Diaspora. By default, the categories are Family, Friends, Work, and Acquaintances. Sharing is asymmetrical, meaning that by default people cannot see others in their own aspect, nor does it follow that because a user adds a person to their aspect that the reverse is true (Figure 3).

This level of control is markedly different from Facebook and is best illustrated by an example. Imagine Diaspora user Allan phones in sick at the office to spend the weekend in London with his wife, Marie, and his son, Harry. To avoid

being exposed for malingering, he can choose to share his holiday photos with only people within his Family aspect rather than with his colleagues. Marie, in turn, may have her own Family aspect, which includes Allan, Harry, and her sister, Stella. If she shares her holiday photos, these will be visible to her sister as well as her husband and son.

Cross-Posting

Given the huge market share of social network titans like Twitter and Facebook, it may be difficult to force the masses to join those who wish to make the switch. Mercifully, it's possible to register with Diaspora pods that support cross-posting to major platforms such as Twitter, Tumblr, Facebook, and WordPress (Figure 4).

The specific setup steps are available on Diaspora's Wiki, but in simplest terms this involves authorizing an app similar to a Twitter or Facebook bot to post on a user's behalf. This should mean each time that new content is shared on Diaspora it will also be posted by the corresponding app. The privacy of posts



Figure 4: The despora.de pod supports cross-posting to Twitter. Note the app can also read tweets on users' timelines and track new followers.

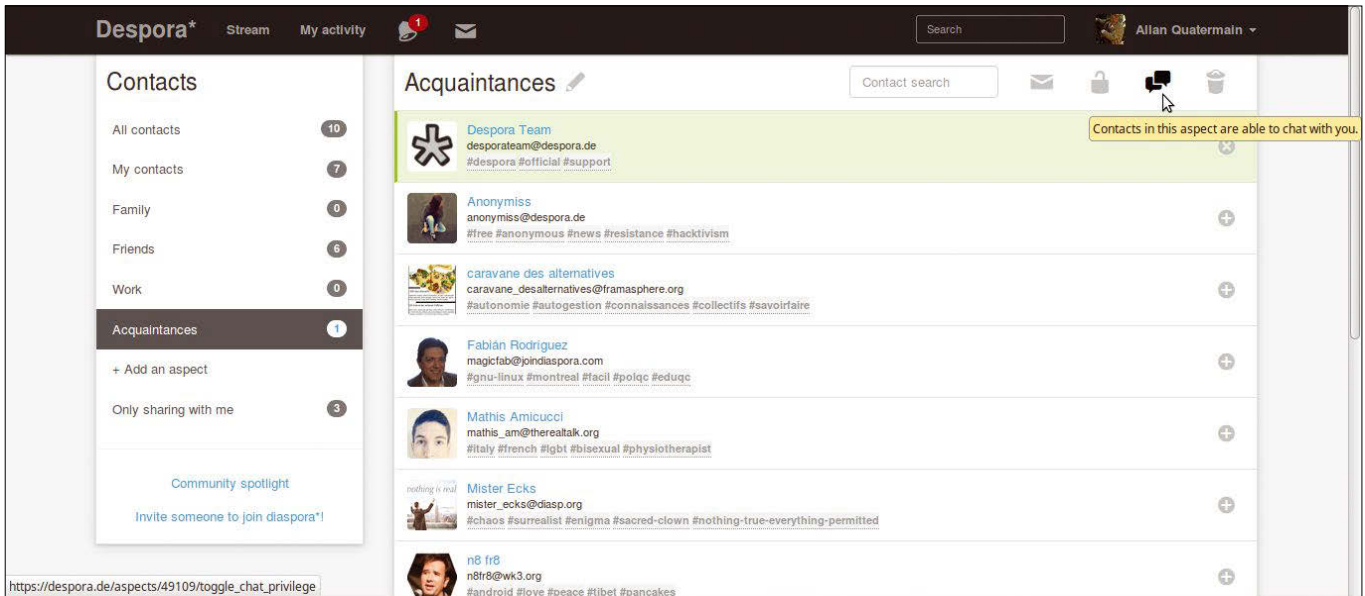


Figure 5: Click the chat icon at the top right to allow chatting with people within each aspect. For acquaintances, this is switched off by default.

is determined when initially setting up the app. For example, all posts may be visible only to the account's Facebook friends.

Setting Up a Diaspora Pod

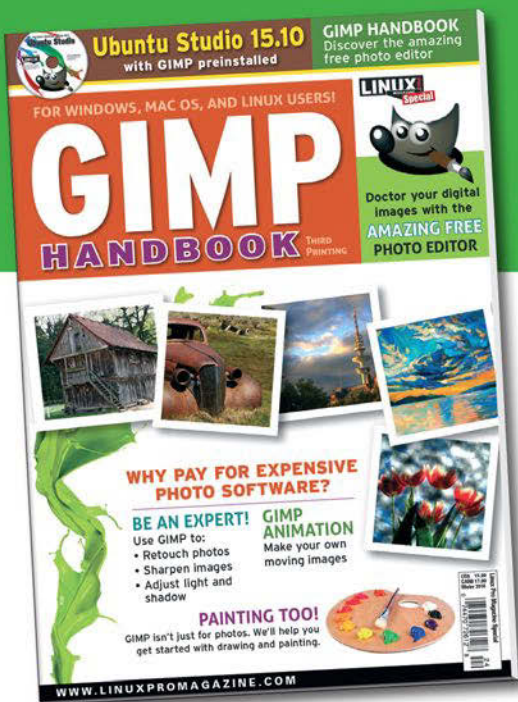
For readers whose appetites have been sufficiently whetted, simple setup guides for running one's own Diaspora pod are available on the Foundation's website. Debian-based

distros such as Ubuntu are supported, as well as CentOS, Gentoo, Arch, and Hamara Linux [13].

Whichever distribution is employed, the hardware requirements aren't very onerous. The website recommends at least 1.5GB RAM, 1GB swap space, and a decent multicore CPU. The amount of HDD space required will vary naturally depending on the number of users and the content shared.

Shop the Shop

shop.linuxnewmedia.com



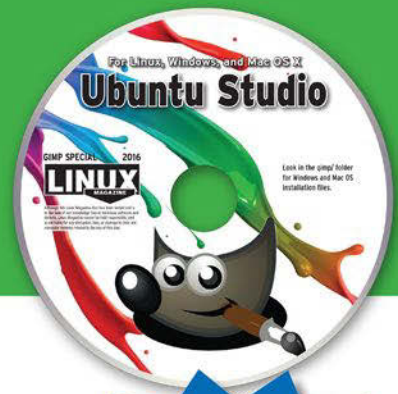
GIMP HANDBOOK

SURE YOU KNOW LINUX...

but do you know GIMP?

- Fix your digital photos
- Create animations
- Build posters, signs, and logos

Order now and become an expert in one of the most important and practical open source tools!



Order online: shop.linuxnewmedia.com/specials



FOR WINDOWS, MAC OS, AND LINUX USERS!



For Raspberry Pi lovers, the Foundation says Diaspora will run on the Raspberry Pi Model 2 and 3 but is very slow and so not recommended for multiple users [14]. Diaspora requires several different components, so the Foundation recommends using a VPS or dedicated server over shared hosting servers.

In terms of software, Diaspora is written in Ruby. The Foundation recommends using Ruby Version Manager (RVM) to keep Ruby up to date and also to separate it from any other Ruby apps you may have installed. Diaspora is compatible with most app and web servers; however, the most common setup is using Unicorn for the app server and Nginx as the web server.

Users who are serious about running their own Diaspora pod will need an SSL certificate both to encrypt traffic between clients and the browser and to allow federation with other pods. Self-signed certificates or certificates issued by CA Cert won't work. The Foundation recommends obtaining a free TLS Certificate from providers such as Let's Encrypt, WoSign, or StartSSL [15].

Diaspora Drawbacks

Setting up a pod should pose no trouble for anyone with experience or server administration or Rails app deployment; however, there are support channels on Freenode in various languages for anyone who has difficulties [16].

As previously discussed, the admin of the pod has access to all registered users' data. Similarly, content shared by users with people registered to other pods will be visible to those "podmins" as well. There's also no way to isolate a pod from others without modifying the source code, so caution should still be exercised when posting content.

The decentralized nature of Diaspora can be abused, as it was in 2014 when ISIS supporters decided to join Diaspora after being kicked off Twitter. Although in technical terms the Diaspora Foundation could do nothing to stop people from hosting their own pods, they did urge the community

to report any offensive posts and worked with admins of the largest Diaspora pods to make sure extremists' accounts were deleted [17].

Podmins headaches also do not end with terrorism. Admins are specifically forbidden from integrating advertisements into a pod, meaning that each must be run as a labor of love, not for profit. There's also no automatic way to run backups of a server. The Diaspora Foundation recommends making copies of the public/uploads directory to back up images and then googling how to perform a dump of "Diaspora_databases" from the database server, which is less than elegant [18]. Users have also complained that downloading profile data is done via a JSON file, which is only machine readable, meaning users need to upload to a different pod to review what they uploaded [19].

By default, Diaspora doesn't come with an integrated chat feature as is available in Facebook, but the website's wiki does contain instructions to set this up on a pod via Prosody. The chat feature also has to be enabled by users for each specific aspect (Figure 5). Sadly, use of Aspects means that it is currently impossible to create a group for people with similar interests as can be done easily on Facebook [20].

Quality over Quantity

A final word on this social network is that, beyond the code, Diaspora users in the nature of things tend to be more technically competent and privacy conscious. This means that there is a distinct lack of pictures of small animals, encouragements to copy/paste status updates, or dubious videos of celebrities.

Older users may remember the early days of the Internet when a certain degree of effort and knowledge was mandatory to get online. All that extra effort meant people didn't go to the trouble to post unless they had something important to say. This is also an important rejoinder to critics who may compare Diaspora's relatively miniscule user base of 1 million users to Facebook's one billion. Quality trumps quantity every time. ■■■

INFO

- [1] Marauder's Map: <https://github.com/arank/marauders-map>
- [2] What names are allowed on Facebook?: https://www.facebook.com/help/112146705538576?helpref=faq_content
- [3] 7 Controversial ways Facebook has used your data: <http://time.com/46957-controversial-ways-facebook-has-used-your-data/>
- [4] Andrew Lewis on Twitter: <https://twitter.com/andrewlewis/status/24380177712>
- [5] Diaspora Foundation: <https://diasporafoundation.org/>
- [6] FSSN: <http://freesoftware-support.org/>
- [7] Diaspora Wiki: https://wiki.diasporafoundation.org/Choosing_a_pod#Why_do_I_need_to_choose_a_pod.3F
- [8] Pod Uptime: <https://podupti.me/?advancedview=true>
- [9] Framasphere: <https://framaspHERE.org/>
- [10] Diaspora Wiki: https://wiki.diasporafoundation.org/FAQ_for_users#What_do_the_tags_on_my_public_profile_do.3F
- [11] Diaspora Foundation: <https://diasporafoundation.org/about#privacy>
- [12] Diaspora Wiki: https://wiki.diasporafoundation.org/FAQ_for_users#How_do_I_use_the_search_box_to_find_particular_individuals.3F
- [13] Diaspora Wiki <https://wiki.diasporafoundation.org/Installation>
- [14] Diaspora Foundation – Installation <https://wiki.diasporafoundation.org/Installation/Debian/Jessie>
- [15] Diaspora Foundation – Installation <https://wiki.diasporafoundation.org/Installation/Debian/Jessie>
- [16] Diaspora Wiki – How we communicate https://wiki.diasporafoundation.org/How_we_communicate#IRC
- [17] Islamic State fighters on Diaspora <https://blog.diasporafoundation.org/4-islamic-state-fighters-on-diaspora>
- [18] Diaspora Foundation – FAQ for Pod Maintainers: https://wiki.diasporafoundation.org/FAQ_for_pod_maintainers#Can_I_add_advertisements_to_my_pod.3F
- [19] GitHub: <https://github.com/diaspora/diaspora/issues/7147>
- [20] Diaspora Wiki – Integration/Chat: <https://wiki.diasporafoundation.org/Integration/Chat#Configuration>

Drupal 6 Long Term Support from Tag1 Quo*

Tag1 Quo Security Alerts

- Enjoy peace of mind at a low monthly fee from the experts you trust
- Covers all Drupal 6 Core and Contrib modules
- Centralized security reporting for all your sites
- Host anywhere and everywhere!

tag1 QUO

Free trial with coupon:
LINUX

To sign up visit
tag1quo.com/linux



Exploring the Windows-like Linux distros ChaletOS and Zorin OS

Quick Switch

ChaletOS and Zorin OS are two Ubuntu derivatives designed for users who are migrating from Windows. *By Erik Bärwaldt*

Microsoft's massive effort to force Windows 10 on users is causing a whole new round of migration to Linux. If these Windows users decide to make the switch, the first distribution they encounter might be a high-profile desktop system like Ubuntu and Mint. But behind the headlines, the Linux environment is teeming with choice. More than 200 Linux derivatives vie for users' affections, and if you aren't getting what you want, you can always try another Linux.

In an effort to cover the complete Linux experience, we're always ready to explore any inventive Linux alternative. Many Linux distros are created to address a specific need. The question of how to make Linux look like Windows is particularly interesting – and important, if the goal is to attract more Windows users to the open source community. I tested two Ubuntu derivatives, ChaletOS [1] and Zorin OS [2], that are designed to facilitate easy migration from Windows.

ChaletOS

ChaletOS, which was created in Serbia, is now available in its fourth major version. The current version is based on Ubuntu 16.04 and Debian 8, inheriting the 16.04 long-term support that lasts until April, 2021.

Although ChaletOS uses the slim Xfce for its desktop, the system looks quite a bit like a clone of current Windows versions (Figure 1). However, the developers allow you free choice over the appearance of the desktop, so you can customize the appear-

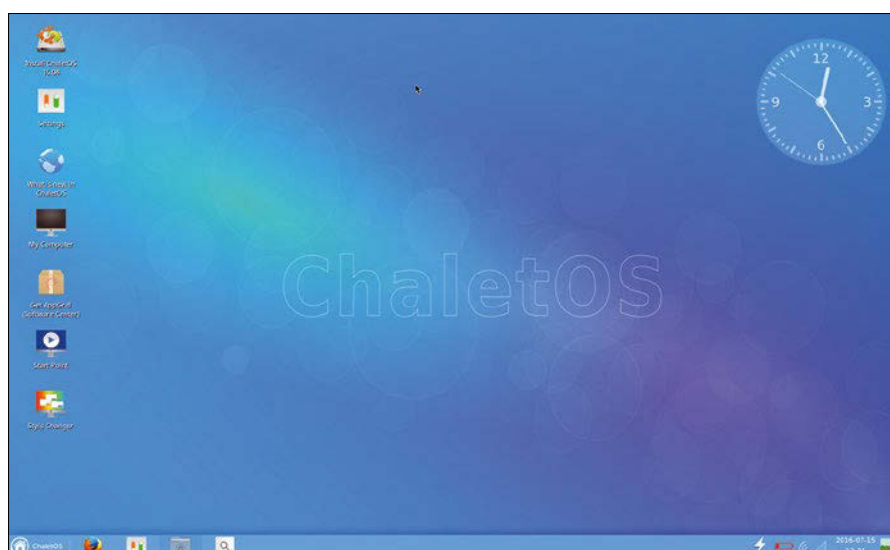


Figure 1: Xfce doing it differently: The ChaletOS desktop operates like a clone of the current Windows versions.



cal capabilities you'd expect from a desktop Linux.

Firefox and Gimp are present at first boot, although you won't find LibreOffice in the default configuration.

The VLC video player and Audacious music player are among the multimedia applications preinstalled in the distribution.

You can use Brasero to burn CDs and DVDs, and you will not need to load any special codec files manually, because VLC already provides the key codecs.

Alongside the usual set of desktop applications, ChaletOS supplies some of its own flourishes. The *Great Little Book Shelf* and *Great Little Radio Player* are both the work of Dejan Petrovic, the main developer of ChaletOS. The radio player, which you will find in *Application Center | Multimedia*, plays radio broadcasts from the Internet, and the *Great Little Bookshelf* in *Application Center | Office* helps sort PDF documents. The tool displays individual PDF files, with a small image of the front page in list view.

You can manage packages with the well-known Synaptic package manager. ChaletOS also supports other package

ance of the work environment to suit your wishes in the operating system's *Style Changer*. ChaletOS offers several dozen pre-made styles (Figure 2).

Like almost all Ubuntu derivatives, ChaletOS lets you test the system by booting from a DVD into Live mode. If you opt to install, click on the *Install ChaletOS 16.04* desktop icon from a Live session or install from the boot manager at boot time. ChaletOS comes in both 32- and 64-bit variants.

With its agile Xfce desktop, ChaletOS is a good option for older computers that don't have enough memory to run big desktops like Gnome, KDE, or Unity. Although the lightweight Xfce offers fewer options than Gnome and KDE, the ChaletOS developers have taken pains to simplify configuration. In the *Style Changer* dialog, you modify the desktop appearance using templates, and you configure the Conky system monitor with the click of a mouse. For changes configured through the *Style Changer*, such as hardware management tasks, you can still use the conventional Xfce *Settings* menu.

Software Inventory

ChaletOS keeps the number of pre-installed programs within manageable limits, although it does provide all the typi-

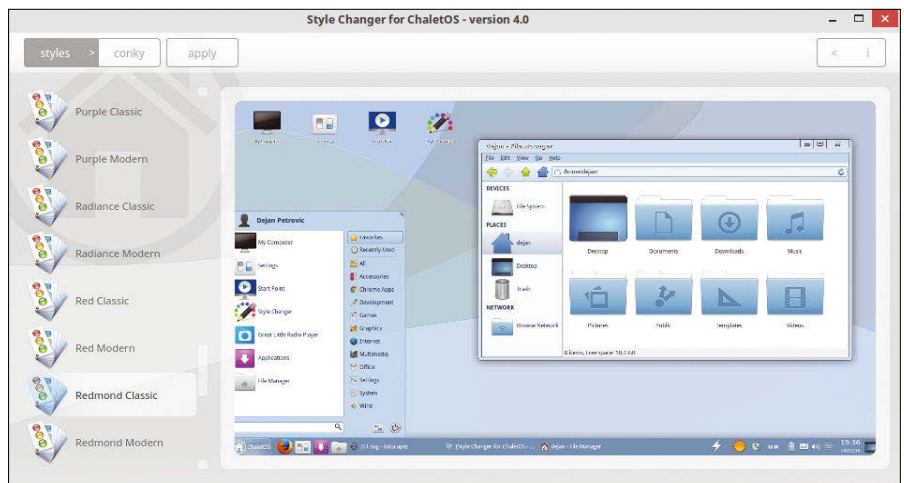


Figure 2: ChaletOS is very versatile thanks to the Style Changer.

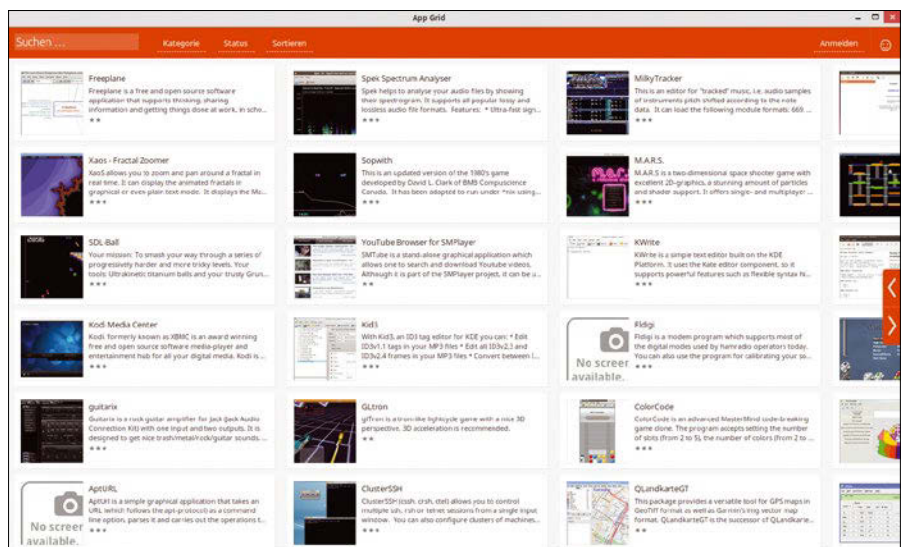


Figure 3: The program management tool App Grid provides an attractive alternative to the slightly dusty Synaptic.

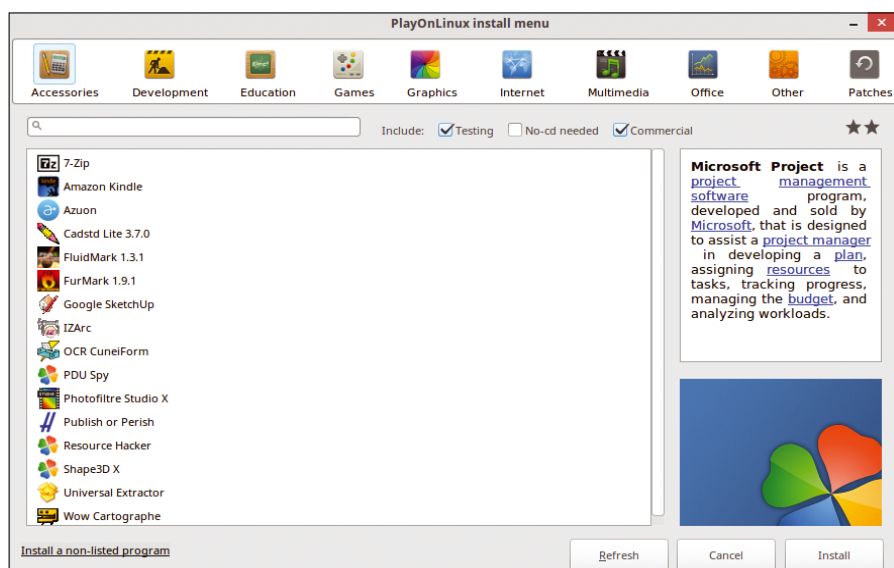


Figure 4: The PlayOnLinux front end lets you install Windows software on Linux computers by simply pressing a button. Wine works as a back end.

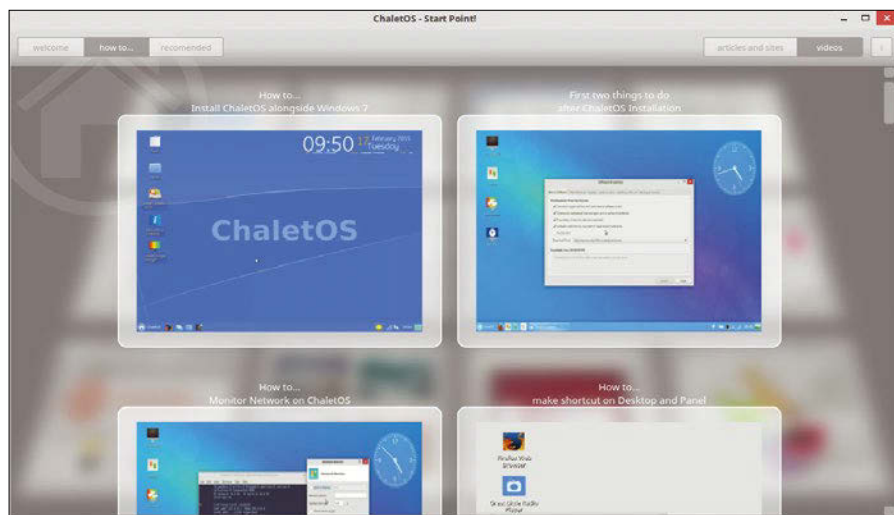


Figure 5: As an alternative to text help files, ChaletOS videos easily orient new users to the system.

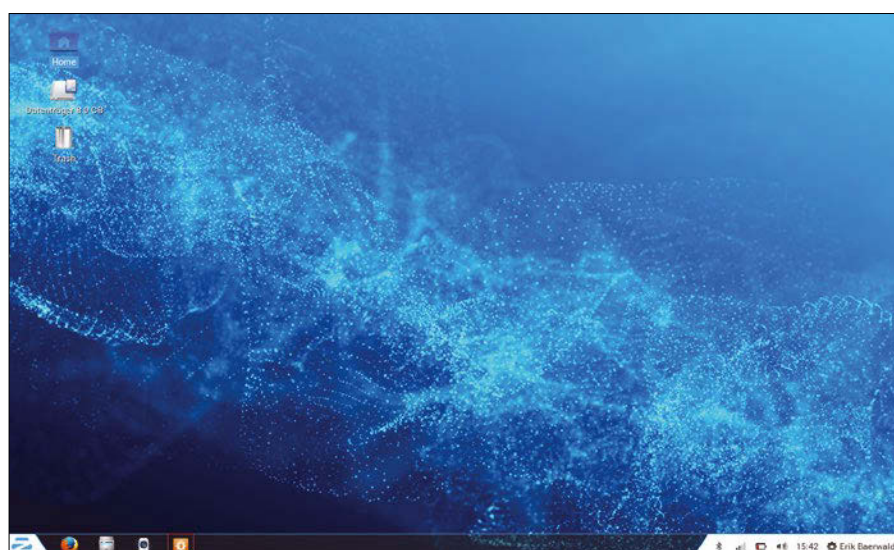


Figure 6: Zorin OS boots into a Gnome 3 desktop with some 3D effects activated, although they do not noticeably affect performance.

management programs, such as Ubuntu Software Center, which is no longer maintained by Ubuntu. You can also try out the App Grid package tool, which is intended as an alternative to the Software Center for Ubuntu 16.04 and later systems. Click the *Get AppGrid* icon on the desktop to install App Grid (Figure 3), which will then appear in the *System* menu. Like Ubuntu Software Center, App Grid lists available applications with thumbnails, which you can install with a simple click.

Windows Programs

Some games and other Windows programs have no Linux equivalent. ChaletOS provides the Wine run-time environment, so users can run their Windows programs on Linux.

Keep in mind that Wine does not emulate a complete Windows system, meaning that not all Windows programs will run on it. Apps that require a specific proprietary multimedia technology, for instance, do not fare well with Wine. To determine whether Wine can cope with specific Windows software, right-click on the file in the Thunar file manager and select *Open in Wine Windows Program Loader* from the pop-up menu.

The first time the Wine environment is called up, the automatic initial setup begins, which reloads Mono and Gecko packages from the network and integrates them into the Wine run time. Then, the Windows software's setup routine starts. If the setup routine does not work as expected, the probable cause is an incompatibility preventing the software from working on Wine.

An easier way to manage Windows applications in ChaletOS is with PlayOnLinux (*Games | PlayOnLinux*). PlayOnLinux [3] lists Windows programs sorted by category in a graphical Wine front end. Most of the options are known programs that have already been successfully tested on Linux with the assistance of Wine. PlayOnLinux sometimes loads the requested Windows applications directly from the Internet (Figure 4), but in some cases, you'll need to provide the application yourself.

Guides

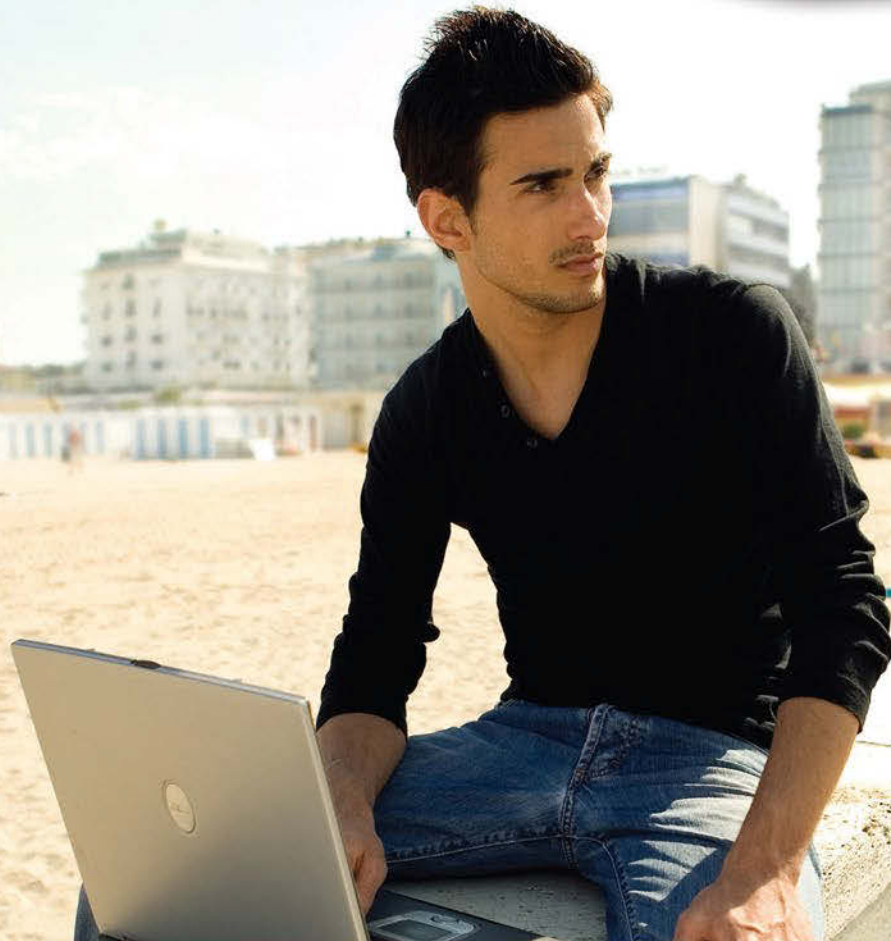
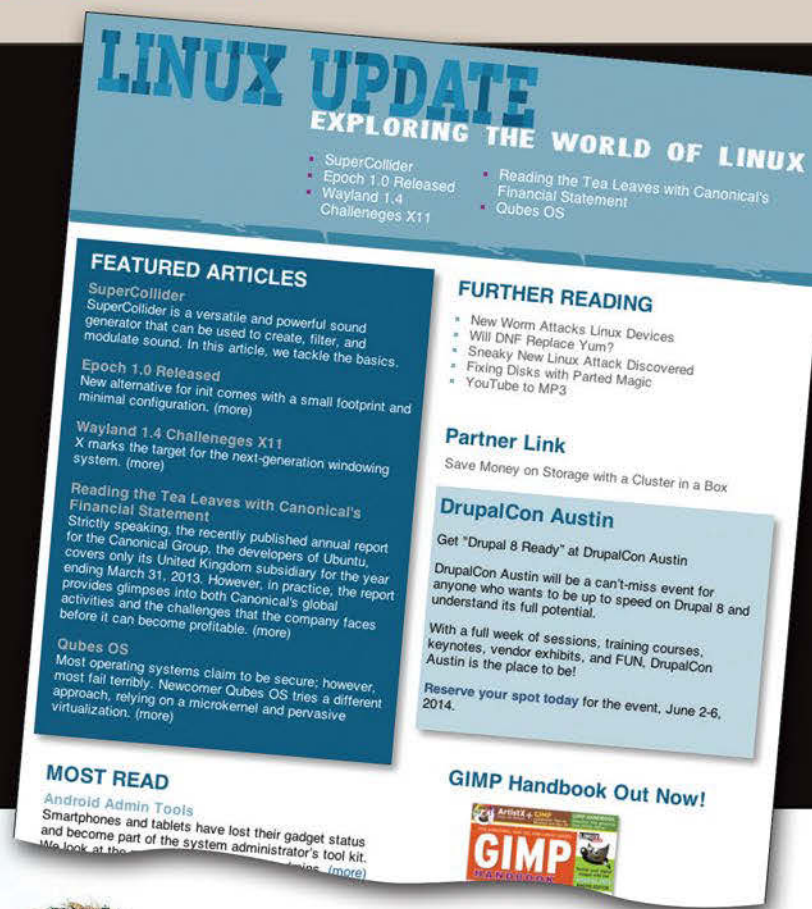
ChaletOS is breaking new ground with documentation. Click the *Start Point* icon

LINUX UPDATE

Need more Linux?

Our free Linux Update newsletter delivers insightful articles and tech tips to your mailbox twice a month. You'll discover:

- Original articles on real-world Linux
- Linux news
- Tips on Bash scripting and other advanced techniques
- Discounts and special offers available only to newsletter subscribers



www.linuxpromagazine.com/mc/subscribe

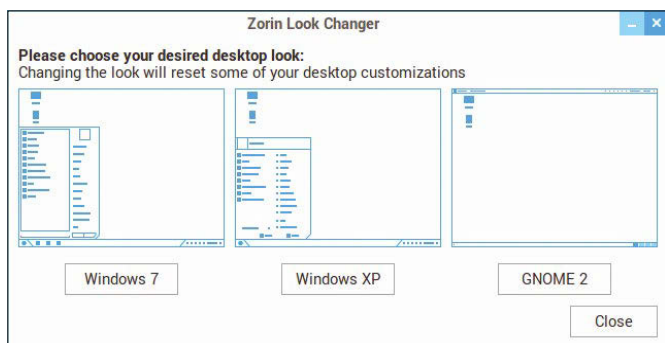


Figure 7: Adapt the desktop layout as you please in the Zorin Look Changer dialog.

to call up an attractive application that displays the most important aspects of the system in video sequences. ChaletOS also offers guides depicting how some essential PC technologies function (Figure 5).

The ChaletOS project offers forums where users exchange information [4], and you can visit the project's Google+ channel [5], which offers video help along with discussions.

Zorin OS

The Zorin OS system from Ireland is also well suited for defectors from Windows. Zorin OS is available for download in different versions. The Core, Lite, Educational, and Educational Lite versions are free; Business and Ultimate editions are available for around EUR10. Most versions come in 32- and 64-bit variants. At around 1.4GB, the 64-bit core version 9 is surprisingly compact. Zorin OS, like ChaletOS, is based on Ubuntu, although it is based on version 14.04 from 2014. Canonical provides long-term support for Ubuntu 14.04 until April 2019.

In spite of the name, the Core edition is a full operating system, equipped with plenty of applications. The system is tailored for the desktop user and offers several customization options. The developers made efforts to keep Zorin OS as lean as possible to ensure it runs smoothly, including on older computers – the Lite version with an LXDE desktop is best suited for low-end hardware.

The operating system starts as a Live system, unless you launch Ubuntu's Ubiquity installer during the boot process. If you choose Live operation, the system launches into an elegant and modern Gnome 3 desktop with some 3D effects activated (Figure 6). The 3D effects did not result in a noticeable loss of speed.

Zorin OS makes it easy for newcomers from other systems to maintain a familiar working environment. You can select the desktop layout by choosing between the (preset) Windows 7 look and a Windows XP-based or Gnome 2-based appearance in *System Tools | Look Changer*.

Use the Zorin Look Changer dialog (Figure 7) to modify the menu structure and the arrangement of the panel strips. *Zorin Theme Changer* (also under *System Tools*) modifies the appearance of windows. Along with the conventional variants, you

will also find options with high-contrast, dark background colors, facilitating good readability on notebook displays.

Software

Zorin OS differs from ChaletOS in that it brings with it all the main applications out of the box. The default software inventory includes LibreOffice, Firefox, Gimp, and the Thunderbird email client. Firefox comes preinstalled, and you can change the browser to Chrome, Opera, or the slim Midori. To switch the browser, go to *Internet | Zorin Web Browser Manager*.

Zorin OS provides access to common repositories for installing new software. In addition to Synaptic, you'll find a visually adapted version of the Ubuntu Software Center, which is simply called *Software Center*. The Zorin OS software inventory is a little smaller than the ChaletOS inventory, because the Ubuntu 14.04 variant is missing some of the packages included in the 16.04 edition.

Updates

You are advised to bring the system up to date after installation, because Ubuntu 14.04 is already quite out of date. Just click *System Tools | Administration | Software Updater*. The system searches for updated packages in the repositories and suggests these packages for installation. You can launch the update by clicking *Install Now* (Figure 8).

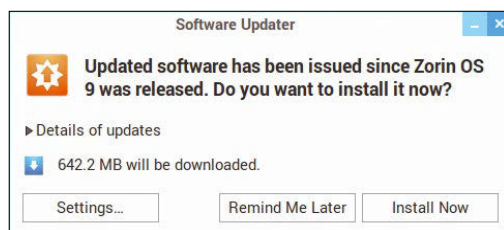


Figure 8: Thanks to update management, you can keep the system up to date with a few clicks of your mouse.

Windows Software

Like ChaletOS, Zorin OS offers the Wine run-time environment for Windows programs, integrated by default.

It also provides the Gecko and Mono add-on packages, so you do not need to download them manually. Additionally, Zorin OS has PlayOnLinux at its disposal as a graphical front end for Wine.

As with ChaletOS, complex software packages still overwhelm the Wine environment. Various reference works function smoothly, but Wine was stumped by Encyclopaedia Britannica, which as an intricately designed interface filled with a large array of multimedia content.

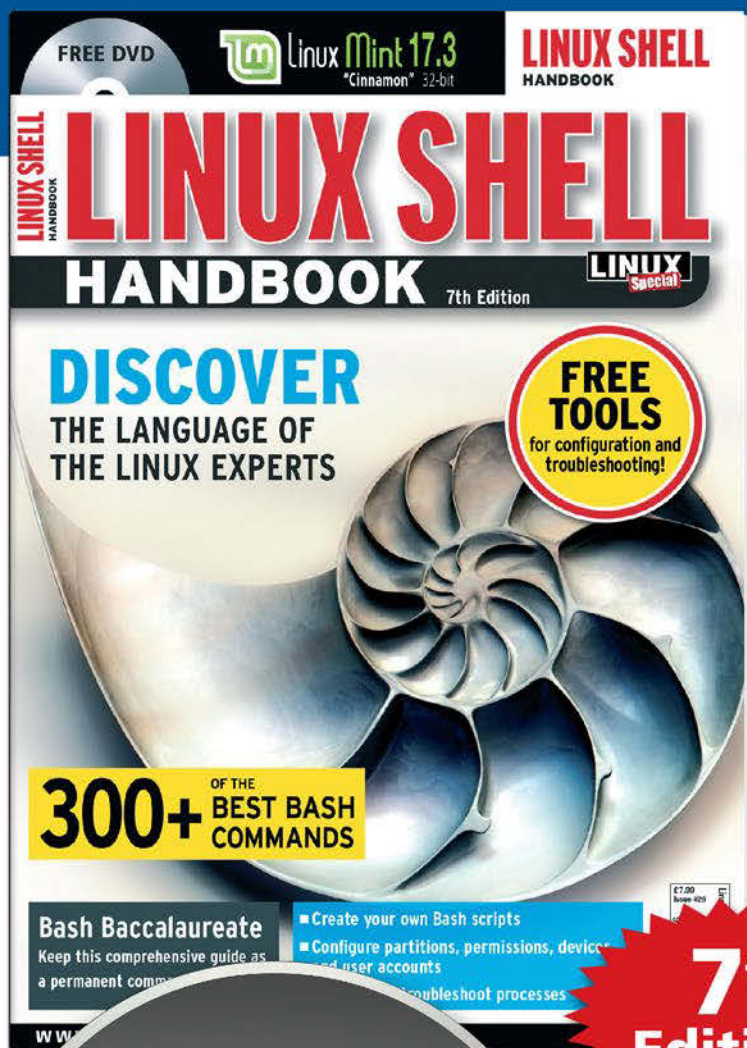
Conclusion

ChaletOS and Zorin OS demonstrate the versatility of Linux. Both systems make emigrating from Windows easy, and in many cases, the pre-installed Wine compatibility software allows you to use the Windows programs you still need. ChaletOS and Zorin OS let Windows defectors get used to Linux without a lot of special training. ■■■

INFO

- [1] ChaletOS: <https://sites.google.com/site/chaletoslinux/home>
- [2] Zorin OS: <http://ZorinOS.com>
- [3] PlayOnLinux: <https://www.playonlinux.com/en/>
- [4] ChaletOS forum: <http://chaletos.info>
- [5] ChaletOS on Google+: <https://plus.google.com/communities/108630943169932456503>

EXPERT TOUCH



Linux professionals stay productive at the Bash command line – and you can too!

The Linux Shell special edition provides hands-on, how-to discussions of more than 300 command-line utilities for networking, troubleshooting, configuring, and managing Linux systems. Let this comprehensive reference be your guide for building a deeper understanding of the Linux shell environment.

You'll learn how to:

- Filter and isolate text
- Install software from the command line
- Monitor and manage processes
- Configure devices, disks, filesystems, and user accounts
- Troubleshoot network connections
- Schedule recurring tasks
- Create simple Bash scripts to save time and extend your environment

The best way to stay in touch with your system is through the fast, versatile, and powerful Bash shell. Keep this handy command reference close to your desk, and learn to work like the experts.

7th Edition!



FREE DVD INSIDE!

Linux Mint 17.3
"Cinnamon" 32-bit

ORDER ONLINE:

shop.linuxnewmedia.com/specials

Five screencast programs tested

Action!



To make audiovisual content easy to understand, you can use screencast programs, of which Linux has a few. *By Erik Bärwaldt*

Multimedia learning has long been established in companies and government agencies. Screencasts, which reproduce the screen content of computer systems as video sequences, along with spoken instructions from the tutor, are an important tool. Whether they are simple operating procedures in word processors or complex commands in a higher programming language, screencasts can contribute to the understanding of interrelations in information technology.

Problematic

In the Linux world, many free projects have engaged in screencast software development. However, some obstacles make it difficult to use: For example, Linux uses two sound architectures – ALSA and OSS – and no fewer than three

current sound servers in the form of ALSA, PulseAudio, and Jack. Two different multimedia frameworks compete for the task of handling recording, playback, and encoding/decoding of audio and video data: GStreamer and FFmpeg. Add to that an almost unmanageable number of video and audio codecs. Because some screencast programs also depend on specific graphical component libraries like GTK+ or Qt, they will not integrate seamlessly with just any old desktop environment.

Apart from these differences, before you can use a screen recorder, you also need to note that the functionality of individual applications differs; this means that not all programs may be fit for the intended purpose: Some screencast applications lack the ability to record content from windows, which makes it im-

possible to use them on virtual machines. Others cannot grab sections, and some can only record video sequences, but not audio content.

For this article, I only looked at those programs that can handle sound and video recordings and allow them to be used universally, which includes capturing windows. Another criterion is active development of the project.

The Candidates

The test candidates were the fairly basic Kazam, the old hand recordMyDesktop, the comprehensive new development SimpleScreenRecorder, and the intuitive Vokoscreen. Krut, published under the GPL, was also tested as the only Java program in the field.

In addition to the feature scope (Table 1), the test attaches importance

TABLE 1: Function Overview

	Kazam	recordMyDesktop	SimpleScreenRecorder	Vokoscreen	Krut
Full-screen recording	Yes	Yes	Yes	Yes	Restricted
Window recording	Yes	Yes	Yes	Yes	Restricted
Section recording	Yes	Yes	Yes	Yes	Yes
Magnification	No	No	No	Yes	No
Audio disable	Yes	Yes	Yes	Yes	Yes
Selectable codecs	Yes	Yes	Yes	Yes	No
Selectable formats	Yes	Yes	Yes	Yes	No
Webcam support	No	No	No	Yes	No
Timer	No	No	No	No	Yes
Countdown	Yes	No	No	Yes	No
Mouse pointer disable	Yes	Yes	Yes	Yes	Yes
Preview window	No	Yes	Yes	No	Yes

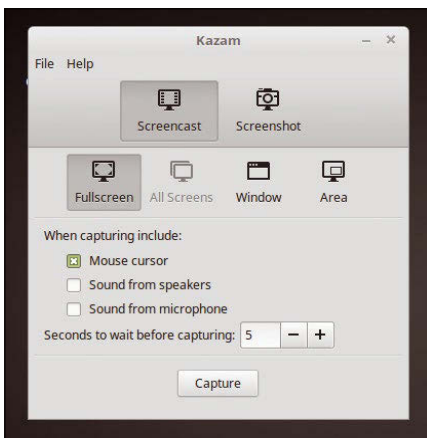


Figure 1: The Kazam interface is immediately usable without training.

to seamless desktop integration and a mature operating concept. I also wanted the programs to use popular codecs – after all, you want the movie sequences you generate to be playable anywhere.

Kazam

The Kazam [1] screen recorder has been under development since 2010; it is based on the GStreamer framework and supports PulseAudio. Kazam comes with a few dependencies and is maintained in the repositories of all the major distributions, which makes installation easy. The software is frugal when it comes to the desktop environment: KDE, Gnome, Unity, and less common environments such as Cinnamon are supported.

After installing with the package manager, you will find a launcher in the Multimedia menu (sometimes called Entertainment, Sound & Video, or something similar). Clicking on the program name displays a simple, intuitive program window (Figure 1).

You can only control the use of loudspeakers and microphones and define a delay until the screen recorder starts. Two buttons in the upper section of the program window let you define whether to use Kazam for screencasts or screenshots.

Open *File | Preferences* to configure the software. The simple options in the *Screencast* tab lets you configure the frame rate and output format (Figure 2). Although *VP8 (WEBM)* and *H264 (MP4)* are available, bear in mind that not all current media players can handle the *VP8 (WEBM)* standard. The storage location for the video can also be defined in the Preferences dialog.

The two video formats make Kazam universally deployable: The *VP8 (WEBM)* codec delivers good image quality at lower resolutions and is suitable for web applications or mobile devices. The *H.264 MP4* standard is more likely a solution for professional movies.

After starting the recording, the software shows a camera icon in the system tray of your desktop. Clicking on it opens a context menu in which you can stop or pause the recording. When you stop recording, the program window appears on the desktop. In the Settings dialog, Kazam offers to automatically save the recorded video sequences in a freely definable directory. If this option is enabled, you need not worry about saving.

Results

In my test, Kazam delivered stunning image qual-

ity. As with all screencast programs, it is advisable before the first recording to adjust the volume to avoid feedback or a sound level that is too loud. The screencasts showed significant differences in terms of storage requirements in the *WEBM* and *MP4* formats, despite using the same frame rate and resolution. Whereas a screencast recorded in *HD* quality in the *WEBM* format, including the audio recording, occupied about 5MB storage space, a recording of equal length in the *MP4* container format needed around 10MB.

Annoyingly, when recording in the *VP8 (WEBM)* format, the audio signals caused problems. Short blackouts occurred at intervals of a few seconds; this did not happen in the *MP4* format. You should perform test runs of professional desktop recordings to identify possible malfunctions as soon as possible.

recordMyDesktop

RecordMyDesktop [2] is one of senior screencast programs and is available from the software repositories of many Linux distributions. The installer creates a launcher in the Multimedia

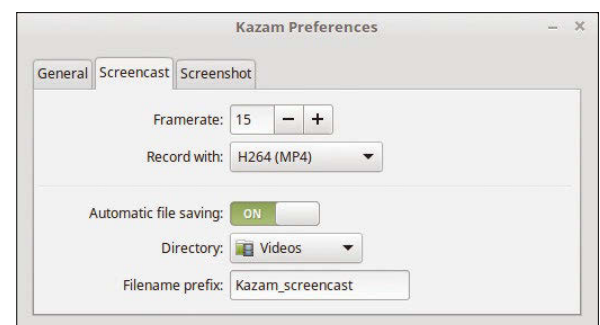


Figure 2: Kazam's configuration dialogs are straightforward and free of ballast.

menu. In addition to the program package, you will want to install the associated binary package for Qt or GTK+ based desktops to have a user interface that integrates seamlessly with the desktop environment.

The software then opens an unusual-looking application window: On the left, you see your screen in a scale-down form, and on the right, two sliders let you adjust the quality of the video and audio recording. Sound can be turned on or off by clicking the button in front of the slider (Figure 3).

Control Concept

RecordMyDesktop expects you to set the areas to record manually in the scaled-down display in the program window. On mousing over the display, the mouse pointer turns into a cross, and you drag a frame around the area you want to include by holding down the left mouse button. The frame is shown in red and keeps a red border after releasing the mouse button. If so desired, you can grab just one window on the screen by clicking *Select Window* and clicking the desired window.

After defining areas or a window, you start the recording by pressing *Record*. The software then retreats into the system tray and shows you a black square that you use to stop the recording.

After stopping the recording, you see a window in which the data encoding progress can be tracked (Figure 4). If you *Cancel* this process, you lose the recording. After encoding, the program window pops up again.

A Question of Settings

On the start screen, the *Advanced* button below the sliders lets you define the quality of the video and audio recording in a more detailed settings dialog. You can adjust various options: for example,

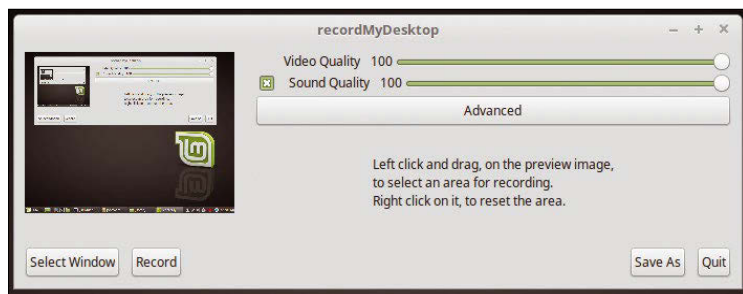


Figure 3: The old-fashioned way: the recordMyDesktop application window with its unusual view.

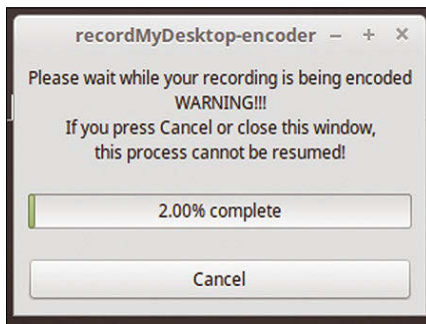


Figure 4: Progress bar: recordMyDesktop encodes the recording in a separate operation.

defining the frame rate in the *Performance* tab. If you have powerful hardware, you can also select to *Encode On the Fly*.

The *Sound* tab lets you set the sampling frequency and the number of channels. If you have a Jack audio server installed on the system, the program will use it if you check *Use Jack for audio capture*. In the *Misc* tab (Figure 5), you can define the recording area and select to hide or display the window decorations. Additionally, you can hide the mostly useful tooltips that appear when mousing over an option.

One convenient feature is that recordMyDesktop lets you select most of the options by checking or unchecking a box, which simplifies configuration of the software. Brief info bubbles keep you informed so that you can make the necessary decisions.

Hands-On Test

RecordMyDesktop stores the recorded content in files with the OGV extension, which means they are stored in the OGG container format that uses free codecs. In the test, I appreciated the fairly small file size: In the default settings with 15 frames per second (fps) and an audio sampling frequency of 22,050Hz, a one-minute video sequence

took only about 3MB of disk space. At 30fps and a sampling frequency of 44,000Hz, the foot-

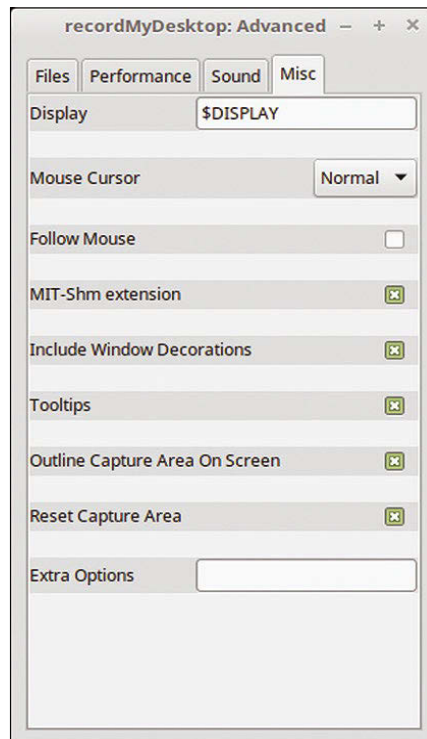


Figure 5: The recordMyDesktop advanced settings tabs are intuitive.

print increased to approximately 8.4MB. Thus, long screencasts have a pleasingly small footprint.

One thing that struck me in the test was that the difference in quality between using 15 and 30fps and a higher audio sampling rate was almost imperceptible compared with the defaults. In both cases, the software ensured uninterrupted recording of screencasts without jitter and dropouts.

A feature I did not like was the slow encoding of the recording. In the default settings, the tool slowly converts the raw material to OGV containers after recording. The encoding can take longer than the actual recording if you also increase the video frame rate and audio sampling rate, even on high-performance quad-core systems working at full CPU capacity. RecordMyDesktop is thus not genuinely useful for longer sequences like training videos.

SimpleScreenRecorder

A new development named SimpleScreenRecorder [3] is by far the most comprehensive screencast program in the test in terms of functionality. You have to add Maarten Baert's PPA to install the software. Instructions are in the *Download* section of the main webpage.

After launching from the Multimedia menu, the tool shows you a settings window (Figure 6) where you initially set up the recording area. Your choice of screens, a freely definable rectangular area, or a window are the available options. Also, you can define the frame rate and the audio system, where *PulseAudio* is the default. Additionally, the entire screen is the default grab area; based on the system settings, the program computes the correct resolution.

On the second page of the configuration dialog, you specify the file name and directory in which to save the screencast file, as well as the container formats and codecs. The number of alternatives is impressive. Matroska (MKV), MP4, WebM, and OGG are the container format choices, with H.264, VP8, and Theora for video codecs and Vorbis, MP3, and AAC for audio codecs.

You can use *Other* formats for both the audio and video codecs assuming the appropriate libraries are available on your system. Also, the audio recording rate can be set in kilobits per second (Kbps), with 128Kbps as the default. For ease of configuration, the software displays some

very detailed info messages for each of the options when you mouse over them.

Summary

The next page of the dialog lets you set a recording hotkey and displays information about the current screencast, which begins when you press the *Start recording* button at the top of the screen. SimpleScreenRecorder also offers a preview function with the *Start preview* button that displays a scaled-down screen in the application window.

Below the preview monitor, you will find a meter for the microphone input. The signal level of the recording is already displayed in idle mode, thus helping you avoid excessively loud or quiet recordings. You can use the sound system preferences to adjust the input sensitivity of the microphone to suit your requirements if the needle fails to deflect or jumps off the scale.

While recording, the software shows the current data and status messages at the bottom of the program window (Figure 7). To avoid the dialog getting in the way when recording starts, click its icon in the system tray; right-clicking on the

icon opens a context menu where you can start, cancel, or save the recording, even when the program window is minimized. The *Show window* option restores the program window to the desktop. When you are done with the screencast, press the *Save recording* button to store the encoded file at the defined location. Encoding is extremely fast.

Profiles

For professional use, the application also lets you create and load settings profiles, which eliminates the need for manual configuration work. It should be noted that the program stores the profiles screen by screen for the configuration dialogs. While working on the first screen of the dialog, click *New* in the Profile section at the top of the window, specify a name for the profile, and configure the desired settings. On the second page, click *Save*; you can also create another profile on this page by clicking *New*.

Once the profiles are stored as desired, you can retrieve them separately on each settings page from the selection list in the Profile section. Confusingly, each time you save a new profile you are

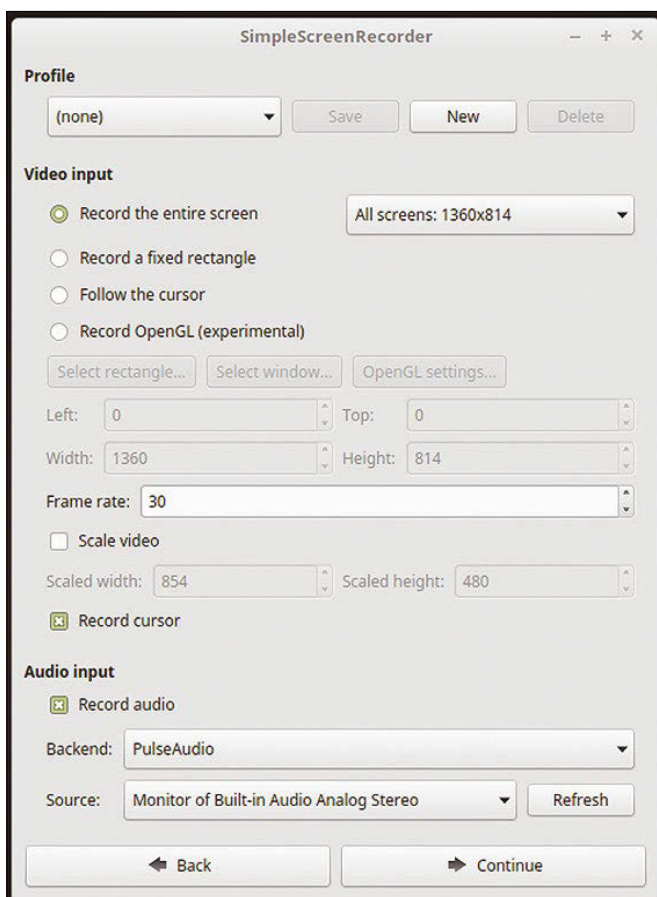


Figure 6: Configuring SimpleScreenRecorder is complex.

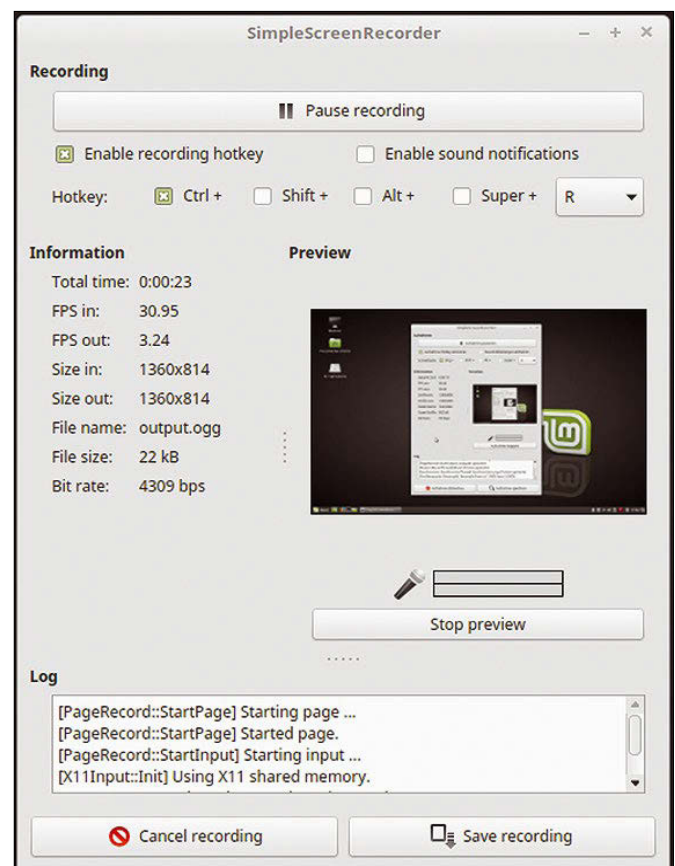


Figure 7: SimpleScreenRecorder displays a plethora of information while recording, if desired.

asked whether you want to overwrite it. Just say *Yes* here, because it does not affect the save operation. You can access preset profiles on the second page (e.g., for Livestream and YouTube), as well as delete unused profiles.

In Practice

SimpleScreenRecorder showed no weaknesses in the lab. The software created space-saving screencasts that impressed with good video and audio quality. The configuration dialogs can be bypassed with the use of preconfigured profiles. SimpleScreenRecorder is therefore primarily for those who frequently make screencasts and rely on tried and tested settings. Because of the variety of formats offered, the software is also suitable for heterogeneous environments, where not all playback systems will support all formats.

Vokoscreen

Vokoscreen, which is still going through some serious development work, is also found in the repositories of the established Linux distributions. Additionally, the source code is available on GitHub [4]. The developers also offer a comprehensive portable version weighing in at around 36MB as a `tar.gz` archive [5] for distributions that lack binary packages. You need to unpack the archive in a directory of your choice, which creates a large executable file of 95MB; then, working at the command line, type the `vokoscreen.run` command.

As with all screencast programs, Vokoscreen is configured in a central window; screencast recordings are controlled from an icon embedded in the panel. The program window differs from the usual visual conventions (Figure 8). Six tabs for the various control groups top the main

window, with controls at the bottom in the form of buttons for playing back, pausing, and stopping a recording. In the tabs, you select parameters for the audio and video codecs, the audio system, and the container format to use. Vokoscreen does not have a menubar.

Settings

In the first tab, you can enter the basic options for the area to record, including the choice of screen for full-screen recording, a window, or an area. As a special feature, the software offers a *Magnification* option, which significantly increases the size of the content in the area being captured. Even the size of the area to be magnified can be defined in a separate dialog. *Countdown*, *Showclick*, and *Showkey* functions round off the options for the recording. The software applies the selected options without the need to save them separately.

In the second tab, you define the basic options for your existing audio system. Vokoscreen works with both the ALSA and PulseAudio sound servers. Unchecking the *Audio* box lets you disable audio recording.

Vokoscreen was the only screencast application in the test to detect a headset connected to the computer via Bluetooth, although the microphone, a Sennheiser MM 100 headset, still failed to cooperate with Vokoscreen. If you choose the ALSA sound server instead of the default PulseAudio server, you can specify the correct audio system from a selection list.

In the third tab, you define the output format and set the frame rate of the video re-

coding. Several high-quality alternatives can be selected as the video codec, depending on its availability on a particular system; the H.265 codec even supports 4K resolution. Vokoscreen can also create Ultra HD footage, and in addition to the H.264 codec, MPEG4 and the slightly exotic free HuffYUV codecs are available. The screencast program offers the MKV, MP4, and GIF container formats for saving data.

In terms of audio, Vokoscreen does not offer many alternatives: The default is the MP3 codec, but also offers OGG Vorbis and AAC encoding. In contrast to the other candidates, you cannot adjust the sample rate for the audio codecs, although the frame rate for video recordings is definable. Vokoscreen also offers the option of not recording the mouse pointer in this dialog.

In the fourth tab, you define the storage path for the screencasts and specify the player on which to play back the recordings. The software selects a playback program used on your system, although you can change the default if you want. The audio framework is also defined in this dialog, and you can determine whether the program is shown in the system tray. Unfortunately, the icon display in the system tray did not work reliably: Although I had no problems in the test under Linux Mint 18 with the Cinnamon desktop, the icon did not show up in the system tray on Rosa Linux R6 and R7 with the KDE desktop



Figure 8: Vokoscreen's application window couldn't be easier to navigate and use.

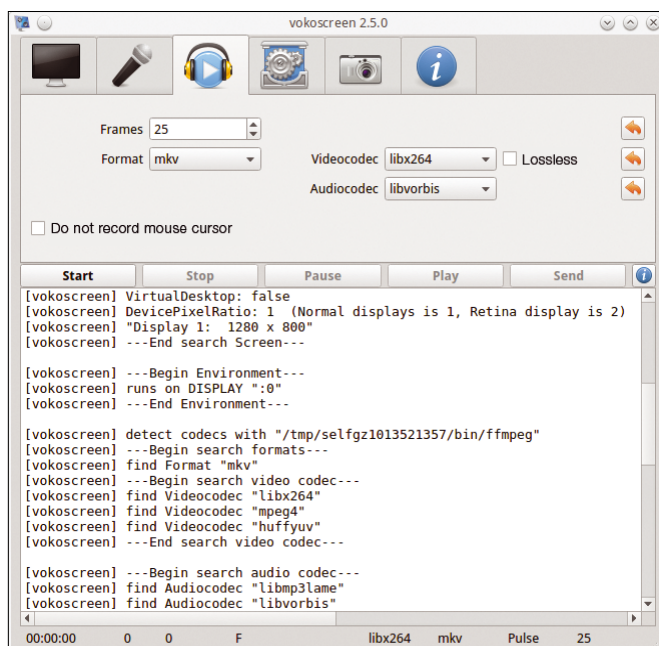


Figure 9: The log display helps you solve problems in Vokoscreen.

and with version 2.4.18 from the Rosa repositories – which makes it difficult to control the software – although the error did not occur with the portable version of the program.

Integrated Webcam

The fifth tab lets you activate and control a webcam so that you can integrate the camera recording into the screencast. If no webcam supported by Linux is present on the system, the corresponding dialogs in this tab are grayed out.

When the webcam is switched on, right-clicking on the webcam image calls up a context menu that offers options, such as defining the size of the webcam window; one option specifies that the webcam image remain in the foreground, even if you have several desktops.

Vokoscreen provides a useful function in the form of the log display, which tells you – in a way that is understandable even for non-experts – about the root cause of any problems and provides information on the program's various actions. The log display can be enabled

and closed in each tab by clicking on the info icon at the bottom right of the main program window (Figure 9).

Hands-On

As expected, Vokoscreen proved to be uncomplicated in the lab. Recordings worked from the outset with the preset parameters, without interruptions or jitter. The image quality was satisfactory, and the file sizes were pleasingly compact.

The lack of a program icon in the system tray to control the software on some desktops did affect the ease of use, however: To record the desktop in such cases without the Vokoscreen window, you need to check the *Vokoscreen minimized when recording starts* option in the fourth tab and then control the minimized program with shortcut keys, where Ctrl+Shift+F11 is *Stop* and Ctrl+Shift+F12 is *Pause*. Vokoscreen still responds to these shortcuts with the window minimized.

I also noticed a problem while the program was running that was related to some versions of the *libv4l* library. This

library, which is necessary for the correct use of webcams in Linux, does not support the RGB24 pixel format used by Vokoscreen in some variants, meaning that the webcam is not accessible. The webcam window in Vokoscreen opens, but remains black.

This problem manifested itself especially in three versions of Mageia 5. The operating system's obsolete Libv4l library failed to motivate the webcam to work on any of the three tested versions.

Krut

The fifth test candidate is something out of the ordinary: As a Java program, Krut is published under the GPL and thus a candidate for the review. You can pick up Krut as a ZIP archive from the project website [6], unpack the archive into any directory, and change to the newly created Krut subdirectory. All the program files are located there.

To call Krut, you need to have Java Runtime Environment 1.5 or higher in place on the operating system. In our lab, I had no trouble running Krut with the current Java Runtime Environment

LOST YOUR BOOKSTORE? LET US BE YOUR BOOKSTORE

Browse our shop for single issues of *ADMIN*, *Linux Pro*, *Linux Magazine*, *Raspberry Pi Geek*, *Drupal Watchdog*, and *Ubuntu User* – delivered right to your door.

■ shop.linuxnewmedia.com/single

Better yet, subscribe, and you won't need a bookstore.

■ shop.linuxnewmedia.com/subs



shop.linuxnewmedia.com

DIGITAL AND PRINT EDITIONS AVAILABLE!

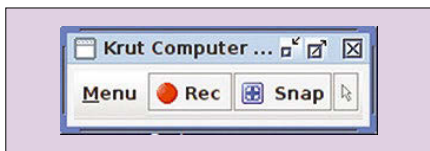


Figure 10: The control window in Krut is very unobtrusive.

versions 1.7 and 1.8, regardless of whether you have the runtime environment from Oracle or the OpenJDK, as enabled by default on the latest Linux distributions when you install the operating system.

You can then launch the screencast program in a terminal by typing

```
java -jar krut-0.9.4.jar
```

A small window in the typical Java program design opens that has only three buttons: *Menu*, *Rec*, and *Snap*. With the help of the *Snap* and *Rec* buttons, you can create screenshots and screencasts, whereas clicking the *Menu* button opens a window with some basic settings (Figure 10).

Options

To access the configuration menu, click on *Menu* | *Settings/Save Files*. Krut offers a variety of options for configuration, although they are not as detailed as in SimpleScreenRecorder or Vokoscreen. Therefore, everything fits in a single window (Figure 11). The software works well with the hardware detected by Linux.

Krut stores the video, audio, and screenshot files separately in their own subdirectories, so you specify a separate path for each type. In the settings window, you can also configure the frame rates for recording and playback separately and define whether you want mono or stereo recordings. Optionally, you can specify the bit depth of the recording, as well as the sampling frequency. A slider adjusts the encoding quality. Modified settings require confirmation in Krut and are applied when the window is closed.

Definitions

For Krut to create a recording, you need to define the area to grab. Window and full-screen settings do not exist, and Krut does not use webcams, should one be present on the system. To define the recording area, you click on the iconized mouse pointer to the right of the *Snap* button in

the small control window, and with a Ctrl+Left-click, drag the mouse pointer from the upper left to the lower right corner of the target area. The target area turns a semi-transparent green. After defining the target, the recording is started in the usual way. If you want a preview (e.g., to check the target area), check the *Menu* | *Preview Window* option.

After starting a recording, Krut records until the *Stop* button in the control window is pressed. Then, a small window displays the encoding, which is a pleasingly quick experience. Krut creates a MOV file that contains both audio and video data in the defined storage directory. Separately, the software stores a sound file in WAV format. Krut does not allow you to change file formats.

Practical Problems

In practice, defining the recording area turns out to be unreliable in Krut: The software failed to integrate the panel strips into the image in various desktop environments, so full-screen recording was not possible. The window recordings only delivered the desired results after defining the recording area multiple times. The video and audio recordings produced were of a good quality, though, and synchronized correctly.

Encoding was quick, but the file sizes in the test were significantly larger, even for a medium encoding quality, than those produced by the other candidates.

Conclusions

The saying “where there is light, there is also shadow” also applies to screencast programs, which meet vastly different requirements in Linux. Kazam and recordMyDesktop are suitable for occasional work, although recordMyDesktop has an outdated operating concept that needs some getting used to. Kazam showed weaknesses in audio recording in the test.

SimpleScreenRecorder is the best choice for professional users. Thanks to the ability to create profiles, this software in particular facilitates the process of creating training videos for which you want to use the same parameters.

If you are looking for a very solid all-around application with a modern interface and intuitive operation, however, Vokoscreen stands out from the test field. You can see that the developers carefully considered which functions are important for professional screencasts and which are not. The software interface is accordingly uncluttered, but without having to compromise on the program’s feature set.

Krut is for those who need portable, platform-independent screencast software. This program has a learning curve, and the recording target areas did not always work in the lab. ■■■

INFO

- [1] Kazam: <https://launchpad.net/kazam>
- [2] recordMyDesktop: <http://recordmydesktop.sf.net>
- [3] SimpleScreenRecorder: <http://www.maartenbaert.be/simplescreenrecorder/>
- [4] Source code for Vokoscreen: <https://github.com/vkohaupt/vokoscreen>
- [5] Vokoscreen: <http://linuxecke.volkoh.de/vokoscreen/vokoscreen-download.html>
- [6] Krut: <https://sourceforge.net/projects/krut/>

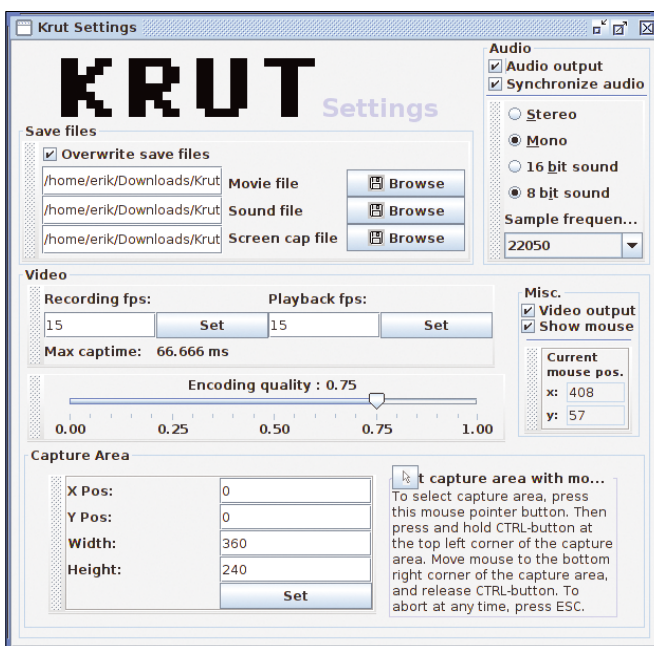


Figure 11: Fully loaded: Krut offers all the settings in a single window.



ENIGMA

A USENIX CONFERENCE

SECURITY AND PRIVACY IN CONTEXT

Join a diverse mix of experts and enthusiasts from industry, academia, and government for three days of presentations and open sharing of ideas.

Our focus is on current and emerging threats and defenses in the growing intersection of society and technology. Our goal is to foster an intelligent and informed conversation with the community and with the world, including a wide variety of perspectives, backgrounds, and experiences.

The full program and registration are available now.

ENIGMA.USENIX.ORG

JAN 30-FEB 1 2017

OAKLAND, CALIFORNIA, USA





Debian's long-term support experiment

Well Supported

The Debian project is extending its famous development process to offer long-term support. *By Ferdinand Thommes*

Debian Linux, which calls itself “The Universal Operating System,” is a huge and popular Linux variant. Debian was one of the first Linux distributions, and it remains one of the largest, with over 43,000 software packages. Unlike many other leading Linux distros, Debian is not backed directly by a company, and it is managed democratically by the many volunteers who populate the Debian mailing list.

Despite its reputation as an all-free, counter-cultural collection for hackers, Debian is also quite stable and reliable, which makes it a viable option for many corporate networks. But after many years of Linux in the enterprise, admins have a pretty clear idea of what they want: a system that will operate for several years without requiring an upgrade. Rolling out a new operating system in the enterprise can take many months, and the process is quite complex and prone to complications. Better to make such events as infrequent as possible.

In 2014, the Debian developers woke up and realized the recent trend for Long-Term-Support (LTS) releases had left them behind. Because Debian is not backed by a company that can make money on selling support contracts (like

SUSE, Red Hat, and Canonical), they had never gotten around to implementing some form of long-term support.

Before 2014, each Debian release was supported for one year beyond its “end of life,” which was heralded by a new release. That support mainly meant integrating and providing security updates and bug fixes. Since Debian released a new version approximately every two years, the traditional Debian release schedule meant that each new version would be maintained for three years.

A three-year support period is too short for many companies. When Ubuntu, which is based on Debian, began to offer an LTS edition, the option of running Debian in the enterprise started to lose its appeal, and companies began to migrate to other systems. For example, in 2014, Spotify migrated 5,000 servers from Debian to Ubuntu LTS [1]. Red Hat and Suse also offer LTS editions that appeared to have snatched some of Debian’s thunder.

To regain momentum in the enterprise space, the Debian security team decided in

February 2014 [2] to experimentally release Debian LTS for then-current Debian 6 “Squeeze.” The plan was to extend the support period from three to five years for Debian 6, 7, and 8 (Figure 1). The main problem: The approximately 1,000 official Debian developers were stretched to full capacity with what they were doing already.

The need for additional developer hours resulted in the idea of asking companies (and users who are interested in the long-term support for Debian) to remunerate individual developers directly on an hourly basis for working on LTS updates. The extended support was initially only offered for x86 systems with 32-bit and 64-bit architectures; the support now includes the *armel* architecture and *armhf* for “Wheezy” (see Table 1).

Debian’s LTS support does not include support for all of the software: Around

Lead image © elwynn, 123RF.com

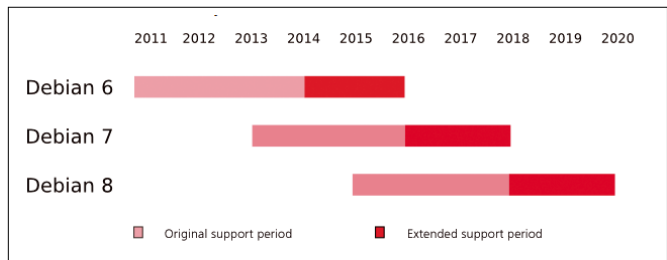


Figure 1: The new extended support period provides two additional years of updates for each Debian release.

TABLE 1: Debian LTS Support Period

Version	Supported architectures	Period
Debian 6 "Squeeze"	i386, amd64	up to 29 February 2016
Debian 7 "Wheezy"	i386, amd64, armel, armhf	up to 31 May 2018
Debian 8 "Jessie"	i386, amd64, armel, armhf (possibly others)	May 2018 up to April/May 2020
Debian 9 "Stretch"	i386, amd64, armel, armhf (possibly others)	2020 up to 2022

/etc/apt/sources.list.d/, as well as to the Apt configuration. Today Debian LTS is instead updated through the normal security repository:

```
http://security.debian.org/
wheezy/updates main
```

40 packages in the web application and virtualization sector are excluded, since support for these tools cannot be guaranteed for the entire period. These tools include web browsers and applications such as KVM or Xen. The *debian-security-support* package automatically monitors the status during each package installation and informs the user.

After completing the preparatory work, the then-current Debian 6 "Squeeze" became Debian's first LTS version. Its successor, Debian 7, has a correspondingly long support period: "Wheezy" was released in Spring 2013, and its support does not end until 31 May 2018. Currently, Debian is planning to offer Debian LTS support for the still-current Debian 8 "Jessie." The same applies to Debian 9 "Stretch," which will probably ship in the spring of 2017.

One-Stop Shop

The infrastructure for handling LTS tasks is provided by Freexian [3], a consulting company operated by French Debian developer Raphaël Hertzog. Hertzog publishes a monthly report on the current status, the tasks taken, and the hours worked by the individual developers. Any developer who receives money for work on Debian LTS is required to submit a report on a monthly basis.

The first official paid work for Debian LTS [4] was completed in July 2014. The report published at the time [5] showed 21 hours for two developers; the following month, the two developers had already contributed 32 hours. Two years later, in September 2016, the figures grew to 152 hours provided by 13 developers [6].

A text file on the Debian website provides an overview of the outstanding LTS work [7]. This file includes activity on packages, which sponsors are allowed to prioritize. As the number of sponsored hours increases, sponsors can book direct contact with the developers. It is also possible to apply test suites to cover the specific needs of each sponsor during updates of prioritized packets.

A developer hour costs the sponsor US\$75, plus the administration costs for Freexian (Figure 2). To be eligible to contribute to Debian LTS as a developer, the candidate must be a Debian developer or Debian maintainer and prove that he or she has already released security updates for Debian.

Technically Simple Solution

Debian LTS was created as an additional repository that users needed to add to the source list, /etc/apt/sources.list or

The transition from Debian "stable" to Debian "unstable" thus only triggers a team change in the background: The security team that ensured the release of security updates during the release's lifetime hands the responsibility over in the background to the LTS team at the end of the support period. Nothing changes for the user, who simply installs the usual security updates via the package manager.

Conclusions

The success of Debian LTS proves that Debian is still a preferred candidate for the enterprise due to its proverbial stability and well-respected security policy. The list of sponsors currently includes 38 companies. The list extends from major league players such as Toshiba and GitHub down to smaller sponsors like Linux Hotel or hoster BitFolk. Debian itself trusts in its system and has already promised long-term support for Debian 9 "Stretch," which will not take effect before 2019. ■■■

INFO

- [1] Spotify migration (German): <https://raphaelhertzog.com/2014/07/16/spotify-migrates-5000-servers-from-debian-to-ubuntu/>
- [2] LTS announcement: <https://lists.debian.org/debian-devel-announce/2014/03/msg00004.html>
- [3] Freexian: <https://www.freexian.com>
- [4] Debian LTS information: <https://www.freexian.com/services/debian-lts.html>
- [5] First report: <https://raphaelhertzog.com/2014/09/10/freexians-first-report-about-debian-long-term-support/>
- [6] Most recent report: <https://raphaelhertzog.com/author/rhertzog/>
- [7] Pending packages: <https://anonscm.debian.org/viewvc/secure-testing/data/dla-needed.txt?revision=HEAD&view=co>

Funding level (time per month)	15 min.	30 min.	1 h.	2 h.	3 h.	4 h.	6 h.	1 day	2 days	3 days
Benefits (details)										
Packages to prioritize	✓ Weighted on the amount sponsored									
Thanked as sponsor:	✗	✓ Bronze			✓ Silver		✓ Gold		✓ Platinum	
Private mailing list	✗				✓					
Direct contact to LTS staff	✗							✓		
Submit your own test cases				✗						✓
Prices										
Monthly payment	N/A			€255 \$300	€340 \$400	€510 \$600	€680 \$800	€1360 \$1600	€2040 \$2400	
Quarterly payment	N/A		€255 \$300	€510 \$600	€765 \$900	€1020 \$1200	€1530 \$1800	€2040 \$2400	€4080 \$4800	€6120 \$7200
Biyearly payment	N/A	€255 \$300	€510 \$600	€1020 \$1200	€1530 \$1800	€2040 \$2400	€3060 \$3600	€4080 \$4800	€8160 \$9600	€12240 \$14400
Yearly payment	€255 \$300	€510 \$600	€1020 \$1200	€2040 \$2400	€3060 \$3600	€4080 \$4800	€6120 \$7200	€8160 \$9600	€16320 \$19200	€24480 \$28800

Figure 2: Prices and services for Debian LTS.

Ardour DAW: Fifth Generation

Symphony No. 5

The fifth generation of the free Ardour digital audio workstation is easier to operate, offers more Lua support, and comes with a variety of improvements and extensions.

By Hartmut Noack

Just six weeks after the announcement of a forthcoming new Ardour version, the Ardour project released the fifth generation of its free digital audio workstation (DAW) [1]. The impressive list of new features does not promise a total revolution of the proven music production suite, but a variety of improvements and extensions definitely have

revolutionary potential. I investigated how the software works in practice.

Saddle Your Keyboards

The free Ardour 5 software is not only available from Ardour's GitHub repository in the form of the current source code [2], the project also offers tried-and-tested installation packages – for Linux, Mac OS, and now, starting with Ardour 5, Windows – in return for a single donation in an amount that you get to choose.

If you want to avoid the fairly complex process of building Ardour yourself from source [3], you might want to acquire such an installation package. Although Ardour can be installed from the repositories

you ask questions about problems with the distribution packages. The packages built by distribution maintainers always contain small deviations from the current state, over which the Ardour team has no influence.

The Ardour project is entirely funded by end users. The previous industry sponsors only support the project with logistics and through a very laudable exchange of information; Ardour developers no longer get a salary from this corner.

The installation package copies the application, as well as various components supplied as new Lua modules to `/opt/ardour-5<x>/`. Different versions of Ardour can be set up in parallel,

but each generation creates its own configuration files and backups of old projects. If you open a project recorded with Ardour 3 in Ardour 5, the new Ardour only copies the old project to a numbered backup and then opens the original file (Figure 1).

AUTHOR

Hartmut Noack works in Celle and Hanover, Germany, as a lecturer, author, and musician. When not sitting in front of his Linux audio workstation, he's hanging out on web servers. You can find some CC-licensed examples of his audio work at <http://lapoc.de>.

of all the major distributions, the packages offered by the project itself are generally more up to date and often more stable than the packages for Debian and openSUSE. Also, the developers on the Ardour.org forum might be slightly irritated if



Figure 1: A project recorded in 2011 with Ardour 2 loads in Ardour 5 without problem; Ardour makes a backup copy for the old version as a precaution.

When first launched, Ardour displays a configuration tool for the desired audio interface, even if the preferred audio server Jack is already active. You can also run Ardour on a second sound card with ALSA while Jack manages other applications. If you select Jack, the dialog in Figure 2 tells you that Jack is already running and that Ardour is connecting to it.

The tools offered on startup also include a new wizard for calibrating the audio system. It measures the latency from digital processing that a signal experiences between input and output. Ardour adjusts internally to this latency and can thus avoid unwanted shifts in recordings.

The tool displays various additional pieces of information about the signal, such as the phase shift and distortion because of incorrect inputs and faulty wiring. It is strongly recommended to send a measurement signal to the input, such as a smooth sinusoidal

tone. Fluctuations typical of guitars can very quickly change the output.

Visibly New

At first glance, Ardour 5 is visually almost indistinguishable from its predecessors, but even with the simplest actions, significant new features show up.

At the top right, new buttons switch between the Editor and Mixer views and the Preferences. Unlike earlier versions, instead of creating new windows, the selected module appears as a tab in the main window. However, the original behavior also remains available: a right click on the new toggle buttons displays a menu (Figure 3), where you can specify that the Mixer occupies its own window, which is a better solution for some users, especially those with multiple monitors.

The tab view comes with a practical side effect: the bar at the top from the Editor view with the transport controls

and time displays is kept in the Mixer tab, which means you can now finally manage these controls on just one monitor when

mixing, without having to learn masses of keyboard shortcuts.

An entirely new element has appeared in the Mixer that you can add to a project as a new track: VCA tracks. They let you intercept the signals of several individual tracks upstream of the master input and adjust the volume across the board. Ardour does not create a submix in the strictest sense; all tracks can be set up, wired, and edited individually using plug-ins, without VCA, just as easily.

The VCA channel acts as a shared volume, and it has an additional function, which its role models in expensive hardware mixer consoles do not have: When you click on the VCA, the Mixer only displays the tracks connected to it. You can thus create a custom VCA channel for the tracks of an individual musician when recording a band – an elegant method for quickly shifting these tracks into the foreground in the Mixer (Figure 4).

Many new tools can be found in the Preferences tool. One particularly striking newcomer is the management of color palettes, which comes with a series of well-thought-out presets. You will find some nice combinations of colors in bright pastel colors.

Cutting and Assembling

Tempo ramps are the most visible new feature in the Editor. Thus far, you could only change the tempo in Ardour gradually during a piece in beats per minute (BPM) units (the number of strokes on the downbeat: the faster the music, the more BPM). The new feature allows for a gradual, natural-sounding tempo slow-down or speed-up (Figure 5).

If you want to use typical sequencer functions with recorded audio material, you would probably appreciate a region for each played note on the timeline. However, it would be extremely time consuming to break apart

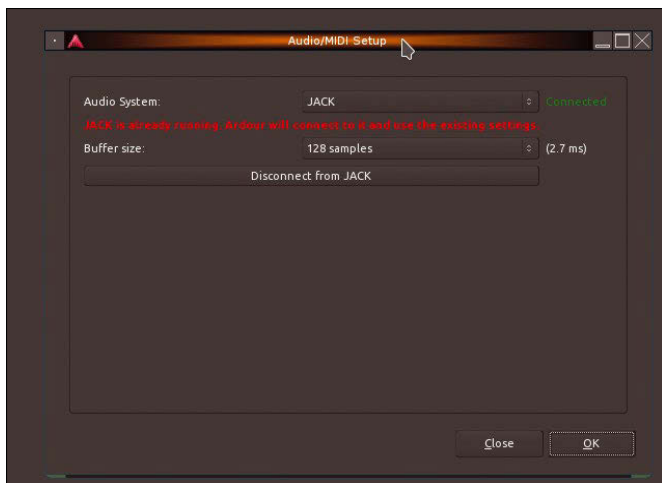


Figure 2: Do not be confused by the flashing red message: The connection to Jack will work after clicking OK.

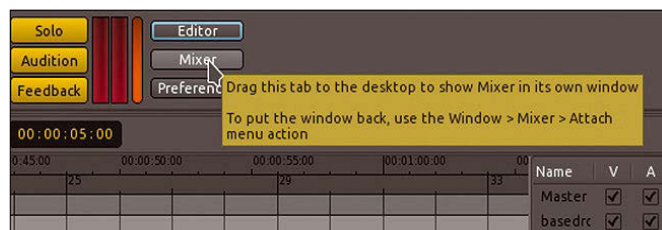


Figure 3: The Editor, Mixer, and Preferences can be displayed both in tabs and in standalone windows.

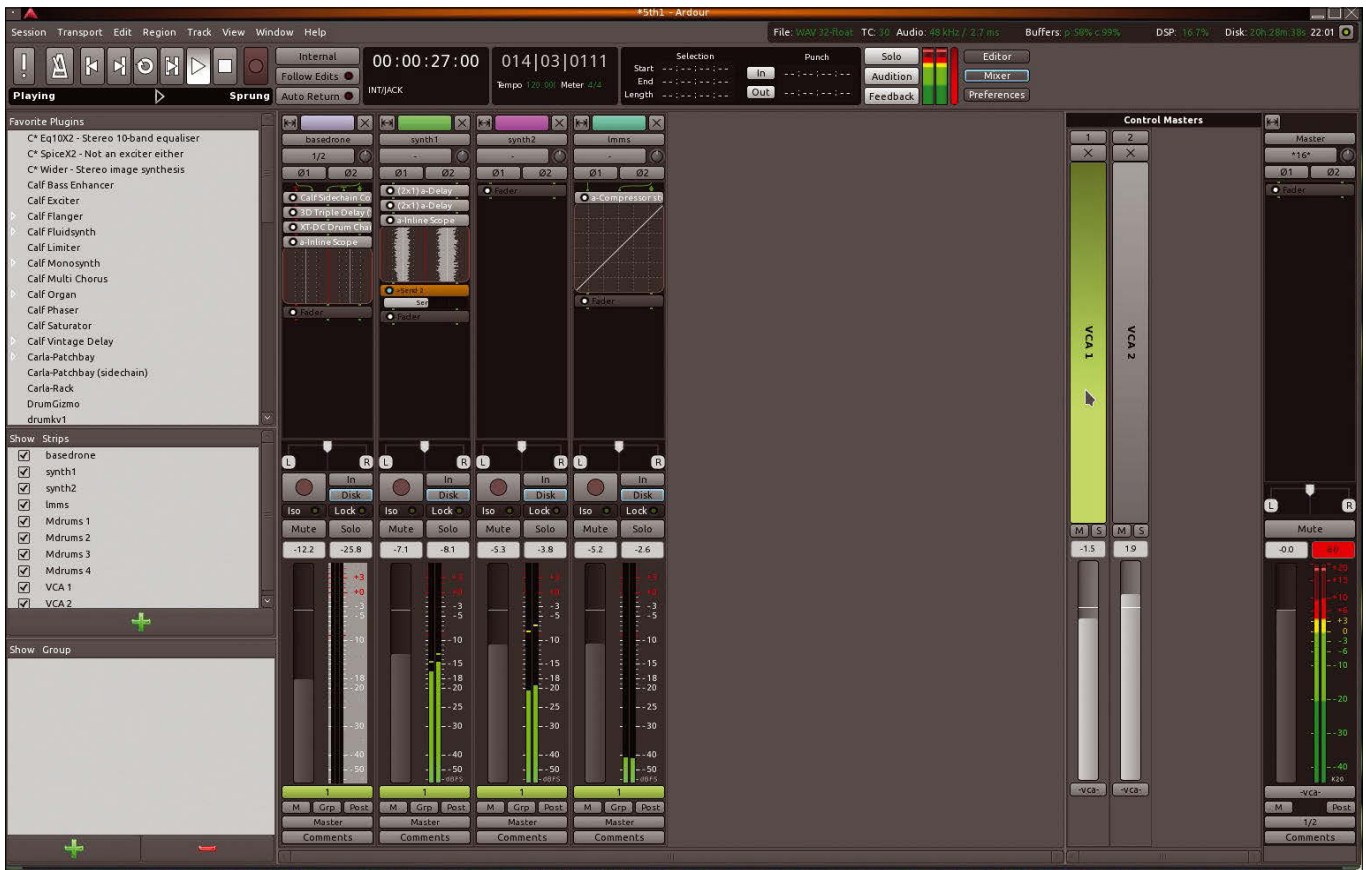


Figure 4: The new Mixer view with two VCA channels. Only the tracks connected to a channel are selected. At top left, you can see the transport controls in the Mixer.

all the recordings manually. That is why Ardour has an automatic cutter with the strange name of Rhythm Ferret for this operation.

Hidden in the Region context menu, this useful animal is found below *<Name_of_Region> | Edit | Rhythm Ferret*. Ardour 5 gave the cutting tool some new options that significantly improve the precision, especially in less percussive material.

Especially when working in the Editor, keyboard shortcuts allow faster, more convenient operation. You can split, normalize, duplicate, and move

individual MIDI notes and audio regions using preset keyboard shortcuts far more easily than with the mouse. Moreover, you can assign key bindings by tab to suit your needs.

Searching by specific functions is not supported, however. It's particularly annoying that when attempting to register an already assigned shortcut key for another function, you receive a terse request to delete the existing combination, although the message does not let you know what function already owns that shortcut combo.

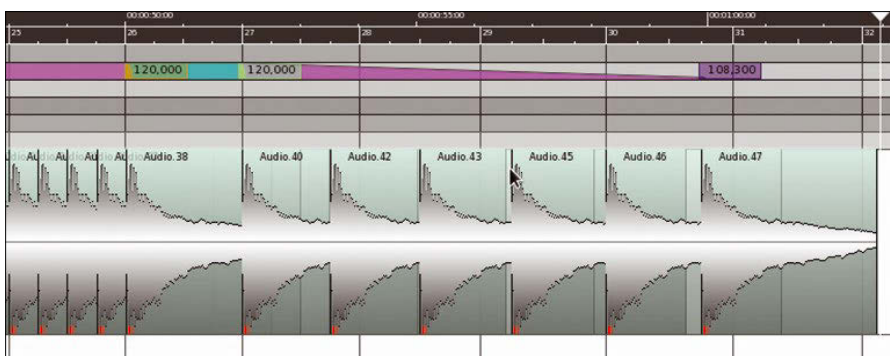


Figure 5: A tempo ramp from 120 to 108.3BPM. Regions bound to bars and beats automatically follow the speed change.

Lua Scripts in Ardour 5

All functions of the Ardour interface can now be controlled through the Lua scripting language [4], allowing you to both automate frequently used processes in macros (Figure 6) and construct your own controls from existing functions.

Lua can drag low-level features of the Ardour core to the surface. Libardour contains various manipulation functions for audio data, which can find the underpinnings for, say, the mixer. Based on them, Ardour-specific audio plugins can be programmed in Lua scripts. Thus, for the first time, Ardour 5 comes with internal audio processors that integrate seamlessly with the application and work just as flawlessly in real time as the well-known mixer elements.

For the near future, the developers are planning a switch in the shortcut menu of the mixer channel that lets you add your own Lua processors, just like normal plugins. Thanks to the real-time-capable Lua design, you can then program the required DSP processors directly into the mixer channel in a targeted way.

The existing internal Lua plugins are found in the normal plugin overview. You will easily find the search filter there easily based on the common prefix `a-*`. Ardour delivers additional Lua scripts in the *Session | Scripting* menu, where correctly formatted Lua scripts that you have written also appear. You need to store the appropriate files in `$HOME/.config/ardour5/scripts/`.

For testing purposes, you can launch Ardour as a Lua console. To do so, install self-compiled builds of Ardour with the `ardour_lua` command in `/local/bin/`. The Ardour manual [5] contains a detailed section on Lua scripting that also contains the basic information needed by developers who are reasonably familiar with script programming and audio.

More and Better Plugins

In modern music software, plugins are responsible for creating and editing sounds with effects. Like its predecessor LV2 and the time-honored LADSPA on Linux, Ardour 5 supports Audio Units plugins on Mac OS and VST modules on Windows and Linux in

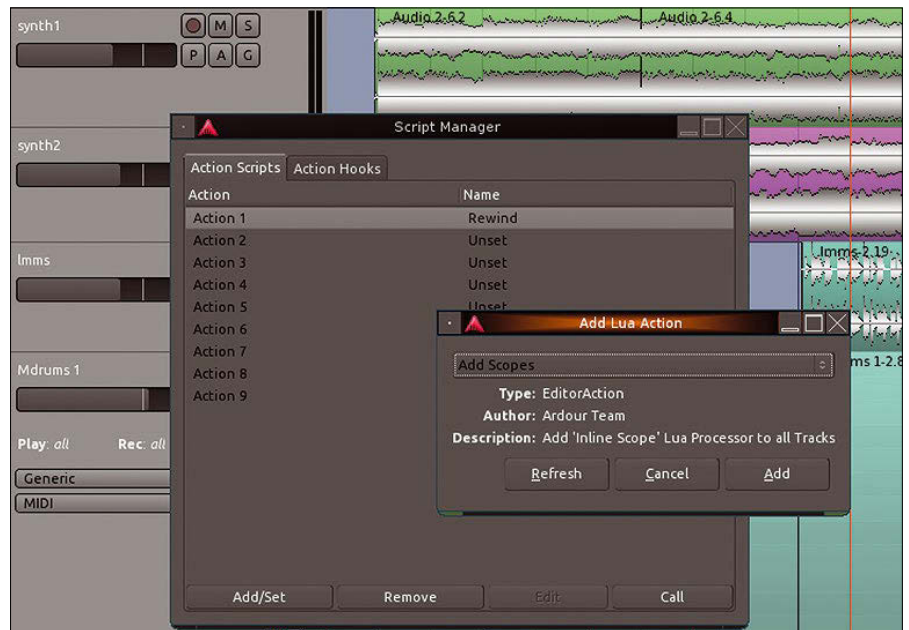


Figure 6: Some supplied Lua macros can be added under *Edit | Scripted Actions | Manage*.

their respective native formats. In principle, Ardour 5 can use VST modules compiled as Windows DLLs on Linux; however, the project advises you to go for the native LX VSTs in the form of `.so` files.

On top of the additional plugin interface, the on-board plugins mentioned in the previous section offer animated sound analyses directly in the Mixer chain (Figure 7). If you compose with MIDI, you can use plugins as sound

Shop the Shop

shop.linuxnewmedia.com



RASPBERRY PI ADVENTURES

COOL PROJECTS
FOR GEEKS OF ALL AGES

RASPBERRY PI ADVENTURES

is a one-volume special edition magazine for curious Raspberry Pi beginners. This easy, hands-on guide starts with an introduction to computers and offers a series of special hands-on projects illustrating many of the most popular uses for the Raspberry Pi.

ORDER YOUR VERY OWN ISSUE!



ORDER ONLINE: shop.linuxnewmedia.com/se27

generators, just like samplers and synthesizers. In our lab, I used SynthV, Calf Monosynth, ZynAddSubFX, and DrumGizmo as LV2, as well as the LXVST plugin Carla Patchbay.

Carla clearly works better than in Ardour 4, which opens up a particularly easy approach for friends of Windows VST modules to use their plugins in Ardour: Carla loads VST DLLs without complaint and integrates them into Ardour sessions. You can thus save yourself the trouble of compiling Ardour from source with Windows VST support.

The method for inserting instrument modules into a track is much improved: Ardour 5 reduces the procedure to a dialog, where you simply confirm that you want to replace the module and accept all the virtual wiring (Figure 8). (See also the “Through the Side Door Easily” box.)

The buyers of the official Ardour installer are treated by Ardour 5 to a special plugin treat in the form of 16 high-quality effects as demo versions from Ardour’s long-time industry sponsor Harrison Consoles. These include EQ and dynamic processors, as well as a very usable, first-class stereo echo/delay. The plugins can be used as usual, but after about 10 seconds, the message *Please buy license* hides the user interface. If

you want to use the module without this interference, you can purchase the entire set for a remarkable \$399 on the Harrison website [6].

SUPPORT FOR ACTIVE MUSICIANS

Despite very limited resources, the Ardour project has the ambition of offering professional support. But project manager Paul Davis, in particular, has an approach to this that appears a little chilly to some users. He does away with empty promises, backslapping, and wordy flattery and makes no major differences between new users and long-time fans who have already made various donations. Instead, he and his team focus on the solutions to problems.

In our lab, two difficulties occurred for which I looked to the project site for help. The first was a regression with the trim tool: In Ardour 5, triggering this was tricky. The problem was difficult to explain and could not be represented easily with a screenshot, because the real problem lay in the dynamic response of the interface. After some back and forth, a user uploaded a video of the problem in the Ardour bug tracker.

Davis’s response was typical of the style of communication with Ardour: “Fixed in git.” What this translates to for mere mortals: “We have fixed the problem and

Scope for Improvement

New releases of complex programs often struggle with teething pains, and Ardour 5 is no exception. However, the

tested the fix in our current source code. If you want to use Ardour without this problem, you need to recompile the program from the current source code or wait for the next official update.”

The second problem related to an unfinished, but remarkable, new feature. Under circumstances that were difficult to comprehend, the behavior of the mouse pointer tool for moving regions in the timeline simply changed. Instead of moving the region, it continually created copies of a region when dragged. In the *Edit / Undo* menu entry, I found an entry for *Undo: Drag region with brush*. Aha! A new paintbrush that nobody had ever heard of that you cannot turn off.

In the bug tracker, I discovered that this function is supposed to be triggered by pressing *B* when dragging the region; however, Ardour does not notice when you let go of the button. This problem harvested a very rare “Sorry for the error” from Davis. I assume that this new feature, now fully functional, will feature prominently in the next update of Ardour.



Figure 7: Scope, Spectrogram, and Compressor from the Ardour plugin collection show their output directly in the Mixer tracks.

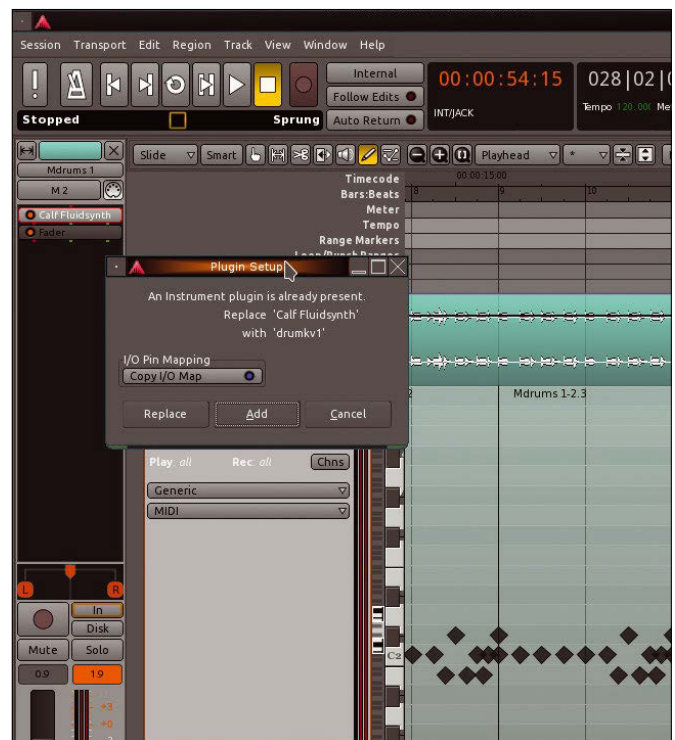


Figure 8: The drumkv1 plugin replaces Calf Fluidsynth at the push of a button.

problems are confined to small niggles. For example, the interface of the new pin configuration tool reacts pretty sluggishly but does not restrict its function.

As with other music production suites, caution is advisable when using third-party plugins. Virtually all serious problems during the test occurred in connection with these external modules; in all cases the root cause was a sound generator in the MIDI tracks. Synthesizers and samplers are by nature quite complex and add another potential source of error, which you tend to underestimate: You can load presets and sample data from almost completely uncontrolled sources.

For example, a session crashed reproducibly in a test in which I downloaded

a new drum kit file for Calf Fluidsynth SoundFont player while the module was playing notes with another SoundFont. The same experiment caused no more than a brief buzz with two other SoundFont files. Because Calf Fluidsynth is generally considered very reliable, the SoundFont file must have been the problem; this was also reproducible. (See also the “Support for Active Musicians” box.)

For all crashes – in running sessions, at least – the built-in recovery function in Ardour 5 worked perfectly. In version 4, I saw isolated cases in which this useful feature failed. In about 30 hours of test work in Ardour 5 with about a dozen (mainly deliberately induced) crashes, not one minute of work was lost

thanks to disaster recovery recovering everything.

Conclusions

Ardour 5 justifies the generation jump in its version number thanks to many useful new features. The still very simple Lua functions promise many in-depth extensions for Ardour 5 in the future. The absolutely outstanding stability, despite the version jump, was a particularly positive feature of the test. Ardour 5 is no less solid than its proven predecessor, but it is better. ■■■

INFO

- [1] Ardour 5 release notes: <https://community.ardour.org/node/13842>
- [2] Ardour on GitHub: <https://github.com/Ardour>
- [3] Ardour build instructions: https://ardour.org/building_linux.html
- [4] Lua scripting language: [https://en.wikipedia.org/wiki/Lua_\(programming_language\)](https://en.wikipedia.org/wiki/Lua_(programming_language))
- [5] Ardour Manual – scripting documentation: http://manual.ardour.org/lua-scripting/brain_dump/
- [6] Harrison web shop: <http://harrisonconsoles.com/site/store-mixbus-plugins.html>

THROUGH THE SIDE DOOR EASILY

Sidechaining involves controlling a parameter of an effect by audio signals that do not otherwise need to be heard in the mix. Thanks to its uniquely flexible cabling options, Ardour has already supported sidechaining for quite some time. However, setting up the popular effect has been quite cumbersome and tedious. Ardour 5 now offers a new tool that makes sidechaining an intuitive process (Figure 9).

Right-clicking on a plugin in the mixer channel shows a new entry named *Pin Configuration*. You can add inputs to those provided by a plugin and thus experiment with sidechains not originally intended in a plugin. You can select an existing audio output in the project for each of the inputs. The tool creates a new *send* output, which you can then use like self-specified sends.



Figure 9: The Calf Sidechain Compressor is ideally suited for creating popular pump effects where the beat of a bass drum rhythmically down-controls a synthesizer section.

Compress JPEG images with Lepton

Shrink It!

Lepton lets you compress your photo files to make room for more. *By Thomas Leichtenstern*

Although more sophisticated image formats have existed for some time [1], JPEG is still one of the most common. Whether you are working on a smartphone, compact camera, or professional photo device, you'll find an option for generating images in JPEG format. The flood of JPEG images caused by so many millions of users snapping photos actually sucks up lots of storage space on file hosting servers. This problem led cloud provider Dropbox [2] to look for a better way to store photo images. The result of this effort is the new Lepton image format, which reduces the space requirements for JPEG images by up to 25 percent. Lepton works losslessly, meaning the image remains exactly the same after you reconvert it (Figure 1). Dropbox says it has already applied the new format to 16 billion images and has thus saved multiple petabytes of storage space. The Dropbox cloud transparently converts JPEG images to Lepton format and changes them back into the original format as soon as someone accesses them.

Unlike JPEG, Lepton doesn't reduce the visible image – image reduction often results in known artifacts with JPEG. Instead, Lepton is designed to reduce the data needed for the image information using the Middle-out algorithm, a type of

arithmetic coding that is also used in the VP8 video format. A Dropbox blog post provides a detailed technical explanation about how the Lepton format reduces the size of a JPEG file [3].

Installing Lepton

Dropbox has additionally released a tool, also called Lepton, that you can use to convert files to and from Lepton format. The Lepton compression tool is available under the Apache 2.0 license. Ubuntu, Debian, openSUSE, and several other Linux distributions provide Lepton packages. You can also download the source code from GitHub [4] and compile Lepton from source (Listing 1) If you choose to compile, you'll need the Git and Cmake tools, which you can install with:

```
sudo apt-get install git cmake
```

The Lepton README [5] suggests a few other possibilities for obtaining Lepton, but Git or a package installation are likely the easiest options.

In Use

The Lepton syntax is pretty simple. For example, the command:

```
lepton image.jpg
```

converts the specified image into Lepton format. By default, the new file will have the original filename with the .lep extension. The following command:

```
lepton filename.lep
```

extracts the original JPEG from a Lepton file.

If the output file already exists, Lepton doesn't overwrite it but instead creates a new file with an underscore (_) at the end of the filename. You can also specify a filename for the output file.

During our tests, it became apparent that Lepton only processes images up to

LISTING 1: Compiling Lepton

```
$ git clone https://github.com/dropbox/lepton
$ cd lepton
$ mkdir -p build
$ cd build
$ cmake ..
$ make -j8
$ sudo cp lepton /usr/bin/
```

```
testbild: bash - Konsole
File Edit View Bookmark Settings Help
kork:/testbild # ll
total 25708
-rw-r--r-- 1 root root 9481401 Sep  5 10:48 testbild.jpg
-rw----- 1 root root 7357296 Sep  5 10:48 testbild.lep
-rw----- 1 root root 9481401 Sep  5 10:48 testbild2.jpg
kork:/testbild # md5sum testbild.jpg testbild2.jpg
cb64b20cd2b86bb5f0d8c494755cfbdb testbild.jpg
cb64b20cd2b86bb5f0d8c494755cfbdb testbild2.jpg
kork:/testbild #
```

Figure 1: Lepton doesn't make any changes to the image when converting and reconvert: The image remains exactly the same.

```
tommy: lepton - Konsole
File Edit View Bookmark Settings Help
tommy@kork: ~-> lepton -listen=2020
/tmp/a43672cce8-607b-e3e7-662850f3d634-f0.uport
16925244 bytes needed to decompress this file
DONE
17808000 bytes needed to decompress this file
DONE
Child 8240 exited with code 0
```

Figure 2: Lepton has a server mode that receives images via TCP.

LISTING 2: Server Mode

```
$ cat bild.jpg | netcat localhost 2020 > /ziel/bild.lep
$ netcat localhost 2020 < /ziel/bild.lep > bild.jpg
```

TABLE 1: Lepton vs Compression Tools

Action	Lepton	Bzip2	Zip	Gzip	Convert WebP*
Compressed file size	7.9	9.9	9.9	9.9	9.0
Compressing time (seconds)	5.0	2.5	0.4	0.5	30.8
Decompressing time (seconds)	2.1	0.9	0.09	0.10	2.4

* WebP works losslessly.

a maximum size of about 17MB; with larger images, the process aborts and the 00M SHORT_READ error message appears. Fortunately, most JPEGs are smaller than 17MB.

Lepton generally works in multi-threaded mode, employing multiple CPU cores if they are available. The `-single-thread` switch disables the multithreading feature, which will significantly slow down the conversion process but will release more resources.

Lepton also has a server mode, which you can run with the command:

```
lepton -listen=2020
```

LISTING 3: Searching a Directory

```
01 #!/bin/bash
02 for jpeg in `echo "${file $(find ./ ) |
03   grep JPEG | cut -f 1 -d ':'}`
04 do
05   /usr/bin/lepton "$jpeg"
06 ### comment out following line to delete JPEGs
07 /usr/bin/mv "$jpeg" jpegs/
08 ### remove comment mark (#) to delete JPEGs
09 # /usr/bin/rm -rf "$jpeg"
10 done
11 exit 0
```

LISTING 4: Converting Back

```
01 #!/bin/bash
02 for lep in `echo "${find . -name "*.lep"}"`
03 do
04   /usr/bin/lepton "$lep"
05 ### comment out following line to delete LEPS
06 /usr/bin/mv "$lep" lep/
07 ### remove comment mark (#) to delete LEPS
08 # /usr/bin/rm -rf "$lep"
09 done
10 exit 0
```

LISTING 5: Adding Inotify

```
01 #!/bin/bash
02 inotifywait -mrq -e create -e moved_to
03   --format %w%f /path/to/JPEG-directory/ | while read FILE
04 FILE2=`echo "$FILE" | cut -d '/' -f 5 | sed s/.jpg//g`
05 do
06   /usr/bin/lepton "$FILE" /path/to/LEP-directory/"$FILE2".lep
07 done
08 exit 0
```

In this case, Lepton listens on TCP port 2020 for incoming connection requests (Figure 2).

Line 1 of Listing 2 sends a JPEG image to the server; the command in line 2 reconverts the file. In this example, the source and target are on the same computer.

Benchmark and Comparison

We compared Lepton with other compression tools by comparing the output file size for a 10MB JPEG file. Table 1 shows the results of various compression programs.

As you can see in Table 1, not a single conventional compression algorithm succeeded in significantly shrinking a JPEG without losing quality. Lepton, however, was able to shrink the file, although it managed just 2 MB/s – far less than the advertised 5 MB/s encoding rate.

In One Go

The benefits of Lepton are most apparent in cases where you need to compress a lot of files. The shell script in Listing 3 recursively searches through the directory from which it was called, looking for JPEG files and automatically converting them to Lepton format. In you comment out

line 4 and remove the comment mark before line 6, the script deletes the output JPEGs, instead of moving them to the `jpegs/` directory. The script in List-

ing 4 converts Lepton files back into JPEG format.

In both cases, it is a good idea to adapt the scripts to local conditions. For example, if you want the scripts to automatically convert the images, replace the period character after `find` with the path to the directory where the scripts are located. Then create an appropriate Cron job that executes the script at the prescribed times.

Monitoring Folders

If you want to add even more automation, you can use Inotify tools [6] to monitor a directory for changes. Inotify works like a daemon that continuously checks whether anything has changed in the directory. If a JPEG file is added to the directory, you can thus automatically convert it to Lepton format. The script in Listing 5 gives you an idea of how to use Lepton with Inotify.

Be sure you don't put the LEP directory in a folder monitored by Inotify, or you will trigger an infinite loop.

Conclusions

Saving 20 to 25 percent of the space for a JPEG file might not seem like a great deal, but in today's mass of ever-larger image collections, the savings could quickly add up to several gigabytes of disk space.

Lepton is generally much better than all other solutions at compressing JPEG images, although the shortage of good documentation makes using the program difficult. Support for the Lepton format in common image-processing programs and image viewers is still pending, but it shouldn't take too much longer. ■■■

INFO

- [1] WebP: <https://developers.google.com/speed/webp/>
- [2] Dropbox: <https://dropbox.com>
- [3] How Lepton works: <https://blogs.dropbox.com/tech/2016/07/lepton-image-compression-saving-22-losslessly-from-images-at-15mbs/>
- [4] Lepton with GitHub: <https://github.com/dropbox/lepton>
- [5] Lepton README: <https://github.com/dropbox/lepton/blob/master/README.md>
- [6] Inotify Tools: <https://github.com/rvoicilas/inotify-tools>

The sys admin's daily grind: Searx

Peppered with Hits

It goes against the economic rationale to assume that commercial search engines have the best interests of users at heart when it comes to data protection and use. Sys admin Charly has found an alternative. *By Charly Kühnast*

Many Linux tools with low version numbers seem astonishingly mature and stable – and they are. Searx [1], which recently bounced up to 0.10.0, is definitely one of them. The developers describe it as “a privacy-respecting, hackable meta search engine.” If you enter one or multiple search terms in the search box, Searx forwards the request to up to 70 online sources. The results show you the sources from which they originate.

You can individually sort the sources by topic or disable them, if needed. In the General field, you will find the major players: Google, Bing, DuckDuckGo, Yahoo, Wikipedia, and Reddit. For images, the results come from 500px, Flickr, or Deviant Art. If you are looking for music, the program taps into Deezer, Google Play Music, Spotify, SoundCloud, or half a dozen Torrent sites. IT gives you GitHub, Stack Overflow, and the excellent Arch Linux Wiki.

The Science category may be quite thinly populated, but it has some jewels in the form of Wolfram Alpha, Google Scholar, and Microsoft Academic. Base

(the “Bielefeld Academic Search Engine”) specializes in searching for and within scientific work. For example, a quick query for “adenosine triphosphate” provides three times more hits than a Google science search on *scholar.google.com*. If you like, you can prioritize results from open access sources.

Data Protection and Cookies

Searx also loads more hits when you scroll through the results. What's new in the current version is the ability to restrict the search to a specified time period (Figure 1). The developers are particularly proud of the fact that their tracker dog handles the user data with care. The Python source code is open, and Searx does not store user profiles. By default, Searx does not even store cookies. That only happens if you change settings and save the changes.

A couple of dozen Searx instances [2] worldwide are available for global use by everyone. The development team itself also runs an instance [3]. And, if you don't trust external services, you can simply build your own Searx server. Minimal hardware such as a Raspberry Pi is sufficient for this, and there is a how-to online [4]. There is no escaping the facts: Searx is amazingly mature for version 0.10.0. ■■■

INFO

- [1] Searx project:
<https://asciimoo.github.io/searx/>
- [2] Searx instances:
<http://stats.searx.oe5tpo.com>
- [3] Instance run by the developer team:
<https://searx.me>
- [4] Installing Searx on a Raspberry Pi:
<https://asciimoo.github.io/searx/dev/install/installation.html>

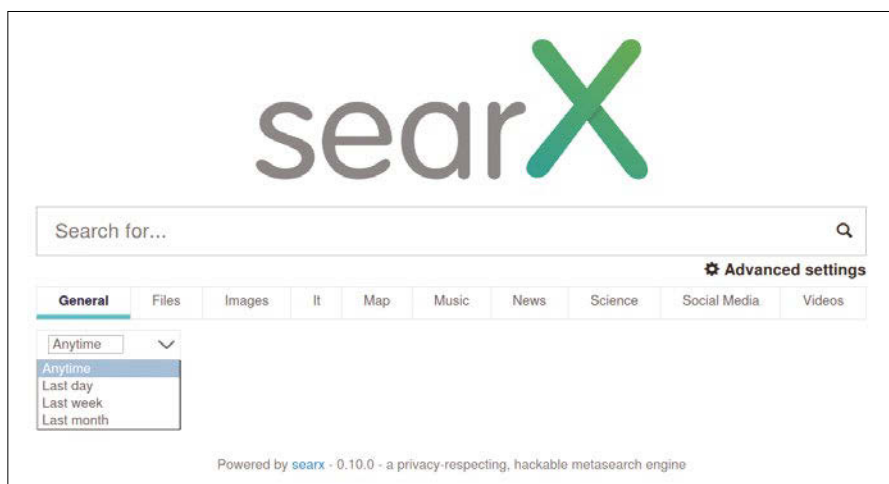


Figure 1: The latest version of Searx, which stands out because of its good data protection, can also restrict results to a specific period of time.

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.



Now Appearing on

APPLE NEWSSTAND

New age convenience...

Our inspired IT insights
are only a tap away.

Look for us on
Apple Newsstand
and the iTunes store.

Download
a FREE issue of
each publication
now!





Z-Wave: Basics of wireless technology for the smart home

HOUSE RADIO

Z-Wave connects components in the smart home and ensures that remote commands from the control station take effect on real household appliances. We look at how it works, its range, the security of the protocol, and some basics that every Z-Waver should know. *By Mike Schilli*

The door lock opens when the owner's mobile phone approaches the entrance. The light in the apartment switches on when the motion detector senses a being walking around and the light sensor determines that it is night. The garage door closes automatically when the car drives in, and a controller detects that the absent-minded driver once again forgot to press the close button on the garage gate before leaving through the back door, so the gate now closes itself.

These automatic actions are no longer a smart home fiction; they are available right now. However, before pressing the "Buy" button, house owners have to decide which method to use when sensors

are picking up signals and report them to the controller, which then in turn initiates actions to control real physical household devices.

Simply Wireless

Not everyone is likely to build their own new home so they are able to route cables in the walls for all kinds of playful extravagances; wireless solutions work around these limitations for the common man. In the 1990s, hobbyists who had a knack for tinkering, mainly in the United States, began to switch lamps or even household appliances on and off with a technology called X10. Signals were transmitted either via power lines or wireless RF technology. But X10 did not offer reliable communication; it was often impossible to say with certainty whether a device had actually received a transmitted signal.

In the United States, where imperfection and unreliability are more easily accepted, the product fared quite well, but

in Europe, it never caught quite on. Along with other wireless alternatives (e.g., ZigBee [1], EnOcean [2], and good old WiFi), the Z-Wave Protocol was launched in the 1990s [3] by a Danish company named Zensys. In 2008, chip maker Sigma Designs [4] acquired the rights to Z-Wave.

The industrial consortium Z-Wave Alliance, to which some 300 corporations with approximately 1,000 certified products belong today, guides the development and the certification of new products. Hobbyists can build small solutions for little money that they then can extend piece by piece to a professionally automated smart home with dozens of components.

Acknowledgement Needed

On a wireless Z-Wave network, the brains of the system are called controllers. They send commands to actuators, called slaves, which then initiate actions. As an example, a laptop with a USB stick

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>.



Figure 1: The Z-Wave DSA02203 USB stick controller by Aeon Labs lets a Linux system send wireless commands to Z-Wave slaves.



Figure 2: The Z-Wave slave, in the form of the Aeon Labs DSC06106 network switch, receives commands for switching the consumer on and off wirelessly.

like the one shown in Figure 1 acts as a controller and uses a wireless radio connection to control the mains switch in Figure 2, the slave, which turns electrical consumers such as a lamp or another computer on and off.

To set up new gadgets on the network, you will usually find a button to press on the device, for example at the side on the mains switch in Figure 2. This causes the device to advertise its presence on the network and any listening controllers, such as the USB stick in Figure 1, whom the user also switches to a special integration mode. The controller then picks up the beacon from the new gadget and teaches it its new name on the network, making sure that it knows which wireless commands it needs to pick off the air and respond to going forward.

The process is similar for deregistering a device from the network; the user typically has to confirm on the unit itself, usually by pushing a button, that the device is now leaving the collective.

Once integrated, the controller and slave rely on a handshake protocol in which the slave confirms the receipt of commands sent by the controller. When the slave's ACK reaches the controller, as shown in the top part of Figure 3, the controller knows that the slave has received and validated the packet with the command. It says nothing about whether the slave was able to execute the command or whether the desired effect has occurred. A subsequent status request sent by the controller to the slave can obtain clarity on the matter.

Until recently it was only possible to learn the details of the protocol by pur-

chasing an expensive SDK from Sigma Designs, but the company finally released the Z-Wave specification online in September 2016 [3]. The Z-Wave Consortium still insists on the certification of new equipment, so nobody can simply dump new Z-Wave devices on the market. To certify a new gadget, you need to register as a commercial development partner. However, the specification allows hobbyists to write their own control software, for example for the controller in Figure 1.

Fault Tolerance

In the case of an unreliable wireless connection, it can happen that, as shown in the middle part of Figure 3, the command from the controller does not even reach the slave, or that the slave receives the command and sends an ACK, but it does not make its way back to the controller (bottom of Figure 3). In both cases, the controller simply assumes that the command failed and repeats the procedure up to two times. If the third attempt also fails, the controller reports an error and assumes that the connection to the slave is down.

Bridging Unreachable Distances

Depending on the nature of the house, the Z-Wave signal between the transmitter and receiver can traverse a number of rooms. The range varies, depending on the version of the device and thickness of the walls to be crossed, and is usually around 30 to 100 feet. Z-Wave's proprietary mesh network allows greater dis-

tances to be bridged, however. In addition to their primary tasks as switches or sensors, many Z-Wave devices also act as routers for other signals that a controller wants to send to recipients that are out of range.

For this to happen, the controller tells the devices to locate and report their neighbors during initialization. If the controller then wants to send a message

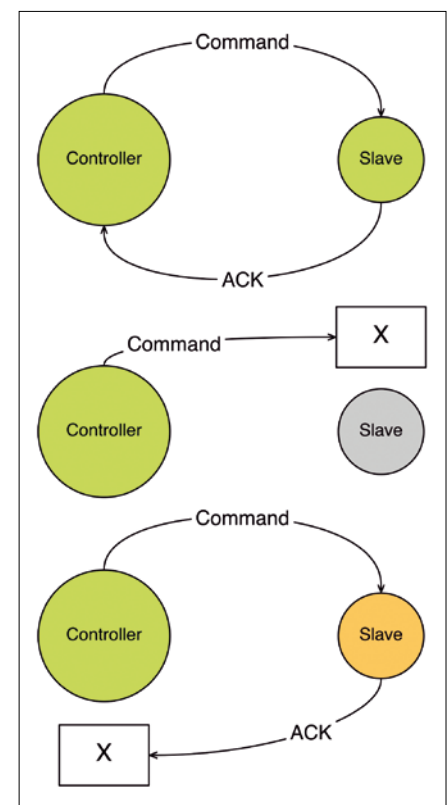


Figure 3: Z-Wave's handshake protocol requires the slave to confirm the commands sent by the controller by returning an ACK.

to one of these devices later on, but can't because of the limited range, it sends a packet with the routing information to the device that previously reported this neighbor. The device then forwards the package, on the basis of the received routing table, to the target, which can subsequently communicate back to the controller along the same path in the opposite direction (Figure 4).

Based on this procedure, Z-Wave builds a mesh network between the devices that can dynamically adjust its routes in case of an error. Each device on the network receives its own node ID; all the devices in a home are grouped under a home ID. This means that a particular device knows with certainty that a packet whizzing around in the ether is intended for it, and not for another node on the mesh network.

Power Supply

Z-Wave devices come in three different categories in terms of the power source and corresponding performance profiles. Devices with a steady power supply via a socket outlet permanently listen for wireless signals and process them immediately on receipt. They therefore often provide router functions for other devices in the mesh.

In the case of battery-powered devices (e.g., door locks), this is not easily possible if you want the battery to last for a reasonable amount of time. The device wakes up at regular intervals, checks for radio signals in the air, and then starts to decode and possibly process them.

The third category includes devices, such as remote controls, that do absolutely nothing in idle mode and only wake up and send signals at the push of a button. They are not usually reachable for a controller in the opposite direction.

Regional Differences

The release of radio spectra is subject to the control of national governments, so Z-Wave devices operate on different frequencies depending on the country. For example, Z-Wave devices in the United States send on 908.4MHz, whereas devices in Germany use 868.4MHz. Although a transmitter and a receiver from the same country will work elsewhere, the combination cannot be legally operated on an unapproved frequency. Instead, users need to purchase devices with the certified frequencies for their countries.

Do the Waves Cook Your Brain?

The question still to be investigated is whether the wireless connections affect the health of the smart home's inhabitants. According to Dr. Christian Paetz [5], Z-Wave devices emit about 1/1,000th of the energy emitted by a mobile phone during a phone call. Moreover, Z-Wave controllers and switches are only sporadically active and are not usually found close to body; their use around the house should thus be safe.

Is It Secure?

In 2013 the manufacturer of a certified Z-Wave door lock made a blunder in

the implementation of the AES encryption they used, which enabled attackers to remotely open the lock and report to the controller that the home was locked even though it was open. Otherwise, no vulnerabilities are currently known.

That said, the security analysis of the Z-Wave protocol is still in its infancy, and the bad guys will certainly be pushing forward their efforts given the increasing spread of such a lucrative target.

Outlook

Of course, this short introduction, by its nature, only scratches the surface. For a more detailed discussion, the book by Paetz [5] was until recently the only available public source. More recently, however, the public specifications by Sigma Designs on the Z-Wave standard [3] provide interested users with more in-depth information directly from the manufacturer, especially at a hands-on technical level.

The practical application of a lamp controller with Linux was demonstrated at the beginning of the year in an article on [6]. Experience has shown that many hobbyists initially start with two or three Z-Wave components. But as users discover a taste for the automatic functions, it's been reported that small installations tend to be growing slowly until they reach the level of comfort provided by a smart home. ■■■

INFO

- [1] ZigBee: <http://www.zigbee.org>
- [2] EnOcean: <https://www.enocean.com/en/technology/>
- [3] Z-Wave public specification: <http://dev-z-wave.pantheonsite.io/design-z-wave/z-wave-public-specification/>
- [4] Z-Wave Networking Basics: <http://dev-z-wave.pantheonsite.io/wp-content/uploads/2016/08/APL13031-2-Z-Wave-Networking-Basics.pdf>
- [5] Paetz, Christian. *Z-Wave Basics*. Amazon Digital Services LLC, 2015, <https://www.amazon.com/Z-Wave-Basics-Christian-Paetz-ebook/dp/B00Y6GXCRC>
- [6] "Z-Wave" by Mike Schilli, *Linux Magazine*, issue 184, March 2016, pg. 54, <http://www.linuxpromagazine.com/Issues/2016/184/Perl-Z-Wave>

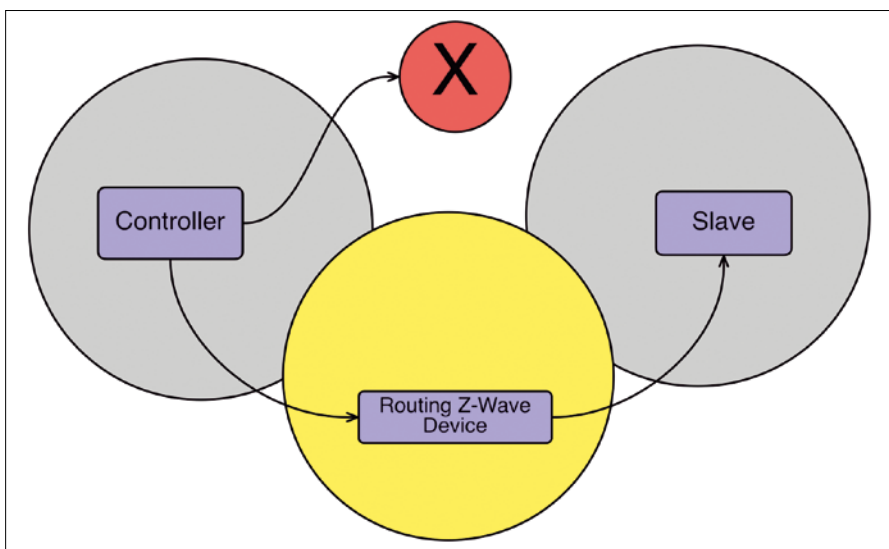


Figure 4: The controller cannot reach the slave, but a third Z-Wave device acts as a router and enables a communication path.

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

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



We're also in
Google Play
and the
iTunes Store!

Download on the
App Store

GET IT ON
Google play



iTunes Store



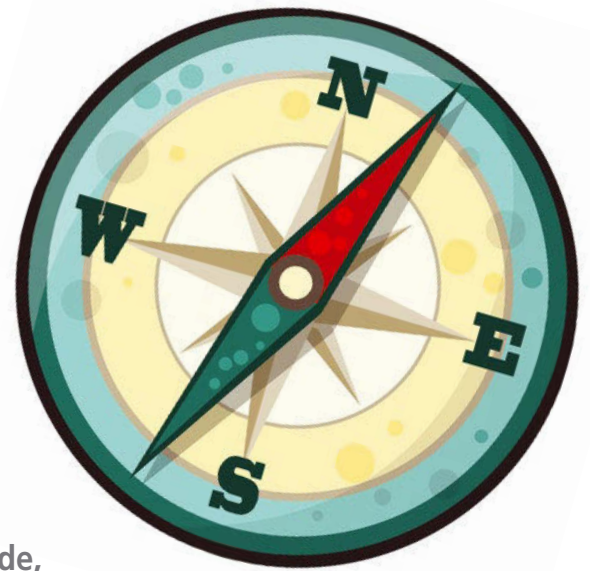
Google US



Google UK

Source code browsers

Navigators



If you've ever struggled to get a sense of someone else's code, the right tool could save you hours of grepping. *By Valentine Sinitsyn*

Open source is all about code. Contributors read tons of code, of which they have written only a small fraction. Being able to comprehend a program is crucial to the contribution process, and free software is all about contribution. In other words, you need tools to read and understand code.

These tools are called “source code browsers” or “source navigators.” Linux has many of them, and normally they fall in two large categories. Older ones implement their own (simplified) parsers to recognize language symbols, such as function definitions, and record their location in the source code. This works fast and reasonably well, yet most programming languages have complex grammar that simplified parsers can't fully embrace. Newer browsers rely on the tool set to build an abstract syntax tree (AST) [1]. This makes indexing more precise, but also slower and more cumbersome to generate. Choosing one approach over another depends on the situation, and I hope this text provides you with some guidance.

Ubiquitous Ctags

Ctags is the de facto standard for source code indexers in Linux. As the name suggests, it builds on the “tag” concept. Put simply, a tag is a syntax construct that has an index entry, such as a class, function, or macro definition. This index comes in a so-called “tags” file, and the main purpose of the non-interactive ctags command is to generate tags files from source-code trees. Tags files have a well-defined format, and virtually all code editors in Linux understand it.

Despite the “C” prefix, Ctags supports a wide range of programming languages.

The output of

```
ctags --list-languages
```

varies between Ctags flavors, but C/C++/C#, Java, Perl, Python, and a few dozen others are usually there. Ctags guesses a programming language by the file's suffix. If it guesses wrong, you can override its choice with `--language-force`. For each programming language, Ctag recognizes different kinds of tags, and

```
ctags --list-kinds
```

displays known tags for each supported language.

To build an index, you supply ctags a list of files to consider. For example, ctags `-R` treats subdirectories recursively; `-f` sets the output filename. This should be enough in most scenarios. However, because Ctags is “neither a preprocessor nor a compiler” [2], it may get certain things wrong. If this is your case, use the `-I` switch to ignore or substitute identifiers that require special handling.

A tags file is text, yet it was designed to be machine-readable. Ctags can also build a human-friendly tabular cross-reference if you call it as

```
ctags -x <other arguments>
```

For each tag, Ctags reports its name, kind, and line number in the source code. The original output is in fact quite similar, except it stores not a location, but the EX editor command to use to find the tag. Most often, it's a regular expression search, but Ctags provides `--excmd` and few other com-

mand-line switches to adjust this behavior.

The description above applies to the original Ctags. In so-called “Etags mode,” the file format is different, and `--excmd` and friends are just ignored. “E” was for Emacs originally, but now many other programs (e.g., Midnight Commander's internal editor) recognize tags in this format (Figure 1). You can often tell the format by the filename: tags is for Ctags, while TAGS is for Etags.

The most popular Ctags implementation to date is Exuberant Ctags [3], and it is likely what your distro ships. It provides both ctags and etags commands; you can also enable Etags mode with the `-e` switch to ctags. Universal Ctags [4] are also gaining momentum. As the Exuberant Ctags homepage suggests, it isn't actively maintained now, whereas Universal Ctags attempts to continue the development and sports completely rewritten C/C++, Python, and HTML, as well as many new parsers (e.g., for Rust). The downside is you'll probably need to compile the program yourself. Luckily, the homepage describes this process in detail.

Compared with original Ctags, which indexes where the tag was defined, Ultimate Ctags can also track where it was referenced: see the *Reference tags* section on the *Docs* page. This brings Ultimate Ctags on a par with the second nominee, Cscope.

Venerable Cscope

Can you imagine software written in the PDP-11 era that still remains in use today? Can you imagine that the software was made free (as in speech) thanks to an infamous SCO Group predecessor? Meet Cscope [5]: a C code browser with some (limited) support for C++ and Java, born

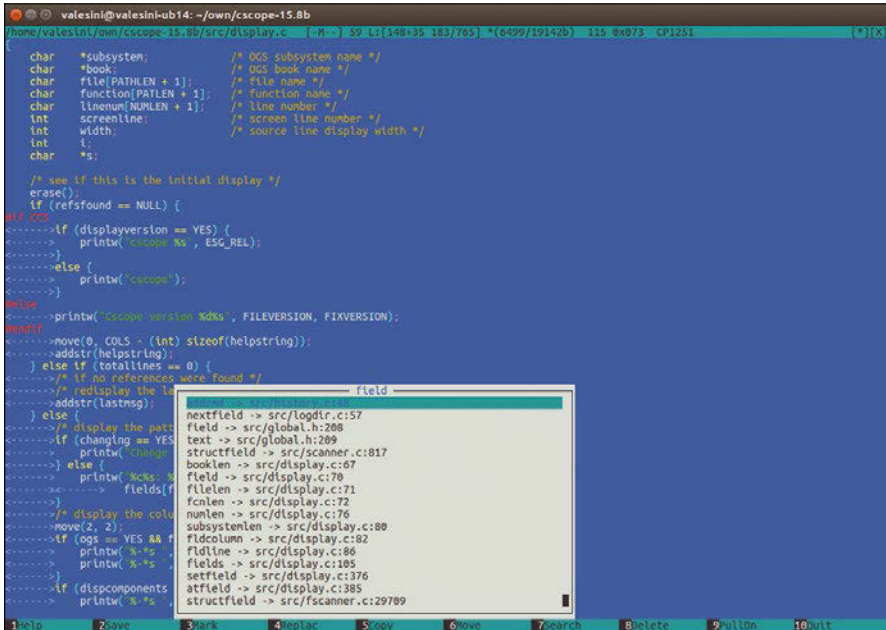


Figure 1: With a tags file, even humble mcedit looks much more IDE-like. Just don't forget it wants an Etags format.

in Bell Labs, and open-sourced by Santa Cruz Operation in 2000. Cscope was briefly introduced in a *Linux Voice* cover feature last year [6], and now it is time to pay this tool the respect it truly deserves.

Cscope should be available in your package manager. Before you start using it, you'll need a cross-reference database for your source code, which you can build separately with `cscope -b`. This is not a must, because when you launch Cscope's curses-based interface, it automatically indexes all C, Bison/Yacc, and Lex files in the current directory. Add the `-R` switch to recurse into subdirectories, which is usually what you want to do. Some projects even provide dedicated makefile targets to generate a Cscope cross-reference database. For example, `make cscope` in Linux kernel source code produces a so-called inverted index that makes symbol lookup a bit faster. Should you want to achieve the same effect on your own, run `cscope -bq`. Also, consider `-k` to enable the "kernel mode." In this mode, Cscope doesn't look into standard locations like `/usr/include`, because kernels (and other low-level code) don't use them. On startup, Cscope detects changes to source code and rebuilds the cross-reference as necessary. This makes subsequent launches faster. Note you still need to tell Cscope where to look for source code, even if the database already exists. To trigger a rebuild from within Cscope, type `Ctrl + R`.

Cscope records not only where symbols were defined, but also how they were used, so you can find all expressions that

ACK

Source-code files are text. Language tokens are pieces of text. The first tool that springs to mind when you think about searching for text is Grep.

You can surely use `grep` to navigate source code, but you have a better alternative: `ack` [7]. `ack` is pure Perl (Are you scared yet?), and you can install it from your distro's package manager or via CPAN.

What makes `Ack` a better `Grep`? Two things: It's fast, and it's designed to search code, which means fewer keystrokes for common tasks. `Ack` ignores non-code directories (e.g., `.git` or `.svn`), backup files, and the

involve a given variable, or functions calling a given function, or functions that the given function calls. Many other tools restrict your searches to C language identifiers. In Cscope, you can `grep` for arbitrary text strings and regular expressions (see the info box titled "Ack"). For dessert, Cscope can look up a file by name or find all files that `#include` the specific header.

Cscope's curses-based interface splits the screen into halves. You enter search terms in the lower half and get results in the upper (Figure 2). Cscope supports POSIX extended regular expression syntax. Filenames allow partial matches while C symbols don't. Putting `foo` in the *Find this file* field matches `foo.c`, `foo.h`, and `foobar.c`. Putting `foo` in the *Find this C symbol* field matches the first, but not the second, expression below:

```
void foo();
int foobar = 1;
```

The `Tab` key lets you switch halves, and you select fields with arrow keys. Some

like, and it doesn't need `-R` to recurse into subdirectories. In a multilanguage project, you can tell it to look only in Python sources with `ack --python`. `Ack` sports Perl regular expressions (guess why) and happily highlights matches it finds.

`Ack` can't do the semantic analysis that `Clang` can – nor can it brew your coffee. However, none of the tools I cover here can do a free-text search (except Cscope), so `Ack` certainly deserves being in your toolbox. Don't forget to share the `~/ .ackrc` snippets you found most useful in Linux Voice forums [8]!

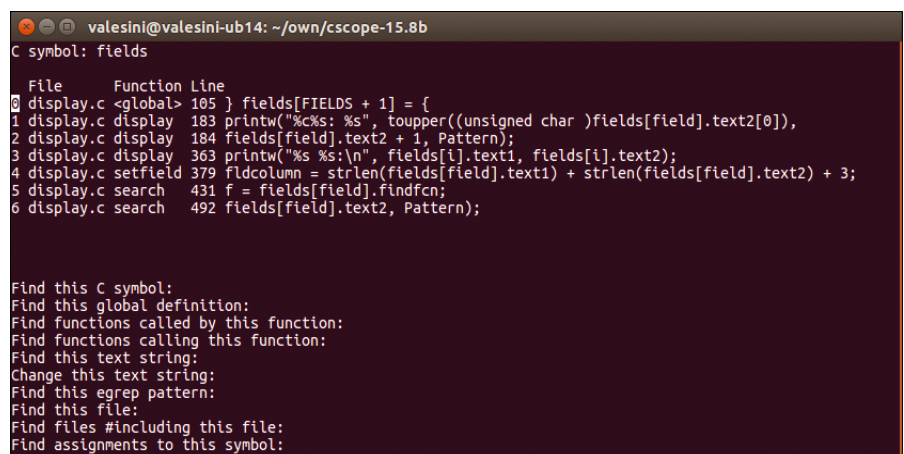


Figure 2: Cscope is a powerful tool that easily copes with the Linux kernel. Here, it browses its own source.

symbols, such as ^, are reserved (see `cscope(1)` [9]). To enter them, first, type \ as an escape character. For each search result, Cscope displays the location (file, function, and line number) and some context. It also assigns the result a single-letter hotkey you can type to open it in the editor (`$CSCOPE_EDITOR`). The spacebar switches search results pages. You can save the results in a file with > or >>. Should you need them later, load this file with < or `cscope -F`. To refine the results, type ^ or |. Both filter through an external shell command. Entering ^ replaces the original results, whereas | simply displays filtered lines and keeps the results untouched.

A few other hotkeys are available. `Ctrl + C` toggles case sensitivity. `Ctrl + Y`/`Ctrl + A` repeat your last search, and `Ctrl + B`/`Ctrl + F` do the same, yet in a search field above or below the current one. This comes in handy if you typed your query in a wrong box. For those accustomed to GNU Readline, history support in Cscope may feel limited, and it probably is. Pressing ? brings the help page, and `Ctrl + D` exits Cscope.

The man page [9] describes more hotkeys and command-line switches. I suggest you spend some time learning them, because it greatly improves your Cscope experience. Cscope also runs in line mode or as a Vim extension. I leave exploring those options as an exercise to a curious reader (i.e., you).

Woboq Code Browser

Once you understand the traditional tools, you can compare them to Clang-based alternatives. Naturally, this limits support to C/C++, but Clang is a real C/C++ compiler, so it should have no problem handling even the most convoluted syntax constructs, provided they are correct.

On the other hand, if you call Clang to index your code, you should supply it all the information the build system (CMake, Autoconf) normally does. This is not the case with Ctags or Cscope, which can simply scan files one by one, looking for specific patterns, such as function declarations. For Clang, build

information usually comes via a JSON compilation database (`compile_commands.json`). CMake introduced this format first, and in a nutshell, it contains the list of source files and exact commands used to build them.

Woboq Code Browser [10] builds on Clang and produces a set of annotated HTML pages showing a project's source code. A bit of JavaScript makes them interactive, and no code is required on the back end; yet, you'll probably want to serve these pages with a web server, because most browsers don't allow Ajax requests to `file://` URLs by default. (That's a security flaw.)

You'd want to compile Code Browser yourself, because it probably hasn't made its way to your distro repositories. It uses CMake, which you'll need to tell where to find the `llvm-config` tool on your system. On Ubuntu, it's at `/usr/bin/llvm-config`; otherwise, the process is straightforward.

How you build `compile_commands.json` depends on the build system of the project you are trying to index. If it's CMake, the day just got better, because you only need to use

```
cmake -DCMAKE_EXPORT_COMPILE_COMMANDS=ON
```

For other build systems, say Autoconf or Qmake, Woboq Code Browser provides a `fake_compiler.sh` helper script. However, there is a better tool: Build EAR (Bear), which you'll probably find in your package manager, and it's available online [11]. Bear sets `LD_PRELOAD` to inject a dynamic library that traces calls to the compiler and collects command-line arguments. To use Bear, you just need to prefix a `make` invocation with bear:

```
bear make
```

The bear `--help` command lists a

few available options. With the JSON compilation database ready, Code Browser can index your project (Listing 1).

This implies you did an in-tree build of Code Browser. The `$BUILDDIRECTORY` argument is where `$PROJECTNAME` (your project) is built, and `$VERSION` is the project's version. The `$OUTPUTDIRECTORY` argument should be set to wherever your web server looks for static HTML (e.g., `~/public_html/$PROJECTNAME`). The `-a` switch tells `codebrowser_generator` to process all source code found in `compile_commands.json`. The second command builds `index.html` for each subdirectory in the project, and the last line copies scripts and stylesheets.

The end result is worth the fuss. You may choose a theme of your liking (Qt Creator/KDevelop/Solarized) to feel at home. Mouse over a symbol to see a pop-up box containing the description and references. For global symbols, the reference kind (e.g., *value read* or *address taken*) is also shown. Click on a symbol to jump to the declaration. Location history is also supported, yet it is bare bones, with no way to clear the history and no indication as to which files history items belong. Similarly, the sidebar on the right contains definitions collected from the current source code file (Figure 3); however, you can't tell whether the definition is a function, variable, or type.

Another downside is licensing. Code Browser is dual-licensed (CC BY-NC-SA 3.0 [12] and proprietary), so I feel it's okay to use the open source version to index open source code, as long as you keep the Woboq branding. For anything

LISTING 1: Code Browser Indexing

```
./generator/codebrowser_generator -b $BUILDDIRECTORY -a -o
$OUTPUTDIRECTORY -p $PROJECT:$BUILDDIRECTORY:$VERSION
./indexgenerator/codebrowser_indexgenerator $OUTPUTDIRECTORY
cp -rv ./data $OUTPUTDIRECTORY/./data
```



Figure 3: To showcase Code Browser, Woboq publishes indexes for several well-known software projects, including the Linux kernel.

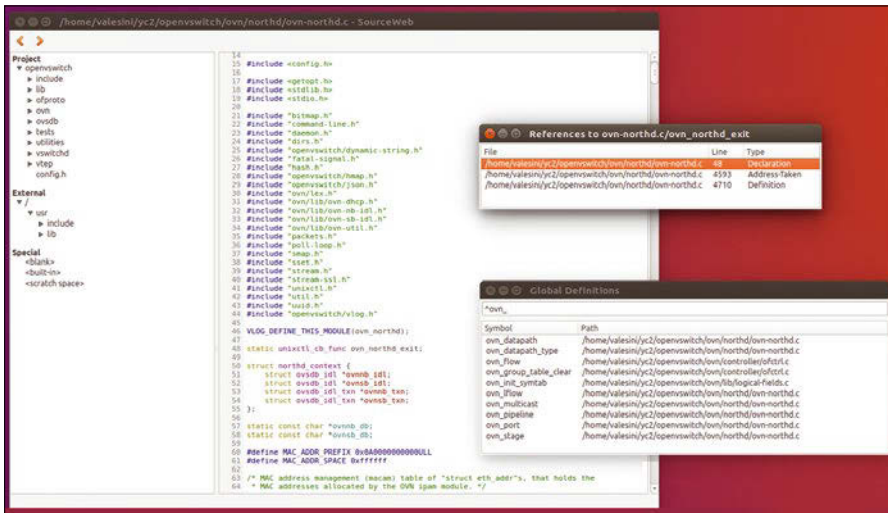


Figure 4: SourceWeb is a functional yet pretty basic source-code browser that just needs some love. And a new maintainer.

else, you'll probably need a commercial license. You should contact Woboq directly if you are serious about Code Browser deployment.

Abandoned SourceWeb

SourceWeb [13] is another example of a Clang-based technology. Again, you'd probably want to build it yourself. You'll need GCC, Clang, and Qt development headers, which should be readily available in your package manager.

Indexing a project with SourceWeb is a two-stage process. First, you create the JSON compilation database. For projects not using CMake, SourceWeb provides its own btrace tool. It works reasonably well and relies on the same mechanisms as Bear. Choosing one over another is mainly a matter of taste. However, btrace doesn't build compile_commands.json directly: it generates a logfile you'll need to postprocess with the sw-btrace-to-compiledb command. You might prefer Bear to skip this extra step.

Then, you run

```
sw-lang-indexer --index-project
```

in the project's build directory to generate an index file. The latter is a binary database that the SourceWeb GUI uses to do lookups, much like tags or cscope.out. This will take some time. For a large codebase, you might want to add --incremental to retain per-file indexes and speed up future index rebuilds.

When you have an index, run sourceweb index to launch SourceWeb navigator's Qt-based GUI. You see the project outline in the left pane; files are opened in the right (Figure 4). You can browse for files, global definitions, and symbols. To this end, SourceWeb supports regular expressions. Right-click on any symbol in the source code viewer and choose Cross-references. You'll learn how the symbol is used. This is a feature set you'd expect from a code browser, yet it's pretty basic. Moreover, the application misbehaved once and

crashed during startup another time on an Ubuntu 16.04 LTS box.

SourceWeb feels interesting but somewhat incomplete, especially if you think about its GUI. And it's a pity because the project never saw official release and now seems to be unmaintained. It is about 15K source lines of code (SLOC) and BSD-licensed, so if you are looking for a pet project, please give it a chance.

Here you are. The four options I just covered aren't the only ones available in Linux. Yet, I think they are the most promising. Personally, I use Cscope and Woboq Code Browser most often, but your mileage may vary. To help you decide, I summarized the tools in Table 1, so you can compare them side by side. Shall you decide to join the Universal Ctags team, or pick up SourceWeb? Please let us know! ■■■

INFO

- [1] "Code compilation" by Valentine Sinityn, *Linux Voice*, issue 22, January 2016, pg. 94, <https://www.linuxvoice.com/download-linux-voice-issue-22/>
- [2] ctags(1) man page: <https://linux.die.net/man/1/ctags>
- [3] Exuberant Ctags: <http://ctags.sourceforge.net/>
- [4] Universal Ctags: <http://ctags.io>
- [5] Cscope: <http://cscope.sourceforge.net>
- [6] "System Calls," *Linux Voice*, issue 29, August 2016, pg. 20
- [7] Ack: <http://http://beyondgrep.com>
- [8] LV forums: <http://forums.linuxvoice.com>
- [9] cscope(1) man page: <https://linux.die.net/man/1/cscope>
- [10] Woboq: <https://code.woboq.org>
- [11] Bear: <https://github.com/rizotto/Bear>
- [12] CC BY-NC-SA 3.0: <https://creativecommons.org/licenses/by-nc-sa/3.0/>
- [13] SourceWeb: <http://rprichard.github.io/sourceweb/>

TABLE 1: Source Code Navigators

Name	Version (last release)	Active	License	Parser(s)	Interface	Language support	References [a]
Exuberant Ctags/ Universal Ctags	58 (7 Sep 2009)? [b]	No/Yes	GPLv2	Custom, regex-based	Command line [c]	Various	No/Yes
Cscope	15.8a (4 Apr 2015)	Yes	BSD	Custom	Curses [d]	C/C++/Java [e]	Yes
Woboq Code Browser	2.0.1 (10 Aug 2016)	Yes	Dual	Clang	Command line/ Web	C/C++	Yes
SourceWeb	? [b]	No	BSD 2-clause	Clang	Command line/Qt	C/C++	Yes

a. The tool not only jumps to definitions but also shows where (and how) they are used.

b. No official release at the time of writing.

c. Generator. Different front ends are available for browsing.

d. Third-party GUIs are available.

e. C++/Java support is limited.

Klaus Knopper answers your Linux questions

Ask Klaus!

By Klaus Knopper

Controlling Brightness

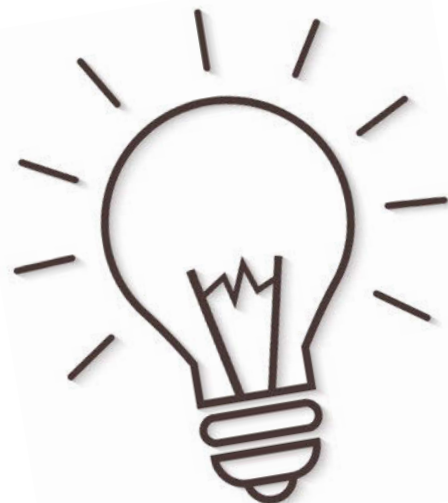
? On my laptop, the function key combinations that are supposed to change the display's brightness are not working with Linux. Is special software necessary in order to activate those function keys?

💡 Most manufacturers of notebooks use a non-standard keyboard assignment for brightness, WiFi enable/disable, and so on – “non-standard” meaning that there really isn't much of a standard keycode available for the extra functionality. Proprietary operating systems use “drivers” to map extended keyboard events to certain actions, as is done similarly in most Linux distributions using the ACPI daemon with vendor-specific configuration files in `/etc/acpi/events` and `/etc/acpi/actions` provided by packages such as `eeepc-acpi-scripts` and `acpi-support` (which contain known mappings for different notebook brands). Finding the correct settings and writing your own scripts for handling events can be a very complex task, but in some cases, you have an easier way to solve this problem.

Boot options can change the default behavior in the Advanced Configuration and Power Interface (ACPI) firmware to handle requests for brightness internally, instead of passing the key event to the operating system. One example for Knoppix is:

```
knoppix64 acpi_osi=Linux ↵
acpi_backlight=vendor
```

The `acpi_osi=Linux` option chooses some Linux-friendly settings (also concerning power management), and `acpi_backlight=vendor` attempts to wire key presses of the “brightness” function keys directly with the backlight controller,



Designed by Freepik

which may or may not work better than the supplied `acpi-support` scripts.

If you can confirm that these boot command-line settings work as expected, you might want to add them to the `APPEND` or `KERNEL` boot options in your favorite bootloader.

Forensic Mode

? What is “forensic mode” in Knoppix? Knoppix 7.1.1 has just been released, and I've read about a “forensic” mode that is only briefly described in the `knoppix-cheatcodes.txt` file:

```
knoppix forensic
Don't use swap and mount read-only
```

What does it really do?

💡 Forensic mode was requested by experts that work in computer forensics (i.e., analysis and recovery of hard disk data – even previously deleted data – as part of rescuing lost data or for evidence in criminal prosecution). For this task, all changes to the stored data are absolutely unwanted. Knoppix does not normally write to hard disk, with two exceptions:



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



1. Configured Linux swap partitions that are not currently used as hibernation space will be reused.
2. Disk partitions containing filesystems that are mounted by user choice can be written to.

Adding the keyword `forensic` to Knoppix boot options will automatically disable the use of existing swap partitions and write-protect all detected partitions on the block layer using:

```
blockdev --setro <partition>
```

Therefore, in forensic mode, attempts to mount partitions read/write will not work unless the user “unprotects” the partitions first using

```
blockdev --setrw <partition>
```

(e.g., when saving rescued data to an external hard disk). If certain filesystem types require being set read/write in order to attempt a repair of the superblock, a copy can first be made safely by means of `dd_rescue`, so that the original data is never changed.

For normal usage scenarios (i.e., taking advantage of writable media for saving data transparently from the running system), forensic mode is rather impractical. ;-)

Broadcom WiFi Drivers

? I can't help noticing that you changed the default driver for some Broadcom chipsets from the free, kernel-included drivers to the “non-free” `broadcom-sta` driver (`wl`) in Knoppix 7.7.1. What is the advantage or disadvantage over the free drivers?

💡 Although the standard kernel drivers for

Broadcom WiFi chipsets usually work well enough for daily use, some users have reported slow or unstable WiFi connections, even with the vendor-supplied firmware copied to `/lib/firmware`. The `broadcom-sta-dkms` driver package, which is provided in Debian as a self-building package, contains code and firmware provided by the Broadcom chip manufacturer with some license restrictions that might disqualify the driver in Debian main, in the sense of not being “open source,” even though partial source code is provided.

On the one hand, I can well understand that following regulations and legal restrictions for WiFi in different countries encourages “unchangeable” drivers. On the other hand, the closed source parts of the driver make it difficult to check for security flaws or unwanted functions, and it is also disallowed to modify the driver for supporting chipsets from other vendors, making it “proprietary,” which I'm somewhat reluctant to accept in Knoppix.

Nevertheless, test results from some of the Knoppix beta testers report an enhancement of throughput and bandwidth gain – a download speed factor of up to 2x – using the `broadcom-sta` driver as a replacement for the kernel internal driver. Also, network disruptions because of weak signals seem to occur much less often.

For these practical reasons, we gave the `broadcom-sta` driver another try, but with the option to switch back to the free driver by changing preferences in `/etc/modprobe.d/broadcom-sta-common.conf`. ■■■



Designed by Flaticon

GOT CLUSTER?



Tune in to the HPC Update newsletter for news, views, and real-world technical articles on high-performance computing.

hpc.admin-magazine.com/Newsletter

Configuration files for converting audio files

Simple Conversion

Learn how to convert your music files to Ogg Vorbis for casual use or to Flac for more refined playback. *By Bruce Byfield*

M P3 is the most common audio format in computing. However, free software boasts two superior audio formats: Ogg Vorbis [1], a lossy format like MP3 but with better compression and sound quality, and Flac [2], a lossless high-quality format, whose files are usually several times larger than MP3 or Ogg Vorbis. Casual listeners might want to convert music files to Ogg Vorbis, while audiophiles might prefer Flac. Neither conversion is difficult, but both require some organization.

In both cases, the first difficulty of conversion is that it requires several dependencies. The overall process is run by `abcde` [3], a CD reader like `cdparanoia` [4] for ripping CDs, the `oggenc` codex [5] for Ogg Vorbis conversion, and the Flac codex for Flac conversion.

BRUCE BYFIELD

Bruce Byfield is a computer journalist and a freelance writer and editor specializing in free and open source software. 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>

Depending on your distribution, some of these dependencies may be installed by default, but you should check before attempting any conversion. You may also require an Internet connection.

All conversions are made with `abcde`. However, the second difficulty is that you will probably require a dozen or more options to make any conversion, opening countless opportunities for mistakes. Once you've done one conversion, the Bash history can speed the progress, but an even greater help is a custom `/etc/abcde.conf` file that contains commonly used options, allowing you to concentrate only on unusual circumstances as you structure the command.

`Abcde` installs with its own `/etc/abcde.conf`, but you should look over all its options, not just the ones for a particular conversion, and read the comments above each option. For convenience, you might even want to write custom `abcde.conf` files for conversions to each format, using the `-c FILE` option to point to the one you want to use. However, most settings are probably the same no matter what the conversion, so you can usually manage with a single file, with separate entries – one per line – for each conver-

sion you do regularly. Some fields in configuration files you may choose to ignore as unimportant.

To avoid confusion, remember that manually entered options override all configuration files, custom configuration files override one in your home directory, and one in your home directory overrides the default one in `/etc`. By editing the `abcde` configuration files carefully, you can greatly reduce the length of the command you need to enter, perhaps to something as basic as:

```
abcde -c /etc/abcde-ogg.conf Z
DISK/FILEPATH
```

However, the options used in a conversion can be far more numerous. What follows are the configuration settings most likely to be useful. The purpose of the settings not mentioned here can usually be understood from the comments.

Important CDDB Options

An `abcde` configuration file falls into several categories (Figure 1). These are not named, but, if you choose, you might add a comment to identify them. Most of the fields in each category are commented out, so you will have to remove

Lead Image © swvx, 123RF.com

```
# System defaults for abcde version 2.2.x
# Nothing in this file is uncommented by default.
#
# If you wish to override these system-wide settings, create your own
# .abcde.conf file in your home directory.

# CDDB options
# Choose whether you want to use CDDB or Musicbrainz. Default is CDDB
#CDDBMETHOD=cddb

# If you wish to use a different CDDB server, edit this line.
# If you just wanted to use a proxy server, just set your http_proxy
# environment variable - wget will use it correctly.
#CDDDBURL="http://freedb.freedb.org/~cddb/cddb.cgi"

# The CDDB protocol level.
# Right now 5 is latin1 output and 6 is UTF8 encoding.
#CDDDBPROTO=6
```

Figure 1: An abcde configuration file begins with options for how the command interacts with the CDDB database of published CDs.

the hash sign at the start of a line if you want a field to activate.

At the top of the file is CDDB Options. The Compact Disc Database (CDDB) [6] is a database of published musical disks. The two most commonly used copies of the CDDB are CDDB or Musicbrainz, with CDDB set as the default in the CDDBMETHOD field. Alternatively, by setting the CDDBUSELOCAL field to y, you can use the database shipped with the CD instead. A third option is to set NOCDDBQUERY to y, which will open a blank CDDB file for you to edit instead.

Should the CD's list of tracks differ from the one in CDDB, or if no entry for the CD exists, you have the option of specifying whether to send updates in CDDBSUBMIT or of refusing to send updates by setting the NOSUBMIT= field to n.

The CDDB section of the configuration file also sets the tags or metadata for each track, setting whether they include

```
# List, separated with a comma, the fields we want the parsing function to
# output. Defaults to YEAR and GENRE, for a complete list of fields provided by
# CDDB.
# The fields are not case sensitive. Actually, "y,g" will work as fine as "Y,G"
# or "YEAR, GENRE"
#SHOWCDDBFIELDS=year,genre

# Specify the style of encoder to use here -
# oggenc, vorbize - for OGGENCODERSYNTAX
# lame, gogo, bladeenc, l3enc, xingmp3enc, mp3enc - for MP3ENCODERSYNTAX
# flac - the only supported for FLACENCODERSYNTAX at the moment
# speexenc - the only encoder for SPEEXENCODERSYNTAX
# mpcenc - encoder for MPCENCODERSYNTAX
# faac, neroAacEnc - for AACENCODER
# opusenc - for OPUSENCODER
# default is a valid option for oggenc, lame, flac, speexenc, mpcenc, faac and opus.
# Currently this affects the default location of the binary, the variable
# to pick encoder command-line options from, and where the options are
# given.
#MP3ENCODERSYNTAX=default
#OGGENCODERSYNTAX=default
#FLACENCODERSYNTAX=default
#SPEEXENCODERSYNTAX=default
#MPCENCODERSYNTAX=default
#AACENCODERSYNTAX=default
#OPUSENCODERSYNTAX=default
```

Figure 2: The second section of an abcde configuration file contains settings for working with different audio formats.

title, artist, composer, album artist, album, genre, year, and comments. These tags are used on some music players as different ways to list tracks, but if, for example, you never choose music by genre or year, you can omit these tags from the converted files by entering a comma-separated list of tags you do want in SHOWCDDBFIELDS.

Codecs and Other Resources

The next section of an abcde configuration file defines the tools used for the conversion (Figure 2). The important of the fields in this section define the encoder to user. The field for Ogg Vorbis is OGGENCODERSYNTAX=, and the one for Flac is FLACENCODERSYNTAX. In most distribu-

raspberrypi GEEK

Raspberry Pi Geek Newsletter SUBSCRIBE NOW – FREE!

Every two months you'll get a sneak peak at the newest *Raspberry Pi Geek* to hit the presses. Read about new projects for beginners and advanced users, tips and tricks, hacks, and the best distributions to try out on your Raspberry Pi!



```
# Keep the wav files after encoding. Set it to "y" and remove "clean" from
# the list of default actions, since we purge the temp directory as default.
#KEEPWAVS=n

# Track padding: force abcde to pad tracks using 0, so every song uses a two
# digit entry. If set to "y", even a single song encoding outputs a file like
# 01.my_song.ext
#PADTRACKS=n

# Define if you want abcde to be non-interactive.
# Keep in mind that there is no way to deactivate it right now in the command
# line, so setting this option makes abcde to be always non-interactive.
#INTERACTIVE=n
```

Figure 3: Part of the abcde configuration file showing options for output.

tions, both can be set to default, which means that oggenc and Flac are used. However, you can also enter the path to any other application that will convert the file. Other formats have similar fields, such as MP3ENCODERSYNTAX= for MP3 conversions. Give the location of the ripping program and pass any extra options.

Directly below this section, you can set the options for each encoder. This is probably the handiest part of the configuration file, because it allows you to enter the most common options once for all conversions to the format.

Should you want the occasional different option, you can enter them manually when you type the command. It can be followed by OUTPUTTYPE=, which takes any legitimate extension for an audio file, including .ogg and .flac.

For Ogg Vorbis conversions, you will need to add -q NUMBER, with 1 being the lowest quality and 10 the highest. Another useful option is --resample NUMBER, which sets the sampling rate. A sample rate of 44,100 is equivalent to a low-quality MP3, while 96,000 is equivalent to a high-quality MP3, although in both cases the Ogg Vorbis file should have high-quality sound. A sample rate of 192,000 is the highest sample rate that makes a difference in quality on most music players. You can add other options as seems appropriate.

For Flac, useful options include -0 to -8 to set the level of compression, with -8 being the highest, and -V, which verifies all conversions. You can also use --picture=FILE SPECIFICATION if you want to include album covers with a file. Flac does not have a man file – presumably because it is technically a codex, not a command – so instead refer to the help on the website.

Similarly, the CD reader to use is set in CDROMREADERSYNTAX – usually to cdparanoia. Although other entries are also

acceptable, some of them, like Flac, are specific to a format. If cdparanoia is chosen – and, generally, there is no reason why not – you can enter options in CDPARANOIAOPTS=.

Output Options

The final section of a configuration file sets details about the output (Figure 3). Most likely, you will have no need to change:

```
ACTIONS=cddb,playlist,read,encode,
tag,move,clean
```

However, you might want to use OUTPUTDIR to specify the destination for converted files, instead of saving them to the current directory.

Another useful set of options sets how single and various single and multiple tracks are named with fields. Track names are built of building blocks, and you can choose which ones to include:

```
OUTPUTFORMAT='${OUTPUT}/
${ARTISTFILE}-${ALBUMFILE}/
${TRACKNUM}.${TRACKFILE}' VAOUTPUTFORMAT=
'${OUTPUT}/Various-${ALBUMFILE}/
${TRACKNUM}.${ARTISTFILE}-${TRACKFILE}'
```

Because these are the names that will appear on music player lists, I recommend making them as short as possible so you do not have to endure endless horizontal scrolling and truncated listings. For consistency's sake, I also suggest using the same pattern for albums by both single and various artists. You can use the same naming conventions to create external playlists with .m3u extensions.

At the very bottom of the installed configuration file is a script that removes illegal characters from names, including spaces. Other fields set niceness, or the relative share of system resources parts of the conversion process use, and the paths to scripts and applications, al-

though neither should be necessary on modern systems.

It is followed by a collection of random options, such as PADTRACKS=y, which tags track numbers using two digits instead of 1, and EJECTCD=y, which completes conversions by ejecting CDs.

Other Considerations

Except for the codecs and options, you can write similar configuration files for MP3s and other audio formats. For a complete – but not necessarily definitive – list, see Andrew Strong's detailed examples [7].

Whichever format you use, setting up configuration files may take time, especially in choosing the options to include. You may even want two configuration files for the same format – for example, one configuration file for Ogg Vorbis files with a low sampling rate, suitable for listening to on the bus, and another for the same files at a higher rate that you can appreciate in your living room.

Consider, too, how you will play the converted files. If you are using a \$50 music player, then there is little point in a 32-bit Flac file except to take up space, because you are unlikely to notice the audio quality. By contrast, a small but increasing group of players, such as the Fiio X7 [8], which retails for just under \$600, is wasted if used for low-quality MP3s, even though the playback may be better than anticipated.

Such considerations explain why fine-tuning abcde configuration files is worth the effort. You may take several tries to produce the results you want, but if you care enough about music to make a point of using free formats, the time you spend will seem worthwhile. ■■■

INFO

- [1] Ogg Vorbis: <http://www.vorbis.com/faq/>
- [2] Flac: <https://xiph.org/flac/>
- [3] abcde: <http://lly.org/~rcw/abcde/page/>
- [4] cdparanoia: <https://www.xiph.org/paranoia/index.html>
- [5] oggenc codex: <http://www.rarewares.org/ogg-oggenc.php>
- [6] CDDB: <https://en.wikipedia.org/wiki/CDDB>
- [7] abcde: CLI music CD ripping: <http://www.andrews-corner.org/linux/abcde/index.html>
- [8] Fiio X7: <http://www.fiio.net/en/products/45>

Subscribe now!

FREE DVD

chapeau 24

archlinux

LINUX MAGAZINE

GEOTAGGING
Add location data to digital images

NOW INCLUDING LINUXVOICE

GEOTAGGING
Add GPS coordinates to your images, and plot your images on maps

Gimp and Neural Networks

PowerShell in Linux
Could it really compete with Bash?

Cordova
Build mobile apps with HTML5 and JavaScript

Greg Kroah-Hartman
"Total world domination was our goal."

FOSS Picks

- Draw cartoons with OpenToonz
- VoxelShop retro

Tutorials

- Ansible meets Docker
- Terminal multiplexing with Tmux

WELCOME LINUXVOICE READERS!

WWW.LINUX-MAGAZINE.COM

Don't miss a single issue of the magazine that delivers the in-depth technical solutions you'll use everyday!



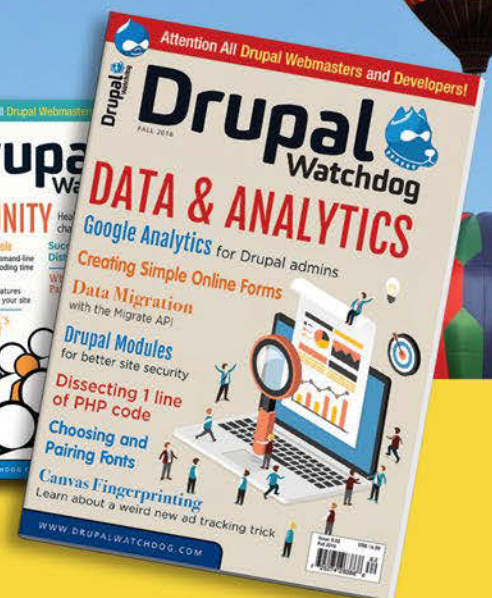
shop.linuxnewmedia.com/subs



RISE HIGHER

EACH ISSUE OF DRUPAL WATCHDOG OFFERS TOOLS, TIPS, AND BEST PRACTICES FOR BETTER DRUPAL WEBSITES.

NOW
PUBLISHED
4 TIMES
PER
YEAR!



Renew or subscribe now!

SUBSCRIPTIONS NOW AVAILABE WORLDWIDE!

Visit <http://drupalwatchdog.com/subscribe>



Ben Everard

I'm an Ubuntu user, so I upgrade my distro every six months. I could stick with the two-yearly release cycle of the Long Term Support (LTS) series, I suppose, but every time I do that I get the feeling I'm missing out on something. The thing with six monthly releases is that not much actually changes in just 26 weeks. It's not enough time to craft, polish, and release major pieces of software, so in general, every release looks quite similar to the last. Of course, big changes do happen, but they're

usually broken down into many smaller changes so you barely feel them happening (well, apart from the introduction of Unity). I've particularly enjoyed reading Mike's article looking back over 20 years of KDE this month, as it's helped me step back and see just how much has changed and how much better the free software world has become. I did, however, feel quite old when I realized that the plucky new desktop I remember from my early days as a Linux user is now beginning its third decade.

But let's not dwell too long on our age – nostalgia's not as good as it used to be anyway – and come bang up to date with Valentine's excellent article on inotify. I discovered this tool earlier in the year (though I've come to learn a lot more about it this month), and it's one of those things that I never knew I needed until I discovered it. Now all my scripts know exactly what the filesystem's up to. These are just two of my favorite articles this month – there's loads more as well. Turn the page and bathe in the pool of Linux knowledge that awaits.

– Ben Everard



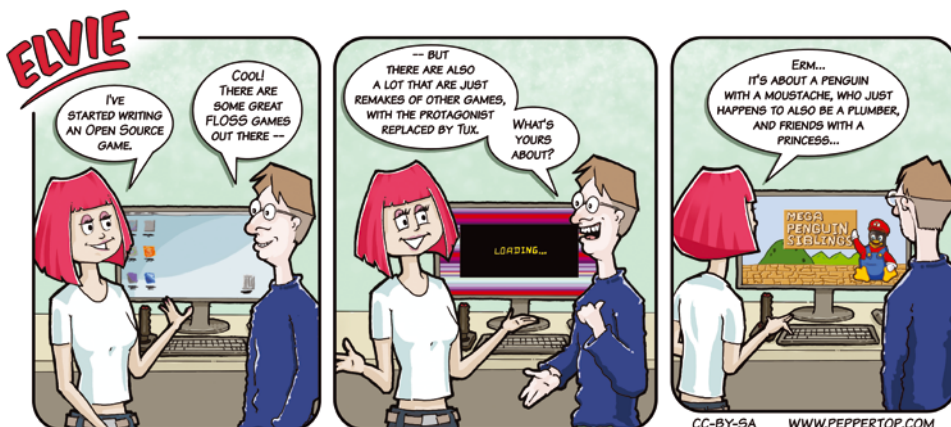
Andrew Gregory



Graham Morrison



Mike Saunders



LINUXVOICE

SEOScerer's Apprentice 66

Simon Phipps
SEO "experts" who game the system can generate unpleasant results.

Cyber-Glossary 67

Andrew Gregory
The UK's National Cyber Strategy doc is worth every penny of the free download cost.

20 Years of KDE 68

Mike Saunders
The "Kool" desktop is almost as old as Justin Bieber – and far more stable.

FAQ – Connected Car 72

Ben Everard
Automotive Grade Linux: Will 2017 be the year of the rolling Linux?

Core Tech – Filesystem Monitoring 74

Valentine Sinitsyn
Notification APIs in Linux.

FOSSPicks 80

Graham Morrison
Browse the Internet with QupZilla, add function-specific prompts to Vim and your shell with Powerline, and install a graphical front end for your arcade emulator with Attract Mode.

Gaming 86

Michel Loubet-Jambert
Build the ultimate Magnum Opus for a Mad Max stripped of his possessions.

Tutorial – Collectd 88

Ben Everard
The collectd tool harvests and stores your system stats so you can use them for fabulous graphs.

Tutorial – Wine 92

Mike Saunders
Decant your vintage Windows apps into a better vessel known as Linux.

NEWS ANALYSIS

The Linux Voice view on what's going on in the world of Free Software.

Opinion

The SEOscerer's Apprentice

Is that "SEO expert" really helping? BY SIMON PHIPPS



Simon Phipps is ex-president of the Open Source Initiative and a board member of the Open Rights Group and of Open Source for America.

Have you ever considered hiring an "SEO expert" to "improve your Google rankings"? I don't mean the type that advises you on how to ensure all the express and implied metadata on your website is correct. That seems a reasonable thing to do, and it shouldn't take a magician to help you get it right.

No, I mean the type who want to actively manipulate your rankings through external means.

Those "external means"



Exactly why the community plugin and template repository had accepted all these fake templates is a great question for another time.

"We found thousands of links pointed to our site. This was caused by an SEO team we hired. We wanted to get our templates shared to more users via your site and hired an SEO team to upload our templates to your site. But they are too aggressive and put too many links inside."

Exactly why the community plugin and template repository had accepted all these fake templates is a great question for another time. The point is that stuffing hundreds of links into fake templates or plugins and uploading them to a community repository is sociopathic and ought to have rung alarm bells.

The lesson is that SEO "experts" who offer to game open source communities for you to "improve your rankings" are scammers, and using their services is

bad for your business. You might see an instant boost in some sort of metric the SEO scammer offers to improve, but any sort of sociopathy will eventually come at a social cost. The reason is is that Google has both automated and manual checks for breaches of the guidelines on site linking. Once you have unleashed the demon of spammed links to your website, you are only a short distance from a manual action against your site, and then you will need to try to unpick the damage that has been done.

This template-stuffing issue isn't the only example. There are a constant stream of requests to the moderators of mailing

lists begging for removal of old postings that were sent by or on the advice of "SEO experts" and are now poisoning search engine ratings rather than improving them. These requests mostly have no effect, since list moderators rarely agree to remove old posts – see the Apache policy for example. Even if they do agree, the many mirrors of community mailing lists are unlikely to be affected.

At the heart of the problem is the confusion of metrics with the results they purport to illustrate. Gaming effects that seem to move metrics in the short term is never smart; one day the rules will change, and your virtual gain will become a real loss. The best solution is to not play the game in the first place. ■■■



Cyber-Glossary!

BY ANDREW GREGORY

The UK's National Cyber Security Strategy doc is worth every penny of the free download cost.

Another day, another government report. Adorning my desktop right at this moment is the UK's National Cyber Security Strategy document, launched at the beginning of November 2016 by the Chancellor and Under-Treasurer of Her Majesty's Exchequer, Philip Hammond. It's grabbed my interest for a number of reasons.

First of these is the budget. £1.9bn has apparently been earmarked, though the history of government IT spending should tell us to expect it to cost a lot more. (£1.9bn is also, by an amazing coincidence, the same amount that Her Majesty's Revenue and Customs reckons is being lost in dodged tax by the mega-wealthy.) Lest we forget, the deluxe NHS healthcare records system, commissioned by the UK government at an estimated cost of £2.3bn, was canceled after nine years of work at an eventual cost of £12bn. This was the costliest IT cock-up in history, but the current revamp of the

welfare system will overtake it soon unless it's canceled, as the welfare revamp has already cost £12.8bn – for a system that will have only 25,000 users.

The second is the glossary. This wonderful section is a thing of beauty, including no fewer than 28 variations on the word "Cyber." For example, it defines a cyber-physical system as one with "integrated computational and physical components." That sounds like my car to me, and my watch, and my computer, and my phone, and all those webcams that have been turned into a giant bot – in fact it sounds like anything made after 2012 that uses electricity. Any definition as broad as that is functionally useless.

Third, there's no mention of the words "Linux" or "Windows." The document completely ignores the vast differences in hackability that exist between different operating systems. The need to replace legacy systems is mentioned though, and I

think this gives us a clue as to where a large chunk of the money is going to be spent. Rather than name Windows as inherently insecure; the Internet of Things as completely wrongheaded; and the weakest link as human failing, we're going to spend £1.9bn getting consultants to replace instances of Ye Olde Windows XP with Windows 10, with a couple of settings changed and a couple of ports closed by default. There will be bespoke secure IRC channels that only work with Internet Explorer, and there will be a load of knighthoods handed out to the CEOs of the companies fleecing the taxpayer, and some tasty board-level positions for the politicians that signed off on the spending.

Meanwhile, script kiddies will laugh and keep on trucking, and makers of every gadget going will add Internet access with a default, hard-coded admin password of `password123`. Well done Britain. Well done. ■■■



IT Highlights at a Glance

Too busy to wade through press releases and chatty tech news sites? Let us deliver the most relevant news, technical articles, and tool tips – straight to your inbox.

Linux Update • ADMIN Update • ADMIN HPC • Raspberry Pi
Keep your finger on the pulse of the IT industry.

Admin and HPC: www.admin-magazine.com/newsletter

Linux Update: www.linuxpromagazine.com/mc/subscribe

Raspberry Pi: www.raspberry-pi-geek.com/mc/subscribe

Desktop Through the Decades

We look at the ups and downs of the past two decades of the “Kool” desktop.

MIKE SAUNDERS

Those of us who started using Linux back in the late 1990s remember all too well that it wasn't a pretty operating system on the desktop. Sure, we had a bunch of tools like Netscape and Gimp at our disposal, and a variety of window managers, but everything felt like a bit of a kludge. Nothing was really well integrated, every app had its own look and feel, and while we still preferred it to the other operating systems of the time, we never really felt comfortable in it.

That's not to say it was all bad. FVWM users, for example, were rocking an insanely configurable window manager that wiped the floor with Windows and Mac OS. Users who did most of their work in text terminals were having an awesome time. The GNU/Linux

combination with XFree86 on top was a workable desktop operating system for geeks with time to invest, but for newcomers it all looked rather clumsily bolted together.

Fortunately, one group of hackers had the foresight to do something about this. Instead of having a window manager developed by one team, and a file manager by another, and a text editor by yet another, all using different toolkits and config files and themes, wouldn't it be great if all these bits and pieces were worked on in unison? Wouldn't

it make more sense for a desktop environment and all its common utilities to be developed by a single project, to encourage consistency and code reuse?

Well, in the late 1990s, two projects came to life that attempted to do just that: KDE and Gnome. The latter was started in response to licensing issues with the former, and we'll cover it another time; but, here we're going to focus on KDE [1]. There's an especially good reason to look at the past and future of KDE – the desktop environment recently celebrated its 20th birthday. 20 years! For some of us, it hardly feels like yesterday since Matthias Ettrich and company decided to tackle one of the biggest issues with Free and Open Source Software (FOSS) on the desktop.

Humble Origins

The problem of desktop inconsistency wasn't just limited to Linux, though. Other Unix-ish operating systems had the same flaws, and a few attempts were made to fix them. CDE, the Common Desktop Environment, began as a joint venture in 1993 between HP, IBM, and Sun Microsystems – all major Unix vendors – in an attempt to unify the experience across different platforms (Figure 1). And it

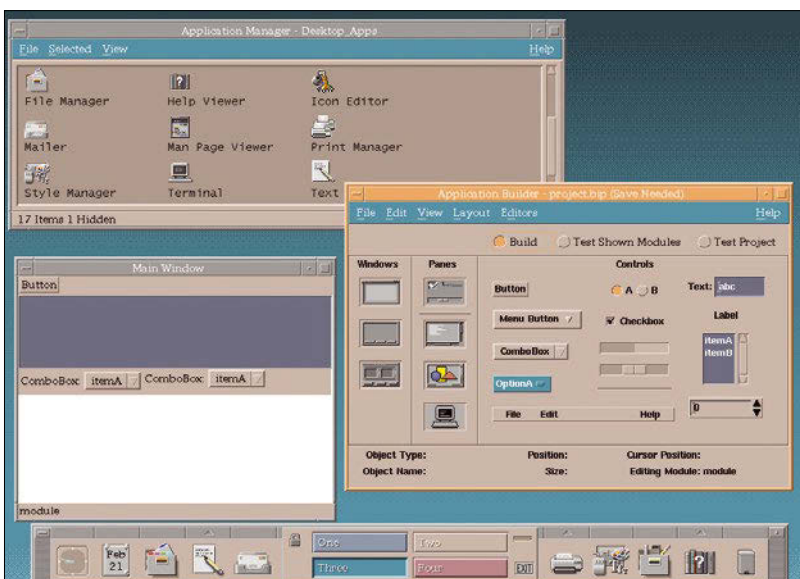


Figure 1: CDE was an early attempt to provide a consistent desktop on Unix and influenced initial KDE releases. (Image: <http://tinyurl.com/jahbz2e>)

did a reasonable job, offering a bundle of utilities built on the Motif widget set. Still, because none of the proprietary Unix variants ever achieved mainstream desktop popularity (Mac OS aside, if you count that), CDE never became a household name. (CDE was open sourced in 2012 [2].)

The KDE project was born in October 1996 by German student Matthias Ettrich, who was already known in the Free Software community for his work on the LyX front end to the LaTeX typesetting system. Ettrich chose the name KDE as a reference to CDE, given that the projects had similar goals, with the K originally standing for Kool. That moniker didn't last long, however, so the official name swiftly became the K Desktop Environment – with the K meaning nothing. (At least the team didn't try to make it a doubly recursive initialism...)

Ettrich's original announcement of the project, and justification for its existence, is fascinating reading, just like Richard Stallman's original post about GNU or Linus Torvalds' "just a hobby" message about his home brew kernel [3], which, in a nutshell, says: Linux doesn't really have a GUI. It has a bunch of largely unconnected tools (window managers, widget toolsets, editors, etc.) that don't really work together. Let's fix that.

Around that time, a new graphical toolkit was starting to gain popularity: Qt. Ettrich's KDE announcement heaped praise on Qt, noting that it was a "revolution" for programming on the X Window System (the base graphical layer of most Unix-like OSes). Also, Ettrich highlighted the fact that Qt was "open source" – an issue we'll return to later – so he recommended it for building KDE. To create a viable desktop, various components would be needed: a panel (like a taskbar), file manager, mail client, terminal, window manager, help browser, and other utilities.

Building the Foundations

So mailing list discussions kicked off, plans were made, and developers started hacking on code. The journey ahead would be tough, yet enough people thought it was worth it to take on the challenge. Indeed, the team showed a lot of foresight by establishing a non-profit organization to represent the KDE community in 1997, called KDE e.V., even before the first release had made it out of the door. At the time, this was seen as an audacious and unusual move – why put all this effort in to setting up such an organization, when hardly anyone is using the software yet?

For one thing, KDE e.V. reassured existing and potential contributors that code, trademarks, and donations (cash and hardware) would belong to the community as a whole. This wasn't a fly-by-night project to grab some cash or boost the egos of a few developers. KDE e.V. was set

up as an association under German law, lending extra credibility to the project and making it easier to organize events and get support from sponsors.

Interestingly enough, this model soon became much more common in the Free Software community. As soon as LibreOffice forked off from OpenOffice(.org), for instance, the team behind it established The Document Foundation, a similar non-profit entity. Then we have the Mozilla Foundation, the Gnome Foundation, the FreeBSD Foundation, and many others. In a world of patent trolling, copyright rip-offs, and GPL violations, these foundations and organizations can provide useful backbones for FOSS projects.

One issue that came up in the early days was licensing. As mentioned before, the desktop was built on the Qt toolkit, but this was a controversial move. Qt's source code was available to look at, but the Open Source Initiative wouldn't bless it as truly "open source" because modified versions could not be distributed. With Qt 2.0, the situation was improved slightly when the toolkit switched to the Q Public License – but it still wasn't compatible with the GNU GPL.

There was also concern that if Trolltech, the creator of Qt, was bought out by another company, the toolkit could end up becoming closed source, leaving KDE with no option but to use outdated versions. After a lot of discussions and flame wars in the FOSS community, a compromise was made: a "KDE Free Qt Foundation" was established that guaranteed that Qt would be re-released under the permissive BSD license if Trolltech (or a parent company) didn't release any more open source versions. So, although the QPL license wasn't ideal for everyone, this backup plan allayed many of the initial fears.

The First Release Arrives

So as we've seen, the KDE project made some major steps forward before even shipping a 1.0 release. By the time that happened on July 12, 1998, expectations were high: Some Linux users had tried prerelease versions, while others had seen screenshots in magazines and were eagerly awaiting a proper polished, integrated, and consistent desktop for Linux. At long last. Some pundits were sceptical that such an effort could even be successful, while others promised that the KDE team could rip their `.fvwm2rc` files from their cold, dead hands.

KDE 1.0 wasn't the prettiest desktop out there and, compared to some Enlightenment themes of the time, it looked rather retro with its chunky, Motif-esque buttons. Unusually, this first release didn't place everything in a single panel along the bottom of the screen, as we came to expect in later KDE and Gnome releases. Instead, the default layout included a taskbar along the top of the

screen – which was seen as screen wastage by some users. On the other hand, it did give KDE a distinct look and helped to avoid accusations that it was just ripping off the design of Windows 9x.

Initial opinions of KDE 1.0 were positive. Journalist Bill Cunningham described the release at the time as “the sharpest, coolest window display I had ever seen on any operating system.” He also praised the ability to configure every aspect of the desktop’s appearance with a few mouse clicks – in contrast to other window managers, which typically required lots of poking around inside text files.

However, as expected of a first release, KDE 1.0 had its bugs and limitations. Seven months later, KDE 1.1 arrived with a boatload of fixes and updates: It was more robust, had performance improvements, and looked slightly better as well thanks to a new K logo (for the main menu) and other visual tweaks. Linux distributions started including KDE, with SUSE Linux doing an especially notable job of showcasing the new desktop.

Not everyone in the Free Software community was happy, though. The Free Software Foundation was especially concerned about KDE’s usage of Qt and decided to tackle the issue on two fronts: first, by creating a separate desktop environment that would “use free software exclusively.” This GNU Network Object Model Environment (Gnome) was initially pieced together from various existing components. Early Gnome releases paled in comparison with KDE in regards to consistency and spit-shine – but they provided an alternative for those who wanted to avoid Qt.

But the Free Software Foundation was pragmatic as well and encouraged the community to create a fully free Qt-compatible toolkit called Harmony. This was “designed to make it possible to run KDE software without using Qt,” but never reached any kind of advanced stage as Qt was

relicensed under the GNU GPL in 2000, thereby settling the issue once and for all. (See the “KDE 1.x Classic” box for more details.)

Brand Repositioning

So after some bumps in the early road, KDE had a fairly smooth ride throughout the 2000s. KDE 2.0 arrived in October 2000 and was widely regarded as a major step forward for the desktop; it was followed by 3.0 in April 2002. However, it took another five years for KDE 4.0 to arrive, and this release had a few detractors. The switch to Qt 4.0 had meant that some older applications were dropped, and the updated Plasma desktop and various subsystems were seen as immature (Figure 2).

In response, the KDE team pitched 4.0 as an early adopter release – in other words, it wasn’t meant to be viewed as a final, polished, end-user-ready product. Consequently, the KDE 4.x series went through many revisions, and some hard core KDE fans only described it as truly ready for widespread usage at versions 4.2 or even 4.3 (See the “Trinity Desktop Fork” box for more details.)

In addition, KDE went through a rebranding in November 2009: “KDE is no longer software created by people, but people who create software.” Instead of KDE being a single bundle of software to make a desktop, new terms were introduced, such as KDE Plasma desktop for the bulk of the interface, KDE Applications for the various tools,

Trinity Desktop Fork

As mentioned in the main text, KDE 4.0 was a huge update and not received positively by everyone. Some KDE fans wanted the 3.x branch to stay alive, so a bunch of developers and other community members started the Trinity Desktop Environment [9], a fork of KDE 3.5. Unlike many forks in the Free Software world, Trinity didn’t fizzle out after a few weeks of bike-shedding and arguments on mailing lists.

No, it actually managed to get usable software out of the door and work on updates and fixes, much like the MATE team did with Gnome 2.x. The most recent version of Trinity arrived in early 2016 – although that was a minor bug fix release, and the last major update came in December 2014.

But does Trinity have a future? The Git commit list [10] shows the work of a small team that’s mostly fixing bugs and updating supporting libraries. So we don’t expect any major planet-shaking new features in Trinity’s future – it just needs to keep KDE 3.x alive and usable for those who love it.

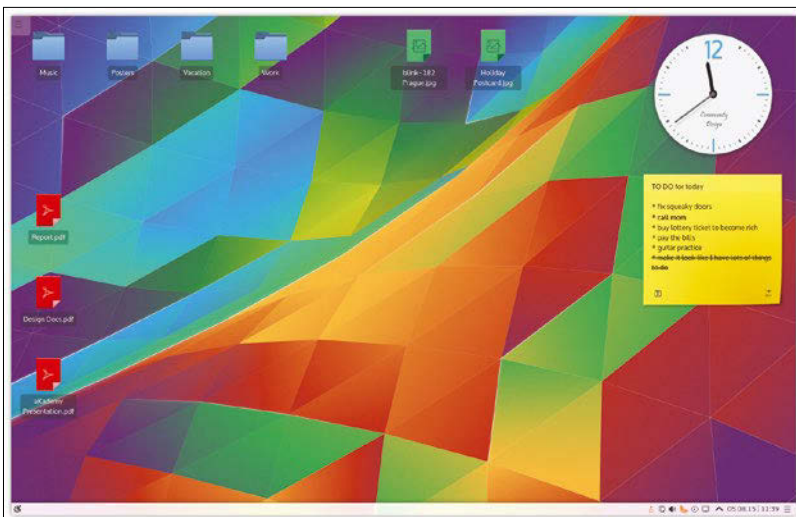


Figure 2: Here’s KDE Plasma from a recent release – you can see how far it has come over the last 20 years...

KDE Platform for libraries and subsystems, and so forth. All together, a release was called a KDE Software Compilation. Some KDE fans objected to the new naming scheme; others recognized it as a necessary step for such a large and all-encompassing project.

What's to Come – And How You Can Help

Currently, KDE is using a regular number-based versioning scheme for Plasma, along with Ubuntu-esque dates for the KDE Applications bundle (Figure 3). (At the time of writing, the latest version of KDE Applications was 16.08.1 – so a minor revision to the version initially released in August of 2016.) Schedules for upcoming releases can be found online [4], but note that there's no roadmap for Plasma 6 or KDE Frameworks 6. In a way, that's a good thing; we all like to read about big new features and major changes coming down the pipeline, but we also like to see projects work on refining and stabilizing the current codebase before rushing off to work on the Next Big Thing.

Of course, the best way to shape the future of KDE is to get involved with the project. If you're a coder and know way around C++ and Qt, see the Get Involved page [5] for some pointers – but there are plenty of non-development tasks to help with as well. For instance, the Visual Design Group [6] works on the look and feel of KDE (and its websites) and is looking for contributions from aspiring designers and artists. Similarly, if you speak other languages than English, you can add translations [7] or work

on docs [8]. If you've never contributed to an open source project before but get involved with KDE, let us know your experiences! ■■■■

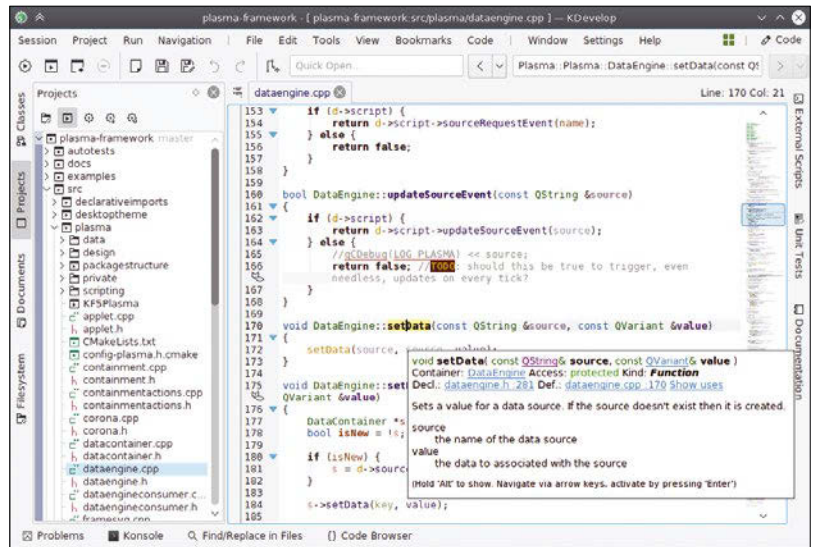


Figure 3: Some KDE applications have become so popular that they're often used with other desktops or window managers – like KDevelop.

Info

- [1] KDE: <https://www.kde.org/>
- [2] CDE: <http://cdesktopenv.org>
- [3] Just a hobby: <https://www.kde.org/announcements/announcement.php>
- [4] Schedules for upcoming releases: <https://community.kde.org/Schedules>
- [5] Get Involved: https://community.kde.org/Get_Involved/development
- [6] Visual Design Group: https://community.kde.org/Get_Involved/design
- [7] Translations: https://community.kde.org/Get_Involved/translation
- [8] Documentation: https://community.kde.org/Get_Involved/documentation
- [9] Trinity Desktop Environment: <https://www.trinitydesktop.org>
- [10] Trinity Git commit list: <https://www.trinitydesktop.org/patches/>
- [11] KDE Restoration Project: <http://www.heliocastro.info/?p=291>

KDE 1.x Classic

Whenever we advocate FOSS, one of the benefits we talk about is having access to the software forever. You don't need to worry about your programs not working with the next operating system release – just recompile the code! Of course, in practice it's not that simple. Libraries change, compilers change, init systems change, and getting an old FOSS app from the 1990s to build on a modern Linux distro can be a pretty major exercise in frustration.

Still, it's technically possible, and the KDE Restoration Project [11] is a great example of this (Figure 4). 20 years after KDE 1 was released, you can now run it on a modern Linux distro – thanks to a bunch of patches and build system improvements from the project. To get it running, you may have to download the source and build it yourself; hopefully binary packages for mainstream distros will be available by the time you read this.

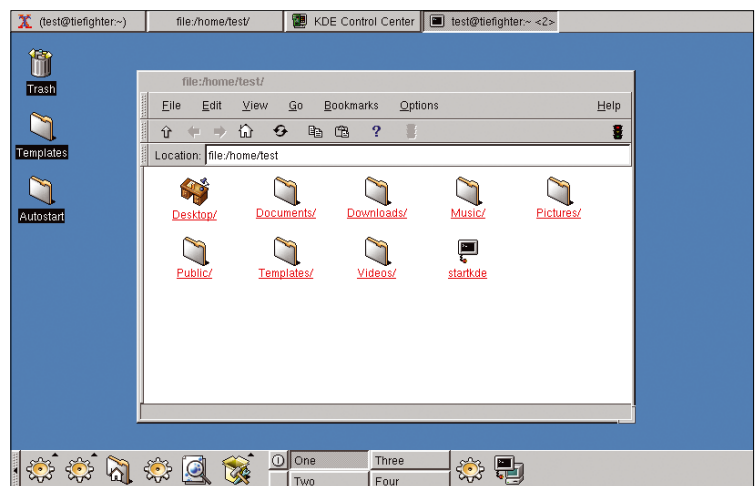


Figure 4: Thanks to the KDE Restoration Project, you can run KDE 1.1 on modern Linux distros for retrotastic fun.

FAQ

Connected Car

We answer your questions about the Automotive Grade Linux project.

BEN EVERARD

Q Linux in a car? Excellent! I've always thought that a steering wheel and pedals were a bit antiquated compared with a keyboard. Will I be able to choose between Vi and Emacs key bindings for the controls on my next car?

A Well first of all, you need a history lesson – it's the keyboard that's antiquated. The QWERTY keyboard originates from the Sholes and Glidden typewriter produced by E. Remington and Sons in 1873, whereas the steering wheel didn't appear until 1894. Second, Automotive Grade Linux (AGL) [1] isn't about taking complete control of the car with Linux but creating a standard platform for the various computers that already exist on cars. The first goal is the in-vehicle infotainment system, but the developers hope that it will support instrument cluster, heads-up display, telematics, and autonomous driving in the future.

Q Autonomous driving? To be honest, I've long suspected I could script my drive to work ...

A Well, let's not get ahead of ourselves. Autonomous driving is more an aspirational goal than something being worked on at the moment.

Q OK, so the first goal is infotainment – what is that?

A The computer and display system on most cars around at the moment fulfill two

separate tasks – they let the driver know what's going on with things like the speedometer, navigation system, rear-facing cameras, and that sort of thing. They also keep everyone entertained through the journey with music (and of course, the Linux Voice podcast), or possibly with videos for those in the back seats. People who wear suits and sell cars know these functions as infotainment because apparently saying information and entertainment is five syllables too long.

There have been two releases so far in 2016 (with the latest being 2.0, aka Brilliant Blowfish), and this goal of an infotainment system is well on its way to becoming reality.

Q It's on version 2.0 already? What cars is it running in?

A The numbering is a bit misleading – it might be called version 2.0 but it's not quite ready for production use just yet. There are more releases planned for 2017, but we're unlikely to see AGL in any production cars before 2018, so you may have to wait a couple more years before getting open source software in your car.

Q AGL might be open source, but will I get any benefits from the openness? Won't the software just be so locked into the car and so unchangeable that it might as well be proprietary?

A Most manufacturers will undoubtedly lock their cars' computers down so that users can't change the software on them. Much as we'd like this to lead to a hackable car, that probably won't happen. The biggest advantages to us – as car users – is that open source is a better way of developing software. Rather than each manufacturer working on proprietary code and reinventing the wheel, they can all pool their resources and create one really good system that we can all benefit from.

At the moment, project members include Mazda, Toyota, Honda, Nissan, Ford, Land Rover, and Subaru, as well as a host of IT companies, and the project lives inside The Linux Foundation umbrella (Figure 1). Together, these companies represent a lot of auto industry might, and if they can all work together, we should see a significant improvement in the area of in-car computers.



Figure 1: The AGL logo includes a not-so-subtle reminder that this is a Linux Foundation project.

Of course, the software itself is freely available both as source code and in images for different devices, including Raspberry Pis and MinnowBoards. Even though you might not be able to access the computer in new cars, this will mean that you can build new systems and retro-fit them to old cars. The wiring could be a little complex, but if you leave the current information system in place, AGL could be a great option for building a more complex entertainment system for your motor.

Q **If AGL allows me to build my own in-car computer, does that mean that I can kit out my aging VW Golf as a sort of Batmobile/James Bond car gadget-fest?**

A We're not on the Batcave technical advisory committee, but if we were, we're strongly recommend investigating AGL for future development. However, no amount of gadgets or smooth user interfaces is going to turn a Golf into an Aston Martin.

Q **There's nothing stopping me from using any Linux distribution to build an in-car computer. Why should I use Automotive Grade Linux?**

A Probably the most unique thing about car computers is their user interface. You need to be able to do everything while driving down the motorway at 70mph (and possibly chasing down Gotham City's ne'er-do-wells), so everything has to be really intuitive and easy to use. The second big difference is that in-car entertainment computers bring together some quite unusual hardware. For example, there aren't many other computers that have to operate both analog radio receivers and handle GPS navigation, and that's without getting into the CAN bus and controlling the air conditioning. Basically, what we're trying to say is that there's nothing fundamentally unique about Automotive Grade Linux, but it is tuned toward the particular needs of in-car computers.

For more simple entertainment systems, AGL is designed to work well with the particular needs of cars, such as the ability to have one control system in the front and multiple display screens in the back. It's not impossible to set this up with careful tuning of your X Window configuration, but it's a lot easier to use a distro that's set up ready for this type of use. On a more practical level, AGL uses Wayland to provide better graphical performance for low-powered devices such as the sort you're likely to embed in a car dashboard.

Q **With high-profile security issues in Linux over the past couple of years – Shellshock and Heartbleed to name but two – isn't it a bit risky to put Linux in cars?**

A Obviously, a car without a computer is going to be more secure from an IT perspective than one with a computer, but you miss out on lots of really great features such as satellite navigation, advanced engine management, and entertainment features. Provided it's properly configured and kept up to date, your car's computer shouldn't pose any significant security risk. Obviously, "properly configured and up to date" is somewhat vague. After all, as a purchaser, how can you know that the system is secure? At least with Automotive Grade Linux, the software is created in public by a large community of engineers, and bugs can be found and fixed by anyone.

No software is immune to security problems, whether open source or not, but the Linux community has shown time and again that it can respond promptly and responsibly to security issues to protect users. It's too early to know for sure that the car manufacturers will take advantage of this and push security fixes out to fix any problems. There are always unknowns with IT security, but we'll feel far safe in cars we know are running up-to-date open source software.

Q **What distro is AGL based on? I've always been a Debian user, so will I be able to apt-get new software?**

A AGL is built on Yocto, which isn't really a distro but, rather, a framework for building embedded distros. Like AGL, Yocto [2] is a Linux Foundation project, and it's used in everything from industrial cooling systems to smart TVs. The project is based around the OpenEmbedded build framework that makes it really easy to build images for different devices. It will, however, be a little unfamiliar to people used to more traditional Linuxes, so you shouldn't expect to be able to use Yum or Apt to grab more software. It's perhaps better to think of it more like Android, in that it doesn't provide a typical Linux environment, but it's still based on the Linux kernel.

Q **Sounds interesting, but do I have to have some embedded hardware, such as a Raspberry Pi to test it out?**

A For the full in-car setup, you'll get most features if you run on a Renesas R-Car Porter board, which is a hackable embedded board that's designed specifically for developing car-based hardware. However, reference images are available for the Qemu virtualized environment and the Intel MinnowBoard Max, as well as an additional Raspberry Pi image created by the community. You can download these images from the AGL wiki [3].

If you have something else in mind, all the source code is available, and it's reasonably straightforward to use the Yocto project tools to build AGL for other hardware. ■■■

Info

- [1] Automotive Grade Linux (AGL): <https://www.automotivelinux.org/>
- [2] Yocto: <https://www.yoctoproject.org/>
- [3] AGL wiki: <https://wiki.automotivelinux.org/>



Valentine Sinitsyn develops high-loaded services and teaches students completely unrelated subjects. He also has a KDE developer account that he's never really used.

CORE TECHNOLOGY

Prise the back off Linux and find out what really makes it tick.

Filesystem monitoring

If you were using Linux back in 2006, you will remember the desktop search tool Beagle (Figure 1), which was notified when you changed your files so it could re-index them. Modern file managers also rely on notifications to update their displays when files are created, deleted, or renamed (Figure 2), unlike earlier file managers that counted on the user to refresh the display (Figure 3). Now, think of the ClamAV open source antivirus software. If you try to open a file containing malware, you expect an on-access scanner to ban it. In this case, notifications aren't enough; ClamAV needs to be an active part of the process, allowing or denying certain operations. Happily, Linux can handle both cases. The downside is, it does so with two separate APIs, and you can't just choose one over another.

Figure 1: Beagle hasn't released a new version since 2009, yet it's remembered as a "textbook" example of inotify usage. (The figure [1] is reproduced under Apache License, Version 2.0 [2].)



Inotify

The first API I'll look at, and the first one to appear in Linux 2.6, is inotify. It went mainline with Linux 2.6.13 in 2005. Inotify was the workhorse behind Beagle's indexer. It replaced an older filesystem monitoring technology (see the "dnotify" sidebar), improving it in several ways.

First, inotify replaced a cumbersome signal-based notification mechanism with a pollable file descriptor from which you just read events. This makes event loop integration a breeze. It also waived a need to retain an opened file descriptor for each directory you monitor. To do so, inotify introduced three new system calls: `inotify_init(2)`, `inotify_add_watch(2)`, and `inotify_rm_watch(2)`.

Your code starts with `inotify_init()`, which returns a file descriptor acting as a handler to an in-kernel event queue. A newer variant, `inotify_init1()` accepts the extra `flags` argument. Passing `IN_NONBLOCK` here opens the descriptor in non-blocking mode, saving you an `fcntl(2)` call. `IN_CLOEXEC` flag is a similar shortcut:

```
int fd = inotify_init();
if (fd < 0) { /* Handle the error */ }
```

Then, you add watches for filesystem objects of interest with the `inotify_add_watch()` system call. It accepts three arguments: an inotify file descriptor, a pathname, and a set of flags (or "mask") telling the events in which you are interested. `IN_CREATE` fires when an entry (think a file or a subdirectory) is created in the directory you watch. `IN_OPEN` is reported when a file is opened, followed by `IN_ACCESS` or `IN_MODIFY`, when the contents are read or changed. Later, when the file is closed, the kernel sends either `IN_CLOSE_WRITE` (if the file was opened for writing), or `IN_CLOSE_NOWRITE`. You can capture both with `IN_CLOSE`.

Some flags carry a `_SELF` suffix, like in `IN_DELETE_SELF`. They apply to the monitored directory itself, not its children. In particular, `IN_DELETE_SELF` is reported

dnotify

Inotify was introduced with early 2.6 kernels, but filesystem monitoring in Linux is really much older. Its first incarnation, dnotify, appeared in Linux 2.4.0 back in 2001.

The dnotify API was a step forward: Original approaches to the problem involved polling directories for changes, which, was very inefficient. However, dnotify's design made an API cumbersome and was not easy to use. Although later approaches introduced separate system calls, dnotify relied on `fcntl(2)`.

Signals are used for notifications, and they are somewhat difficult to treat correctly because they convey little information (not even the name of a file triggers the event), and you can't integrate them easily into event loops, although `sigprocmask(2)` file descriptors mitigate the issue to some extent. Dnotify forces you to retain an opened descriptor for each filesystem object you monitor. Moreover, it has no notion of events that rename a file, leaving programmers to figure it out by comparing two directory trees. If I recall correctly, Dropbox once offered a similar puzzle to candidates seeking an engineering position within the company (i.e., it's not trivial).

Dnotify is still available in the latest kernels, but with `inotify` and `fanotify`, there is little sense to use it except in legacy code.

when you remove a watched directory. The kernel then reports an `IN_IGNORED` event for it. Moving a directory can also generate `IN_DELETE_SELF` if it occurs across filesystem boundaries, but normally, it produces a sequence of two events: `IN_MOVED_FROM` and `IN_MOVED_TO`.

The `inotify(7)` man page lists all supported flags. Here, I just pass `IN_ALL_EVENTS`, which – you guessed it – captures everything:

```
int wd = inotify_add_watch(fd, argv[1], IN_ALL_EVENTS);
if (wd < 0) { /* Handle this one as well */ }
```

An `inotify_add_watch()` returns a so-called “watch descriptor.” It matches events to watched filesystem objects and can be used to “unmonitor” them later with `inotify_rm_watch()`. If anything goes wrong, `inotify_add_watch()` returns -1 and `errno` is set appropriately.

Now you wait for an inotify descriptor to become readable. In a real application, this happens in an event loop. In Listing 1, I just spin in `read()`:

It makes sense to use large buffers capable of storing multiple events for performance reasons. A struct `inotify_event` represents a single inotify

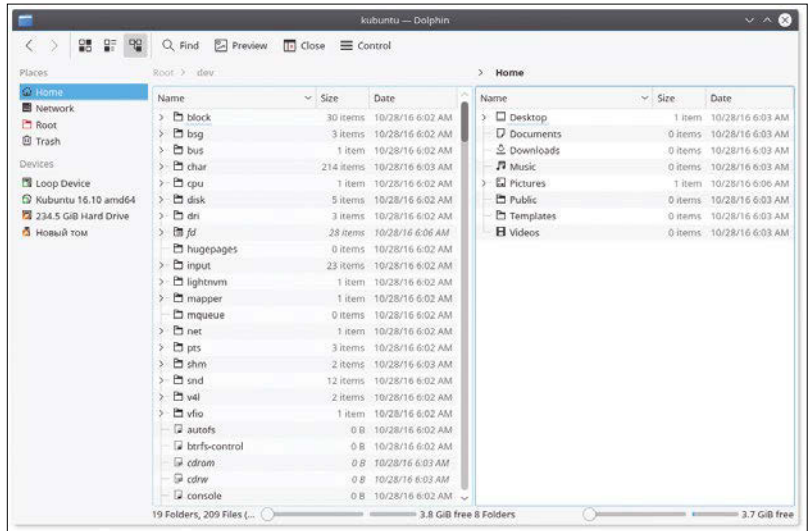


Figure 2: Many file managers, including Dolphin, rely on notifications to keep directory content current.

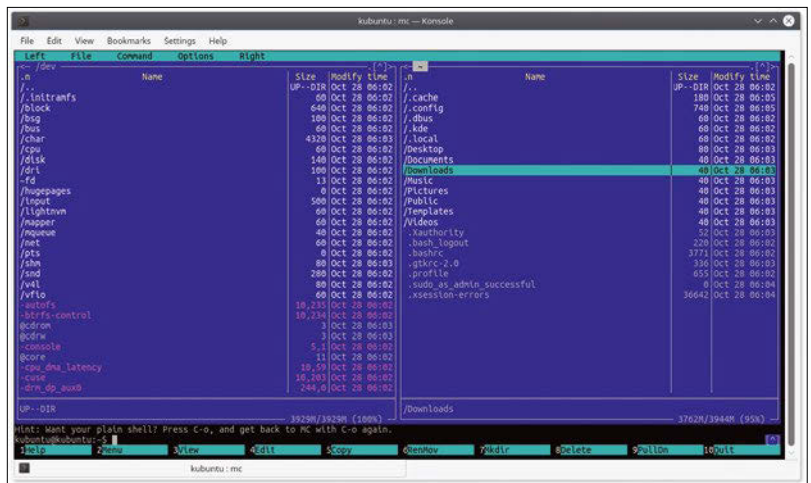


Figure 3: The fabulous Midnight Commander is an exception – you reload contents yourself with `Ctrl+R`. Patches, anyone?

event. The `wd` field contains the watch descriptor, which you can map back to a pathname (see `inotify(7)` for details).

```
if (ev->mask & IN_OPEN)
    printf("IN_OPEN ");
/* Handle other events here */
```

Listing 1: Read Event

```
01 struct inotify_event *ev;
02 char buf[4096], *p;
03 int n;
04
05 while ((n = read(fd, buf, sizeof(buf))) != -1) {
06     for (p = buf; p < buf + n; p += sizeof(*ev) + ev->len) {
07         ev = (struct inotify_event *)p;
08         /* The rest of code follows */
09     }
10 }
```

The `mask` tells what exactly has happened. Besides the `IN_*` flags you supply to `inotify_add_watch()`, it may contain the aforementioned `IN_IGNORED` or `IN_Q_OVERFLOW` if the in-kernel queue has overflowed. The queue can store up to 16,384 events by default. Although the size is adjustable via `/proc/sys/fs/inotify/max_queued_events`, there is always a limit. Otherwise, you leave your system open to a local denial of service (DoS) attack. When a queue overflows, the kernel discards further events (keeping memory usage constrained) until an application empties the queue or destroys it.

```
printf("%s ", ev->mask & IN_ISDIR ? "directory" : "file");
if (ev->len)
    printf("%s", ev->name);
printf("\n");
```

If the event pertains to an entry within the watched directory, `name` is its name, and `len` is the name's length. For subdirectories, `IN_ISDIR` is also set in the `mask`. Finally, `cookie` is an arbitrary but unique integer that links `IN_MOVE_FROM` and `IN_MOVE_TO` events (not shown here) together.

Despite all its goodies, the inotify API is somewhat limited: It doesn't support recursive operations, so to monitor a directory, including all its children, you'd have to add watches one by one. Keep in mind that the directory can change while you install watches, so your code should anticipate possible races.

Rename events pose another difficulty. Their two-fold split is natural from the kernel's point of view, but it leaves your code guessing as to whether it will get matching `IN_MOVE_TO` and `IN_MOVE_FROM` events. What if you don't monitor a directory to which the object moved? A common solution is to wait for `IN_MOVE_TO` for a few milliseconds. If it's absent, you conclude it won't appear at all. Although not ultimately robust, this approach is reported to produce accurate results 95%-99% of the time.

Inotify doesn't convey a PID for the process doing changes and provides no mechanism for access permission decisions. This doesn't mean inotify is flawed or should be deprecated. Much the opposite, many Linux applications rely on it; yet, there is some room for a more advanced API that addresses at least some of these issues.

Fanotify

Fanotify is such an API. It made a debut in Linux 2.6.36, almost five years after inotify. Conceptually, fanotify is similar to inotify, yet somewhat closer to the low-level kernel API they both use. It introduces a set of system calls to obtain a file descriptor and "marks" filesystem objects as being watched. Unlike inotify, fanotify can monitor a mounted filesystem as a whole. Monitoring a sin-

gle directory recursively is still impossible, though. Whereas the inotify file descriptor is read-only, its fanotify counterpart is writable, which is how you tell fanotify your access permission decisions.

To create a fanotify file descriptor, use the `fanotify_init()` system call:

```
int fd = fanotify_init(FAN_CLASS_CONTENT, O_RDONLY);
if (fd < 0) { /* You know what to do */ }
```

Compared with `inotify_init()`, calling `fanotify_init()` usually implies root privileges. The call accepts two bitmask arguments. The first defines fanotify behavior; `FAN_CLASS_CONTENT` or `FAN_CLASS_PRE_CONTENT` are required to handle permission events. If a single file has multiple watchers, `FAN_CLASS_PRE_CONTENT` wins and gets a chance to modify the file's data; `FAN_CLASS_CONTENT` runs next, so it sees the contents in their final form (hence the name). The default, `FAN_CLASS_NOTIF`, runs last and can't be used with permission events.

If you want a fanotify descriptor to be non-blocking, add `FAN_NONBLOCK`. To waive limits for the in-kernel events queue size and the number of watches, use `FANOTIFY_UNLIMITED_QUEUE` and `FANOTIFY_UNLIMITED_MARKS`, respectively. Keep DoS attack scenarios in mind if you use these arguments.

When a file produces some event, fanotify opens a new file descriptor and hands it over to the userspace code. The second argument to `fanotify_init()` tells how exactly to do it. It's the same as `flags` in `open(2)`. Here, you're not going to modify files, so read-only access is sufficient.

Next, you start adding marks. The `fanotify_mark()` system call multiplexes adding, removing, and flushing marks. An equivalent watch descriptor is not available in fanotify, so the call just returns zero if everything went okay:

```
int err = fanotify_mark(
    fd, FAN_MARK_ADD | FAN_MARK_MOUNT,
    FAN_OPEN_PERM, AT_FDCWD, argv[1]);
if (err) { /* Sorry, this didn't work out this time */ }
```

The fanotify file descriptor is `fd`, and `FAN_MARK_ADD` adds a mark. The whole mounted filesystem (`FAN_MARK_MOUNT`) is being monitored for file open permission events (`FAN_OPEN_PERM`). The last two arguments define an object to monitor. You'll find quite a few more possibilities in the `fanotify_mark(2)` man page. `AT_FDCWD` is a special value for the current working directory's file descriptor. If `argv[1]` is not an absolute path, it is treated as being relative to the current directory (`.`). This isn't utterly important here when installing a mount point mark, so any directory residing on a filesystem has the same effect.

Compared with inotify, fanotify's assortment of events might feel limited. At present, creating, deleting, and removing events are not supported: You can

watch files and directories being opened, accessed, and closed, and that's it. Moreover, `mmap()` generates no events. Fanotify isn't an inotify replacement; instead, it focuses on cases such as malware scanning and hierarchical storage management.

Now you can start looping for events again. Fanotify represents events as `struct fanotify_event_metadata`. In theory, it varies in size, so fanotify provides some macros to aid iteration (Listing 2).

Among the `struct fanotify_event_metadata` fields, three are most interesting: `mask`, `fd`, and `pid`. The `mask` bitfield stores `FAN_*` flags, as already discussed, plus `FAN_Q_OVERFLOW`, indicating an overflow event (unless the queue is unlimited). The `pid` process identifier generated the event, and `fd` is a file descriptor for an object to which the event pertains. The second argument to `fanotify_init()` dictates whether it is writable or not. Because there is no `metadata->name`, to retrieve it from `metadata->fd`, you just read a symbolic link in `/proc/self/fd`:

```
char path[PATH_MAX], real_path[PATH_MAX];
snprintf(path, sizeof(path), "/proc/self/fd/%d",
         metadata->fd);
int path_len = readlink(path, real_path,
                       sizeof(real_path));
if (path_len < 0) { /* That's an error */ };
real_path[path_len] = '\0';
```

Imagine you want to ban access to files containing `EvilSignature` in their first 1KB of data. Real antivirus software is far more sophisticated than that, but the big picture stays the same (Listing 3).

Operations on `metadata->fd` don't generate further fanotify events (there would be an infinite loop otherwise). To convey a permission decision, you write an instance of `struct fanotify_response` to fanotify's `fd`. You must set `r->fd` to the file descriptor in question, and `r->response` is either `FAN_ALLOW` or `FAN_DENY`. Fanotify handles permissions events in first-in, first-out fashion, and until you reply, the requesting process remains blocked. If access is banned, the process gets an `EPERM` error. However, when your fanotify application terminates, all unhandled permission events are granted implicitly. Keep this in mind if you design security software.

The above example was intentionally simple. Refer to the `fanotify(7)` man page for a more elaborate version.

Flying Higher

Having a native API for filesystem monitoring is good, but sometimes not exactly what you want. It might be too low-level for Python code, or it could be too Linux-specific for cross-platform development. For these situations, higher level libraries wrap platform specifics in an easy-to-use interface.

If getting filesystem notifications in Python code is all you need, consider `pyinotify` [3], best

Listing 2: Looping for Events

```
01 struct fanotify_event_metadata *metadata;
02 struct fanotify_response r;
03 char buf[4096];
04 int n;
05
06 while ((n = read(fd, buf, sizeof(buf))) != -1) {
07     metadata = (struct fanotify_event_metadata *)buf;
08     while (FAN_EVENT_OK(metadata, n)) {
09         if (metadata->fd >= 0 && (metadata->mask & FAN_OPEN_PERM)) {
10             r.fd = metadata->fd;
11             r.response = FAN_DENY;
12             /* The rest of the code goes here */
13         }
14         metadata = FAN_EVENT_NEXT(metadata, n);
15     }
16 }
```

Listing 3: Banning Files

```
01 const char EVIL_SIG[] = {'E', 'v', 'i', 'l', 'S', 'i', 'g'};
02 char sigbuf[1024];
03
04 memset(sigbuf, 0, sizeof(sigbuf));
05 if (read(metadata->fd, sigbuf, sizeof(sigbuf)) >= 0 && !memcmp(sigbuf,
06     sizeof(sigbuf), EVIL_SIG, sizeof(EVIL_SIG))) r.response = FAN_ALLOW;
07 write(fd, &r, sizeof(r));
08 close(metadata->fd);
```

described as “inotify, the Python way.” It supports both Python 2 and Python 3. To install watches with the `add_watch()` method, you use a `pyinotify.WatchManager` instance (usually a singleton). The `pyinotify.Notifier` class is a central dispatcher hub. `Pyinotify` provides a simple blocking and threaded notifier and integrations for popular Python asynchronous frameworks, such as `asyncore/asyncio` modules and `Tornado`. When an event fires, `pyinotify` runs a defined handler (a Python callable). If you want to chain handlers, consider using `pyinotify.ProcessEvents` instead of plain functions or lambdas.

On the other hand, `python-fanotify` [4] is best described as Python bindings to the native C API. This module comes from Google and has zero Python code inside except for `setup.py`. Documentation is missing as well (sans two examples), which is probably not a big deal. The API stays the same, except you prefix identifiers with fanotify plus a dot and rename them to match Python standards; so, `fanotify_init()` becomes `fanotify.Init()` and `FANOTIFY_EVENT_NEXT()` translates to `fanotify.EventNext()`, because Python has no notion of macros. For dessert, try `Watchdog` [5]. As opposed to the first two libraries, which wrap a single API,

this one abstracts several OS-dependent mechanisms, such as `inotify` on Linux and `kqueue` on FreeBSD. To use the library, you create a `watchdog.observer.Observer` thread object. Watchdog detects your target platform and chooses the appropriate notification mechanism automatically. Then, you implement an event handler, which is a class that inherits `watchdog.events.FileSystemEventHandler` and overrides instance methods like `on_moved()` or `on_created()`, which are probably self-explanatory. Now, you “schedule” monitoring for a specific directory with `observer.schedule()`. Watchdog can recursively monitor by passing the `recursive=True` keyword argument. Finally, spawn the monitoring thread with `observer.start()` and start receiving notifications.

Watchdog also provides the `watchmedo` command-line tool for your shell scripts. Watchmedo executes shell commands in response to various filesystem events and serves as a reference for how to use the library in a real-world project. ■■■

Info

- [1] A screenshot of Beagle’s UI: <https://commons.wikimedia.org/wiki/File:Beagle-search.png>
- [2] Apache License Version 2.0, January 2004: <https://www.apache.org/licenses/LICENSE-2.0>
- [3] pyinotify: <https://github.com/seb-m/pyinotify>
- [4] python-fanotify: <https://github.com/google/python-fanotify>
- [5] Watchdog: <http://github.com/gorakhargosh/watchdog>

Command of the Month: **inotifywait**

Watchmedo isn’t the only command to mate filesystem notifications with shell scripts. The `inotifywait` command, along with its cousin `inotifywatch`, is the de facto standard in Linux. Both come in a single package, often called *inotify-tools*.

The purpose of these tools is to wait for filesystem events in selected directories then dump some statistics. The difference is `inotifywait`’s output is easy to parse (and is configurable with `--format` and `--csv`), whereas `inotifywatch` prints a human-readable table (Figure 4).

The command-line syntax is also rather similar. You supply a path you want to monitor (either a file or a directory). The `-r` switch enables recursive operation. To exclude certain pathnames, use the `--exclude` key, which accepts a regular expression. Events to monitor are specified with `-e`. By default, `inotifywait` captures a single event, but you can override this with `--monitor/-m`. In this mode, the command

executes forever. To do the same thing, but dump events to a file rather than stdout, use `--daemon/-d`. This doesn’t apply to `inotifywatch`, which lasts until you interrupt it with Ctrl+C or a time out specified with `-t` and the number of seconds.

Study this snippet from an `inotifywait` session:

```
$ inotifywait -rm -e create,access /tmp
/tmp/ CREATE tmpfBncckr
...
/tmp/mc-val/ CREATE extfs1C1AQYMathJax.js
/tmp/mc-val/ ACCESS extfs1C1AQYMathJax.js
...
```

Here, I instructed Midnight Commander to open a ZIP archive and viewed a file in it. The output spans a few dozen lines: `/tmp` is a busy place on a live Linux system. ■■■

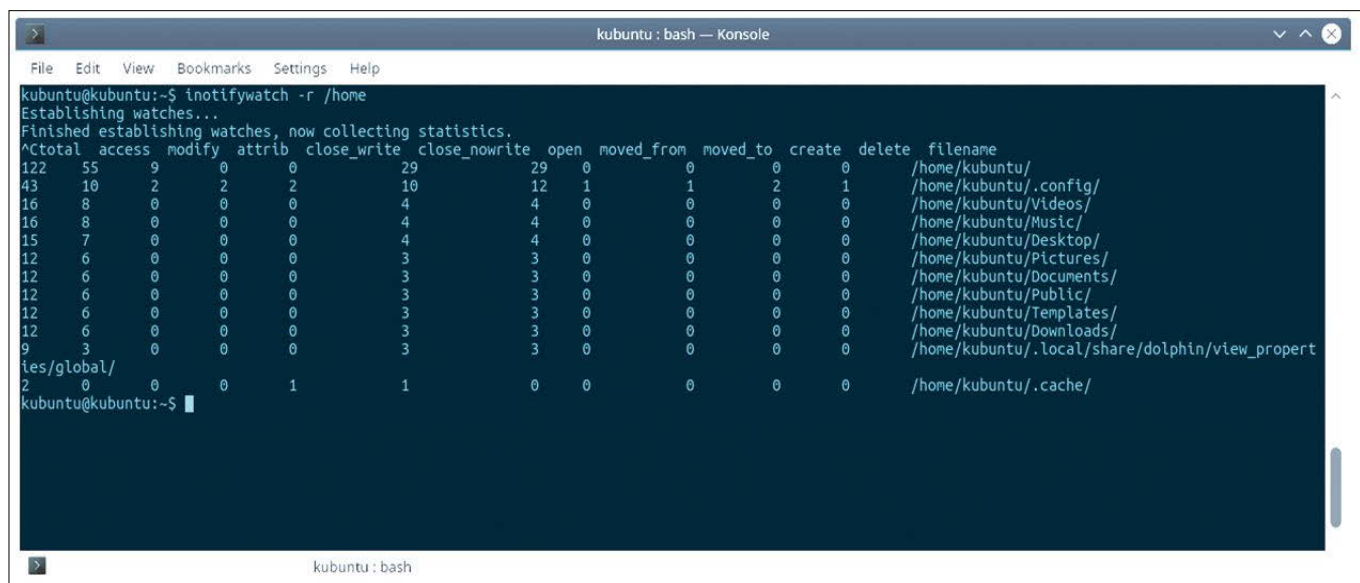


Figure 4: Inotifywatch can gather you some quick stats on what’s is going on in your `/home`, while you are out.

Shop the Shop

shop.linuxnewmedia.com

Want to subscribe?

Searching for that back issue you really wish you'd picked up at the newsstand?

Discover the past and invest in a new year of IT solutions at Linux New Media's online store.

shop.linuxnewmedia.com

DIGITAL & PRINT SUBSCRIPTIONS



SPECIAL EDITIONS



FOSSPicks

Sparkling gems and new releases from the world of Free and Open Source Software



Graham tears himself away from updating Arch Linux to search for the best new free software. **BY GRAHAM MORRISON**

Web Browser

QupZilla 2.0.1

Mozilla Firefox does a wonderful job, and it's a fundamental part of the future of the open web, but that doesn't mean you can't look at alternatives. Chromium, for example, is a legitimate open source version of Google's Chrome, even when Google's motivations toward the web are uncertain, and I'm happy to see Gnome's Epiphany is now being actively developed again. My current favorite is Qutebrowser, because navigating the web with Vim shortcuts is both addictive and productive. The only hitch with Qutebrowser is that the switch to the superior QtWeb-

Engine rendering engine is still in development – Qutebrowser's lead developer has just started a Computer Science course at the University of Applied Sciences in Rapperswil, Switzerland, which means that until the release of Qutebrowser 1.0, we're still looking for the perfect QtWebEngine browser. And QupZilla is a major contender.

What I really like about QupZilla is that it integrates so well with your chosen desktop, whatever that may be. It adopts themes and icons and provides native notification, helping it to feel just like a Gnome, KDE, or Xfce desktop application. This is particularly

important for KDE users, because there's not a particularly strong Qt-based browser that can compete with the GTK-based Firefox. QupZilla gets very close.

There is a very capable bookmarks manager, an overly comprehensive preferences pane, a full history overview, and even built-in adblock. Extensions include the wonderful Greasemonkey, KWallet, mouse gestures, and Flash cookie management. Although I don't want to emphasize this feature, it also includes Flash support for those vestiges of the Internet that still need this ancient artifact. I even appreciate the way the developer has defaulted to Duck Duck Go by default and the inclusion of a speed dial for quick web access.

Quick may even be QupZilla's best feature. Version 2.0 was updated to use Qt5 rather than Qt4, not only bringing much better web compatibility and rendering speed, but also all the many major improvements that have been made to Qt5 over the years. Despite offering nearly every option you could need for a browser, it still feels much lighter on your system. A non-scientific test showed that it took almost half of Firefox's 840MB of RAM when showing the same number of tabs with the same content, for example, and its rendering and page loading feels equally snappy. You do lose Firefox's add-on compatibility, and, despite its name, the project has nothing to do with Mozilla, but I'm happy with that compromise for 99 percent of my browsing needs, especially on older hardware.

QupZilla 2 is the closest thing to an open source alternative to Firefox I'd be prepared to recommend to users hitting the performance limit with Firefox, with the hope they'd still feel they were using a current generation browser, which means this release and its subsequent updates make it a major success. ■■■



1. Panel views. Bookmarks and viewing history are embedded within the main UI. **2. Tabbed view.** Tabs appear above the URL, although everything is configurable. **3. Adblock.** Stop advert adverts loading with integrated adblock. **4. Preferences.** Dozens of options for fine-tuning both the performance and the operation of QupZilla. **5. Extensions.** Several Mozilla extensions have been ported; plus, you can create URL rules. **6. QtWebEngine.** Great modern rendering with brilliant desktop integration. **7. Windows.** Break out panels into the surrounding space.

Project Website
<http://www.qupzilla.com/>

Boot Editor

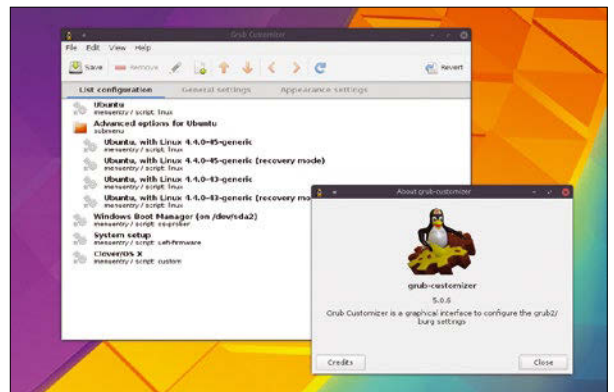
GRUB Customizer 5.0.6

One thing technological progress hasn't made any easier is booting. If you're running a single version of Linux, it's not too bad: Simply make sure you booted with UEFI if you want the GRUB boot manager installed using UEFI and that kernel updates update GRUB accordingly. Most of the time this is all handled automatically. However, things quickly get more complicated with more than one Linux installed, or even multiboot to Windows or OS X occasionally. Each operating system wants its own slice of EFI and UEFI, boot folders, and kernels, which won't always be automatically recognized by your distro when it updates GRUB.

At this point, you normally have to start messing with GRUB's configuration files, and because

most distros are now using a sub-derivative of GRUB 2, those configuration files need to be edited and updated in two stages – not a simple procedure. That's why I'm happy GRUB Customizer is still being developed. As much as I love editing configuration files by hand, a visual tool makes it so much easier and far less likely to make a boot-blocking mistake. It lets you add new boot entries manually, change the theme and resolution of your boot menu, remove the timeout, and change the list order.

Advanced options allow you to change boot flags and edit the specific options for each boot sequence. I successfully used GRUB Customizer to add both an auto-detect option for a Windows boot partition and a custom chain



Don't risk editing GRUB's boot files by hand – use a graphical configuration tool and blame the developer instead!

loader to detect a Cover EFI installation I use to boot OS X. Performing either of these tasks from the command line would have taken considerable research and left me with a system only booting Windows! ■■■

Project Website

<https://launchpad.net/grub-customizer/>

Software Synthesizer

ZynAddSubFX 2.5.4

This project had been languishing on SourceForge for years, but it's been resurrected, with new features added almost monthly. This is brilliant because ZynAddSubFX, a software synthesizer you can play on your desktop, sounds wonderful. To the uninitiated, this may seem easy to accomplish. Standard audio synthesis is well understood and relatively easy to model in software, but the nuance and subtlety that makes one implementation sound so much better than another is just as difficult to pinpoint in software as it is in hardware, and this is something ZynAddSubFX has – the nuance and subtlety of a classic synthesizer.

The synth itself supports three different sound engines. There's the subtractive sound of the clas-

sic Moog design, an additive engine for more experimental and atonal and microtonal timbres, and a standard set of FM sounds, too, plus a PAD engine for creating lovely rich and broad swathes of sound. None of this is really important because it's the results that matter, and the output from this synth, which includes effects to make everything sound much more professional, is fabulous.

ZynAddSubFX is capable of huge sounds that would fit into any style, and it offers control over every parameter. There are banks of example sounds for instant gratification and an advanced mode, so you can dive into the real details of a sound. This new release updates the toolkit when using it as a plugin (and ZynAddSubFX still works



Don't let its terrible name put you off; ZynAddSubFX is instant gratification for your ears.

well standalone) and fixes plenty of bugs. It also comes hot on the heels of the previous release, and that's what's so great about this renaissance in ZynAddSubFX – we're just glad it's back and getting better. We can't wait for version 3.0. ■■■

Project Website

<http://zynaddsubfx.sourceforge.net/>

CLI Upgrade

Powerline 2.5.1

If you've watched any presentations by prominent Linux developers over the past 12 months, you can't help but have noticed that many of their command prompts looked different from those of us mortals. Instead of plain old ASCII with perhaps a cleverly configured status line, you would see tiny hieroglyphics, icons, and chevrons, all rendering in 24-bit color. These colorful snippets of text and symbols update according to what the hacker is doing. While editing a text file in Vim, for example, these symbols show which mode the user is editing in, along with line numbers and document percentages. The default prompt in Bash gets visual cues that make it easier to see each command history, while even the humble Tmux gets hardware monitoring, local time, and Git repository status.

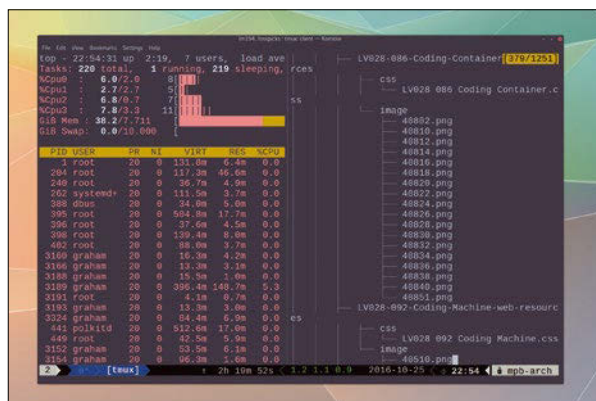
All of this is coming from a single tool called Powerline. Powerline started out as a humble status bar for Vim, adding a little visual flair and functionality without changing its classic console

aesthetic. But Powerline's minimalist and functional approach was soon extended by its users to include many other command-line tools, and the project soon outgrew its text editing roots.

The new project is a unified version of the old Powerline, offering an upgraded prompt plugin for Zsh, Bash, Tmux, iPython, Awesome, i3, Qtile, and, of course, Vim. The function of the prompt will depend on where you're using it, but Powerline will typically show your current location, automatically truncated to fit whatever window size you're using, complete with icons and

Powerline started out as a humble status bar for Vim, adding a little visual flair and functionality

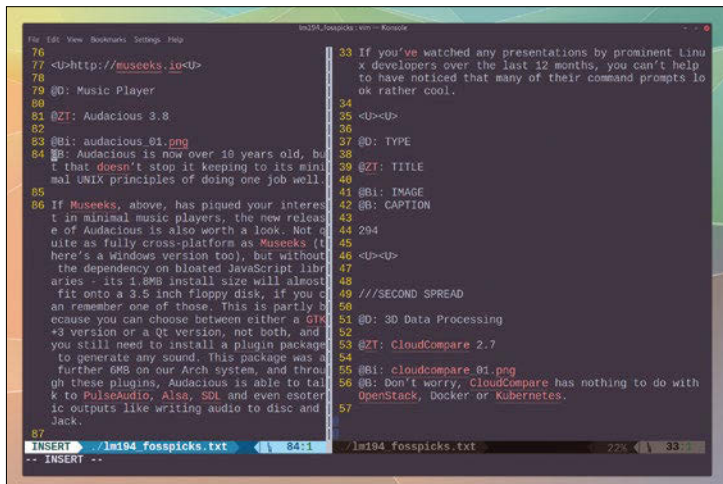
colors to help each session differentiate itself. However, this is just the default, and being a command-line tool, it's been designed to be modified. It's simple enough to write your own status renderer, for example, and thanks to the various configuration files and color



If you use the command line extensively, you need to install Powerline.

schemes using JSON, it's easy to modify the appearance of any plugin without even delving into the code.

Installation is relatively straightforward but not simple. As a prominent Python tool that works with both Python 2.7 and Python 3.x, it can be installed from PyPI with the pip command, but you also need to install a set of the customized fonts that are used to populate the status bar itself, which is how Powerline manages to appear like a graphical shell within the shell. You then configure your respective tools by editing their configuration files, usually by adding a link to a script or environment file installed alongside PowerShell. You can then modify and augment the installation to suit how you typically use the command line. The end result is a small concession to modern screens, colors, and design but a genuine enhancement to the command-line experience, without giving away any of the command-line functionality we're all finding so refreshing after all these years of using desktops. If you find yourself using the command line more and more and need a tiny piece of eye candy to enhance the view, Powerline is a perfect addition. ■■■



The project has become wider reaching, but the Vim plugin is still central to Powerline's enhanced functionality.

Project Website

<https://github.com/powerline/powerline>

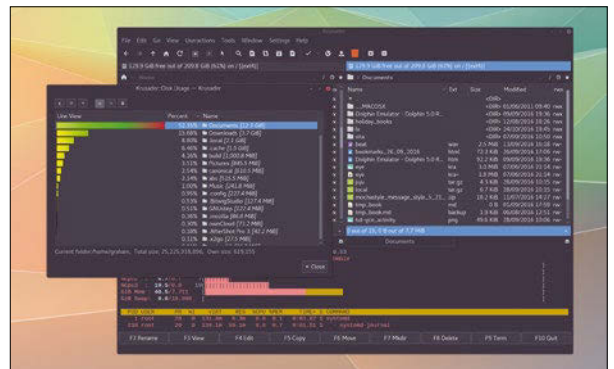
File Manager

Krusader 2.5

It might be the 21st century, but developers are still working on the idea of what makes the perfect file manager. On the one hand, cloud storage has brought stateless file management, such as the kind you find in Google Drive. On the other hand, we're all still accumulating hundreds of files on our local machines. Neither are we happy to trust entirely on the perpetuity of the cloud and its ability to recall an ancient photo with just a couple of tags or search terms, nor are we happy to trust our own organizational skills with local storage. That's where file management comes in, and KDE is particularly well furnished. It's default file manager Dolphin is brilliant. It lets you create and search for metadata, quickly preview all kinds of file types, and split the

view and out of the way. But Krusader, which after four years has finally made it to Plasma/KDE 5, is also a worthy alternative that will perhaps appeal to users who liked Konqueror.

On its first run, it will look for the tools it needs to do lots of clever things. KRename, for example, is used for intelligent multiple file renaming, and both KMail and Thunderbird are used for email access. Integration with GUI diff applications, including KDiff3 and Kompare, is good. Krusader is very good at not reinventing any wheels. The default view is a graphical equivalent to the command line and veritable Midnight Commander, and it's from this that Krusader takes most of its inspiration by using so many external tools. The view is split, mimicking the source/



One of the best thing about Krusader is that you can use it to browse your filesystem as root.

destination paradigm of the command line, along with quick access to the function buttons at the bottom of the view. When you get used to it, it's quick and powerful and I'm really happy to see developers still experimenting with files and modern frameworks. ■■■

Project Website
<https://krusader.org/>

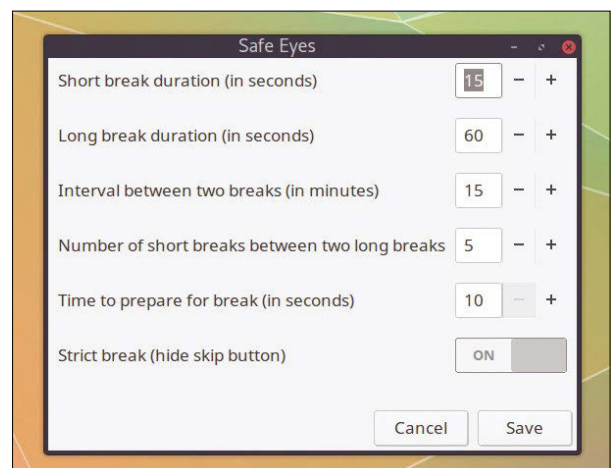
Breaks Reminder

Safe Eyes

Every time we visit the optician, we're asked how many hours a day we spend staring at a screen. Each time we want to tell the truth, but we don't want to quite admit to the extreme lengths of optical distress we put our eyes through. Rather like a 60-a-day smoker saying they only smoke a couple of packs, we usually only admit to staring at the screen for 12 hours or so. We don't like to think about the real number, whatever that might be, but there's one thing we do agree with the optician about – regardless of how much time you spend in front of a screen, it's imperative to have a break. Yet, this sad affliction and addiction that many of us suffer, where our lives aren't complete without the letters F and J beneath our fore-

fingers, also inhibits our ability to remember to take them, which is why we all need some small tool to remind us.

Rather than being a tool that puts a couple of eyes on your desktop that follow your pointer movements around, Safe Eyes is a tool to remind you to take a break. When installed and running, it sits quietly in the background, counting the seconds you spend browsing YouTube before you need to take a rest. You can obviously change the interval between these breaks, the time to prepare for a break, and the duration of short and long breaks, and Safe Eyes will not only remind you but also prompt you to "Roll your eyes" or "Walk for a while." You can even disable the keyboard and the ability to



Sadly, the Safe Eyes applet icon doesn't follow your pointer around the screen.

cancel a break. It might sound a little patronizing, but that's exactly what we needed to break our Reddit reverie and get up for a few moments, and our eyes felt all the better for it. ■■■

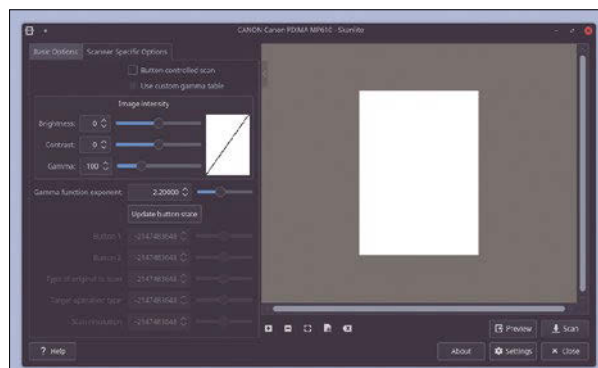
Project Website
<https://github.com/slgobinath/SafeEyes>

Image Scanner

Skscanlite 2.0.1

Scanners are a little like the printers they're so often attached to these days – they're an old technology whose usefulness keeps them in many of our offices. Despite nearly all communications being digital, there still seems to be a need for physical documents and their resultant photocopies and digital copies, which means we still need to use scanning software to control the hardware that makes these copies. Thanks to the SANE back end, Linux has had excellent scanner support for years, and most graphical applications, such as Gimp, will let you import images whenever you need them. But, many more applications are no longer being developed, especially when you need a perfunctory office appli-

cation that will just let you get the job done. Skscanlite has been KDE's default scanning tool for a while, and this is a major update (plus bugfix) that finally sees this essential tool make it to Qt5. This is important because at 600 or 1,200dpi, scanning can still place quite a toll on your system's memory and storage, and the latest versions of Qt are much better at handling this than the old versions. I also really like Skscanlite's UI, as it does a good job at mimicking the similarly perfunctory utilities that come bundled with both Apple's OS X and Microsoft Windows, which is ideal for when Linux is being used in an office, which is where this is going to be most useful. As you'd expect, you can preview the page you're about to scan, adjust resolution,



Until printers and scanners finally cancel themselves out, Linux is great at ensuring old hardware scanners still work.

and process the image as it's being scanned, saving you time when you need to use OCR or send a signed document onward. Skscanlite works exactly as you'd expect, and I'm genuinely thankful that the developers have taken the time to future-proof a fundamental tool when similar tools have been left languishing. ■■■

Project Website
<https://www.kde.org/applications/graphics/skscanlite>

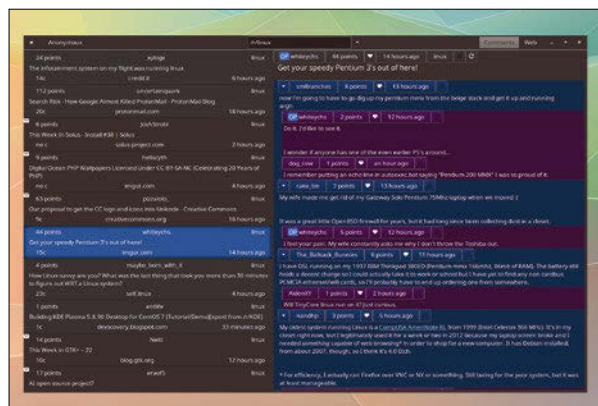
Time Waster

Something for Reddit

In my experience, most applications that attempt to take a successful web forum, such as Reddit, and squeeze its functionality into a traditional desktop application fail. This is because these services have grown popular purely because they're part of the world wide web. Remove the ubiquity of the browser and you lose an interface that's tested by millions of users millions of times a day.

Web interfaces are seldom perfect, but successful ones keep out of the way and let the community get on with it. However, there are exceptions to this. In particular, I'm a huge fan of Materialistic, a beautiful Android application that aggregates Hacker News content. Thanks to its material design, and the way it tabs between the original article, the comments, and the new items themselves, it's a huge

improvement over the functional web interface, and my preferred way of reading those stories. Reddit is a different beast entirely, of course. Diode on Android is great if you're strapped for bandwidth, but I've yet to find a desktop application better than Firefox. Something for Reddit is a great attempt, especially if you've got a touch screen. Built around GTK and WebKit2 rendering, it's much like an augmented version of the website or a Reddit Firefox extension. You can log in, quickly access your favorite subreddits, and either view the comments or read the original article, a function that's quicker than the web interface. The design is lovely, with both a dark and light theme, and the colors used for comment demarcation are a big upgrade over the defaults on Reddit. It also uses Vim



"Finally, something good comes from Reddit"; discuss.

keybindings for rapid navigation. If you already spend too much time on Reddit, don't install this app. If you need more friends placing little green soldiers everywhere (*/r/GreenDawn*), install this app. ■■■

Project Website
<https://github.com/samdroid-apps/something-for-reddit>

Emulator Front End

Attract Mode 2.1.0

We've been running a Manjaro-based arcade machine for a while. It's built into an original Bubble Bobble cabinet from the 1980s, but the PCB that once ran the classic Bubble Bobble has been replaced with our Linux PC, now capable of running all kinds of games from all kinds of systems. To provide easy access to all these systems from the joystick interface, our preferred front end has always been a great tool called Wah!cade.

Wah!cade is a highly configurable graphical interface that lets you easily create or use themes to access your game collections. With MAME arcade game emulation, for example, it's relatively easy to use Wah!cade

to construct a script that will show you a game's history, its original PCB design, and how the controls are displayed, as well as letting you switch between emulators and favorite lists of games.

But Wah!cade requires a fair amount of configuration, and its development pace has slowed in recent years, so when we upgraded the screen in our arcade cabinet, we thought that rather

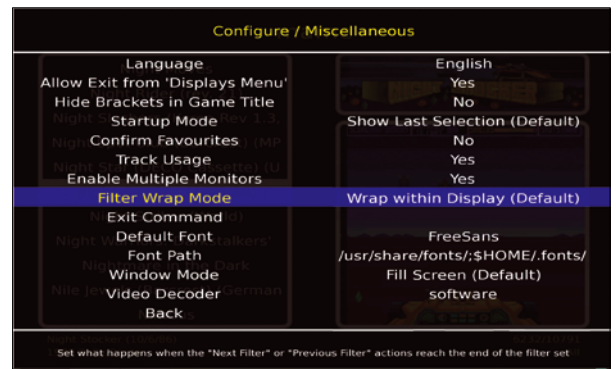
than rescript our Wah!cade theme, we'd upgrade the software we used as a front end, too. Unlike with Windows, there aren't a huge number of front ends to choose between, but there is one significant new option that wasn't available a couple of years ago, and that's Attract Mode.

The first thing you discover when you install Attract Mode is that it's even more configurable than Wah!cade. Despite being written in C++, its comprehensive theming engine interprets layout files written in the Squirrel programming language. This is a light, high-level object-oriented language that doesn't seem difficult to learn, and editing prebuilt themes should be no different

The first thing you discover when you install Attract Mode is that it's even more configurable than Wah!cade.

from editing JavaScript. But even if you never open a layout file yourself, Attract Mode looks brilliant without any further tinkering.

The default theme is an animated Pac-Man background, for example, and works well with



Almost everything about Attract Mode and the emulator you're using can be configured from within the menu system with a joystick.

most screens and aspect ratios. You can easily use the on-screen menu while the application is running to change many of the settings and switch between several of the included alternative themes. Any one of these will fit most screen ratios and resolutions automatically, which is ideal if you're running a screen with vertical orientation. Themes from other users simply need to be dropped into the `themes` folder for them to appear. Attract Mode is also fast and cross-platform, and it includes both Arduino and Raspberry Pi support, along with Windows and OS X. The community is also very active, with many users sharing and developing their own themes.

Getting started is as easy as installing the package and running the application. First time users are guided through an initial menu tree for adding details about which emulators you want to run. MAME is obviously well supported and is the emphasis for most themes, but Attract Mode supports almost every emulator you can think of, and it's a perfect interface to these and their games if you're building a standalone gaming machine, even if that's a Raspberry Pi and an old joystick. ■■■



Attract Mode is named after the attention-grabbing demos and sound cues that lonely arcade machines autonomously ran in cinema lobbies.

Project Website
<http://attractmode.org>

GAMING ON LINUX

The tastiest brain candy to relax those tired neurons



BY MICHEL LOUBET-JAMBERT



Michel Loubet-Jambert is our Games Editor. He hasn't had a decent night's sleep since Steam came out on Linux.

The Vulkan API has generated plenty of exciting news recently. The developers behind the immensely popular Unity game engine released a preview version of the editor with support for Vulkan, which means we should see Unity games using Vulkan in the not-too-distant future. Likewise, the porting house Feral Interactive (Tomb Raider, Mad Max, and Shadow of Mordor) has stated that it will be releasing Vulkan games in the future, although the recent *Deus Ex: Mankind Divided* uses OpenGL.

Meanwhile, at Valve's Steam-DevDays, their Vive VR system has been shown off finally running on Linux using none other than the Vulkan API, rather than OpenGL. Interestingly, this was done on Kubuntu rather than SteamOS, although there were also signs that Valve has far from abandoned the platform. Numerous Steam Machines set up at the event were running SteamOS, including *Rocket League* and some of Valve's own games.

Fans of the hugely successful *Civilization* franchise were somewhat bemused when the sixth iteration was released without Linux support, after getting *Civilization V* some time back and *Civilization: Beyond Earth* as a day 1 release. The porting house Apyr has said that the feasibility of the port is still being analyzed, and we should hear news about it soon; so, fingers crossed. This hasn't stopped a few enthusiastic gamers from sending goodies to their offices on a couple of occasions, presumably to guilt-trip a port out of them.

Mad Max

A literal sandbox

Web: <http://store.steampowered.com/app/234140/>
Price: £24.99/\$29.99/EUR29.99

This classic post-apocalyptic franchise has made its way off the big screen and onto Linux in what is one of the most expansive sandbox games we've seen to date – and certainly the most sandy.

The game starts off with Max losing all of his possessions, which lays the groundwork for a story revolving around building up what he lost. His most prized possession is his car, and building the ultimate machine (the *Magnum Opus*) from scratch and slowly upgrading it is incredibly satisfying. The game's other mechanics are numerous, including upgrading strongholds and Max's abilities to clear regions of threats by completing a number of different challenges, such as racing and brawling through enemy strongholds. Needless to say, the game is extremely varied, providing a range of things to do both on foot and behind the wheel.

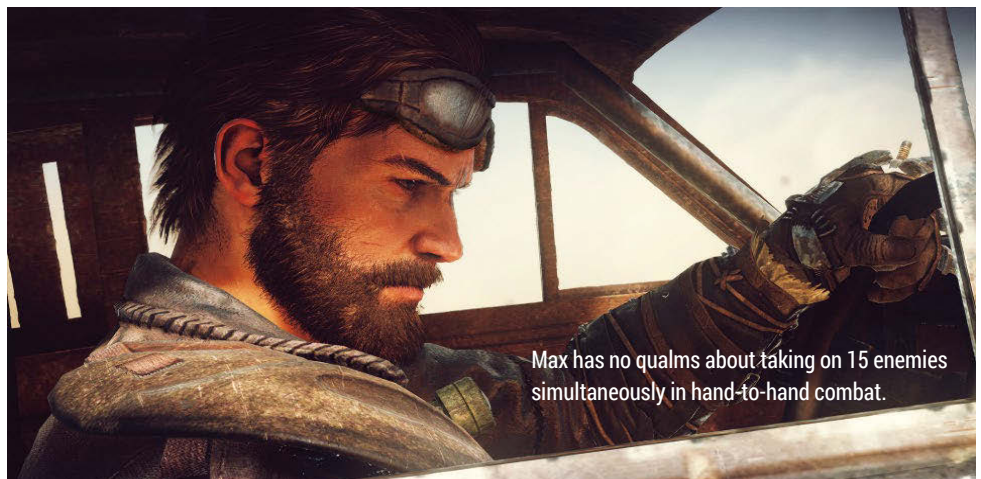
Where the game does fall short is in the limitations of its interactions. NPCs are repetitive and lack any personality, limiting the world's immersion and lore, which is mostly explained through an encyclopedia-like sys-



Car combat is exhilarating, especially as the car gains new abilities and weapons.

tem rather than characters and dialog. This also makes exploring somewhat less enticing, since the odds of finding any meaningful interaction and exciting side-missions are next to none. That being said, the game has some good ambient storytelling through its populated areas and abandoned relics from a fallen civilization. It also does a good job in conveying the scarcity of resources in its world, with water, fuel, and ammunition hard to come by, while having to feast on rodents and old dog food show just how much of a struggle it is to survive in this harsh world.

The graphics for this game are superb, with the barren wastelands managing to look fantastic. The lighting in particular makes the game look extremely cinematic, particularly at sunset, and rocky cliffs give impressive views of what little is left of Max's world. Plenty of gameplay hours are available that stay entertaining throughout, making *Mad Max* highly recommended. ■■■



Max has no qualms about taking on 15 enemies simultaneously in hand-to-hand combat.

Aragami

A visually stunning stealth game

Web: <http://store.steampowered.com/app/280160/>
 Price: £14.99/\$19.99/EUR19.99

This third-person stealth ninja game allows the player to make use of shadows to navigate around the medieval Japanese world and sneakily assassinate enemies. The visuals, although not the most detailed, make use of a very effective cartoonish style, which looks absolutely fantastic without hitting your graphics card with a sledgehammer. That said, it has some performance issues related to optimization, although many of these have been patched out gradually. The Japanese-style music also compliments the graphics well to create a decent atmosphere.

The story revolves around summoning of the lead character and uncovering his past, which isn't all that fantastic, but the



Aragami's visual design and satisfying mechanics more than make up for the lackluster story.

game makes up for it in the gameplay department. Moving around stealthily and taking out enemies with a sword or demons is incredibly satisfying, and the game cleverly makes use of Aragami's cape, which glows to show the effectiveness of the current cover, leaving the interface uncluttered. The stealth mechanics are key because being seen leads to almost instant death, and killing enemies is actually optional.

Overall, this is a very good option for those looking for a stealth game, and although a little short, it's still a very satisfying experience. ■■■

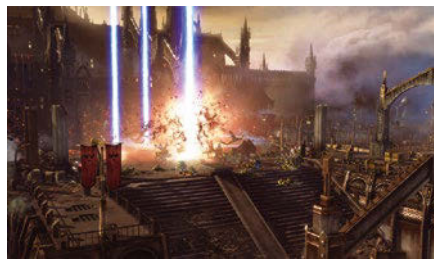
Warhammer 40,000: Dawn of War II

... and the standalone expansions

Web: <http://store.steampowered.com/app/15620/>
 Price: (original game) £14.99/\$19.99/EUR19.99

Warhammer and real-time strategy (RTS) game fans alike have a lot to be excited about with the simultaneous release of three games: Dawn of War II and the Retribution and Chaos Rising standalone expansions.

Some very satisfying campaign modes, each tailored to the different races of the world, give many hours of gameplay before even venturing into the multiplayer realm. The game mixes up traditional RTS elements by introducing role-playing game (RPG)-style leveling and character upgrades to specific units, as well as item drops from battles for further customization, keeping the game fresh and interesting. Similarly, the use of items' cover mechanics keep the gameplay varied and more tactical than the conventional base-rushing



The game's epic structures and landscape create a perfect backdrop for its large battles.

and victory-through-number styles one might be accustomed to in such games. The expansions add a lot of different content, ranging from game modes to new story campaigns, and deserve to be standalone games.

Although the games were released on other platforms a while back, they hold up extremely well and are a worthy addition to the Linux roster. Chaos Rising sells for £14.99 (\$19.99/EUR19.99), like the original game, whereas Retribution is £19.99 (\$29.99/EUR29.99), although all three are available in a bundle at a lower price. ■■■

ALSO RELEASED...



Doorways: Old Prototype

As the name suggests, this game is a previously unreleased prototype game made before the Doorways horror series, which the developers released for free. It's a good one to download if you're a fan of the franchise or if you're simply strapped for cash but feel like a bit of horror-themed gaming. It's more of a platformer than a traditional horror game, but it has a nice eerie atmosphere.

<http://store.steampowered.com/app/525780>



Higurashi When They Cry

If you're a fan of visual novels, then the Higurashi series is a good one, with its fourth chapter recently released and its former three also on Linux. It has a dark and disturbing story with plenty of murder under the guise of regular high school life in a rural village. Intrigue and twists are plentiful, much like the anime series on which the game is based.

<http://store.steampowered.com/app/526490>



Eventide 2: The Sorcerers Mirror

Experience the joys of Slavic folklore in this charming hidden object game featuring legendary characters like the outlaw Yanosik in a story that involves saving a quintessentially Slavic village from an evil mage. This game provides a satisfying casual experience for those looking for the usual trappings of a point-and-click game, but who don't have the time to sink into or have the patience for long and difficult puzzles.

<http://store.steampowered.com/app/510650>

Cool and Collected with collectd

The collectd tool harvests your system stats and stores them for plotting into colorful graphs.

BEN EVERARD

Why do this?

- Find out all there is to know about your Linux Box
- Optimize your setup for maximum performance
- Generate pretty graphs to impress your technically inept boss

Linux has a host of command-line tools for probing what's going on under the hood. To list just a few examples, there's `top`, which shows what processes are using which resources, `df`, which shows how much disk space is free, and `iftop` to show how much data is flowing through the network.

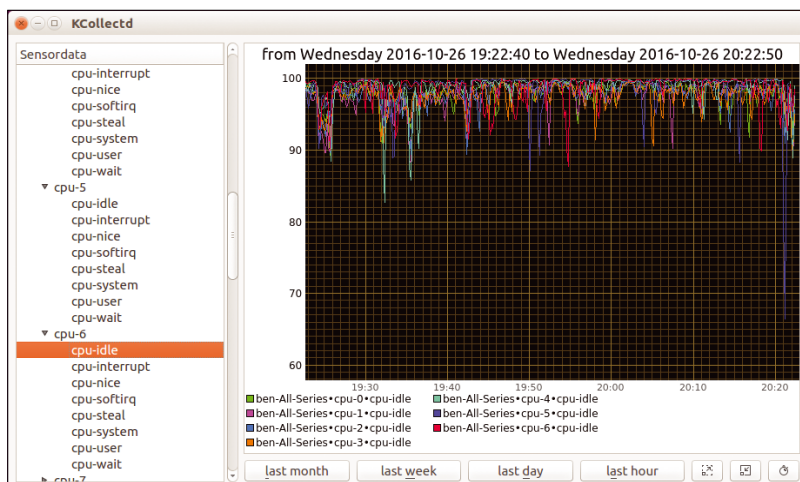
If you really want to get fine-grained details, you can poke about in the `/proc` filesystem that the kernel automatically populates with precise details of everything that happens, but it can be a little too low-level for mere mortals to understand. Although all these tools are useful, they all have one fatal flaw: They only tell you what's happening right now. In this article, we're going to look at `collectd` [1], which hoovers up your system stats and stores them, ready to be interrogated and plotted into colorful graphs.

You'll probably find `collectd` and `kcollectd` (the graphical front end we'll be using) in your package manager. On Debian, Ubuntu, and derivatives, you can install them with:

```
sudo apt install collectd kcollectd
```

On non-Debian-based distros, you might also have to start the `collectd` service to ensure it's collecting details. You can see if it's running with:

Figure 1: Graphing CPU usage. It turns out you could push this machine quite a bit harder.



```
sudo service collectd status
```

If it's not running, you can start it with:

```
sudo service collectd start
```

You can ensure it starts every time you boot your machine with:

```
sudo chkconfig collectd on
```

If your distro doesn't have the `service` or `chkconfig` commands, you'll need to consult the documentation on managing services. At the most basic level, that really is all there is to it. Once you've installed `collectd`, it'll run in the background, and when you launch `kcollectd`, you can build graphs

Networking

In this tutorial, we've looked at monitoring one machine with `collectd`. This works really well, and it's a great addition to your Linux setup. However, `collectd` is also useful for monitoring multiple machines at a time. Using the network plugin, each `collectd` instance can act as either a client sending data or a server collecting data. The server collates data from all the clients, and you can use any of the graphing tools to compare different machines. The basic setup for a client is:

```
LoadPlugin network
<Plugin network>
    Server <server>
</Plugin>
Where <server> is replaced with the IP address or hostname of the server. The basic setup for a server is:
LoadPlugin network
<Plugin network>
    Listen "0.0.0.0"
</Plugin>
```

Typically, this will all happen within a local network, so it's often fine to run without encryption. If you do need to send your data over a public network, then consult the `collectd` man page for details on how to ensure it remains private.

Collectd-web

If you want to use collectd on a server (or other headless box), kcollectd is a little awkward. You could launch it via SSH with X forwarding enabled (by adding the `-X` flag to your `ssh` command) or use collectd's networking options to send the data to a machine with a display, but neither of these options is really ideal. Because collectd just handles collection and storage of data, plenty of alternative graphical front ends are better suited to network access. Perhaps the easiest to use of these is collectd-web.

There's no installation for this software, but you will need a few dependencies, which you can download with:

```
sudo apt-get install librrds-perl \
    libjson-perl libhtml-parser-perl
```

Once this is done, just download the code from GitHub with:

```
git clone https://github.com/httpdss/collectd-web.git
```

Then `cd` into the new `collectd-web` folder and run:

```
python runserver.py
```

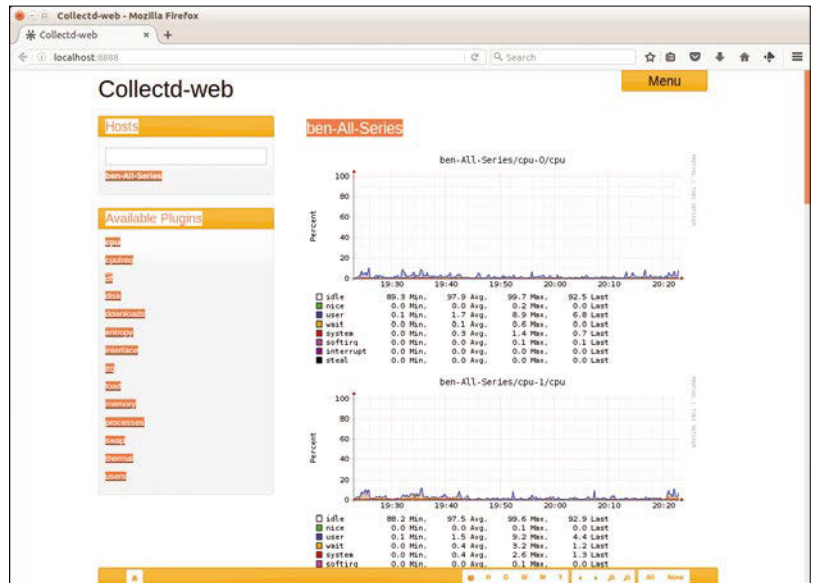
By default, this will run a server on port 8888 that you can view by pointing your web browser to `http://localhost:8888`.

Collectd-web works slightly differently from kcollectd in that it automatically graphs all the data for a given plugin instance (Figure 2).

Although collectd-web does a good job of making it easy to view your data from other machines on the network (Figure 3), power users might want a little more functionality. Some really powerful visualization tools can work on top of collectd, but they can require a bit more setup and are beyond the scope of this article. Here are a couple of our favorites:

- Elasticsearch, Logstash, and Kibana (aka ELK). Collectd can forward data to a Logstash server, which in turn stores it in Elasticsearch. This data can then be visualized with the Kibana web visualization tool. If you use Logstash for analyzing your logfiles as well, this can provide a really powerful way of monitoring everything that's happening on your system. The Timelion plugin to Kibana adds powerful additional graphing tools for really seeing what's going on.
- Grafana can take data from collectd either through Elasticsearch (as above) or via the Graphite database. This is perhaps a little more flexible than ELK if you're not also using Logstash for managing your log setup.

Just as with the plugin, base monitoring is flexible enough to work with almost any system, the selection of front ends means that you can use collectd to work with the system that you have rather than having to orientate your monitoring around collectd. The best advice we can offer on this is to start monitoring with a simple setup and then configure it as you learn the most useful data and graphing tools for you.



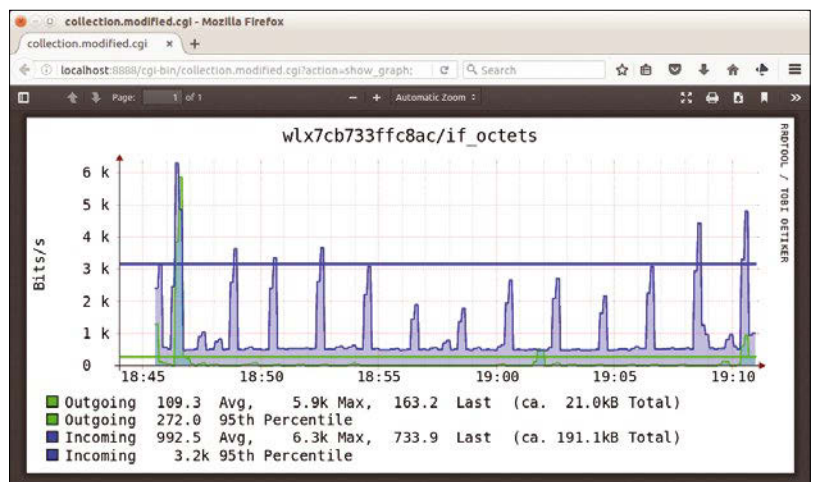
showing how system resources have been used over time.

On the left-hand side of the kcollectd window, a `Sensordata` widget should show a list of all the hosts for which you have data. Initially this will just be the host you're currently running on, but if you set up networking (see the "Networking" box), the other hosts will appear in the list on the server. If you expand the host, you'll see a list of all the plugin instances for which collectd has data. So, for example, the CPU monitoring plugin (Figure 1) monitors each CPU core separately, so you'll see `cpu-0` to `cpu-(n - 1)` (on an n -core machine).

Each of these plugin instances has one or more pieces of data associated with it, so sticking with the `cpu` plugin, if you open `cpu-0`, you'll see data for the various states of the CPU such as `cpu-idle` (the time the CPU's not doing anything), `cpu-wait` (the time the CPU's waiting for disk access), and `cpu-user` (the amount of time executing userspace code). If you drag all of these from the left side of the screen onto the right, you'll get a graph showing what your CPU's been up to. The buttons at the bottom allow you to select the time period to view.

Figure 2: The collectd-web interface makes it easy to access data across the network.

Figure 3: You can export graphs from collectd-web in a variety of formats, including PDF, which makes them easy to share with other people.



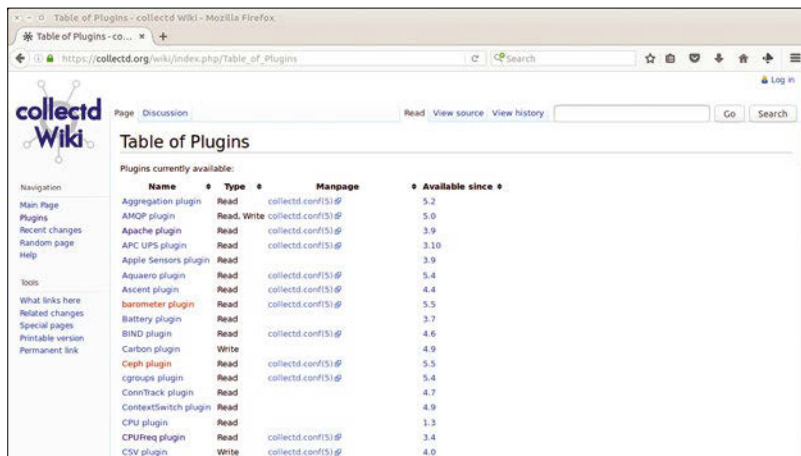


Figure 4: A lot of collectd plugins are available; head to the project’s wiki to find out how to configure them.

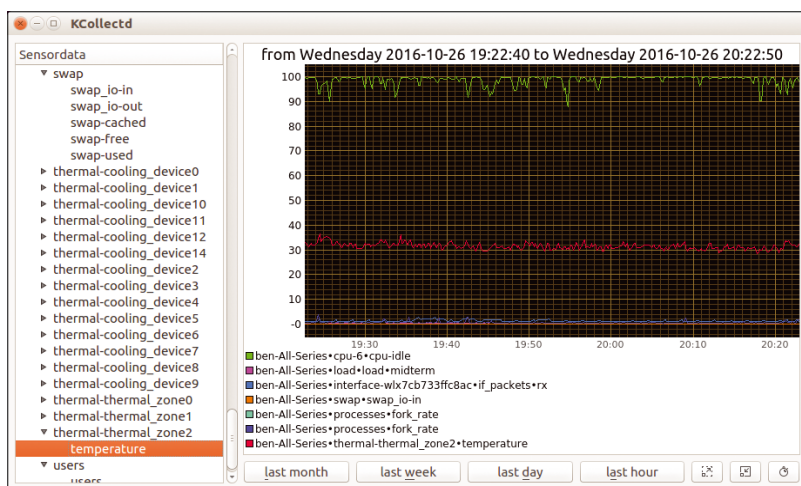
Because this graph has only just been set up, the *last hour* button will be most useful, because the others will be mostly empty.

You can plot more than one graph on a kcollectd window by right-clicking and selecting *add new subgraph*. When you have a view you like, you can save the set of graphs – this only saves the layout, not the data itself, which is stored in collectd. Kcollectd is a good starting point for exploring collectd, but you might want to monitor servers or other machines without graphical displays. A number of other front ends to collectd might be more suitable in this instance. See the “Collectd-web” box for one that’s easy to use.

Every system has different limitations and problems, so there’s no one set of graphs or data that are best for all collectd instances. Instead, collectd is incredibly flexible and lets you set it up for the data you need.

Although collectd will run on most distros with the default settings, you can tweak a lot of things to make it more useful. The collectd config file is usually located at `/etc/collectd/collectd.conf` (although some distros have it at `/etc/collectd.conf`). Open that file in your favorite text editor (you’ll need root privileges), and you’ll find an XML-like format for the configuration with some data (but not all) enclosed in tags.

Figure 5: You can put any data you want together on a graph in kcollectd, even if the units are completely different. It’s up to you to ensure it makes sense.



By default, collectd will collect data every 10 seconds. This is a good starting point, but isn’t always best. If you want more data and have plenty of storage space, you might want to decrease this number. On the other hand, if you’re pulling data together from lots of different machines, you might want to increase this number so you avoid swamping your main collectd instance.

Let’s look at some things that could be particularly interesting to the home PC user. Many modern machines come with the ability to scale the CPU frequency depending on system load. This means that as you run more things, the BIOS increases the clock rate of your processor so it can perform more work, but as the load decreases, the clock rate drops back so that it’s more power efficient. This is a useful feature, but it does mean that it’s not always clear how much of your CPU time you’re using.

Connected to the CPU frequency is the temperature. Faster CPUs run hotter, and if they get too hot, your system might crash or, potentially, damage itself. Hot CPUs can be a sign of insufficient cooling (possibly caused by dust-clogged vents) or simply of the system running faster than it can handle.

The functionality of collectd comes from plugins [2], and the two you need are `cpu-freq` and `thermal` (Figure 4). To load these, just add the following lines to `collectd.conf`:

```
LoadPlugin cpu-freq
LoadPlugin thermal
```

Now restart collectd with:

```
service collectd restart
```

Re-open kcollectd and you should see new options for `cpu-freq` and `thermal` in the sensor data list (Figure 5). You can drag these over to the graph to see what’s going on. If they aren’t there, you might need to install the plugins separately on your distro, so check the package manager for available software.

Neither of these plugins required any configuration to use, but some others require a little bit of tweaking to make sure they are doing what you want. For example, by default, the `processes` plugin will gather information about the number of processes running on your machine, but not details about individual processes. With a little tweaking, you can change that.

You don’t want to grab all the available info, because you’ll quickly fill up your hard drive with information you’ll never need, but it can be useful to pull out particular software to see what’s going on. When using collectd on a server, this could be in-

formation about the various services that are running, but on your desktop, you find that you have most performance issues with the Firefox web browser. Let's change the configuration to see how many resources that's using.

Back in the `collectd.conf` file, you need to add a configuration. There are two ways to find particular processes to match: by name or by regular expression. Because Firefox is always launched with the `firefox` command, you can use this. However, if you need to monitor something that's launched in different ways (or you want to monitor a range of different processes), look at the collectd man page for more details.

The configuration can go anywhere in the configuration file, provided it's below the `LoadPlugin Processes` line.

```
<Plugin "processes">
    Process "firefox"
</Plugin>
```

Once you've made this change, you'll need to restart collectd for it to take effect.

Collectd sucks up vast quantities of data about what's going on in your computer, but it's not always easy to make sense of it. Now, let's look at some of the key information coming in from the `processes` plugin. If you open `kcollectd`, you should now see a `processes-firefox` list; inside is a range of options. The most important of these are the `ps_cputime` values for system and user, which give the average amount of time (in microseconds) the process uses every second. The system entry gives the amount of time spent in system calls, whereas the user entry gives the amount of time spent in userspace. The second most important is the `ps_rss` value – here, `rss` stands for resident segment size, which is the amount of physical memory (not including swap) that the process is currently using.

What we've looked at so far allows us to tune the behavior of collectd using prebuilt features. However, we all have different setups and different requirements, and it's unrealistic to expect a monitoring system to have all the capabilities we need for all our computers. Perhaps the best feature of collectd is that it's easy to write your own add-ons.

Although these can be full plugins that collectd loads, there's a much easier text protocol that you can take advantage of using the `exec` plugin. This plugin will run some command and take values from the output. You can use any programming language that can print text output to add data to collectd. We're going to use a Bash script, but as you'll see, it's trivial to convert this to another language.

One thing we're particularly prone to is filling our `Downloads` directories. Trying out different distros

Listing 1: Monitoring the Downloads Directory

```
01 #!/bin/bash
02
03 HOSTNAME="${COLLECTD_HOSTNAME:-`hostname -f`}"
04 INTERVAL="${COLLECTD_INTERVAL:-10}"
05
06 while sleep "$INTERVAL"
07 do
08
09     USED=$(du -sb /home/ben/Downloads | cut -f1)
10     echo "PUTVAL $HOSTNAME/downloads/counter-used interval=$INTERVAL N:$USED"
11
12 done
```

can quickly overwhelm even large hard drives, so we've made a plugin to monitor the amount of space taken up by just this directory. To find this out, enter the command:

```
du -sb /home/ben/Downloads | cut -f1
```

This uses `du` (disk usage) with the `s` (summarize) and `b` (bytes) flags to get the amount of data in `Downloads`. The `cut` tool then removes everything from the command's output except the first field, which contains the data you're looking for. You need to translate this into the collectd text file format, which means your script should output lines in the form:

```
PUTVAL ben-All-Series/downloads/used-counter 2
interval=10 N:5638101902
```

In this format, `PUTVAL` is the command that sends data to collectd, `ben-All-Series` is the name of the computer this tutorial was written on, `downloads` is the collectd plugin instance, `counter` is the data type, and `used` is the identifier for the data. To create this, you just need to wrap up the `du` command in a loop and an echo statement to send this command to stdout at the frequency passed to collectd in the environmental variable `INTERVAL`. See Listing 1 for details.

That's our collectd system up and running. Using these techniques, you should be able to get almost any monitoring information you need, and this should help you use your hardware more efficiently, whether that's working out the best settings for gaming or performance-tuning the enterprise software in your data center. ■■■

Info

[1] Collectd: <https://collectd.org/>

[2] Plugins: <https://collectd.org/wiki>

Windows Apps on Linux with Wine

Convert your home/office/friends to be completely Microsoft-free with our tips and tricks.

MIKE SAUNDERS

While we all love to shout from the rooftops about the awesomeness of GNU, Linux, Free Software, and open standards, we also have to deal with the awkward reality that Windows is still pretty dominant on the desktop and has an enormous software base. Sure, FOSS equivalents for proprietary Windows apps exist (and new ones are popping up every week), but certain tools simply don't exist elsewhere, and people or companies that want to migrate away from Windows toward Linux might still need to run the occasional Windows program.

Of course, dual-booting is one way to solve this problem, but it still means you need a Windows license and a chunk of your hard drive devoted to it. If you want to be as Microsoft-free as possible but still need to run the occasional Windows app or game, you have another option: Wine. This open source project provides a software compatibility layer that lets you run many Windows programs on Linux – albeit with varying degrees of reliability and compatibility. Wine is a mature piece of software, having been in development since 1993, but given the vast complexity (and closed nature) of Microsoft's operating systems, it has trouble running some apps.

Wine famously means “Wine Is Not an Emulator” – which seems silly for something that ostensibly emulates Windows. But really, Wine intercepts system calls from programs and reworks them for their Linux equivalents. This means that you can often get surprisingly good performance from Wine, because it's not doing the job of emulating a whole CPU; indeed, I've heard from readers who say that some Windows apps run faster on Linux+Wine rather than native Windows itself!

Over the next few pages, I'll show you how to install and configure Wine, exploring some of its workings and a few tricks to save time. If you've got some old Windows games you'd like to dig out, now's the time! Or indeed, even if you're fully Microsoft-free, it's worth knowing the essentials of using Wine in case you need to help a friend or school/club/charity/company transition to Linux while still having some Windows compatibility.

First Steps

The first thing you'll need to do is install Wine from your distro's package repositories. Almost every mainstream distro has Wine available, so it should only take a

```
sudo apt-get install wine
```

or equivalent to get it installed. If you're fairly new to Linux and not yet fully au fait with the command line, you should also be able to find Wine by searching in your graphical package manager, but bear in mind that you'll be using the command line for a few things in this tutorial.

Note that some distros may include older versions of Wine; if you have problems running a particular Windows program, it may be worth updating to the very latest version of Wine because compatibility improves with every release. On the download page [1], you'll find binary packages for various distros. The WineHQ wiki also shows you how to build it from source [2]. Just make sure you remove any Wine binary packages before compiling the source code, to avoid conflicts on your system.

Once you have Wine installed, it's a good idea to test it with a small and fairly simple program to make sure it's working correctly before moving on to more complicated apps. Here, I'll use a Minesweeper clone called MineZ. To begin, download the latest version (minez0.1.2.zip at the time of writing) [3], extract the archive, and run the game:

```
unzip minez0.1.2.zip
cd minez
wine minez.exe
```

If this is the first time you're using Wine, a new configuration will be created – more on that in a moment – but all being well, the MineZ game will appear on your screen and should be playable (Figure 1). There you have it, a native Windows app running flawlessly on Linux!

Working with Installers

Very few Windows programs are available as simple standalone .exe files, however; most of

Figure 1: Once we have this small and simple Minesweeper clone running, we know Wine is installed properly.



Chasing a Moving Target

Given that Wine has been in development for 23 years, you might wonder why it still has issues running some Windows programs. After all, that must be enough time to recreate the Windows API, right? Especially when the project has benefited from commercial backing. Well, there are two main reasons why compatibility is still not perfect.

First, a lot of the Windows internals are not documented – at least, not publically. Details for many API calls are thin on the ground, and Microsoft hasn't been particularly forthcoming with them. Yes, there have been some source code leaks of Windows versions, but Wine developers don't want to go anywhere near them for legal reasons. If they reimplement

something in Wine after seeing original Windows code, Microsoft's lawyers could spring into action. So "clean room" implementations are required, accomplished by studying the behavior of Windows.

Second, Windows is a moving target. If the Wine team had set a goal of supporting, say, Windows 98 with maximum compatibility, they might have reached that by now, and the open source world would have a fully FOSS Windows 98 alternative. However, many other Wine users want support for newer APIs and Windows features, so the Wine team has had to keep chasing this ever-changing goal. It's tough work, so massive kudos to the developers for doing it.

them require some kind of installation. For this example, you'll be using the Windows version of AbiWord [4]. Grab the `abiword-setup-2.8.6.exe` file (or a newer version if available) and run it in Wine:

```
wine abiword-setup-2.8.6.exe
```

This time, a familiar installation wizard will pop up, asking a few questions. To skip through them, just keep tapping Enter. Once the files have been copied over, choose not to run AbiWord immediately, and then the installer window closes. OK, what now? How do you run the freshly installed AbiWord from here?

Well, the first time you ran Wine, it created a minimal installation of a Windows-like operating system (including libraries and tools) inside `.wine/drive_c/` in your home directory. Have a look inside there, and notice the famous `Program Files` and `Windows` directories. After doing a bit of searching, you will find that the AbiWord executable has been installed inside `Program Files (x86)/AbiWord/bin`, so you can run it like so:

```
wine "Program Files (x86)"/Z
AbiWord/bin/AbiWord.exe
```

Note the use of quotation marks here to deal with the spaces and brackets, which are usually annoying to handle at the command line. Et voilà, AbiWord appears as expected, and you could create a desktop or menu shortcut to run the above command (including the full path of `/home/user/.wine/drive_c`) to save you always having to type it.

Bear in mind that because Wine is not an emulator, programs running inside it are not sandboxed away from the rest of the system. They can access files in your home directory like any other native Linux app. So you still have to be aware of security issues, even if a lot of viruses and malware wouldn't have the same effect as on a native Windows system.

You can see this yourself: in AbiWord, go to **File / Open** and navigate to **My Computer**. You'll see that the `C:` drive is mapped to `.wine/drive_c/` in your home directory, whereas `Z:` is mapped to the root (`/`) directory (Figure 2). In many cases this is useful, because you can access other files on your filesystem without having to copy them into a dedicated place, but if you want to run something potentially risky, it's better to do it in a virtual machine.

A last note on installers: Many Windows programs are now supplied in `.msi` format, for which you'll need the `msiexec` utility (included with Wine) followed by the `/i` parameter and the package filename to install:

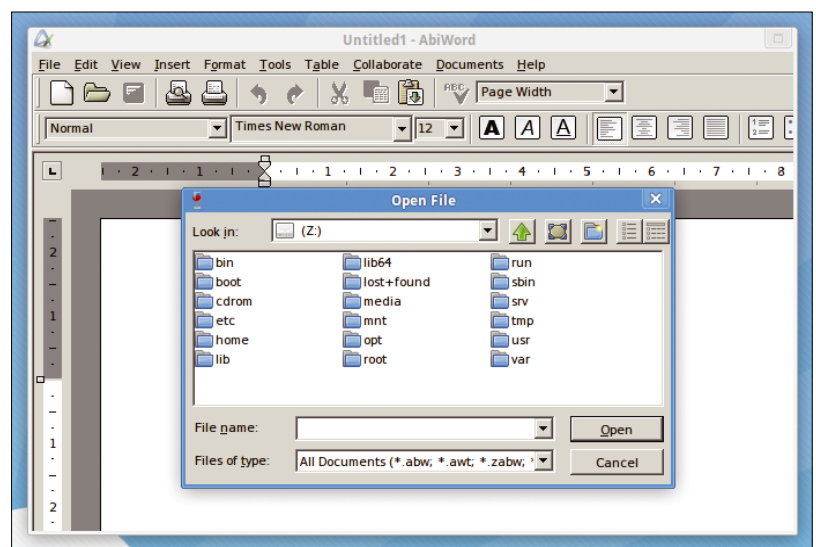
```
msiexec /i <program>.msi
```

You can then locate the program inside the `drive_c` directory.

Advanced Features

At this point, you can start trying to install your favorite old Windows programs and games, but chances are that you'll have some problems.

Figure 2: By default, Wine maps the virtual `Z:` drive to your root filesystem – so bear in mind that Windows apps can still see your Linux files!



Wine is far from perfect (see the “Chasing a Moving Target” box for reasons why), but it’s also highly configurable; you have so many settings to tweak, to (one hopes!) coax some programs into running.

Enter `winecfg` to start Wine’s graphical setup tool, and look at the tabs along the top (Figure 3). Of special interest for compatibility is **Applications**, which should contain a list of programs you’ve installed in Wine. (If not, click **Add Application** and locate them.) In the drop-down list at the bottom, you can choose which version of Windows should be imitated for each app.

Another tab worth looking at is **Libraries**. Wine ships with a number of DLL files that provide much of the functionality of Windows’ own libraries, but with some limitations and bugs. In many cases, you can improve compatibility by using real Windows DLLs – provided you have a legitimate copy of Windows, of course.

To see where Wine places these libraries, go into `~/wine/drive_c/windows`, and then the `system32` or `system64` directories. Enter `ls` at the command line, and you’ll see lots of DLLs created by the Wine team. If you want to use a native library, it’s best to place the native DLL alongside the EXE file of the program you want to run. Then, in the `winecfg` tool, you can choose to override Wine’s version with the native version.

Other tabs to look at include **Drives** (where you can change the mapping of C: and Z:), **Graphics**, and **Audio**. Now, say you’ve done all the tweaking possible to make a program work in your Wine installation, but you want to install another app that requires very different compatibility, library override, and graphics settings. Do you have to poke around inside `winecfg` each time you switch apps?

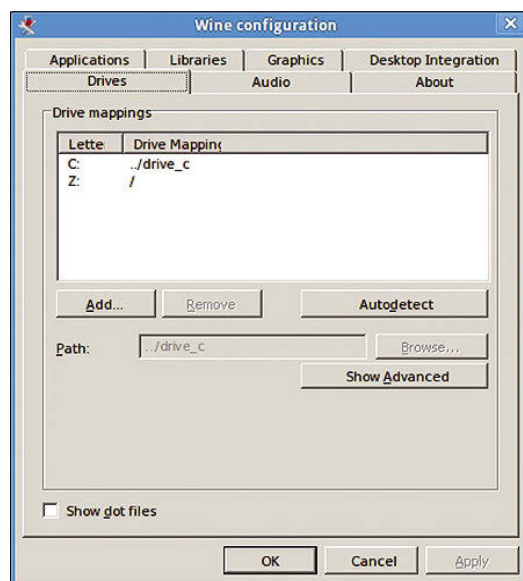


Figure 3: The `winecfg` tool provides a fairly user-friendly front end for fine-tuning your Wine installations.

Fortunately not. Wine includes a feature called “prefixes” (or “bottles”) that let you maintain multiple Wine installations simultaneously. It’s a bit like having separate versions of Windows on your machine. If you’ve downloaded a program called `CoolApp` that you want to install and run in its own Wine installation, you can create a new prefix by setting an environment variable like so:

```
env WINEPREFIX=~/.wine_coolapp wine coolapp.exe
```

This creates a new folder in your home directory called `.wine_coolapp` with all the Windows directory structure, DLLs, and tools as before, but it’s completely separate from your other Wine installation. To use programs inside it (or indeed the `winecfg` tool), just specify the environment variable before the command, as above. These separate Wine installations can take up a fair amount of disk space, however, so don’t forget to remove them when you no longer need an app or game!

A similar environment variable is `WINEARCH`, which chooses whether to run in 64-bit or 32-bit mode. If you’re running a 64-bit distro but having trouble with some Windows apps in Wine, try using `env WINEARCH=win32` at the start of your commands.

Extra Resources

If you’ve looked inside the `drive_c/windows/` directory of your Wine installations, you might have noticed a few `.exe` files. Yes, Wine includes simple versions of Windows Explorer, Notepad, and Regedit (Figure 4). Regedit is especially useful for fine-tuning settings and fixing compatibility issues. You can run it in place by simply entering:

```
wine regedit.exe
```

An especially useful online resource for checking compatibility of programs under Wine is the Wine AppDB [5]. You can see that apps have different ratings, such as platinum (runs perfectly without any Wine tweaking required), gold (runs perfectly but needs some configuration), silver (has minor issues but is otherwise usable), and so forth.

Use the search bar to find the apps you want to use, and check whether other Wine users have made any comments at the bottom of the page; they often contain useful tips and tricks for that specific app. If you’ve had success getting a program to run, but it’s not listed in this compatibility database, click **Submit App** in the menu on the left to add a new entry and describe how well it works and what you had to do.

Last, I’ll give a quick mention to `Winetricks` [6], an “easy way to work around problems in Wine.” `Winetricks` is a script that assists with the installation and setup of some common apps. To get it, click the link for the latest version from the web-

Alternatives to Wine

If classic MS-DOS games float your boat, it's worth checking out DOSBox [7]. This is a fully fledged emulator that includes a virtual x86 processor and basic implementation of DOS, and it is capable of running a huge number of old DOS games. Install it from your distro's package manager and then start it by simply providing a directory as follows:

```
dosbox mydir
```

This starts a new DOS session, with the C: directory mapped to mydir (or whatever you specified). Now you can enter the good old dir command to view files inside the directory or run a program. If you're running a game and DOSBox grabs control of your mouse pointer, hit Ctrl+F10 to get it back.

DOSBox is highly configurable: You can change how fast the emulated CPU runs, customize the virtual video and audio devices, and much more. The DOSBox wiki [8] at is packed with information on fine-tuning the emulator, so take a look. Oh, and it's even possible to run Windows 3.1 inside DOSBox [9].

Another alternative to Wine – or more precisely, a commercial version of Wine – is CrossOver [10]. CrossOver is like a value-added version of Wine with extra compatibility patches, configuration tools, and other bits 'n bobs. So, if you're having trouble getting something to work under Wine or need support, it's worth investigating. Plus, changes to CrossOver are rolled back in to the Wine codebase, so the original open source project benefits.

site, which will display a plain text file called winetricks in your browser. Save this to your home directory and run it:

```
sh winetricks
```

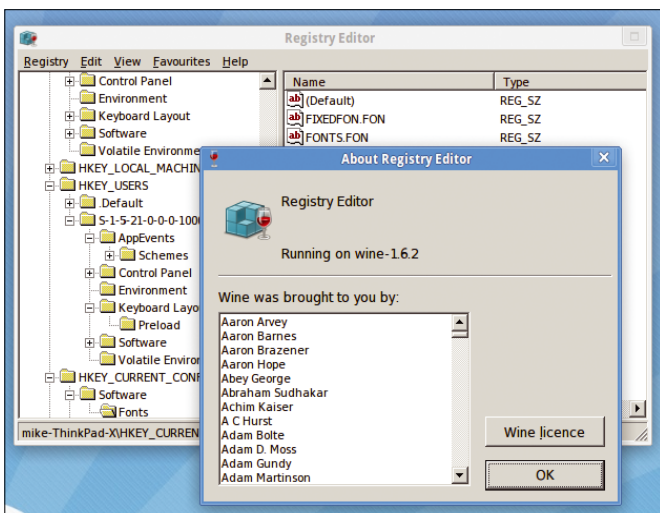
A menu will appear offering to help you install apps or games (Figure 5), so just choose what you want to do and follow the instructions.

Winetricks can save you a huge amount of time if you need to install many programs in Wine, but it's still important to know how Wine works under the hood – hence the “do it yourself” approach described earlier. With this knowledge and Winetricks at your disposal, you're in a strong position to help move friends and colleagues off Windows for

good (see the “Alternatives to Wine” box for more information).

Let us know how you get on! ■■■

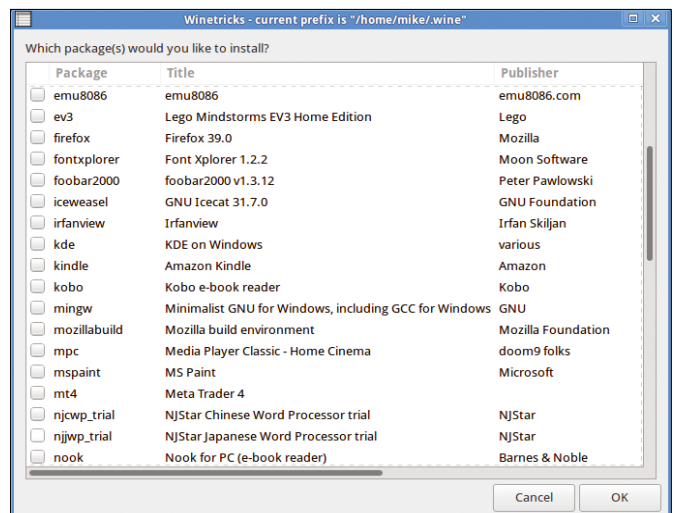
Figure 4: Wine is bundled with various tools such as its own editor for the dreaded Windows Registry.



Info

- [1] Wine download: <https://www.winehq.org/download>
- [2] Building Wine: https://wiki.winehq.org/Building_Wine
- [3] MineZ: <http://windrealm.org/minesweeper/>
- [4] AbiWord: <http://www.abisource.com/download/>
- [5] Wine AppDB: <https://appdb.winehq.org>
- [6] Winetricks: <https://github.com/Winetricks/winetricks>
- [7] DOSBox: <http://www.dosbox.com>
- [8] DOSBox wiki: <http://www.dosbox.com/wiki>
- [9] Run Windows 3.1 in DOSBox: <https://joshmccarty.com/2013/08/run-windows-3-1-in-dosbox/>
- [10] CrossOver: <https://www.codeweavers.com>

Figure 5: Winetricks lets you install various well-known Windows apps in Wine with just a few clicks.



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.



SCALE 15x

Date: March 2–5, 2017

Location: Pasadena, California

Website: <https://www.socallinuxexpo.org/scale/15x/>

Don't miss the 15th annual Southern California Linux Expo at the Pasadena Convention center in California. SCALE 15x is four days packed with more than 100 technical sessions and featuring more than 100 exhibitors. More than 3,600 Linux and technology professionals attended SCALE 14x in 2016. Visit the website to learn more about the event.

2017 HPC for Wall Street

Date: April 3, 2017

Location: New York City, NY

Website: <http://www.flagmgmt.com/linux/>

Register now for HPC for Wall Street for an All-Star Conference Program! This event is designed for forward-thinking industry veterans to get the latest on cloud and data centers. Visit the website to learn more and register today.

DrupalCon Baltimore

Date: April 24–28, 2017

Location: Baltimore, MD

Website: <https://events.drupal.org/baltimore2017>

The Drupal community is one of the largest open source communities in the world. We're developers, designers, strategists, coordinators, editors, translators, and more. Once a year, our community comes together in a US city for one of our biggest events: DrupalCon. This year we are excited to bring DrupalCon to Baltimore. Visit our website to learn more!

EVENTS

Open Source Monitoring Conference	November 29–December 2	Nuremberg, Germany	https://www.netways.de/events_schulungen/osmc/uebersicht/
IT-Tage 2016	December 12–15	Frankfurt, Germany	https://www.ittage.informatik-aktuell.de
DEVCONF.cz	January 27–29	Brno, Czech Republic	https://devconf.cz/
DevConf Panamá 2017	February 1–3	Panama City, Panama	http://devconfpanama.com/#/
SCALE 15x	March 2–5	Pasadena, California	https://www.socallinuxexpo.org/scale/15x/
CeBIT 2017	March 20–24	Hanover, Germany	http://www.cebit.de/home
SPTechCon 2017	April 2–5	Austin, Texas	http://www.sptechcon.com/
2017 HPC for Wall Street – Cloud and Data Centers Show and Conference	April 3	New York City, New York	http://www.flagmgmt.com/linux/
DrupalCon Baltimore	April 24–28	Baltimore, Maryland	https://events.drupal.org/baltimore2017
Check_MK Conference #3	May 2–4	Munich, Germany	http://mathias-kettner.de/
ISC High Performance (ISC 2017)	June 18–22	Frankfurt, Germany	http://www.isc-hpc.com/
AnDevCon	July 17–19	Washington, DC	http://www.andevcon.com/
InterDrone	September 6–8	Las Vegas, Nevada	http://www.interdrone.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.



AUTHORS

Erik Bärwaldt	24, 30
Swapnil Bhartiya	8, 14
Zack Brown	11
Bruce Byfield	60
Joe Casad	3
Nate Drake	18
Ben Everard	65, 72, 88
Andrew Gregory	67
Klaus Knopper	58
Charly Kühnast	48
Thomas Leichtenstern	46
Michel Loubet-Jambert	86
Graham Morrison	80
Hartmut Noack	40
Simon Phipps	66
Mike Saunders	68, 92
Mike Schilli	50
Valentine Sinityn	54, 74
Ferdinand Thommes	38

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.

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 Pettie

Layout

Dena Friesen, Lori White

Cover Design

Dena Friesen

Cover Image

© stillfix, 123RF.com

Advertising – North America

Ann Jesse, ajesse@linuxnewmedia.com
phone +1 785 841 8834

Advertising – Europe

Brian Osborn, bosborn@linuxnewmedia.com
phone +49 89 99 34 11 48

Publisher

Brian Osborn, bosborn@linuxnewmedia.com

Marketing Communications

Gwen Clark, gclark@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 11 67
Fax: +49 89 99 34 11 98

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.

Copyright and Trademarks © 2016 Linux New Media USA, LLC.

No material may be reproduced in any form whatsoever in whole or in part without the written permission of the publishers. It is assumed that all correspondence sent, for example, letters, email, faxes, photographs, articles, drawings, are supplied for publication or license to third parties on a non-exclusive worldwide basis by Linux New Media USA, LLC, unless otherwise stated in writing.

Linux is a trademark of Linus Torvalds.

All brand or product names are trademarks of their respective owners. Contact us if we haven't credited your copyright; we will always correct any oversight.

Printed in Germany

Distributed by COMAG Specialist, Tavistock Road, West Drayton, Middlesex, UB7 7QE, United Kingdom
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.

Published monthly in Europe as Linux Magazine (ISSN 1471-5678) by: Sparkhaus Media GmbH, Putzbrunner Str. 71, 81739 Munich, Germany.

Approximate	
UK / Europe	Jan 02
USA / Canada	Jan 20
Australia	Feb 20
On Sale Date	

Issue 195 / February 2017

Network Attached Storage

Unless you're ready to put everything in the cloud, you still need a place to store files. File storage technology keeps evolving, so we keep watching it. Next month we focus on Network Attached Storage.



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 © faithie, 123RF.com

FLOSSUK Spring 2017

14-16th March, The Studio, Manchester

<http://bit.ly/flossuk2017>

Join FLOSSUK Today

<http://bit.ly/flossukmembers>

flossUK 

OPEN

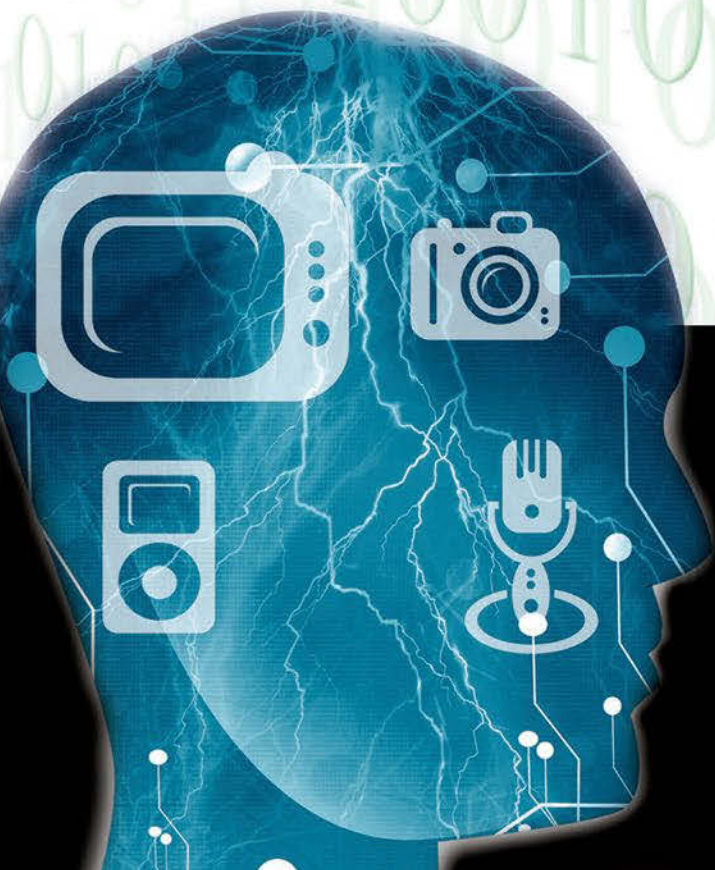
TECHNOLOGY

office@flossuk.org, www.flossuk.org

Free/Libre Open Source Systems

Open Software - Open Hardware

Open Data - Open Rights





MicroBlade

High Density • High Performance • High Efficiency • Cost-Effective
Enterprise, Data Center, Web Applications, HPC and Cloud Computing Solutions

Intel® Xeon® Processor E5-2600 v4/v3 Product Families Supported



New!

3U MicroBlade

Up to 28 UP/ 14 DP Nodes



6U MicroBlade

Up to 112 UP/ 28 DP Nodes

10Gbps



MBI-6128R-T2



MBI-6128R-T2X

Dual Intel® Xeon® Processor E5-2600 v4/v3

New!

HD Graphics
NVMe
10Gbps

New!



MBI-6219G-T7LX



MBI-6119G-T7LX

Intel® Xeon® Processor E3-1578L v5

New!

10Gbps

New!



MBI-6218G-T81X
MBI-6218G-T41X



MBI-6118G-T81X
MBI-6118G-T41X

Intel® Xeon® Processor D

New!

SAS3

New!

New!

New!



MBI-6119G-C2



MBI-6119G-C4



MBI-6119G-T4



MBI-6219G-T

Intel® Xeon® Processor E3-1200 v5



MBI-6118D-T2/T2H



MBI-6118D-T4/T4H

Intel® Xeon® Processor E3-1200 v4/v3



MBI-6418A-T7H



MBI-6418A-T5H

Intel® Atom™ Processor



Intel Inside®.
Powerful Productivity
Outside.



Learn more at www.supermicro.com



© Super Micro Computer, Inc. Specifications subject to change without notice.
Intel, the Intel logo, Intel Atom, Intel Atom Inside, Xeon, and Xeon Inside are trademarks or registered trademarks of Intel Corporation in the U.S. and/or other countries.
All other brands and names are the property of their respective owners.